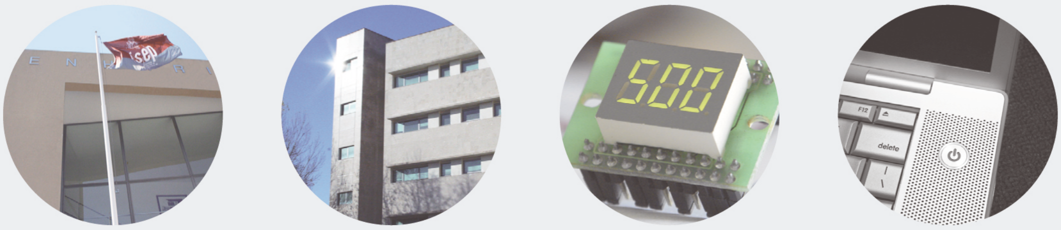
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



5G-Enabled Autonomous Platooning on Robotic Vehicle Testbed

JOÃO FILIPE MATEUS PEREIRA
julho de 2023

POLITÉCNICO
DO PORTO

Politécnico do Porto

Instituto Superior de Engenharia do Porto

5G-Enabled Autonomous Platooning on Robotic Vehicle Testbed

João Filipe Mateus Pereira

Master in Electrical and Computer Engineering
Specialization Area of Autonomous Systems

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto

July, 2023

This dissertation partially satisfies the requirements of the Thesis/Dissertation course of the program Master in Electrical and Computer Engineering, Specialization Area of Autonomous Systems.

Candidate: João Filipe Mateus Pereira, No. 1180526, `1180526@isep.ipp.pt`
Scientific Guidance: Pedro Miguel Salgueiro Santos, `pss@isep.ipp.pt`
Scientific Co-Guidance: John Harrison Kurunathan , `jhk@isep.ipp.pt`
Company: CISTER-Research Center of Real-Time & Embedded Computing Systems
Advisor: Pedro Miguel Salgueiro Santos, `pss@isep.ipp.pt`

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200–072 Porto

July, 2023

“Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.” - Albert Einstein

Acknowledgements

In this section, I would like to thank my supervisor Pedro Santos, Harrison, for the orientation regarding the work, without them nothing would be possible and to Francisco Pires and Luis Lam for the help in several stages.

Furthermore, I would like to thank for the possibility of working in the FLOYD project, it was a very challenging and captivating experience.

Considering this, I thank my family and friends that indirectly push me forward, so I can do the best for my life. This project has helped me grow and I am happy to say so. This thesis, perfectly ends this cycle and I hope it starts a new one, with many more challenges ahead. Concluding this section, I would like to thank my girlfriend, Mariana Carneiro, for being by my side at all times and giving me strength to continue and never give up until everything is on the way it should be. To all my colleagues at CISTER, I thank you for the freedom and well-being you have provided me in the time I have been here. To all, thank you!

Abstract

Humanity is progressively moving towards a more intuitive and technological future. The area of Intelligent and Cooperative Transport Systems has revealed itself as one of the areas in great evolution, through technologies of autonomous driving and intra-vehicle communication. With the main goal of providing accident-free environments as well as optimizing the movement of vehicles on roads all over the world, *Vehicle to Everything* (V2X) communication is very important when it comes to all kinds of vehicular applications. The *CMU/PT FLOYD* project focuses on this area, with the aim of developing new systems for possible future implementation.

In this report, a vehicular application using a 5G-capable module to perform *Vehicle to Infrastructure* (V2I) communications was evaluated. This vehicular application is based on an emergency braking scenario, whereby detecting an approaching vehicle in a place where an accident occurred, a message is sent over the network that is picked up by the main vehicle, triggering braking. It should be noted that this sending will be made through the module with 5G capacity, thus being an innovative application. Complementary to this scenario is the tracking of a vehicle by another vehicle, thus making a more complex emergency braking application with a cooperative platoon. This platoon will be maintained through sensors present in the following vehicle, such as LiDAR and ZED camera. With this, image processing and a sensor fusion was done in order to keep the follower at a safe distance but with the ability to follow the leader.

In order to validate and test this entire solution, robotic testbeds were used as a low-cost solution, allowing a concrete evaluation, with enlightening physical results of the entire application performed.

Keywords: 5G, V2X, Robotic Testbed, Module , Vehicle, Platoon, Image Processing, Sensor Fusion.

Resumo

A humanidade, está a caminhar, progressivamente, para um futuro mais intuitivo e tecnológico. A área dos Sistemas Inteligentes e Cooperativos de Transporte tem-se revelado como uma das áreas em grande evolução, através de tecnologias de condução autónoma e comunicação intra-veicular. Com o objetivo principal de proporcionar ambientes sem acidentes, assim como otimizar o movimento de veículos nas estradas de todo o mundo, a comunicação V2X é muito importante no que toca a todo o tipo de aplicações veiculares. O projeto *CMU/PT FLOYD* centra-se nesta mesma área, com o intuito de desenvolver novos sistemas de possível implementação futura.

Neste relatório, é avaliada assim uma aplicação veicular utilizando um módulo com capacidade 5G para realizar comunicações V2I. Essa aplicação veicular baseia-se num cenário de travagem de emergência, em que ao detetar uma aproximação de um veículo num local onde ocorreu um acidente, é enviada uma mensagem pela rede que é captada pelo veículo principal, despoletando a travagem. De destacar que este envio será feito através do módulo com capacidade 5G, sendo desta forma uma aplicação inovadora. Complementado a este cenário está a realização do seguimento de um veículo por parte de um outro veículo, tornando assim uma aplicação mais complexa de travagem de emergência com um pelotão cooperativo. Este pelotão será mantido através de sensores presentes no veículo seguidor como o LiDAR e a ZED camera. Com isto, foi utilizado processamento de imagem e foi feita a fusão de sensores de forma a manter o seguidor a uma distância de segurança mas com capacidade de seguir o líder.

Com o objetivo de validar e testar toda esta solução, foram utilizadas plataformas robóticas como solução de baixo custo, permitindo assim ter uma avaliação concreta, com resultados físicos esclarecedores de toda a aplicação realizada.

Palavras-Chave: 5G, V2X, Pelotão, Veículos, Fusão de Sensores, Módulo.

Contents

List of Figures

List of Tables

List of Acronyms

ADAS	*Advanced Driver Assistance Systems*
C-ITS	*Cooperative Intelligent Transport System*
C-V2X	*Cellular Vehicle to Everything*
CAM	*Cooperative Awareness Messages*
CISTER	*Research Centre in Real-Time and Embedded Computing Systems*
DENM	*Decentralized Environmental Notification Messages*
ECDF	*Empirical Cumulative Distribution Function*
ESC	*Electronic Speed Controller*
ETSI	*European Telecommunications Standards Institute*
I2V	*Infrastructure to Vehicle*
IEBA	*Infrastructure supported Emergency Braking Application*
IMU	*Inertial Measurement Unit*
ITS	*Intelligent Transport System*
MAC	*Medium Access Control*
MANET	*Mobile Ad-Hoc Network*
MCU	*Teensy Microcontroller Unit*
NTP	*Network Time Protocol*
OBU	*On Board Unit*
PID	*Proportional-Integral-Derivative*
PWM	*Pulse Width Modulation*
ROS	*Robotic Operation System*
RSU	*Road Side Unit*

V2I	*Vehicle to Infrastructure*
V2N	*Vehicle to Network*
V2V	*Vehicle to Vehicle*
V2X	*Vehicle to Everything*
VANET	*Vehicular Ad-Hoc Network*
VNC	*Virtual Network Computing*
VPN	*Virtual Private Network*
WAVE	*Wireless Access in Vehicular Environments*

Publications

M. Araújo, J. Silva, P. M. Santos, H. Singh, D. Gunjal, J. Fonseca, P. Duarte, B. Mendes, Raul Barbosa, P. Steenkiste, S. Sabamoniri, L. Lam, J. Pereira, H. Kurunathan. Demo Object detection under 5G-edge mobility. In Proceedings of the 24th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 12-15 June 2023, Boston, Massachussets, U.S.A..

Chapter 1

Introduction

Over the past decade, there have been monumental advancements in the domains of personal mobility and vehicular transportation systems. Despite safe navigation being the primary driving force behind V2V and *Vehicle to Infrastructure* (V2I) communications, the potential applications of vehicular networks extend far beyond. With vehicles such as cars and trucks playing an increasingly integral role in people's daily lives, integrating software-based intelligence into vehicles has the potential to enhance the user experience vastly. Furthermore, the demand for greater reliability, safety, and entertainment value in automobiles also has led to significant commercial development and support for vehicular networks and related applications.

The ever-growing expectation is that the next generation of vehicular networks will play a crucial role in supporting autonomous driving, traffic management, and infotainment applications. To achieve this, intelligent vehicular networks enabled by 5G systems will integrate various heterogeneous wireless techniques, thus enabling time-sensitive services with guaranteed quality of service and efficient bandwidth usage.

1.1 Context

In today's society, safety and comfort are becoming increasingly important in everyday life. The existence of intelligent infrastructures allows a complement that in the future may become indispensable on roads and cities all over the world. In addition, vehicles increasingly have the ability to receive information and use it to enable more assertive decisions. This information can be provided from intra-vehicle (vehicle sensors) and extra-vehicle (strategically placed infrastructure on the road) communications.

The V2V and V2I communications are examples of technologies that allow the exchange of as much information as possible in order to provide greater safety and stability to the vehicle. For this to happen, several tests have to be done and infrastructures installed in cities and roads worldwide, ensuring the possibility of constant communication and perception of the environment that allows total vehicle automation.

The current level of autonomy available in commercialized cars depends on what the car can see through a varied range of sensors, having some limitations. Introducing a layer of communications into the road networks can produce advantages such as more efficient autonomous operation due to the fact that the vehicles are able to have a more complete knowledge of their surroundings.

Consequently, there are more vehicular applications that can be exploited, automating the vehicle's actions depending on the scenarios in which they are inserted. Vehicular applications are services offered in vehicular contexts to improve traffic safety, efficiency, and sustainability. Two noteworthy vehicular applications are:

- **Platooning** - A sequence of vehicles that follow each other using their sensors (Cameras, LiDAR) and, if available, wireless communication;
- **Infrastructure-supported Emergency Braking** - The *Road Side Unit* (RSU) informs the vehicle to brake to prevent collision with another road-user (vehicle, pedestrian, etc.).

This project leverages the capabilities of 5G technology, such as low latency, high bandwidth, and reliable connectivity, to support advanced platooning algorithms and enable seamless coordination between multiple autonomous vehicles. The test bed also incorporates advanced sensing and perception technologies, such as LiDAR and computer vision, to enhance the perception and situational awareness of the platooning system, as can be seen in Figure 1.1.

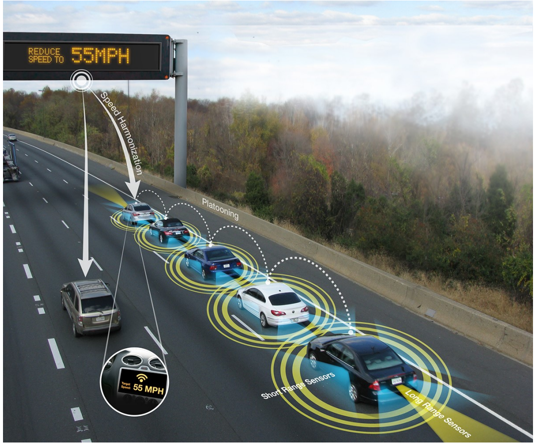


Figure 1.1: Cooperative Platooning [1]

The research and development activities of the project include the design and development of a robotic vehicle test bed, the integration of 5G communication capabilities and advanced sensing technologies, and the development and testing of autonomous platooning algorithms. The project aims to demonstrate the feasibility and effectiveness of 5G-enabled autonomous platooning systems and to provide insights into the potential of 5G technology in advancing the development of autonomous vehicle systems.

1.2 Motivation and Goals

Vehicles can leverage wireless communication for coordination towards a common goal, resulting in increased safety and efficiency. There are multiple radio technologies that can enable such communication, such as IEEE 802.11p for local-area networks and 5G for wide-area networks. Multiple protocol stacks can operate on top of those radio technologies, such as the *European Telecommunications Standards Institute* (ETSI) *Intelligent Transport System* (ITS) vehicular services stack and the traditional TCP/IP stack. The most recent of the two radio technologies, 5G, is currently being rolled out, with commercial 5G modules becoming available and coverage becoming widespread, and several use-cases making use of 5G in vehicular contexts have been proposed [2], although experimental validation is very scarce [3].

The specific application of infrastructure-supported emergency braking in platoons has seldom been addressed. This is a critical and demanding application that can be broken down into two components:

- **Base Station-Platoon Leader communication** - Ensuring the reliable and timely reception of a message from the infrastructure informing the platoon leader vehicle to brake;

- **Follower Awareness and Reaction** - Ensuring that this information is propagated throughout the platoon, via sensors or communication, in a timely fashion so that none of the following vehicles hits its front vehicle.

In [4], the infrastructure-supported emergency braking application has been showcased using a single 1/10 scale robotic vehicle and ETSI ITS/802.11p-capable on-board (OBU) and RSU, with the OBU triggering the physical action of braking on the vehicle after receiving a DENM from the RSU. Several aspects remain to be investigated on both components, notably:

- The performance of 5G for the Base station-Platoon leader link;
- Characterizing information propagation throughout the platoon (via sensors or local wireless links such as 802.11p).

The aim of this thesis is to integrate a 5G-capable module in the robotic vehicles, and implement a concrete vehicular application that leverages cellular communication: emergency braking of the platoon triggered by Internet-hosted services. As this work requires an operational platoon of robotic vehicles, improving the current platooning capabilities available in the robotic test bed is going to be needed. A performance evaluation of the vehicular application was carried out, namely by evaluating if the network response time meets the application timing requirements.

1.3 FLOYD Project

Connected vehicles are set to become a pervasive reality in the next few years. Still, providing seamless connectivity and computing services to vehicles is a challenging task, as it requires their coordinated action and the integration of their respective technologies. FLOYD, aims at building such a technological stack for offering high-performance network/computation services to vehicular users. Through the collaboration between complementary R&D industrial and academic entities, it develops, validate and exploit a set of components and technologies addressing multiple unanswered challenges regarding the integration between vehicular and network communications. These technological research and development lines are designed around two main driving forces: Artificial Intelligence and Platooning Applications. FLOYD explores the potential of Artificial Intelligence in multiple technological domains, by identifying processes in mobility, autonomy, or network operation that follow complex behaviors and thus require or have considerable potential to benefit from an intelligent learning approach. Underlying all the technological development, a down-to-earth, practical-utility use-case of vehicular platooning motivates the developed applications, test bed and demonstrator [5].

This project and its output are framed in the scope of FLOYD – "5G/SDN Intelligent Systems For LOw latencY V2X communications in cross-Domain mobility applications", a CMU-Portugal Large-Scale Collaborative Research project (AAC nº 04/SI/2019, Grant n. 045912) led by Capgemini Engineering and having as partners Altice Labs, Instituto de Telecomunicações, Vortex CoLab, CISTER Research Center/ISEP, and the Carnegie Mellon University.

1.4 Project Contributions

This project, in his first part, improves an existing platooning algorithm, by leveraging image processing from on-board cameras and point clouds produced by a LIDAR for sensor-based platooning, and extended it with wireless communication to enable information. In this manner, it allows better control of the platoon, with a multi-sensor approach being an important step in keeping it safe. The practical test done in the test bed contributes to see better results and check if the implementation is correct. Doing this application was important for future projects in this segment, being one step closer to full autonomous platoon cars.

In the second part, it was done an integration of a 5G-capable module in order to investigate the communications using cellular network. The application is doing an emergency braking of the platoon triggered by Internet-hosted services. For this manner, the car should be connected to the cellular network to receive the message to stop the vehicle. The valuable contribution of using 5G-capable cellular network, because of the scarce implementations and study's, was a test to see if it's usable in vehicular scenarios, evaluating the network responses time.

Having this is mind, the main contribution of this project is an application in the area of autonomous driving, making a vehicular scenario and evaluating all the measures of the application, using the last communication technologies and working with a 5G-capable module.

1.5 Work Plan

The work was performed over a robotic test bed of two 1/10 scale vehicles equipped with camera, LIDARs, among other resources. The use of a robotic test bed is necessary to have a more accurate representation of the application requirements, and provides a more immersive experience into the challenges of designing critical cyber-physical systems. Previous versions of platooning using robotic test beds have used image processing, sonars and range finders, whereas it's propose to evolve to an approach based on image processing and LIDAR; and have used IEEE 802.11p, with 5G not having been evaluated yet [4] [6].

With this in mind, a work plan has been done to have a better understanding of all the steps needed to achieve the goal of this thesis. This work plan was divided into two specific phases, a preparatory and a development phase.

In the preparatory phase, the steps that needed to be done were:

- Learn the fundamentals of 5G technology;
- Learn about multi-sensor approach (e.g., from image processing and LIDAR) for vehicle following;
- Study about the ETSI ITS standards and protocols for cooperative platooning and vehicular applications.

Regarding the development phase, this was a phase of practical elaboration of the studied contents and application in the robotic test bed. The primary steps were:

- Setup operation of 5G module and characterize its performance;
- Collect sensor information and develop control logic for sensor-based vehicle following;
- Set up collection of additional sensor data to enable cooperative platooning;
- Integrate 5G module in scale robotic vehicle;
- Integrate assembly in scale test bed to emulate vehicular application;
- Set up communication exchange between RSU and Car1 using 5G module;
- Evaluate performance of platooning approach in scale robotic platoon;
- Evaluate performance of vehicular application in scale robotic platoon using 5G module.

For a better understanding of the work plan, as well as a better time organization, a schedule that gives the project planning was made to fulfill the imposed goals. This simpler planning can be seen in the figure 1.2.

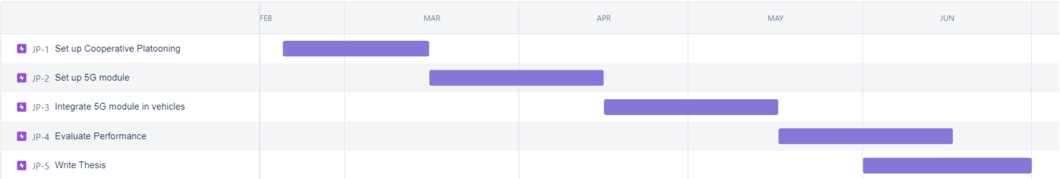


Figure 1.2: Temporal Project Planning

1.6 Dissertation Structure

In addition to the introduction, this dissertation contains five more chapters. In Chapter 2, a review of the state of the art is done. It presents an overview of the impact of vehicular communications technologies, the existent architectures with the main focus on ETSI ITS, and the existent open-source implementations and commercial solutions.

Chapter 3 describes the process of designing the applications to implement in the test bed. The theoretical composition of the system is built, and an analysis of the ETSI ITS messages is done to identify if they could be used for the proposed purpose.

In Chapter 4, it will be explained the implementation of the various applications, with the Emergency Braking and the platooning.

Following that, in Chapter 5, it will be showcased the results obtained from this project, with some discussion about them in the last part.

Finally, in Chapter 6 some conclusions will be obtained from the project, with some possible future work being described.

Chapter 2

State of the Art

This chapter presents a review of the existing projects regarding vehicular communications using 5G-enabled vehicular applications. During this chapter, several topics will be approached, including a overview of vehicular use-cases and the use of communications in autonomous platooning. In the last part of this chapter, the existing projects and studies around the 5G technologies in vehicular applications will be presented, as well as the platooning applications.

2.1 Vehicular Use-Cases

Vehicular communications consist on the ability of individual vehicles being able to communicate between them (V2V) and others (V2X), since road users to infrastructures. There are various use cases for vehicular communication [2], a few of which are illustrated below.

- **Autonomous driving** - Vehicular communications are crucial for enabling autonomous driving, which is arguably the most significant application of this technology. Facilitates cooperative and coordinated maneuvers between autonomous vehicles, such as intersection crossing and lane merging, there by enhancing mobility and making it more efficient;
- **Vehicle Management** - Provides the ability to deliver software and firmware updates to the vehicles over the air without causing inconvenience to the user;

- **Simpler Maneuvers** - Such as Cooperative Lateral Parking, where a vehicle signals to others its intent to perform a maneuver, and Obstructed View Assist, where the host vehicle can obtain a better point of view from an external camera or other vehicles;
- **Safety-Critical** - Essential application of vehicular communication technology. For instance, Cooperative Traffic Gap enables a vehicle to request cooperation from other vehicles before changing lanes;
- **Reports** - Accident reports and patient transport monitoring data can be shared through the network and sent to data centers, such as the traffic control center or hospital, respectively. This allows for more efficient and effective management of critical information, enhancing public safety and welfare.

In Figure 2.1 it can be seen a safety scenario where vehicular communications can be used as a way of helping the traffic control and send a danger message to the cars nearby in case of an accident.

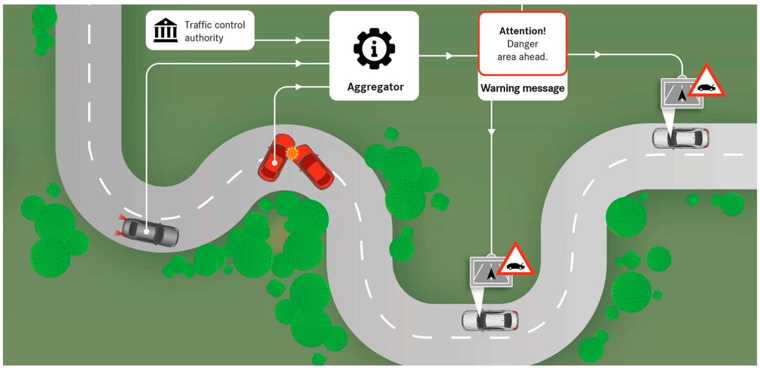


Figure 2.1: Safety Scenario [7]

One of the most important parts of vehicular communications will be to increase efficiency on the roads around the world by controlling all vehicular traffic [8]. Scenarios like intersections, traffic jams, can be optimized and avoided if it is possible to have communication between vehicles and coverage from the infrastructure of all places where there is vehicular movement. In figure 2.2 it is illustrated the communication that can happened in an intersection scenario. With all the vehicles connected to the RSU, the situation can be manipulated so that all vehicles pass through the intersection in the most efficient and fastest way.

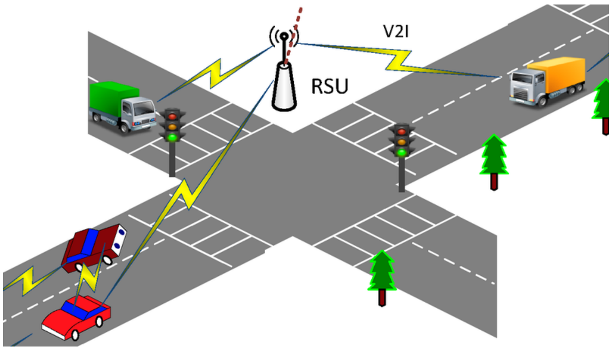


Figure 2.2: Intersection scenario communications [9]

In addition, parameters such as safety and the environment are influenced by the presence of vehicular communications. In the first case, when there is communication, information about the environment around the car will be obtained, making the car more aware of everything around it, helping in situations of poor visibility. In the second case, it will also help the environment, because the whole trip will be made more efficiently, using, for example, cooperative platooning [10].

2.1.1 Emergency Braking

The adoption of driving assistance technologies designed to automate specific driving tasks are used in order to reduce the chances of collisions and mitigate the accidents that happen in the roads around the world. However, it is important to acknowledge that these systems are not flawless and may experience failures under certain specific conditions, like traffic jams that can hinder communications or during football matches that can cause jammed channels [11].

The emergency braking use case refers to a scenario where a vehicle's braking system is activated in response to an imminent or sudden hazard. This feature is typically found in *Advanced Driver Assistance Systems* (ADAS) and is designed to assist the driver in avoiding or minimizing the impact of a potential collision [12]. Once the vehicle's sensors identifies a potential hazard, such as an obstacle or a vehicle ahead suddenly slowing down, the emergency braking system is activated. This prompts the system to engage the brakes with maximum force, aiming to bring the vehicle to a complete stop or substantially reduce its speed. In the figure 2.3 it can be seen two examples of the use of emergency braking in safety-critical scenarios, to avoid collision with a car and a person.

2.2 Vehicular Communication Technologies

Wireless communications used in Safety-Critical situations need to have low latency and great reliability. These operations rely on message exchange, including state monitoring, service data, control packets, and safety messages. The latency of these

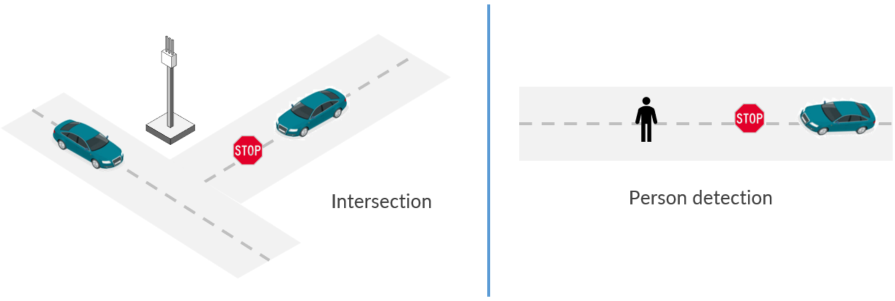


Figure 2.3: Emergency Braking Use Cases

messages can vary from milliseconds to seconds, depending on the intended operation. The data can be disseminated by time, motion, or events within the streets and can be distributed in distinctive ways, as unicast, broadcast or multicast.

2.2.1 C-ITS Architecture

C-ITS is a paradigm of information sharing by vehicles and smart infrastructures that is gaining more and more relevance, resulting in their further development. This communication and information is part of what is called *Vehicular Ad-Hoc Network* (VANET) [13]. In this type of network, most of its nodes are mobile and able to make decisions on their own, therefore being a form of *Mobile Ad-Hoc Network* (MANET) [14]. In the figure 2.4 is represented a typical VANET scenario.

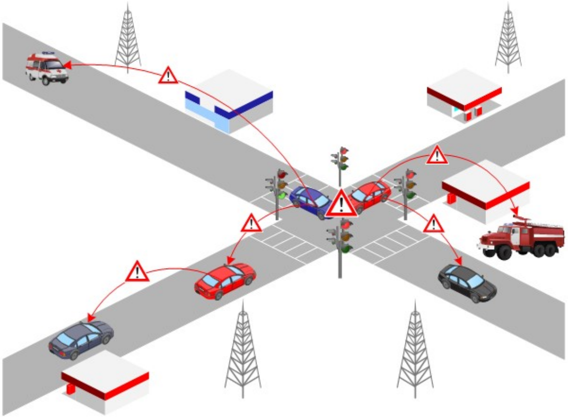


Figure 2.4: Scenario - VANET [15]

Belonging to this vehicular network are the main components such as:

- **RSU** - Fixed, they function as communication units providing information to vehicles within a certain radius of action. Used as sources of information on traffic conditions;

- **OBU** - Are responsible for managing external communications such as V2I and V2V communications.

Regarding the C-ITS structure, the connection types *Infrastructure to Vehicle* (I2V), V2I and V2V are of high importance in this project and can be complemented by other connection types (e.g. satellite), represented in the figure 2.5.

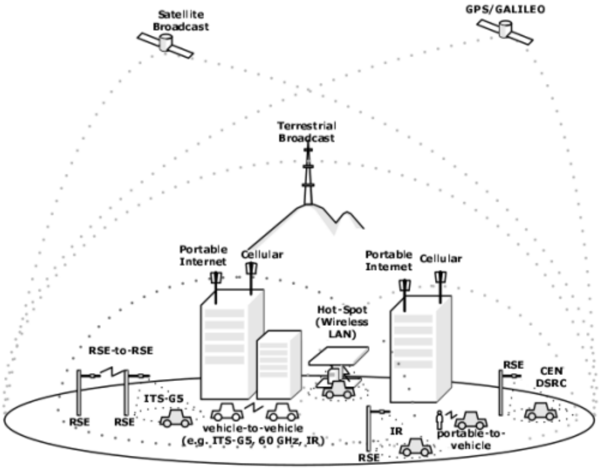


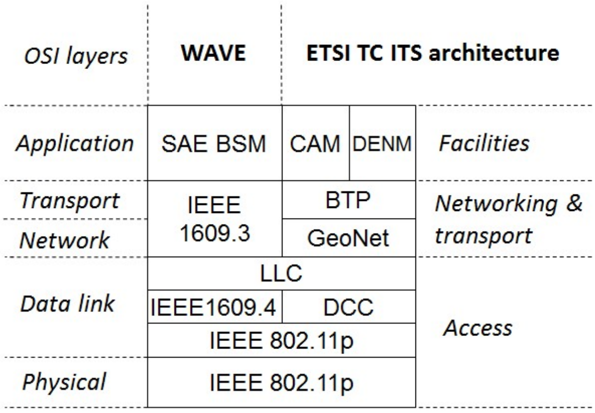
Figure 2.5: C-ITS - Connection types in a VANET network [16]

Finally, a connection to the Internet will enable services that will complement vehicle automation, without forgetting the need to comply with all safety conditions.

2.2.2 Overview of WAVE and ITS-G5 Technologies

The *Wireless Access in Vehicular Environments* (WAVE) and ITS-G5 technologies relate as two protocol stacks that use IEEE 802.11p in their MAC and physical layers. The WAVE differs in that it is used in the United States, while the ITS-G5 is used in Europe.

Regarding IEEE 802.11p, it depicts a technical specification of the 802.11 protocol in vehicular environments. As can be seen in the 2.6 figure, WAVE and ITS-G5 are stacked on top of IEEE 802.11p.



OSI layers	WAVE	ETSI TC ITS architecture		
Application	SAE BSM	CAM	DENM	Facilities
Transport	IEEE 1609.3	BTP		Networking & transport
Network		GeoNet		
Data link	LLC			Access
	IEEE1609.4	DCC		
	IEEE 802.11p			
Physical	IEEE 802.11p			

Figure 2.6: WAVE and ITS-G5 - Comparison [17]

Apart from the various similarities between the two protocol stacks, the main differences lie in the application, network and transport layers, as visualised in figure 2.6, and there is also a different organisation of the layers. Nevertheless, the two protocol stacks have the same objective, being used for all types of vehicular communications.

2.2.3 ETSI ITS-G5

Responsible for all types of communications presented in this project, the ETSI ITS-G5 [18] is the *standard* V2X most accepted by automotive industries in the European area, since in the United States it is more accepted the IEEE 1609 WAVE [19], which will be quite similar to the ETSI ITS-G5 itself in technical terms. Both use IEEE 802.11p for communication between underlying layers.

The high complexity of ITS systems results in a wide variety of them, with various implementations and topologies that allow for example the vehicle to receive important information about the state of the road, traffic and the entire neighbourhood from the RSU. In addition, in figure 2.7 is described the two more important communications in vehicular applications.

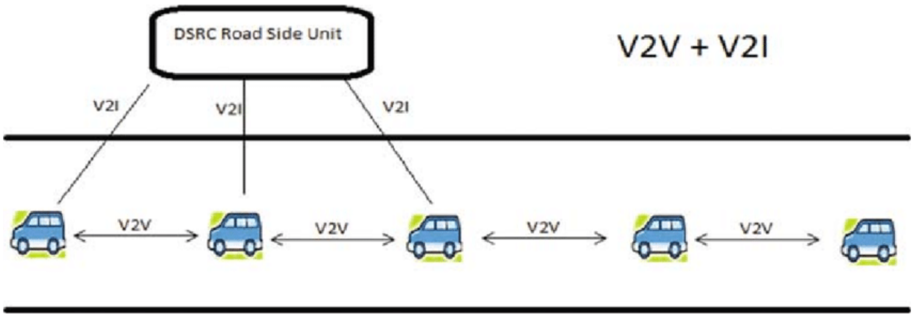


Figure 2.7: ITS Systems Network

Regarding the structure of ETSI ITS-G5, in its constitution there are 6 major groups:

- **Applications** - Application of the ITS scenarios, whether they are used for security, traffic efficiency, etc.;
- **Security** - Responsible for the security functionalities implemented in this standard;
- **Networking and transport** - Contains the functionalities and protocols dedicated to communications ITS;
- **Access** - The access layer provides the means to access the communication medium;
- **Management** - In charge of the proper functioning of all other layers;
- **Facilities** - Responsible for the exchange of information between vehicles and other stations ITS.

Its distribution in the most compact and functional way is shown in the figure 2.8.

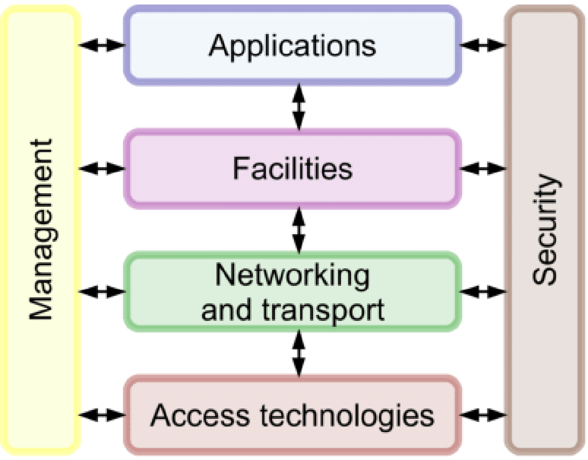


Figure 2.8: Distribution - ETSI Structure [20]

In this project, the main area of action will be in the *Applications* layer and the *Facilities* layer, with the manipulation of the DENM and the exchange of information realized by the vehicles. Now, in order to increase the level of understanding of the exchanged information and because it's the main focus of this work, the *Applications* layer will be addressed.

Applications

As already clarified, there are several types of application scenarios, but there is a special approach to:

- Traffic efficiency scenarios;
- Road safety scenarios.

In this project, both scenarios will be in evidence, as collision avoidance will refer to road safety but there will also be *platooning* which refers to the traffic efficiency application group.

In order to have safety, as well as accuracy in receiving and sending information, communication must have conditions that will be necessary and mandatory to have a functional and successful scenario. These conditions will be:

- Security;
- Reliability;
- Latency.

Due to the need to guarantee these conditions, ETSI created with the help of other users the so-called *Basic Set of Applications*, in which, for each possible use case, the necessary requirements are explicit. As well as the requirements, there are also some of the most relevant examples of scenarios for possible use.

CAM Messages

These messages are used to update a station's location or status information. They are sent periodically by a station to all other stations located within the communication zone specified from the *Basic Set of Applications* [21].

As position updating is one of the main objectives of sending this type of message, its sending time must fulfil certain requirements that ensure that there is no information failure and that everything is updated quickly. These requirements, specified in its structure by the ETSI, have as main factors:

- Minimum message generation of 0.1 seconds;
- Maximum message generation of 1 second;
- Once certain values (speed, angle, distance) are exceeded, messages are sent automatically.

The minimum sending frequency will vary, it can be smaller or larger, depending on the scenario used, and this is also evident in the ETSI requirements.

Therefore, the sending of *Cooperative Awareness Messages* (CAM) can be structured as visualised in the figure 2.9.

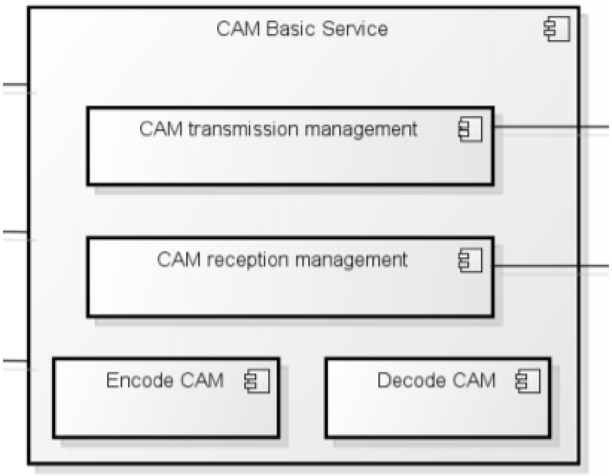


Figure 2.9: CAM - CaService [21]

As mentioned in the figure 2.9, messages are encoded according to ASN.1 notation [22], so a decoder must be available at the destination station to obtain the information.

Finally, ETSI allows the existence of different types of ITS stations that will create different priority hierarchies in the sending of CAM such as:

- **BasicVehicle** –> The private vehicles, serving as a base for the others;
- **basicIRS** –> A *RoadSide Station* that functions as an infrastructure;
- **EmergencyVehicle** –> Which, as the name implies, designates an emergency vehicle such as ambulance, fire brigade, etc;
- **publicTransportVehicle** –> Which is intended for public transport.

DENM Messages

DENM are messages that are triggered from events. Used by all stations, they act as a warning of an event triggered in a certain geographical area.

Before the detection steps are clarified, 4 *containers* can be specified when sending DENM. Each of them will provide information when sending the message. Thus, they are:

- **Management container** –> Mandatory, it contains information about the management of the DENM and its protocols;
- **Situation container** –> Contains information on the type of event detected;
- **Location container** –> Contains information about the location of the event;

- **À la carte container** –> Contains additional information (if needed) about the use case used.

In order to better explain the triggering of DENM, the detection steps and some characteristics will be listed.

1. When the event is detected by the ITS station, an DENM is sent to all stations present in a given geographical area;
2. The DENM is sent as long as the event is occurring and is repeated compulsively at a certain frequency;
3. Termination of the DENM is triggered when the event automatically disappears or when a station ITS the end of the event;
4. The information from the DENM is used by the stations that are interested in the information, and they decide what to do.

The structuring of the DENM is done in the *DenService* and is shown in the figure 2.10, with its most important steps.

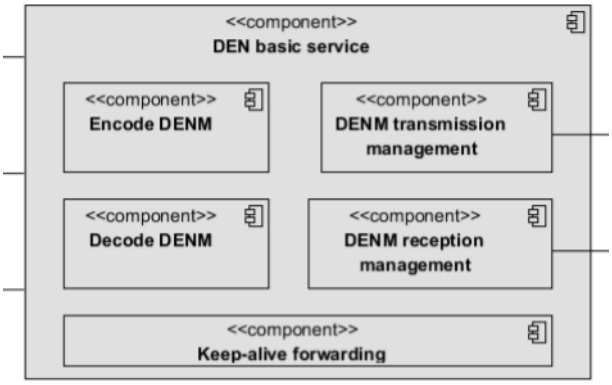


Figure 2.10: DENM - DenService [23]

All the information from the DENM is used by the *Facilities* layer of the ETSI ITS-G5 to update the *Local Dynamic Map.*

Finally, as shown in the figure 2.10, encoding by ASN.1 notation is also used.

2.2.4 IEEE 802.11p Protocol

Some protocols were done with an optimization to be used in vehicular communications. It is the case of IEEE 802.11p standard, a protocol that is built upon on the premise of enabling the best possible vehicular communication architecture. The IEEE 802.11p protocol, which is derived from the 802.11a Wi-Fi standard, has been designed to facilitate V2X communication in the US in accordance with the IEEE

1609 protocol [24]. In Europe, a similar protocol called ITS-G5 has been developed by ETSI [25].

The IEEE 802.11p protocol enables V2V, I2V and V2I [26] communications. In this way, IEEE 802.11p is designated as a set of protocols which guarantee communication between vehicles and infrastructures, leading to greater fluidity on the part of traffic.

In order to achieve its objectives, this standard has a range of regional specifications available for use in a vehicular environment [27], i.e., in an environment where changes occur very quickly.

Regarding its structure, IEEE 802.11p is a tweak on the IEEE 802.11 architecture [28]. It is based on the IEEE 802.11a protocols and has the main characteristics of 802.11q [29]. The main differences are in the *Medium Access Control* (MAC) layer, which had to be changed to make communications between vehicles faster and more efficient as well as also provide a faster communication time due to the high vehicle speeds [30]. In addition, the 802.11p protocol will also be responsible for the physical layer, as can be seen in figure 2.11.

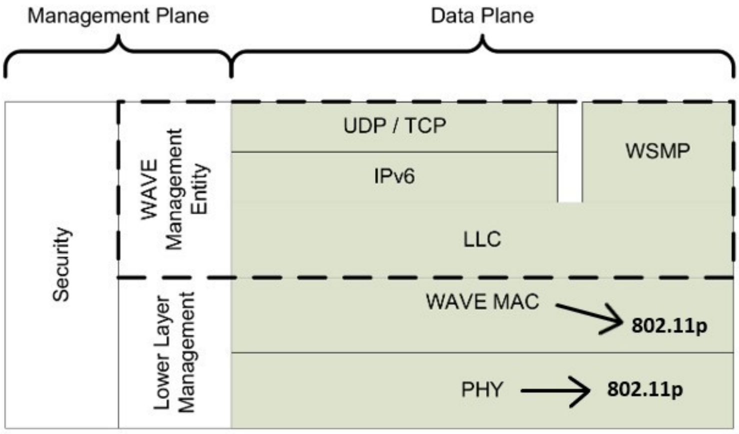


Figure 2.11: Example - WAVE Architecture (USA)

Advantages

Finally, the IEEE 802.11p protocol, in relation to other similar technologies such as for example LTE-V2V, is reflected as more appropriate for situations where there is a large movement of vehicles, allowing a good reception of the packets, with a delay also diminished [31]. With these particularities, it is revealed as a protocol that can be further developed and used in the future for intra and extra-vehicular communications.

2.2.5 C -V2X - Cellular Communications

In 2016, the Third Generation Partnership Project (3GPP) released the *Cellular Vehicle to Everything* (C-V2X) protocol within LTE Release 14. This protocol includes a short-range interface that can be utilized outside cellular coverage, making it a viable alternative to IEEE 802.11p. The short-range LTE-V2X, also known as sidelink, utilizes the same single carrier frequency division multiple access (SC-FDMA) employed by LTE uplink at the PHY and MAC layers. At the upper layers, LTE-V2X is compatible with the existing ETSI ITS and IEEE standards, which is crucial for its widespread adoption.

Architecture

C-V2X supporters advocate that the evolutionary path of cellular communications, transiting from LTE to 5G, will guarantee a long-term solution to road communications. Release 15 of 3GPP introduced the next generation 5G and New Radio (NR) in 2019, which offer significant improvements. However, Release 14 with LTE-V2X will remain the foundation for basic communications, while an additional optional interface with superior performance on various channels will be introduced [32]. In Figure 2.12 it can be seen the architecture used by C-V2X in vehicular applications.

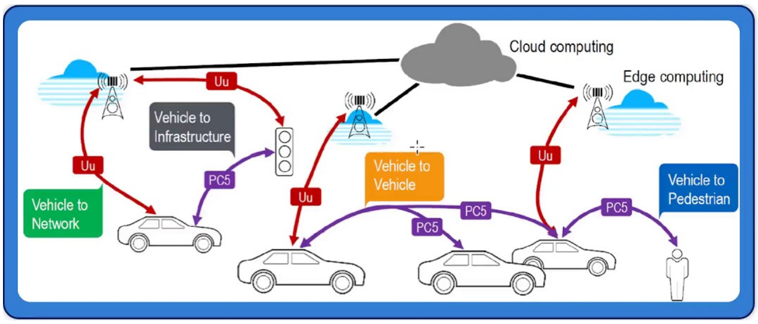


Figure 2.12: C-V2X Architecture [33]

As it can be seen in figure 2.12, C-V2X defines a new air interface called PC5 for V2V, V2I communication. *Vehicle to Network* (V2N) is over the legacy LTE Uu air interface and provides over the top cloud services.

C-V2X is designed for both In-coverage and Out-of-coverage communications. The transmission Mode 3 is defined as when network does the scheduling of resources for vehicles to communicate on (In-coverage) and the transmission Mode 4 is defined as when vehicles autonomously does resource selection based on sensing the environment (Out-of-coverage). There is no involvement of the network in transmission Mode 4.

The most interesting mode is the transmission Mode 4, that can therefore operate in out-of-coverage conditions and results perfectly suited for safety applications.

Despite the current scarcity of experimental deployments and uncertainty about its effectiveness in handling congestion, C-V2X continues to gain momentum day by day. Therefore, although C-V2X is still facing numerous challenges, its prospects for the future remain promising. Combining secure wide area and short-range connectivity in one module, C-V2X is a versatile and cost-effective solution for automakers looking to improve road safety [34].

Practical Scenario

Applying it to a practical platoon scenario, C-V2X facilitates communication between up to three vehicles in a platoon, enabling them to slow down or speed up simultaneously. Additionally, it can be utilized to notify other vehicles and roadside infrastructure of the platoon's presence. Typically, platoons are established on motorways and disbanded when a vehicle exits the motorway, making them highly adaptable. However, for platoons comprising more than three vehicles, relaying information between vehicles takes too long to achieve synchronous braking. Consequently, platoons with more than three vehicles must also utilize the low latency cellular network infrastructure that will be deployed with 5G [34].

2.3 Autonomous Platooning

2.3.1 Background

The Autonomous Platooning is an application by which two or more vehicles circulate on the road in a joint and coordinated manner, so that it ensures the necessary security in any project where it is inserted. As such, it is designated as an ITS application. There are several scenarios that are based on ITS, and a description of some of them is given by the *5GAA Automotive Association* [2]. But even though there are numerous scenarios [35], the most relevant ones to study belong to two groups:

- Road safety;
- Traffic efficiency.

These types of ITS scenarios entail several advantages, among them the decrease in fuel use, the decrease in the number of accidents and the increase in road capacity/efficiency [36].

Autonomous vehicles rely on multiple sensors to control and manage their behavior. However, wireless communication can enhance the safety and reliability of platooning, which is a key aspect of autonomous driving [37]. These services enable users to make better-informed decisions, promote safer practices, and encourage more coordinated and efficient use of transportation networks.

To implement a platooning application, the underlying technologies must be capable of maintaining a safe and minimal inter-vehicle gap while following the lead vehicle in the platoon. The benefits of platooning are directly linked to the distance between platoon members. Smaller distances between vehicles result in higher traffic and fuel efficiency, as more vehicles can be accommodated on the road and aerodynamic drag is reduced [38]. However, decreasing the distance between platoon members can also create safety concerns that must be addressed.

At this moment, there are 2 solutions to implement autonomous platoons:

- **Using only V2V communications** - Fully dependent on V2V communications, these must be assured to be properly working and safe, as any minor issue regarding information reception can be dangerous for a proper platoon maintenance;
- **Using intra-vehicle sensors, i.e, LiDAR and Camera** - Need to be able to properly follow its local leader. The hardware components involved must be very precise and calibrated, i.e LiDAR.

Since communications may fail due to interference or information congestion, the use of intra-vehicle sensors are a good complement to accomplish the autonomous platooning. In the figure 2.13 it can be seen how using this two solutions together are a upgrade to having just one of them.

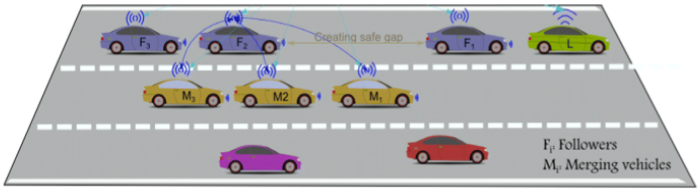


Figure 2.13: Platooning with intra-vehicle sensors and V2V communications [39]

2.3.2 Related Work

One of the projects well known in the platoon investigation it's the Essemble project [40], with companies like Bosch, Brembo and MAN present, the aim of this project is realise pre-standards for interoperability between trucks, platoons and logistics solution providers, to speed up actual market pick-up of (sub)system development and implementation and to enable harmonization of legal frameworks in the member states.

In [41] is presented a solution that fuse the V2V communications with sensor fusion of a radar and camera. Using this method, it prevents the danger of utilize just V2V communications to control the platoon, making use of the intra-vehicle sensors to complement the information received by a vehicle or infrastruture nearby.

The use of image processing and sensor fusion are one of the found solutions that can give the performance necessary to achieve this goal, havind in mind that the components need to be very accurate to not give bad measures that can prejudice all the system.

In the project [42], another solution was used, the authors try to compensate the failures using an adaptive Proportional-Derivative (PD) control to achieve stability in the platoon. They implement a dynamic information flow among the vehicles, applying a predecessor-follower mode. This way, the information is sent from the preceding vehicle to the following two vehicles. There are also sensors to detect the distance and position of the previous car.

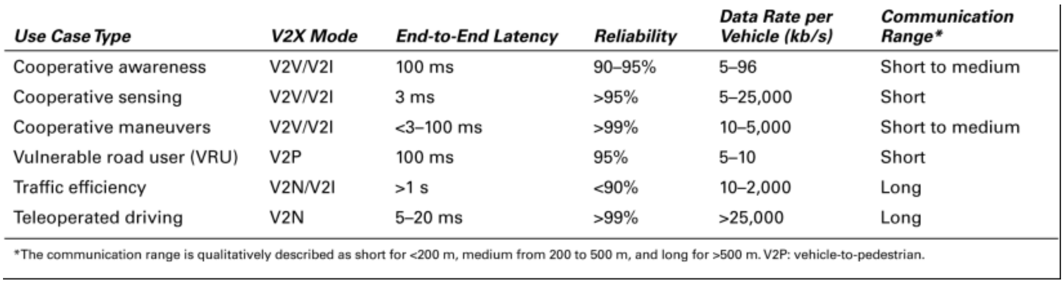
Several studies have been conducted on platooning, and one crucial factor for validating various methods is the practical implementation using a robotic test bed. In this way, it is possible to test all the development done in an environment similar to the one found on roads around the world, using a physical and scaled-down car with the features of a real car [43] [44].

2.4 5G-enabled Vehicular Applications

2.4.1 Background

5G, right now, is one of the most in evidence technologies worldwide. This is because it allows low latency and greater reliability to be used in real-time projects where communications are essential for a secure experience, like, in vehicular scenarios [45].

V2X use-cases may have various scenarios, such as convoy management (platooning), cooperative lane change, and collective collision avoidance. These use-cases collect data from onboard sensors and nearby vehicles to be utilized by autonomous vehicles in handling situations such as efficient traffic flow and emergency maneuvering [46]. 5G based V2X is expected to deploy use cases for supporting different situations as shown in Figure 2.14.



Use Case Type	***V2X Mode***	***End-to-End Latency***	***Reliability***	***Data Rate per Vehicle (kb/s)***	***Communication Range****
Cooperative awareness	V2V/V2I	100 ms	90–95%	5–96	Short to medium
Cooperative sensing	V2V/V2I	3 ms	>95%	5–25,000	Short
Cooperative maneuvers	V2V/V2I	<3–100 ms	>99%	10–5,000	Short to medium
Vulnerable road user (VRU)	V2P	100 ms	95%	5–10	Short
Traffic efficiency	V2N/V2I	>1 s	<90%	10–2,000	Long
Teleoperated driving	V2N	5–20 ms	>99%	>25,000	Long

*The communication range is qualitatively described as short for <200 m, medium from 200 to 500 m, and long for >500 m. V2P: vehicle-to-pedestrian.

Figure 2.14: Use Cases deployed situations in 5G V2X [47]

In Figure 2.14, for the 5G based V2X, some parameters can be observed, such as latency for each of the daily urban cases. One important detail is the difference seen from one use case to another, in terms of latency and reliably.

The reduced latency and increased reliability of 5G, compared to current technologies, will enable new use-cases such as trajectory sharing, real-time local updates, and coordinated driving. V2X communication utilizing 5G technology can support an end-to-end latency of ten milliseconds and an over-the-air latency of one millisecond, in the case of edge computing [48].

Because low latency and high reliability are both key factors in V2X communication, their joint function should be investigated. It is crucial to have precise and pertinent metrics to strengthen the level of service and criteria set by mobile network providers for autonomous vehicles. An important aspect to consider is whether the 5G data transfer technology is suitable for the demands of self-driving cars [49].

2.4.2 Related Work

Concerning projects assessing 5G V2X technology, in [50] were analyzed the requirements and use-cases related to 5G V2X and identified gaps in existing communication technologies. They also provide guidelines on how existing technologies, as well as 5G, can address these gaps. In addition, another of the projects concerning 5G V2X technology, the 5GCAR project aims at developing the specifications of 5G to be recognized as V2X applications enablers as they are still not in the spot due to the hinders in communication networks right now [51].

One of the aspects that are also key in vehicular applications, besides the latency and reliability, it's the security and privacy in the communications. This is also, a case of study that needs to be investigated, because, the 5G technology and the performance requirements of specific application scenarios bring about many security risks. In [52], it's done a study on the principal risks of 5G applications and presents hierarchical solutions for stakeholders to build secure 5G applications. Risks such as inserting false and unauthorized messages and information, and threatening the robustness of the system by jamming the same system are dangerous and can be fatal in some scenarios [53], giving an idea of what it needs to be kept in mind with 5G vehicular communications.

This project, is going to reinforce the investigation of 5G-capable V2X communications using cellular network. Due to the few studies carried out in this manner, this dissertation will distinguish from other projects with the practical implementation and scenario on a real robotic test bed, giving a more accurate study about the capabilities of the 5G-capable technology.

Chapter 3

5G-enabled Communications System for Platoons

In this chapter, the process of designing and describing the applications and the supporting system of this project it's explained. In addition, the main techniques used will be described.

3.1 Applications

3.1.1 Platooning

Platooning refers to a method of vehicle coordination and automation in which multiple vehicles travel in proximity to one another, usually in a convoy or a line formation, while maintaining a consistent speed and distance between each vehicle.

For the platooning scenarios, two techniques are used to maintain the safety distance between the Leader and the Followers.

The first one uses sensors to detect the distance and relative speed of the vehicle ahead, allowing it to adjust the speed of the vehicle accordingly without any kind of vehicle-to-vehicle communication. The second, on the other hand, takes the capabilities of the first one to the next level by incorporating vehicle-to-vehicle communications. With that, enables vehicles to share information with each other, creating a cooperative network of connected vehicles. In this project, it was done a fused implementation of these techniques, with the vehicles being connected using the OBUs and adapting the speed accordingly to the car in front.

Regarding the implementation, two ways were used in order to test the reliability of the sensor based platooning in an emergency braking scenario:

- **Cooperative** - When a message to stop is sent to the leader, he sends another message to the follower car and just then, the car stops;
- **Reactive** - Using the sensors present on the follower car, when it sees the car closer to a certain distance, it slows down or stops.

3.1.2 Infrastructure supported Emergency Braking Application (IEBA)

Furthermore, an even more complex scenario can be realized using cooperative platoons with emergency braking. That is the base of this project and aims to address the safety of all the cars in a platoon. It needs to have low latency in order to avoid collisions between the Leader Car and the Follower Car.

One way of having a more detailed vision in the environment around the car and around a determinate zone is by having an infrastructure. The infrastructure can play a significant role in enhancing the capabilities of in-car systems.

The RSU and Edge Nodes serve as monitoring stations that capture real-time information from the cameras and sensors deployed along the road network. The Edge Nodes are a physical or virtual machine located at the edge of a network. This data can then be processed and transmitted to in-car systems or other connected vehicles. By leveraging this infrastructure-based approach, a more comprehensive and accurate understanding of the traffic situation can be achieved.

In this case, the infrastructure is used in order to enable an emergency action triggered by a car approaching to a zone that is known to have an accident or a car stalled. To do this, the infrastructure is constituted by a Road-Side Camera that will be permanently watching a designated area of interest and will have the necessary range to oversee the whole traffic lane in order to see when will a car approach the zone with a hazard. In the figure 3.1 it can be seen how the Road-Side Infrastructure will operate in the project scenario.

3.1.3 Teleoperation

Teleoperation, also known as remote operation, is a technology that allows the control and manipulation of a system or device from a remote location. It enables operators to interact with and manage equipment or vehicles located at a distance, overcoming physical barriers and geographical limitations.

This application is very important in the area of autonomous driving, because it can be a safety measure to avoid constraints that can come with the development of autonomous driving systems, as well as gives an expanded operational capability.

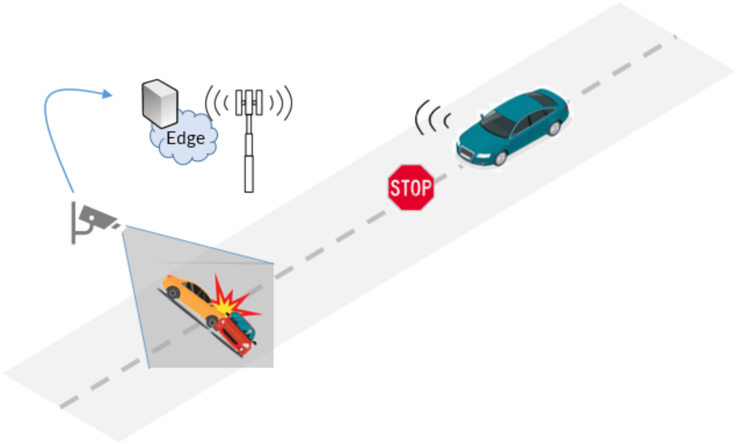


Figure 3.1: Emergency Braking Setup with Edge Node

The use of the Network to maintain control of a car can bring many challenges, with the necessity of low latency communication and a robust infrastructure to ensure a good data transmission.

In this project, it will be presented two types of teleoperation, via VNC using an VNC application and via Network using an Edge Node connected to the Internet.

This last teleoperation application is done using the network provided by the 5G module and the cellular network located on the car and the network of the Edge Node, giving a more detailed way to evaluate the use of a 5G-enabled module in a safety-critical application.

3.2 C-ITS Architecture

In this section, it will be shown the architecture that was used between the infrastructure, the platoon leader and the follower, with a brief description in order to get a better understanding on the communications part of the project.

3.2.1 Road-Side Infrastructure

The scenario that will be implemented, is composed by an infrastructure to handle the communications, the Local Edge Node, that runs the software necessary to enable object detection on the Road-Side Camera. If the Edge Node detects a vehicle that is, e.g., stopped further down the road, any vehicle that gets closer to a region of interest will receive a warning message and deploy the emergency braking procedure.

Edge Node

The Edge Node is composed by an Nvidia Jetson TX2, that is responsible by hosting the object detection service and the hazard advertisement service. This component will be the processing unit of the Road-Side infrastructure, with the object detection running continuously until the car is detected. After that, it will send a warning message via network that will tell the car to stop. It will be connected to the Internet in order to make a connection to the Leader Car via network to do both teleoperation and the IEBA.

Road-Side Camera

Regarding the Road-Side Camera, it's used a ZED Depth Camera, that can provide all the necessary information in order to visualize the approaching of the Car. It will be placed in a position close to the road where the cars will be crossing, being also responsible for giving the distances to the Cars that are approaching.

Hardware-wise, the overall Road-Side Infrastructure is shown on Figure 3.2 with the real components of the Edge Node and the Road-Side Camera.

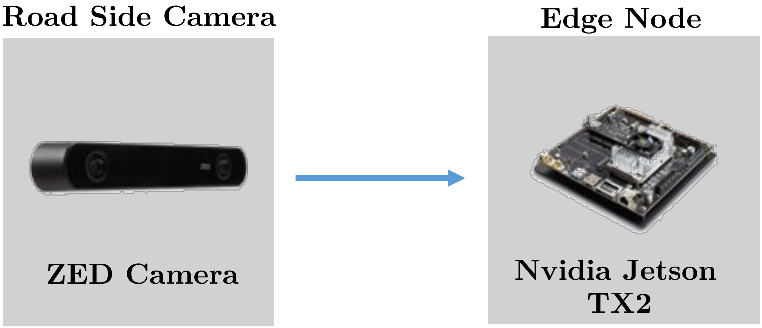


Figure 3.2: Road-Side Infrastructure Components

3.2.2 Autonomous Robotic Vehicles

The availability of an autonomous vehicle platform is crucial for accurately simulating real-world scenarios and replicating the behavior of a vehicle with communication capabilities. At *Research Centre in Real-Time and Embedded Computing Systems* (CISTER), a dedicated test bed for prototyping autonomous driving protocols, has been under development by multiple students over the years. This test bed, built upon the F1/10 platform [54], which is an open-source autonomous cyber-physical platform, offers a comprehensive solution that encompasses the chassis, sensors, computation, power board, and communication architecture (Fig. 3.3).

In this project, two test beds are employed to ensure both tangible results and an exceptionally authentic portrayal of a car's behavior in a safety-critical application. This approach enables the acquisition of physical data that contributes to the

Figure 3.3: F1TENTH Car Test bed [54]

validity and reliability of the application's outcomes. By utilizing these test beds, the project aims to achieve a high level of realism, ultimately enhancing the safety and effectiveness of the car's behavior in critical scenarios.

This test beds have 4-wheel drive, a maximum speed of 60 km/h and rubber tires that simulate a real car, all in a Traxxas chassis Rally Edition. The electric motor is controlled by a *Electronic Speed Controller* (ESC) using *Pulse Width Modulation* (PWM) with a *Teensy Microcontroller Unit* (MCU) being responsible for the interface of all the sensors of the car. For the vehicular communications, an OBU is installed with OPENC2X, enabling the exchange of information between the two cars. The most important component that needs to handle all the processes in order to get the application going and gives the computational power is an Nvidia Jetson TX2, installed on top of the chassis of the test bed. In addition to that, an array of sensors is installed on the car like a LiDAR (Hokuyo LiDAR for Car1 and RP LiDAR for Car2), a ZED Depth Camera capable of stereo vision and an *Inertial Measurement Unit* (IMU). Figure 3.4 describes the hardware architecture.

For the platooning, it was used an object detection algorithm that gives a bounding box and with the coordinates of it, the velocity and the angle of the Follower car will be calculated. In order to have the 5G capable communication, a 5G Module is also installed in Car1 with a SIM8200EA-M2 5G Hat. In the figure 3.5 it can be seen the two test bed cars used in this project.

In the figure 3.6, is shown a diagram, using the real components used to test the implementations, giving a better perception to the used material for this project.

As it can be seen in Figure 3.6, the C-ITS infrastructure is composed by three main components, the Road-Side Infrastructure, the Leader Car and the Follower Car. In addition, the connections between the main components are also described, with the network connection between the Road-Side Infrastructure and the Leader Car, and the connection using the ETSI ITS Stack of the OBUs between the cars.

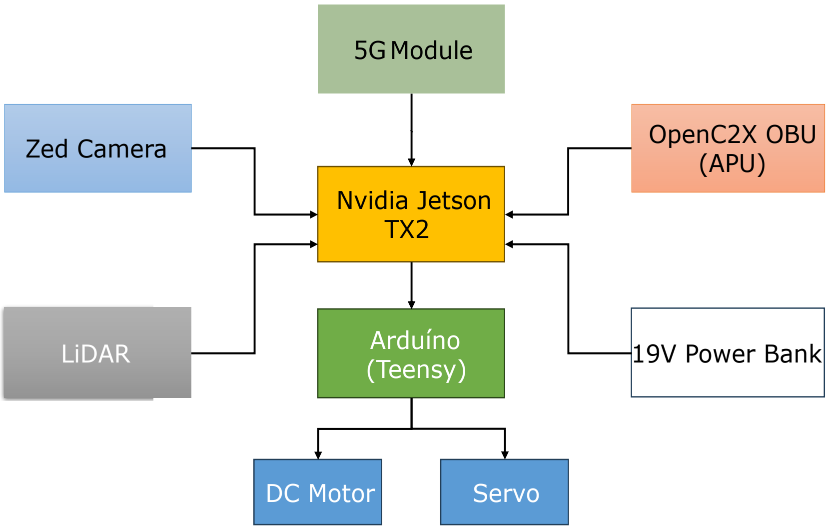


Figure 3.4: HW architecture of robotic vehicles

((a)) Leader Car ((b)) Follower Car

Figure 3.5: Test bed used in the scenario

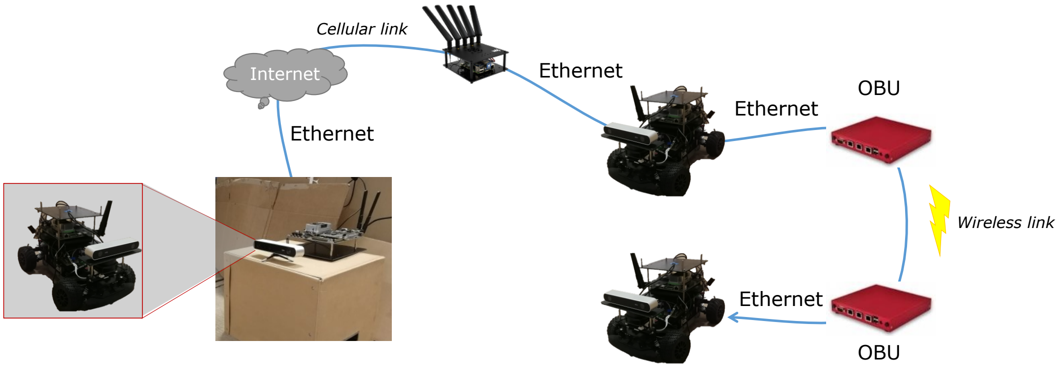


Figure 3.6: Setup of the C-ITS infrastructure with real components

3.3 Operation of the IEBA for Platoons

The Road Side Camera permanently watches a designated area of interest and have the necessary range to oversee the whole traffic lane. Using this, can be made an infrastructure supported application that can guarantee the safety of all users located on that road. With this in mind, a diagram has been made to better explain the system undertaken between all the components on this application, and it can be seen in Figure 3.7.

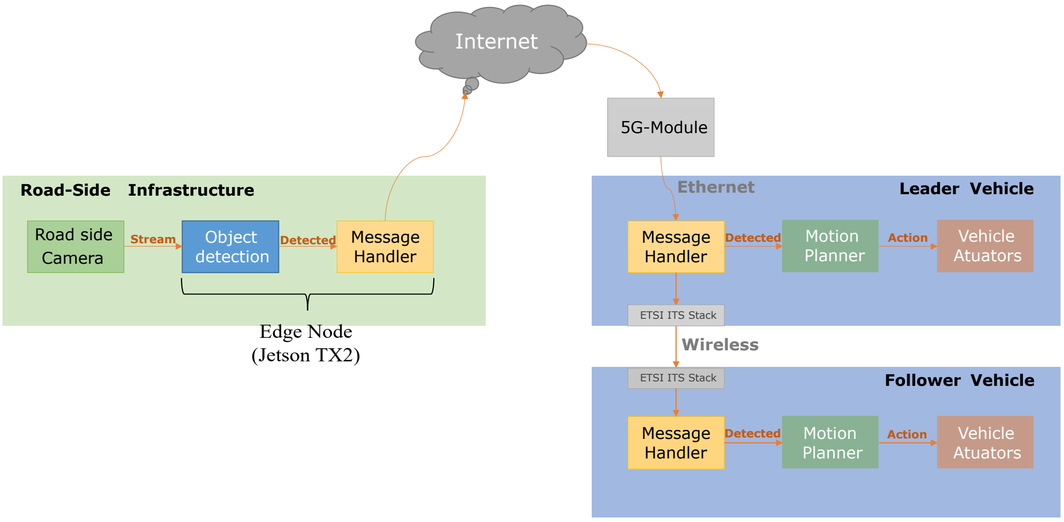


Figure 3.7: Diagram of the IEBA system

As it can be seen in the figure 3.7, a Road-Side Camera streams the video to the object detection software. The object detection software used is YOLOv3 [55], and it will be responsible for detecting a stop sign. A stop sign is used as it showed to be the most reliable option. It is important that the camera can measure distance, to identify in which lane a car might be travelling in and to decide if it is approaching the camera.

After that, a message will be sent over the Internet, using the public IP of each of the components (Car and Infrastructure) that should be received by the car and will trigger the emergency braking, for that reason, it needs to have a message handler in the two parts in order to receive and transmit the messages. For this to happen, the car leader will be equipped with a 5G-capable module that will give him connection to a cellular network and to the Internet. The Edge Node is connected to the Internet.

After that, the stop procedure is going to happen in the vehicle actuators in order to stop the car, first on the Leader and then, using the inter-vehicle communication, on the Follower car as it can be seen on figure 3.7.

The previously Figure 3.7 that represents the IEBA component's connection was adapted to a sequence diagram, as it can be seen in Figure 3.8. On this figure, the position of each timestamp along the sequence of events can be identified.

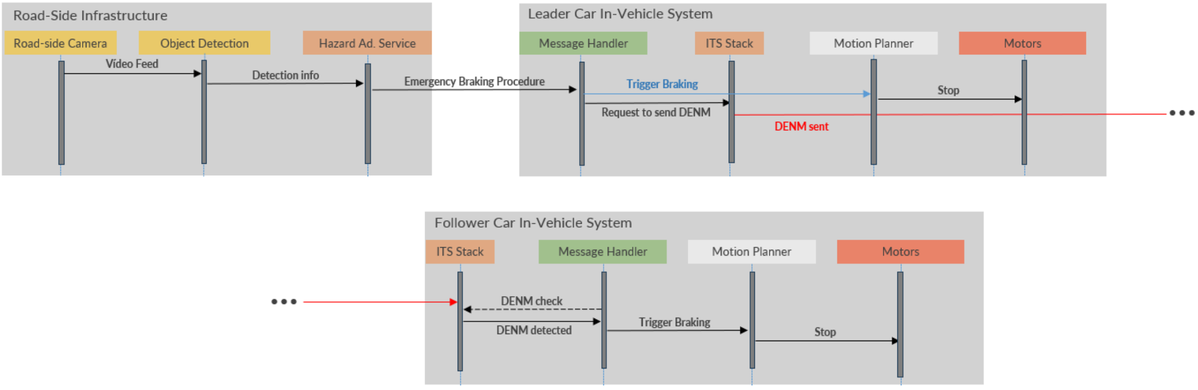


Figure 3.8: Temporal Sequence Diagram

Temporal Model

As it can be seen in Figure 3.8, many services are being used in order to make the IEBA being dependent of the communications between all the components. Having this in mind, the total time of this application can be calculated using a single expression:

$$t_{resp\_app} = t_{frame} + t_{detection} + t_{e2e\_latency} + t_{actuation} \tag{3.1}$$

In the equation 3.1 it's represented all the delay variables of this application. The t_{frame} is the time it takes to process the frames to the object detection software, that can go until 200 ms because the frames are processed at a frame rate of 5 frames per second, giving a maximum of 200 ms of delay. The $t_{detection}$ refers to the time that the YOLOv3 takes to do the detection of the car. The $t_{e2e\_latency}$ it's the time of all the Emergency Braking procedure, from the moment that it sends the messages to the Leader Car to the moment that arrives the DENM to the Follower Car. At last, the $t_{actuation}$ refers to the time it takes to engage the brakes after receiving the stop order.

3.4 Infrastructure supported Teleoperation

For this application, the vehicles, and the Edge Node were used in order to make an infrastructure supported teleoperation. Therefore, the Edge Node can ensure that the vehicles, in a situation where there is a movement blockage, will have a way out, using teleoperation in the Edge Node.

As it can be seen in the figure 3.9, the Edge Node will send the message via network (using the Internet connection of the Edge Node and the Vehicle) with the key command to increase/decrease velocity and the angle to turn, with the Message Handler in the vehicle being responsible to decode the message and do the right action.

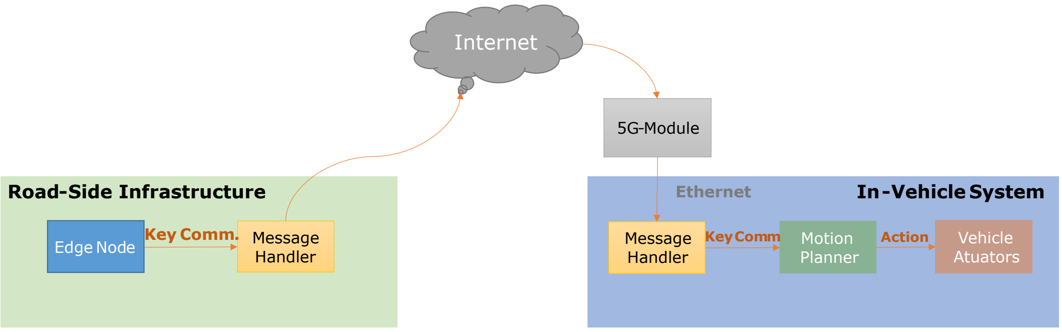


Figure 3.9: Teleoperation Diagram

With this application, more results can be obtained in order to test the 5G-capable network, with the exchanging of multiple messages with multiple commands to the vehicle.

The previously Figure 3.9 that represents the Teleoperation component's connection and actions was adapted to a sequence diagram, as it can be seen in Figure 3.10. On this figure, the position of each timestamp along the sequence of events can be identified.

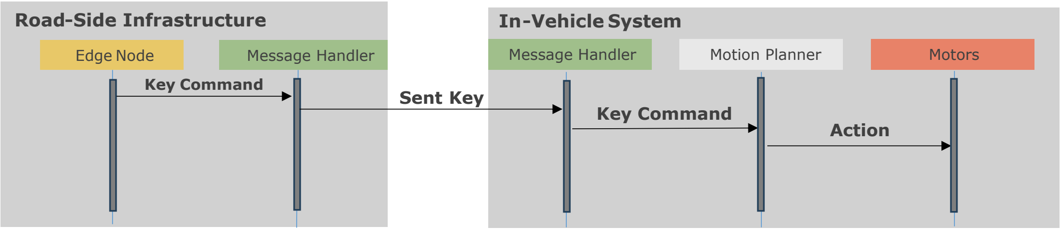


Figure 3.10: Temporal Sequence Diagram – Teleoperation

Temporal Model

As it can be seen in Figure 3.10, multiple timestamps come together in order to get the full application. Having this in mind, gathering all these timestamps, a temporal model can be formed. In the equation 3.2, the calculation of the temporal model of the full application is observed.

$$t_{resp\_app} = t_{send\_Key} + t_{calc\_action} + t_{actuation} \tag{3.2}$$

In the equation 3.1 it's represented all the delay variables of the Teleop. application. The first, t_{send_Key} it's the time that it takes to send the message from the Edge Node to the Vehicle, depending on the network. The t_{calc_action} it's the time it takes to choose the action to do from the received key. Finally, the $t_{actuation}$ it's the time it takes to increase/decrease the velocity or the angle to turn.

3.5 Takeaways

A complex scenario with the Platoon doing an Emergency Braking procedure was described. The Road-Side Infrastructure will be connected to the Internet and will communicate the hazard to the Leader Car (described in Section 4.3), with him being the responsible to transmit to the Follower Car the necessity to stop via 802.11p using the OBU's (described in Section 4.2).

Furthermore, the incorporation of teleoperation is a plus, that is important to evaluate the network behavior and can in a real scenario be useful to manipulate the vehicle in a more difficult environment.

Chapter 4

Implementation

In this chapter, the implementation of IEBA will be showed. Besides that, the platooning implementation will also be explained, ending the chapter with the implementation of the teleoperation.

4.1 Improvements performed over the Robotic Test bed

In this section, it will be showed all the improvements done on the robotic test bed in order to get the cars ready for the applications scenarios, like the design and all the control adjustments.

Having an autonomous vehicle platform at our disposal is crucial for this project, as it allows us to closely simulate real-world scenarios and replicate the behavior of a vehicle equipped with communication capabilities. At CISTER, a 1/10 scale vehicle dedicated to prototyping autonomous driving protocols is being developed for some years by students that are involved in some projects of this institution. As it was said early, this test bed is based on the F1/10 platform, an open-source autonomous cyber-physical platform, that provides chassis, sensors and computation, power board and the communication architecture. Having this in mind, some relevant equipment to this project is now described.

RPLiDAR A1 360°Laser Range Scanner

LiDAR, short for Light Detection and Ranging, offers more than just the ability to scan the surrounding environment within a specific range, depending on the sensor's capabilities. It also enables the measurement of the distance to objects with precision. By emitting laser pulses and analyzing the reflected light, LiDAR systems can accurately determine the distance between the sensor and a particular object or surface. This capability is essential for various applications, including obstacle detection, mapping, and navigation in autonomous vehicles.

For this project, a LiDAR was used in order to get the distances between vehicles and maintain the platooning running smoothly. The LiDAR used was an RPLiDAR A1 360° Laser Range Scanner that is a low cost 360 degree 2D laser scanner (LIDAR) solution developed by SLAMTEC. In figure 4.1 it's displayed the LiDAR used in this project.

Figure 4.1: RPLiDAR A1 [56]

The RPLiDAR-A1 can perform 360º scan within 6-meter range and produce a 2D point cloud data that can be used in mapping, localization and environment modelling. Also, has a scanning frequency that can be configured until 10 Hz maximum. It's a low-cost LiDAR that function very well in indoor and outdoor environments.

In this project, the RPLiDAR A1 will be the only LiDAR used even if there is another LiDAR in the leader car, the Hokuyo LiDAR UST 10-LX. The main objective of this LiDAR is to give the distance to a certain zone of interest that is determinated by the object detection, in order to increase/decrease the follower car velocity if the distance to the leader car increases/decreases.

ZED Depth Camera

The ZED is an advanced stereo camera that offers high-definition imaging capabilities along with precise depth measurement of the surrounding environment. Specifically engineered for demanding applications, the ZED camera excels in areas such as autonomous vehicle control, mobile mapping, aerial mapping, security, and surveillance. With its exceptional performance and versatility, the ZED camera proves to be an invaluable tool for capturing detailed visual data and obtaining accurate depth information in a wide range of challenging scenarios.

In this project, the ZED camera will be used in the leader, follower and in the infrastructure, being an essential part of this scenario. It has several notable characteristics that can be used to setup complex scenarios and is compact and lightweight, making it easily mountable on different platforms, including vehicles, drones, or robotic systems. In figure 4.2 it's displayed the ZED camera used in the multiple platforms of this project.

Figure 4.2: Zed Stereo Camera

The ZED camera's depth sensing technology accurately measures distances within a range of 0.5 to 20 meters, operating at a maximum frame rate of 15 frames per second, delivering smooth and real-time visualization. The software support given by the ZED SDK turns easy the use of this camera, having a rigorous calibration software and an interface to do 3D reconstructions. The python library that comes with the ZED SDK gives an easy way to program and do image processing algorithms.

4.1.1 Design adjustment of the Test bed

In the previous works [4] [6], some changes to the robotic test bed were done in order to accomplish their project goal. But, in a general basis, the work was incomplete and with some inconsistencies. For this reason, some adjustments to the leader and the follower vehicle needed to be done in order to make the cars more reliable, with a more aesthetic look and at the same time optimize the spots where the sensors could be to have the application running smoothly.

Leader Car

Having this in mind, first, the Leader Car was optimized. In the figure 4.3 it can be seen the original state of the leader vehicle without the adjustments.

As it can be seen on figure 4.3, the car was with tapes under the LiDAR and with the camera being held with tapes too. The stop signs and some components were in a cardboard and the position of the sensors were not optimized.

((a)) Leader Car ((b)) Leader Car

Figure 4.3: State of the Leader Car without the adjustments

For these reasons, the design of the test bed was changed, with new platforms to hold the components and the sensors in acrylic being done, the positioning of the camera was changed, no more cardboards, the stop sign was putted in a position closer to the front of the car and a more stable car without loose parts was accomplished. In figure 4.4, it can be seen the leader car on his actual form, with all the adjustments done.

Figure 4.4: State of the Leader Car with the adjustments

Follower Car

Now, for the Follower Car, more work needed to be done in order to optimize the sensors because of the multi-sensor approach used for the platooning and at the same time, give a more aesthetic look of a car to this test bed, without tapes and loose parts. In the figure 4.5 it can be seen the original state of the follower vehicle without the adjustments.

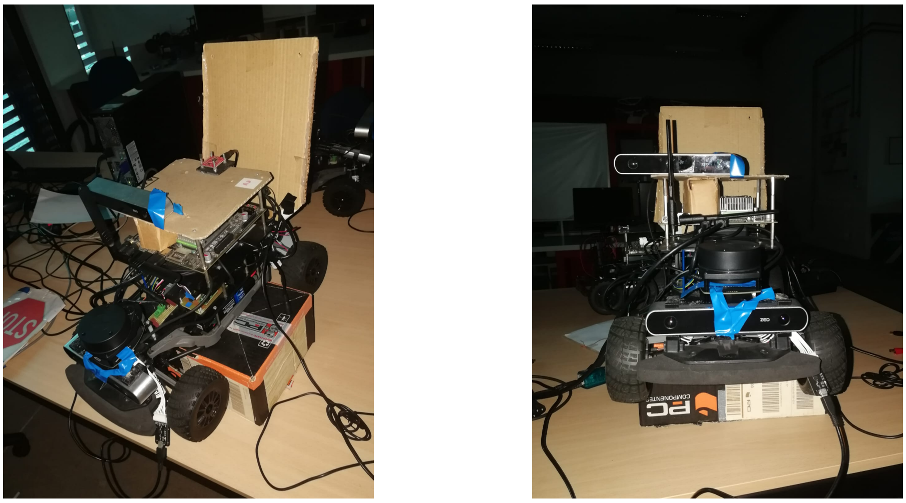


Figure 4.5: State of the Follower Car without the adjustments

As it can be seen on figure 4.5, more cardboards were used and some components were installed in that same cardboards. That was all replaced by an acrylic material and organized in a better way, with some things being removed from the car, like the back cardboard and the second camera that was not being used. The main sensors and components like the LiDAR and the camera were attached, because, in previous works, they were loose on the car, when the car was starting to run, they moved and for this application and the multi-sensor approach that could not happen, because it needs the most accuracy possible.

Therefore, In figure 4.6, it can be seen the follower car on his actual form, with all the adjustments done.

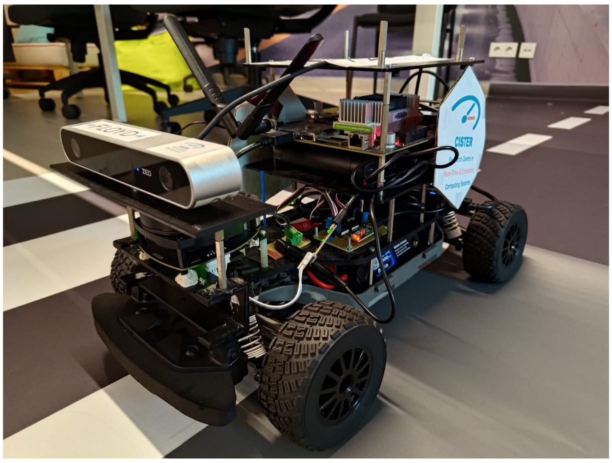


Figure 4.6: State of the Follower Car with the adjustments

4.1.2 Vehicle & Control Description

In previous works [4], a line following algorithm was implemented in order to lap a pre-defined circuit and simulate a car following a road lane. This algorithm uses the ZED Camera installed in the front of the car to detect the white line present in the CISTER floor. The decision to make the following line white was because of the better contrast that a white line does in the brown carpet used in CISTER.

Using *Robotic Operation System* (ROS), it turns out to be simple the implementation of this algorithm. The ZED Camera captures the video, sends him to a ROS topic and then, it's done an image processing algorithm to detect the white line using Canny edge detection to detect the edges between the CISTER carpet and the white line. After that, a progressive probabilistic Hough transform is applied to detect the coordinates of the detected lines. These coordinates are, then, putted in a ROS message and transmitted to the module responsible to do the motion of the vehicle.

In figure 4.7, it is shown the line following architecture and all the modules that make part of it.

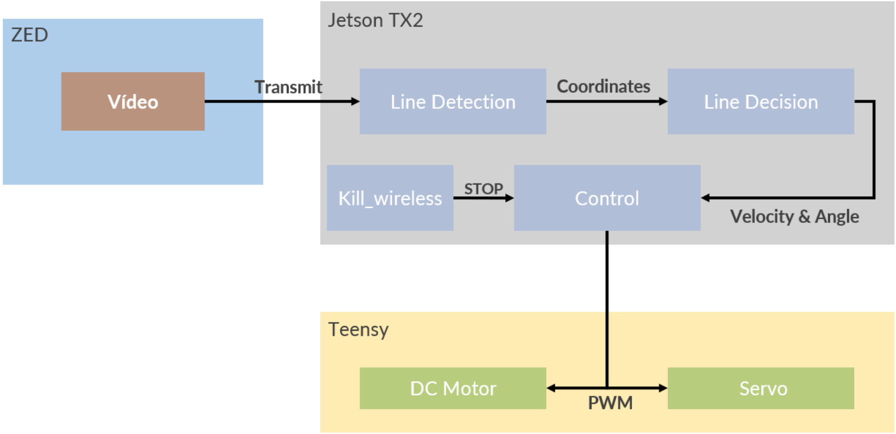


Figure 4.7: Leader Car Control Architecture

As it can be seen in the figure 4.7, the line decision module gets the coordinates from the line detection module via ROS topic, being subscribed to it. The coordinates received are from its own position (x_1, y_1) and from the position that the car aims to be (x_2, y_2). To calculate the angle that the car needs to turn, a comparison between the center of the camera and the direction of the received line, and the travelled distance were used, like it can be seen on equation 4.1. The travelled distance is calculated multiplying the velocity at which the car is going by the time elapsed.

$$Theta\_error = arcsin(\frac{steering\_error}{travelled\_distance}) \tag{4.1}$$

In top of that, a *Proportional-Integral-Derivative* (PID) controller was implemented, in order to minimize the error actions and do a better control of the car, in a more stable way. The proportional component of the system compares the received line's direction with the camera's center, generating a steering angle known as the theta error (θ_{error}). The integral component accumulates the sum of previous θ_{errors}, which is then incorporated into the final output. On the other hand, the derivative component calculates the difference between the current θ_{error} and the previous one, contributing to the overall calculation. All this calculation can be summed up in one equation, that is:

$$Steering\_Angle = K_P \times \theta_{error} + K_I \times \int steer\ dt + K_D \times \frac{\Delta\_steer}{dt} \tag{4.2}$$

After this calculation, this value needs to be sent to a control module that will be connected to the motors and the servo through the Teensy module. In this control module, a transformation will be done to a PWM signal in order to make the car turn what is intended to.

Concerning the emergency stop procedure, the *kill_wireless* will be the script directly connected to the control of the vehicle, that will wait for a specific socket message. The car will be running in the circuit at a velocity of 4.5 m/s and if an emergency message is detected, the vehicle will cut the energy going to the vehicles, making the car stop in the fastest way possible.

4.1.3 PID adjustments

For this project, the previous implementation was maintained but, it was optimized to perform a better following of the line. The giggles in a straight line and in the circuit were evident, and the car was running in a slow speed, for these reasons, an optimization needed to be done. Also, the car was not stable, many times when doing the line following, the car just got lost in some corners because of the oscillating behavior of the car in the circuit.

As said before, the line following algorithm is based on making several straight lines with the orientation and angle of the white line and finally making an average of those angles. But, to get a better control over time, a PID controller was implemented, given a better steering angle that the car needs to do. One of the things that are more time-consuming and difficult to do is optimize a PID controller because of the tests that are needed to do in order to see if the PID values are good or not. For this reason, many tests were done as a way to encounter the best values that could make the car do the circuit lap faster and without losing the line.

The previous PID values used were: $K_P = 3.3$, $K_I = 1.8$, $K_D = 1.2$ and they were resulting in an oscillating line follower. At first sight, it can be seen that the

K_I parameter is set too high, causing instability and oscillation to the car meaning that over time, the behavior of the car following the line will be worse, making one or two corners, but then it will just lose the line.

So, through a fine-tuning of these values and numerous tests done, the final values encountered were: $K_P = 3.6$, $K_I = 1.1$, $K_D = 1.7$. If the speed of the car is increased then, he needs to be more responsive and because of that, the integrative part of the PID controller needed to decrease, and the derivative needed to increase, with a little increase on the proportional part too. The changes done in these values, changed completely the following, with the straight line following being very good, without wobbling and in the circuit being very consistent. To maintain a good line following around the circuit and not letting the angle given by the PID have some extraordinary errors, a limitation was done to have a maximum of 30º in the vehicle turning angle and a minimum of -30º.

Evaluation of the improvement with respect to the previous implementation was done empirically, and no results are reported, as this is outside the scope of this thesis.

4.1.4 Camera Positioning

The two robotic test beds, initially, had a ZED Camera installed in front of the car, without something to held them and just held by some tape glued to the LiDAR. That, made the LiDAR inoperable and at the same time, made the ZED Camera less stable, because with the movement of the car, the camera wiggled.

Because of that, a bit of change to the test bed was necessary in order to have a decent line following and a decent structure. The instable camera, with the tests moved and wiggled, what sometimes made the line following inaccurate, with the car not being in the center of the line.

With this in mind, an acrylic platform was acquired and using some tools present in the CISTER lab, the camera place was changed and secured with nails in the platform, as it can be seen in figures 4.4 and 4.6.

One of the most important points was changing the horizontal position of the camera. As it is being used a ZED Stereo Camera, that has two cameras, just one could be used for the line following algorithm. The one that is used is the right camera and in the previous work, the right camera was centered with the car in order to make the line be under the center of the car. But, that is not really necessary if the center of the camera on the algorithm is changed to a value that makes the car stay with the white line under the middle of the car. That was one of the changes made to the line following algorithm.

4.1.5 Final Line following algorithm

With the changes completely made in terms of car structure, it became necessary to edit the algorithm and optimize it for the new car structure, because any difference in terms of position makes it necessary to change the line following parameters. In this way, in the following algorithm, it's presented the main steps of the script used.

Algorithm 1 Line Decision Script

```
Subscribes to the topic where the coordinates are published
Receives the coordinates and the average slope of the lines from the topic
Do vehicle_control()
Set $error\_steer = (x\_mean - camera\_center)/camera\_center)$
Do steering_control()
Get $\theta\_error = arcsin(error\_steer/dist\_corr)$
Calculation of steering_angle using PID
Publishes the angle and velocity in drive_param topic
```

The line decision algorithm is based in the processing of the parameters obtain from the white line detection, it uses the coordinates obtain from the lines_detection script and the average slope (x_mean) to calculate the angle that the car needs to turn, using the image pixels. When the coordinates are published in the topic, the function *vehicle_control()* is done, obtaining the coordinates and calculating the pixel's error from the line detected to the center of the camera (*camera_center*).

After that, it's done the transformation from the pixel's error to an θ_error in a form of an angle, doing the PID control afterward.

4.2 Platooning

To do a vehicle follow another, it's necessary to detect the leader car and do a steering control along the circuit. An object detection algorithm was used in order to detect the front car. YOLOv3 algorithm was used, and as the robotic test bed cannot be identified as a car (being quite irregular in its detections), a stop sign was used as an example for platooning. In the figure 4.8, it can be seen the output of the acquired image with the ZED Camera using object detection.

Using the bounding box information given by the object detection, the angle of the right corner, left corner and center of the boundary box was calculated using the equation 4.3.

$$Corner\_Angle = atan(\frac{dx}{f}) \tag{4.3}$$

This formula was used to estimate the angle of a point in the camera's field of view, given its horizontal displacement *"dx"* from the center of the image and the camera's focal length *"f"*.

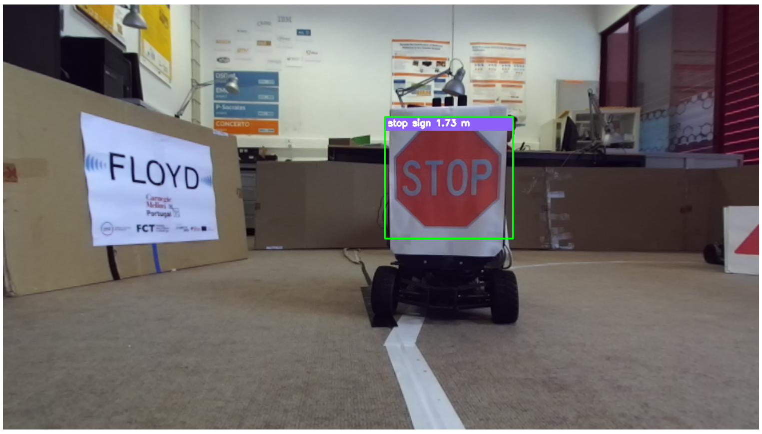


Figure 4.8: Stop Sign Detection

As LiDAR has a measurement radius of about 360º, this calculation was necessary in order to have the angles that the LiDAR needs to measure, erasing the angles that don't matter for this implementation and centering just the distances to the car. After having the distances from the left side to the right side of the bounding box, an average distance to the car, it's calculated and used to control the velocity that the follower car will run.

To have the angle that is necessary to follow the leader car, the angle to the center of the boundary box was used, being calculated with the equation 4.3 and with the displacement being from the center of the image to the center of the bounding box.

In figure 4.9 it can be seen a diagram that shows the relation of the sensors and the main contributions to this multi-sensor approach method.

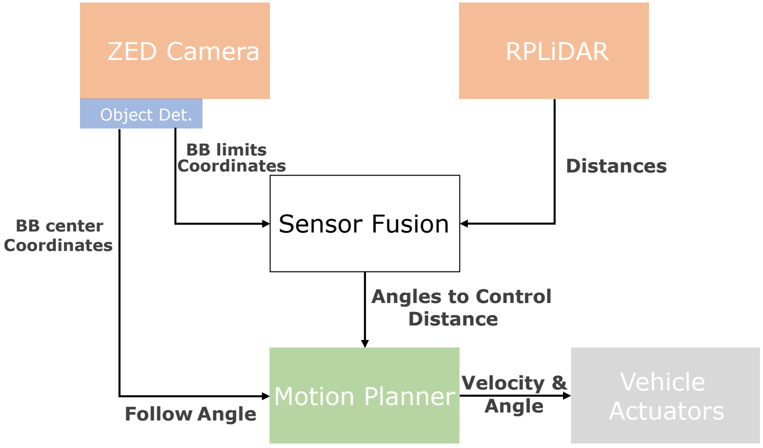


Figure 4.9: Diagram of multi-sensor approach and platooning contributions

4.2.1 PID controller

Coupled to this, a PID controller was implemented to the angle that the follower car needs to turn to follow the leader car, giving a more precise following method to the platooning implementation. Many tests were made in order to have the best PID values that would lead to a better following. The best values found and used were: $K_P = 2.5$, $K_I = 1.1$, $K_D = 1.6$. To maintain a good platooning and not letting the angle given by the PID have some extraordinary errors, a limitation was done to have a maximum of 30º in the vehicle turning angle and a minimum of -30º.

4.2.2 Following Algorithm

In order to get a better understanding of the stop sign follower algorithm done to get a successfully platooning, the algorithm 2 was done, with the more important parts of the code being referenced.

Algorithm 2 Stop Sign Following Script

```
Subscribes to the topic where the Stop Sign coordinates are published
Receives the coordinates and the average slope of the stop sign from the topic
Do vehicle_control() function
Set $dx = (x\_mean - camera\_center)$ and $f$
Do steering_control() function
Get $\theta\_error = atan(\frac{dx}{f})$ for steering control
Calculation of steering_angle using center of the Stop Sign
Calculation of bounding box corners angles
Do velocity control using distances between corner angles
Publishes the angle and velocity in drive_param topic
```

To have a good platooning execution, it is necessary to have a good concatenation between the various scripts used, with constant updating of values and connection between them. The figure 4.10 shows the scripts used for the execution of the platooning, with the information and links exchanged.

As it can be seen in figure 4.10, the RPLIDAR ROS script launched will initialize the RPLiDAR and provide the distances that will be used to control the velocity in the stop sign follower script. This script is the main script on the platooning implementation, doing the velocity control and the angle control to follow the front car. It will also receive the coordinates of the bounding box of the stop sign given by the YOLOv3 algorithm that will be used to do the angle control.

In the end, the talker script will be the script encharged of transforming the velocity values in PWM values, receiving the values from the topics initialized with the launched ROS serial script.

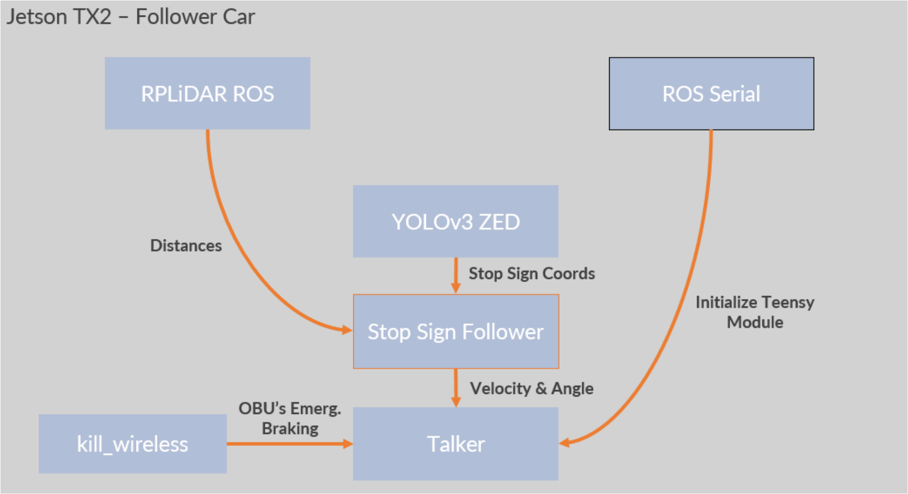


Figure 4.10: Stop Sign Following Control Architecture

4.2.3 Inter-vehicle communication

For the communication between the two vehicles, two OBU running the OPENC2X were used in order to transmit CAM and, when needed, the DENM messages. The OBUs are based in 802.11p and need to be connected to be able to receive and send messages over time. For the purpose of this work, the focus lies on DENM messages. Unlike CAM, which primarily provide general awareness information, DENMs serve the purpose of transmitting specific warnings and alerts about potential dangers on the road. These messages are crucial in ensuring timely and accurate dissemination of critical information to enhance safety for all road users. The OBUs are installed on top of the vehicles, and throughout the circuit they send several CAMs that provide the location of each of the vehicles to one another, the speed of the car and the time when the message was sent.

Regarding the DENM sending, The *kill_wireless* script will be responsible for waiting for the DENM request through the OpenC2X's HTTP server, that is running on port 1188, that can be sent by the Leader Car to the Follower car if a hazard is detected, triggering the emergency stop, slowing down the follower car until it stops.

4.3 Setup of 5G-capable cellular network

In this project, to get a 5G-capable communication vehicle, a module was used that consists of a Raspberry Pi 4B+ that adopts a 5G/4G/3G Raspberry Pi communication HAT with a SIMCOM 5G module SIM8200EA-M2. This module was non-operational and in order to use him, it was needed to flash the Raspberry Pi image from the beginning that has the drivers to use the 5G communication HAT.

5G Module

The SIM8200EA-M2 5G Raspberry Hat supports 5G NSA and SA networking and is designed to enable 5G connectivity on Raspberry Pi devices. This Raspberry Hat adopts the 5G module SIM8200EA-M2 and provides fast data transfer speeds, significantly enhancing the connectivity experience. The SIM8200EA-M2 supports multiple 5G frequency bands and offers flexibility for different regions and network providers. The data rate can be up to 2.4 Gbps (DL) / 500 Mbps (UL) and allows connecting with different 4G or 5G communication module with M2 package [57]. In figure 4.11 it can be seen the 5G module.

Figure 4.11: 5G Module

For this project, the 5G module will be in charge to give 5G-capable internet connection to the Leader car in order to communicate with the infrastructure. For this purpose, it's used a SIM card that will connect to the cellular network given by a network provider.

4.3.1 Raspberry Pi configuration and Ethernet Bridge

In order to get the cellular network in the module, a SIM card from a network operator was needed, an evaluation was also done to get the best network connection in the lab with a SIM Card from various network providers. The choice fell on a SIM card of WOO that uses the NOS cellular network and can give a 5G connection to the user.

After putting the SIM Card in the module, a set of configurations are needed in order to get a network connection. For this purpose, an application called "Minicom" was installed and using AT commands it's possible to configure the module to use the cellular network provided by the SIM Card. The configuration done to the module was:

- **AT+CPIN = PIN CODE** - Unlockes the SIM Card to enable cellular network;
- **AT+CFUN=1** - Turn the radio on (turn off airplane mode);
- **AT+CNMP=2** - Set network mode to auto-seek;
- **AT+COPS = 1,2,"28603"** - Manually define the access to the operator;
- **AT+CPSI?** - Check if the cellular network is online and with 5G.

Doing these configurations, the cellular network can now be used in the Raspberry Pi but, that is not the main goal. The main goal is to use the cellular network in the Jetson that is doing all the processing on the car. For this to happen, it's necessary to do a bridge between the Raspberry Pi and the Jetson. Knowing that the Raspberry will be connected to the Jetson via Ethernet cable, a bridge needed to be done in order to pass the cellular network in the Ethernet interface named *eth0.*

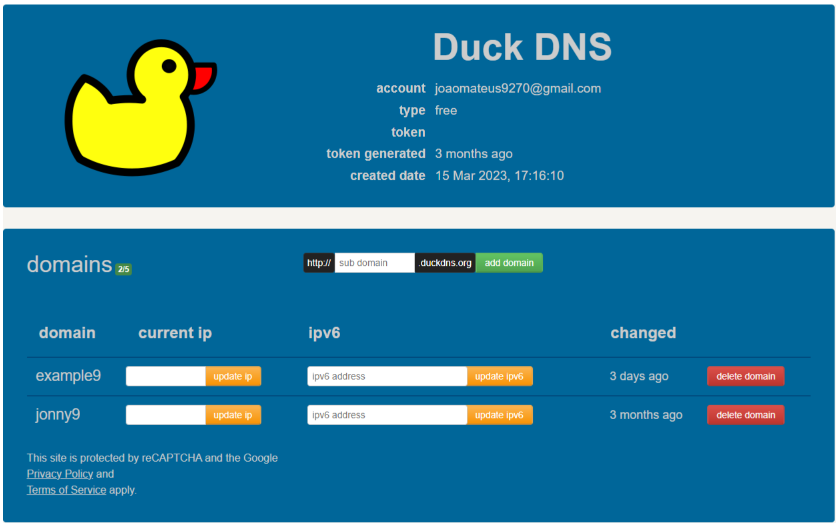
To configure the bridge, a couple of lines were added to the interface's configuration file *(/etc/network/interfaces)* to pass the cellular network to the Ethernet network interface, the lines added were:

```
auto br0
iface br0 inet dhcp
bridge_ports eth0 usb0
```

As it can be seen, a new interface was created named *br0* that does a bridge between the *eth0* and the *usb0.* The last one is the interface where the SIM8200EA-M2 module is connected, and it's the one that give the cellular network to the Raspberry and allows the exchange of mobile data. Doing this, the cellular network it's now available to the Jetson, but one more thing was still necessary, to establish routes in order to communicate correctly and access the Internet using the Ethernet network interface.

4.3.2 Vehicle to Edge Node Communication

As stated earlier, to do the communication between the edge node and the cars it's going to be used the public IP address of each one of the machines. Having this in mind, the public IP address was obtained from the network given by the 5G-capable module, but, from time to time, it could be seen that the public IP was changing, being a dynamic IP address. That is a problem that needed to be solved in order to have the communication between the car and the edge node, because the message needed to be sent to a specific IP address. For that reason, *DuckDNS* was installed and used, being a service that allows you to translate the dynamic IP address of your router, computer or other equipment. In figure 4.12 it can be seen the interface with the domains used by *DuckDNS* to translate the IP address, with the account settings erased in order to preserve the account and privacy.



Duck DNS

account joaomateus9270@gmail.com
type free
token
token generated 3 months ago
created date 15 Mar 2023, 17:16:10

domains 2/5

domain	current ip	ipv6	changed
example9		ipv6 address	3 days ago
jonny9		ipv6 address	3 months ago

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Figure 4.12: DuckDNS Domains and interface

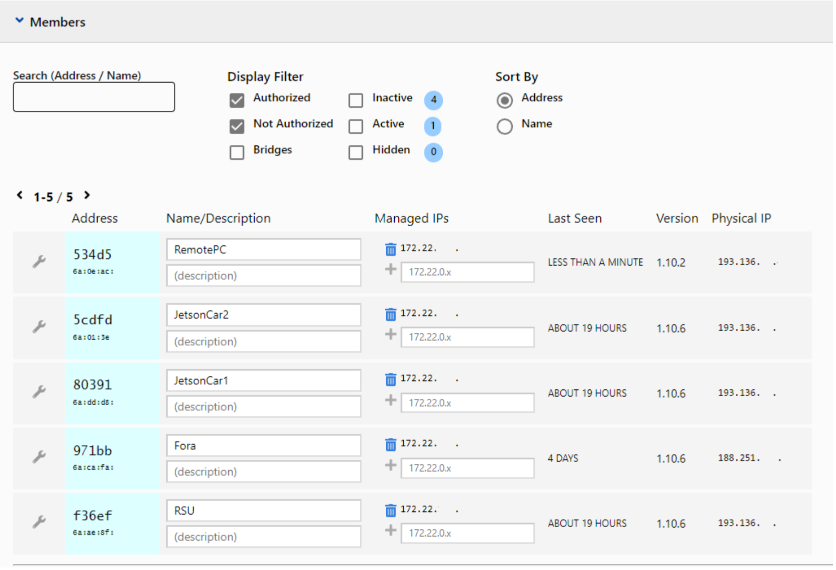
Even so, after doing all of this, a message still could not reach is target, it sent a message from the edge node, but it was not received in the car. That is because of the need of port forwarding between the CISTER router and the cellular network, which is not possible to do in this case since it needs to access the CISTER router to do that.

Keeping this in mind, the solution encountered was using a *Virtual Private Network* (VPN) to create a connection between the Edge Node and the Car Jetson's. The VPN used was the *ZeroTier VPN*, and it was installed to be able to create a tunnel from the Edge Node to Car's Jetson that uses the cellular network through the Ethernet cable. This VPN is easy to use and after making an account, it can be seen the network that all the components are connected, the public IP and the IP address that the VPN gives to the machines (Fig. 4.13).

After this, it is possible to send a message and ping from the edge node to the Car via the IP given by the *ZeroTier VPN* Network, giving a confirmation that the Car with the 5G-capable module has access to the cellular network and can communicate with the edge node using the public IP given by accessing the Internet.

4.4 Local Edge Node

After getting the communication between the car and the Edge Node under way, the application scenario that uses the Edge Node needs to be developed. In order to have awareness of its surroundings and respond automatically to events and passing road users, it was necessary to integrate an object detection system in the Edge



Search (Address / Name)

Display Filter
☑ Authorized
☑ Not Authorized
☐ Bridges
☐ Inactive 4
☐ Active 1
☐ Hidden 0

Sort By
● Address
○ Name

1-5 / 5

Address	Name/Description	Managed IPs	Last Seen	Version	Physical IP
534d5 6a:0e:ac:	RemotePC (description)	172.22. . 172.22.0.x	LESS THAN A MINUTE	1.10.2	193.136. .
5cdfd 6a:01:3e	JetsonCar2 (description)	172.22. . 172.22.0.x	ABOUT 19 HOURS	1.10.6	193.136. .
80391 6a:dd:d8:	JetsonCar1 (description)	172.22. . 172.22.0.x	ABOUT 19 HOURS	1.10.6	193.136. .
971bb 6a:ca:fa:	Fora (description)	172.22. . 172.22.0.x	4 DAYS	1.10.6	188.251. .
f36ef 6a:ae:8f:	RSU (description)	172.22. . 172.22.0.x	ABOUT 19 HOURS	1.10.6	193.136. .

Figure 4.13: ZeroTier VPN Interface

Node. This led again to using the YOLOv3 Object Detection algorithm and the implementation with the ZED Camera.

In the previous work [4], a Jetson NX was used in an Edge Node to process the images received by the YOLOv3 algorithm. But, in this case, a Jetson TX2 was used and not a Jetson NX. One of the perks of using this Jetson it's that the installation of the ZED SDK was already done, in which, includes the ZED Python API, which is important because it is the language of choice of this project and the script that comes with it already does object detection and just needs to be transformed to the application of this project.

YOLOv3 Setup

The supplied Jetson came installed with Ubuntu 16.04 LTS and CUDA 10.2, doing the detections with GPU acceleration. The script that does object detection using YOLOv3 it's in the darkned package of the ZED python API and needs to be compiled in order to be used, being necessary to do some modifications in the configuration of the Makefile:

In order to leverage GPU acceleration with CUDA and cuDNN, the GPU and CUDNN options were activated. This enables faster processing of computations. OpenCV, enables real-time object detection on live video feeds captured from the cameras. Additionally, the libso option is used to compile and build the essential darknet library, *libdarknet.so*, which is necessary for the subsequent Python implementation. The darknet library provides the foundation for implementing advanced deep learning algorithms and models for object detection tasks.

```
GPU=1
CUDNN=1
CUDNN_HALF=0
OPENCV=1
AVX=0
OPENMP=0
LIBSO=1
ZED_CAMERA=1
ZED_CAMERA_v2_8=0
```

After getting this part done, the YOLOv3 object detection algorithm is now compiled and can be utilized running the *darkned_zed_modified.py* script. To run it, the following command was done:

```
OPENBLAS_CORETYPE=V8 python darknet_zed_modified.py -c cfg/yolov3.cfg
-w yolov3.weights -m cfg/coco.data -t 0.3
```

Running this command, the weights, and the configurations of the model can be changed in order to use whatever YOLOv3 weights the user want and get the best detection possible.

Distance calculation

In this manner, the transformation of the *darkned_zed_modified.py* script was done, in which the script waits for a stop-sign detection at a distance of 1.4 meters. Using the ZED Depth Camera and the functions of the ZED SDK, the distance to the stop-sign can be calculated with great precision, with the *GetObjectDepth()* function getting the coordinates of the object. After that, a calculation is done in order to get the true distance to the object detected. That calculation is:

$$distance = \sqrt{x^2 + y^2 + z^2} \tag{4.4}$$

As previously stated, a stop-sign was used to detect the approach of the leader car because of the irregular and unstable detection given to the car by the object detection algorithm, that did not see the car has a car in a regular basis, and it labeled the car with different types of objects in the course of time.

In figure 4.14 it can be seen the precise detection of the stop sign and the distance to it close to the bounding box.

Message Sending

Taking into account the application scenario, a message needs to be sent via network to the Leader Car after the detection of the stop-sign at a distance of 1.4 meters. As

Figure 4.14: Stop-Sign Detection by the Road-Side Camera

said before, the public IP address it's used to connect the two components. Because of this, the *ZeroTier VPN* was also installed in the Edge Node in order to make a tunnel to the Leader Car.

After knowing the given VPN IP address, a socket server and client was set up, with the server being the Leader Car and the client being the Edge Node that will send the message to stop. All of this is done after the detection of the stop-sign, with a message being sent via socket TCP/IP.

In the algorithm 3, it can be seen the most important steps done in the Edge Node script to detect the car approaching.

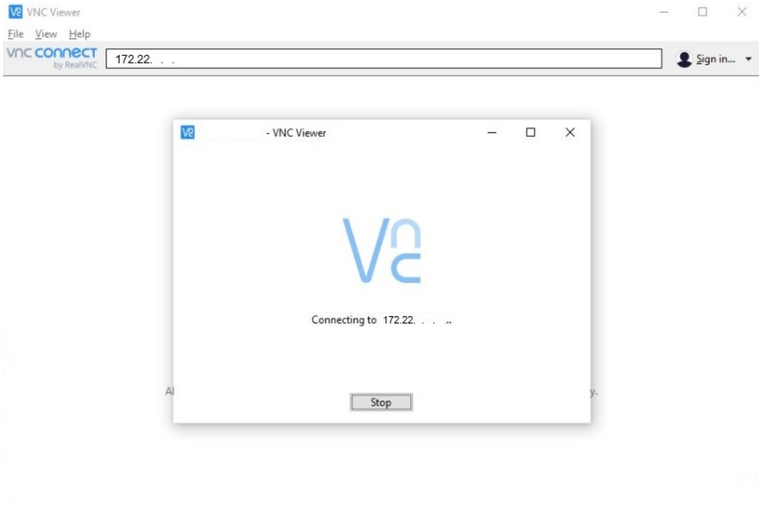
Algorithm 3 Stop Sign Detection edge node

```
Set socket Host and Port and make connection
Configure weights and configs of the model
Configure Camera quality and modes
Get frame from the video and do the detection
Get label and bounding box
Get the object depth coordinates
Calculation of the distance using the coordinates
if label == stop sign and distance < 1.4 then
    Send socket message to stop to the Leader Car via network
    Measure latency
    Get the actual time
end if
Retrieve the frame with the detections and distance
```

4.5 Teleoperation

4.5.1 Via VNC with keyboard

VNC is a graphical desktop sharing system that uses the Remote Frame Buffer protocol to remotely control another computer. This way of getting the control of a remote computer can be used in order to see what is being done in the processing part of the car, the Jetson TX2. This control, it's similar to accessing another computer using an SSH technique, but it operates with the advantage of getting the graphical display of another machine. In figure 4.15 it can be seen the VNC Viewer app used to get the remote control of the processing unit of the car.



172.22. . .

Connecting to 172.22. . . ..

Figure 4.15: Connecting to Leader Car via VNC

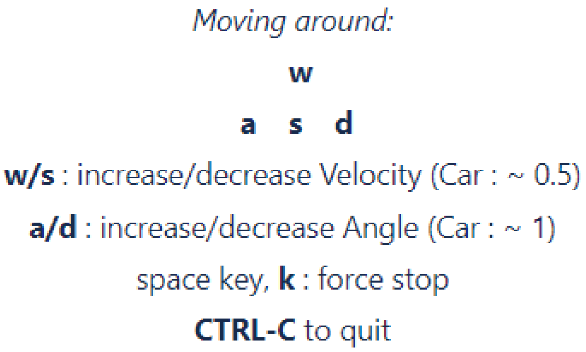
For this purpose, it uses the IP given by the VPN to connect remotely and insert key via keyboard using a script developed. After being connected, it needs to be running the script used to do the teleoperation, based in the Turtlebot teleoperation project. This script needs to get access to the drive parameters, being a publisher of this topic. One of the main changes include the keyboard keys. In order to get all the possible movements that the car could do, some keys needed to be added, like reducing the speed of the vehicle. Also, some tests were done, to see what incremental value would be the best to control the car in a better way. The increases and decreases of velocity and angle are done by steps, that when one key is pressed, the difference will be 0,5 m/s if changing the velocity and will be 1° if changing the steering angle.

4.5.2 Via Network

Teleoperation of cars via network is an emerging field that offers exciting possibilities in remote control and operation of vehicles. By leveraging network connectivity and advanced technologies, it becomes feasible to control a car remotely, enabling various applications such as autonomous vehicle testing, remote monitoring, and intervention in hazardous environments. Teleoperation involves transmitting control commands from a remote location to a vehicle over a network, allowing operators to manipulate the car's movement, steering, acceleration, and braking.

Having this in mind, for this project, a teleoperation technique was also done in order to prevent possible future accidents, as well as control a vehicle in a situation where an autonomous car cannot operate. The teleoperation is done via network, using the 5G module of the Leader Car, getting to see the possibility and reliability of using this in the future in multiple scenarios.

This implementation uses the socket server and client to send the commands to accelerate, brake and do the turns. In this case, the socket server is the Leader Car and the client is the edge node sending the keyboard key's as commands to control the vehicle. The keyboard keys responsible for controlling the car are:



Moving around:
w
a s d
w/s : increase/decrease Velocity (Car : ~ 0.5)
a/d : increase/decrease Angle (Car : ~ 1)
space key, **k** : force stop
CTRL-C to quit

For this implementation is also used the *ZeroTier* VPN, that enables the communication between the two components using the IP address given by the VPN network.

To understand better the Leader Car script responsible by getting the keys via network and changing the velocity and/or steering angle of the car, the algorithm 4 was made.

Algorithm 4 Teleoperation Leader Car Script

```
Set socket Host and Port and make connection
Waits for a key to be pressed
if  key == w then
    Increase velocity by 0.5
    if velocity > MAX then velocity = MAX
    end if
end if
if key == s then
    Decrease velocity by 0.5
    if velocity < MIN then velocity = MIN
    end if
end if
if key == a then
    Increase angle by 1
    if angle > MAX then angle = MAX
    end if
end if
if key == d then
    Decrease angle by 1
    if angle < MIN then angle = MIN
    end if
end if
if key == k then
    Emergency Stop
end if
Publish velocity and angle drive_params
```

Chapter 5

Results

The operation of the IEBA developed is verified, and its performance is evaluated. An essential element of vehicular safety applications is their capacity to provide response times aligned with the requirements of the application. For this reason, the time interval between each component of the system will be analyzed and will be the principal parameter to take into account on this application scenario. Furthermore, more parameters will be analyzed, like the distance to stop the car after getting the emergency message and the distance between the platoon during the course.

Finally, the publication contribution is also explained, showing the contributions of this work to the publication and the main architecture of the integration.

5.1 Applications

5.1.1 IEBA

In this section will be assessed the principal parameters of the IEBA scenario that can influence the final result of this safety-critical application, like the time and latency on the various steps of all the system. Also, the braking distance will be evaluated in a way that the influence of getting the message faster or not can be observed, being an important parameter on road safety systems. In Figure 5.1, we can see the experimental setup that was implemented for this application.

The Leader vehicle has the 5G-capable module on top and the OBU in the middle of the car, and the follower car has just the OBU on top. Initially, the front vehicle

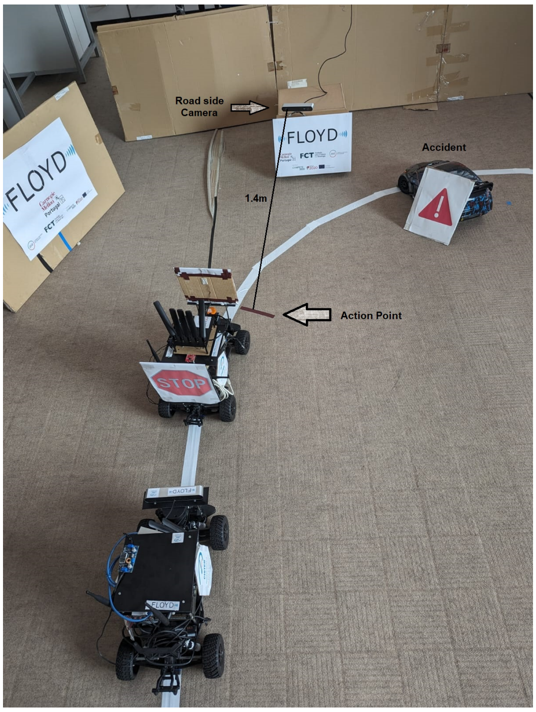


Figure 5.1: Experimental Setup

it's following the white line and, the back or follower vehicle, it's following the leader in a real platooning scenario. After arriving to the Action Point (that is the zone at a distance of 1.4 meters that when the roadside camera detects the car with the stop sign, it will send the emergency stop message via network), the car will engage the emergency brakes in order to stop in the fastest way possible. Following that, the OBU of the leader car will send a DENM to the follower car, making him do an emergency braking too. For this application, a circuit was done that will see the car lapping the circuit and then, when it gets to the Action Point, will engage the application. This circuit can be seen in Figure 5.2.

Before assessing all the inherent parameters to the IEBA scenario, a description of how the parameters will be evaluated is necessary. Having this in mind, the various steps are in Figure 5.3 that allows to understand the multiple factors that can influence the good function of this application.

As it can be seen in Figure 5.3, the scenario will be evaluated in five time steps that are:

- **Step 1** - Do the car detection by the Edge Node with the help of the Road-side Camera, it will include the YOLO processing time;
- **Step 2** - Sending the message to stop from the Edge Node and arrive at the Leader Car via network;
- **Step 3** - Stopping of the Leader Car;

Figure 5.2: Circuit used for evaluation

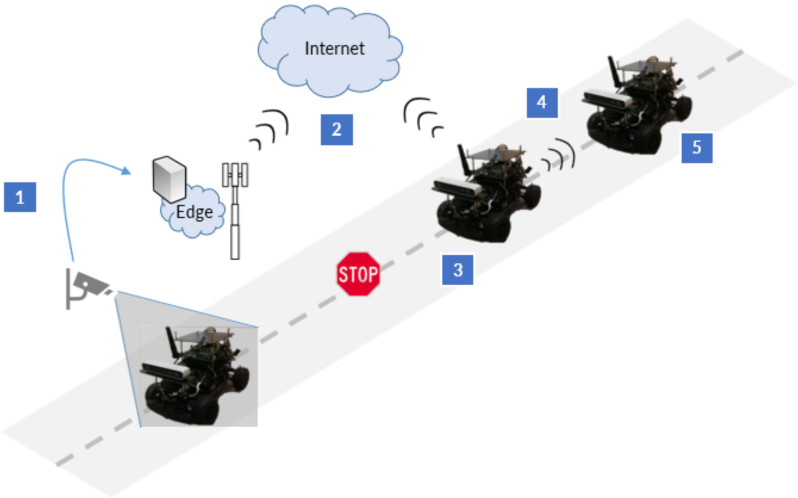


Figure 5.3: Evaluation Steps of the Latency

- **Step 4** - Sending the DENM from Leader Car to Follower Car;
- **Step 5** - Stopping of the Follower Car.

5.1.2 Teleoperation

In addition to the IEBA, a teleoperation application was also carried out. For this application, the most important metric is the time that it needs to execute the command that the user in the Edge Node ordered. This latency needs to be small to prevent too late orders that can lead to accidents and will depend on the state of the network. For this application, the Edge Node was used to connected to the

CISTER network and the Leader Car with the 5G-capable module connected to the cellular network.

To do this application, the Leader Car has a Socket server running that is listening and expecting commands from the client that is connected to him via network. The client will be the Edge Node and, it will be asking for a key to send to the server. After receiving the key, the server script will analyze what was the command sent and will execute the action.

The evaluated measure will be the time from when the key was sent to the moment that the key was received, with the execution of the action being included. With this, the end-to-end application can be measured.

5.2 Experimental Details

Some changes need to be done to the Edge Node, the Cars and the OBUs in order to get accurate results. To erase the time discrepancy, a protocol was used that syncs the clocks of the machines in a server/client way. The server will be responsible for giving the time to the other clients and them just need to be connected to him. The protocol used was NTP and it gives the possibility to reliably collect timestamps.

To get the time intervals between the multiple components, the cars were synchronized to the Edge Node clock using the VPN network IP connection, getting in this case a local clock synchronization. After installing NTP in all the machines and components of the system, a configuration file will be created named *ntp.conf*. In that file, it was specified the server that the machines need to synchronized and after some time, the clocks just sync and the experimental tests can be done.

In figure 5.4, the connections via NTP are described:

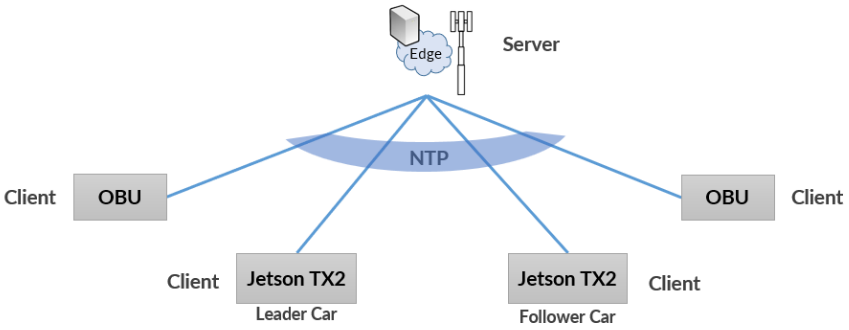


Figure 5.4: NTP Server and Client connections

5.2.1 LiDAR calibration

Having the LiDAR has one of the principal components of the system, with the platooning control being done using his distances, an evaluation was done in order to understand his reliability. The LiDAR used was the RPLiDAR A1, that is a low cost laser scanner and for that reason can be a little untrustworthy in his measures. With this in mind, a test was done with some known distances, measured with a metric tape and, using the LiDAR, these distances will be compared with the distances that the RPLiDAR gives. In table 5.1, it's the numerous measures taken with the LiDAR for a static distance measured with metric tape.

Real Distance	**LiDAR Measurements**	**Units**
30	30.9	cm
40	41.4	cm
50	51.8	cm
60	61.9	cm
70	71.6	cm
80	81.1	cm
90	91.9	cm
100	101	cm
110	112	cm

Mean Error	1.511 cm

Table 5.1: Comparison between real distances and LiDAR measurements

As it can be seen in table 5.1, some error it's associated to the LiDAR measurements, having the trend to increase when the distance is greater. For these measurements, the mean error was 1.511 cm. This error can be because of some error associated to the multi-sensor approach implemented and, at the same time, the uncertainty associated to the LiDAR. The multi-sensor approach of vehicle following uses a technique that can be error-prone with some deviations and, the fact that is done a mean between the values of the right side and the left side of the bounding box, some deviations can also occur and influence these measures.

These errors did not influence the final implementation, being able to do the platooning scenario with the Emergency Braking well, but this associated error in a real application scenario can be problematic and can be the difference between an early braking and an accident.

5.3 Latency Measurements

5.3.1 Network Latency

One way to evaluate the network between the two components of the application can be checking the Round Trip Time (RTT) latency between them using *pings* and then cut that time by half, in order to understand how the network can vary just in one trip time in the numerous positions of the circuit. With this in mind, the Figure

5.5 demonstrates what is the trend of the latency in this communication using 355 *ping* values between the Edge Node and the Leader Car.

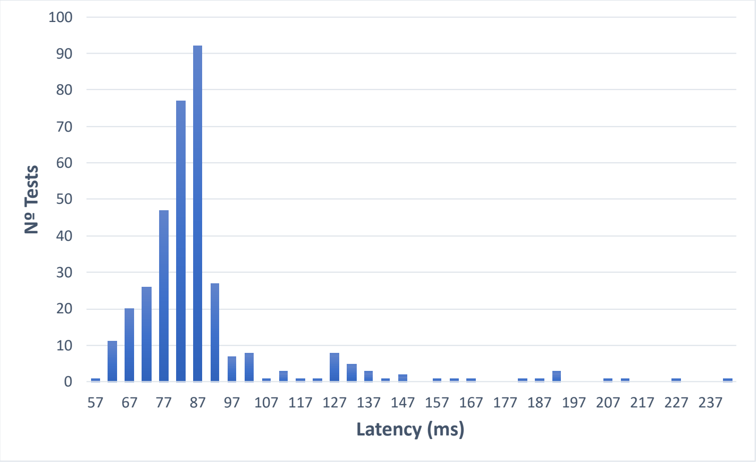


Figure 5.5: Trend of latency from Edge Node to Leader Car

With the help of the figure 5.5, it can be seen the trend of the latency, with the highest frequency of measurements between 67 and 97 ms. In addition to this, a ECDF was also done to better evaluate all the tests done, as it can be seen in Figure 5.6.

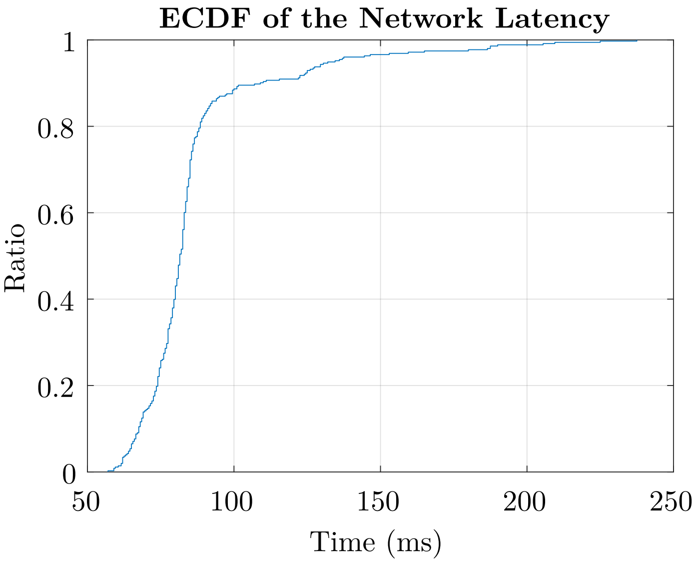


Figure 5.6: Network Latency from Edge Node to Leader Car

As it can be seen in Figure 5.6, the network latency have almost 90% of his values below 100 ms, showing the tendency as it can be seen in Figure 5.5 too. With this network latency, a good emergency braking can be obtained, with a fast response time.

5.3.2 Message Transmission Latency over the 5G Network

IEBA

Having a multi technology application can be a problem because of the latency associated to the numerous communications. With this in mind, first, the 5G-enabled communication between the Edge Node and the Leader Car was evaluated.

For the emergency braking message evaluation, 10 runs were done in order to have a good quantity of results to discuss and verify if using a commercial cellular network can be something to look to in the future. These measurements were made using the NTP between the Edge Node and the Leader Car, saving the time when sending the message and when receiving the message to stop. In order to have a better representation of these results, a ECDF (Fig. 5.7) was done with the time taken to receive the message in the leader car.

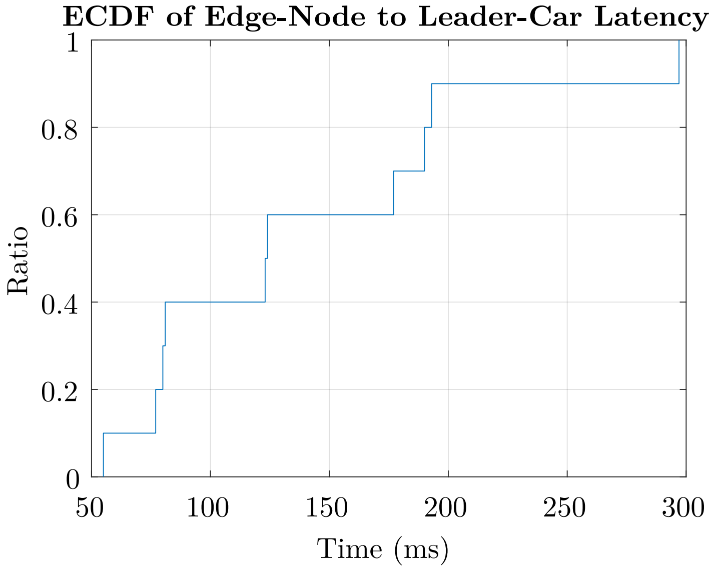


Figure 5.7: Time taken to receive the message in the Leader Car

As it can be seen in Fig. 5.7, for the time that the Leader Car takes to receive the message from the Edge Node almost 90% of the times are less than 200 ms, with the maximum value being 297 ms, that made the car stop much later than the expected, with that being observed in the implementation.

Comparing this with Figure 5.5 that gets the Round Trip Time cut by half from the Edge Node to the Leader Car, the time measures of the Figure 5.7 are between the intervals analyzed in 5.5, with the majority of the values in the 50 ms and 150 ms just for a one-way message. The histogram of Figure 5.8 can prove this point, with the trend of the delay measures.

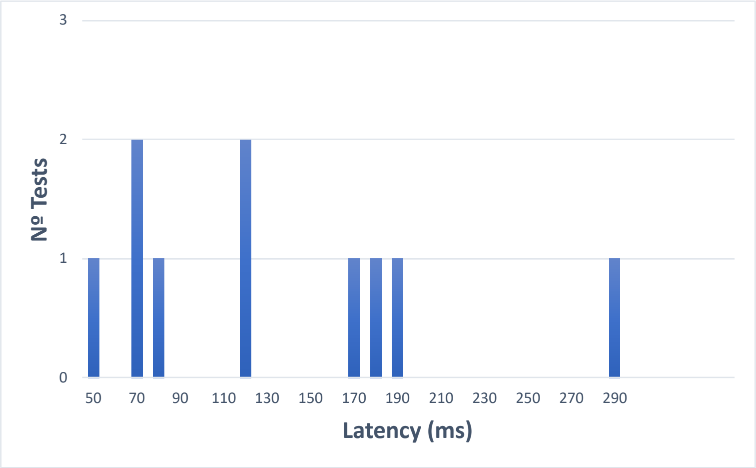


Figure 5.8: Trend of delay to receive message in the Leader Car

Teleoperation

For this application, the latency was measured when sending various commands to the Car and viewing how the car was behaving, getting the times when the car executed the ordered command. In this way, it could be evaluated the viability of using the teleoperation via network when needed. In the figure 5.9, is an Empirical Cumulative Distribution Function (ECDF) that represents the total time between the sent key and the execution of the command.

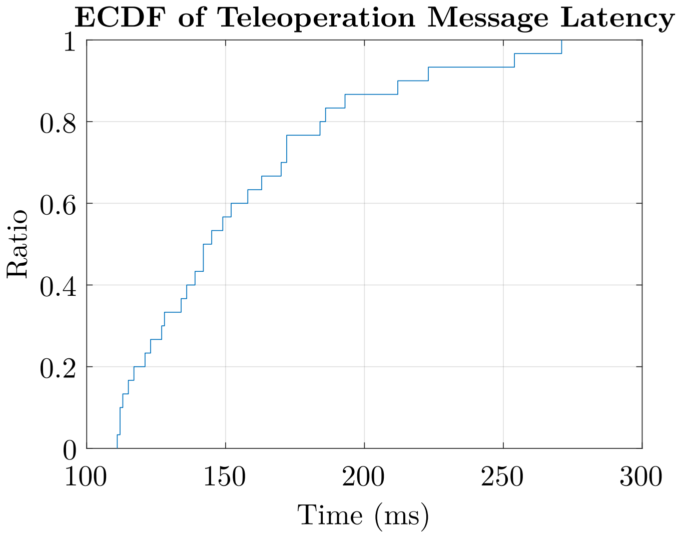


Figure 5.9: Teleoperation Latency ECDF

As it can be seen in the figure 5.9, all the measurements obtained have values above 100 ms, with half of the 30 tests done having values below 150 ms and 80% of the values being below 200 ms. Two measurements had a value above 250 ms, which may have to do with the often instable network cellular signal quality experienced at CISTER.

In the histogram of Figure 5.10, it can be seen the trend of the latency of the teleoperation, that confirms the majority of times being between 120 and 200 ms.

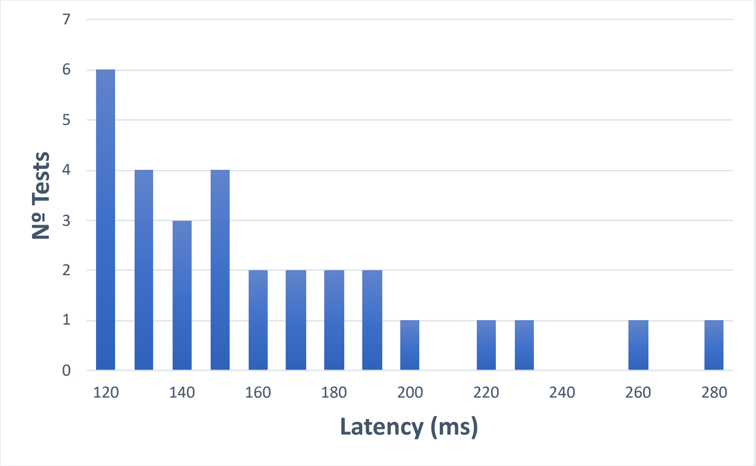


Figure 5.10: Trend of delay to receive Teleop. message in the Car

To understand better the influence of the latency in the teleoperation application, the behavior of the car was analyzed at different velocities doing the turns with a white line below the car. Using this method, the influence of the latency was observed.

At a velocity below 5.0 m/s, the car was doing the curves and the white line was always below the car, but with velocities superior to that, the car sometimes was taking too long to change the angle to turn, getting a bigger offset to the white line, being a little unstable.

5.3.3 Message Transmission time of Cooperative Platooning

For evaluating the Cooperative Platooning, the times that the Follower Car has taken to receive the order to stop the vehicle from the Leader Car were measured. For this evaluation, were made 10 tests in order to verify what was the latency to receive the message to stop via OBUs.

In the Figure 5.11 it's showed an ECDF of the time that the Follower Car takes to receive the order to stop after the DENM was sent by the Leader Car.

These results were obtained using the NTP between all the devices (OBUs and Jetsons), being all connected to just one server, synchronizing the time between them. As it can be observed in 5.11, the time that it takes from sending the DENM from the Leader Car to engage the brakes in the Follower Car it's always inferior to 100 ms, with 80% of the times being inferior to 80 ms.

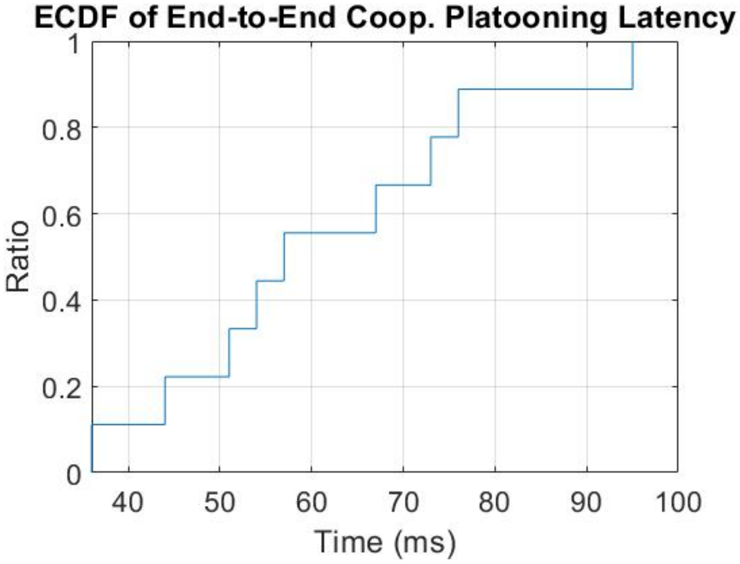


Figure 5.11: Time taken from sending DENM to receive in the follower

With these values, it can be observed that having a cooperative platooning it's a plus in vehicular platooning applications, with a more responsive and secure environment.

5.3.4 Application End-to-End Latency

To do a good scenario evaluation, numerous runs were done to test all the implementation. In this way, the values can be compared, and some unexpected interferences can be evaluated too. Knowing that in safety-critical scenarios the time and latency are the most important parameters and the minimal interference can be fatal.

Having this in mind, in table 5.2 were registered the times of the numerous steps, giving a more precise view of what to expect from a multi technology implementation using a 5G-enabled module and OBUs with 802.11p to do communication.

Steps	**Run #1**	**Run #2**	**Run #3**	**Run #4**	**Run #5**	**Units**
1 - Action Point Detection	52	48.9	55.9	52.3	55.6	ms
2 - Sending message via network	124	193	55	77	80	ms
3 - Stopping Leader	0.42	0.321	0.424	0.307	0.397	ms
4 - Sending DENM to Follower	76	54	51	44	62	ms
5 - Stopping the Follower	0.301	0.361	0.407	0.327	0.299	ms
Total Delay	252.721	296.582	162.731	173.934	198.296	ms

Steps	**Run #6**	**Run #7**	**Run #8**	**Run #9**	**Run #10**	**Units**
1 – Action Point Detection	52.8	54	50.7	57.2	51	ms
2 – Sending message via network	190	297	81	177	123	ms
3 – Stopping Leader	0.485	0.33	0.401	0.332	0.494	ms
4 – Sending DENM to Follower	56	95	67	73	57	ms
5 – Stopping the Follower	0.503	0.459	0.306	0.335	0.329	ms
Total Delay	299.788	446.789	199.407	307.867	231.823	ms

Table 5.2: Time taken in each step of the scenario

Analyzing the results of these tables, it can be observed that the step 2 is the process that takes more time to execute because of the latency between the Edge Node and the Leader Car. That latency is bigger because it is connected via network, having many details that could make the latency increase.

Concerning the total delay, the runs have an average total delay of 256,994 ms, with some runs getting a big delay because of the low network connection on the sending message via network. Having this in mind, a histogram (Fig. 5.12) was done in order to better understand the total delay of the emergency braking application with cooperative platooning.

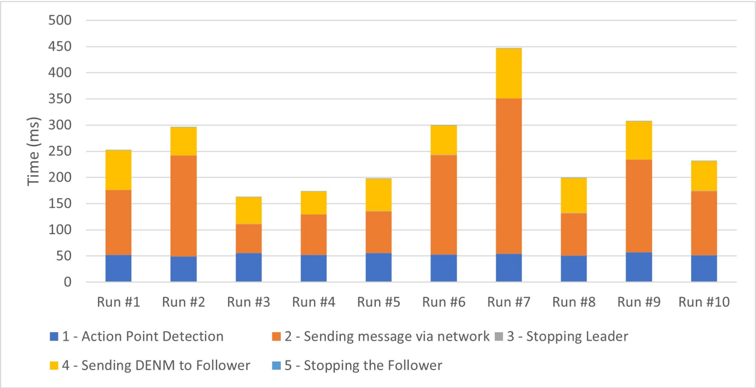


Figure 5.12: Total Delay of the scenario

As it can be seen in the histogram of Fig. 5.12, the end-to-end times are presented for the 10 runs, most of the runs being below a total delay of 300 ms, which can be seen as a good sign for a multi technology application. Another detail noted is the fact that most of the delay comes from sending the message via network, as it was expected.

Another way of analyzing the total delay of the end-to-end application, is using a ECDF to explore the various tests done. In the Figure 5.13, it can be seen the ECDF of the end-to-end application.

As it can be seen in Figure 5.13 and confirming what has been said early, 80% of the total delay times are below 300 ms, with some tests being more dispersed to a maximum delay of 445 ms.

In this implementation, these values are very responsive for an application made via network and with the multiple challenges existent in multi technology implementations, having a need of everything being synchronized in order to see good results.

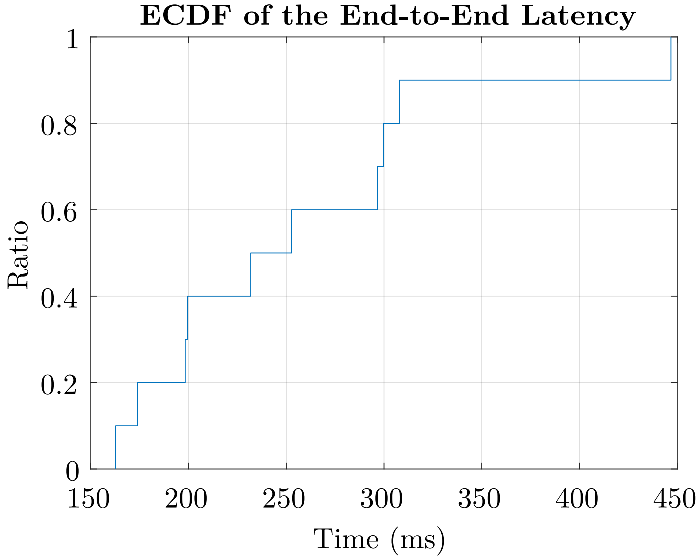


Figure 5.13: Total Delay ECDF

5.4 Following and Braking distance

5.4.1 Following Distance

In order to evaluate the behavior of the platooning throughout the scenario, the distance of the LiDAR that the Follower car uses to adapt the velocity between the Leader and the Follower was saved. With this, it can be seen the distance values oscillating as the Follower gets closer and then moves away of the Leader car. The representation of this behavior can be seen in figure 5.14.

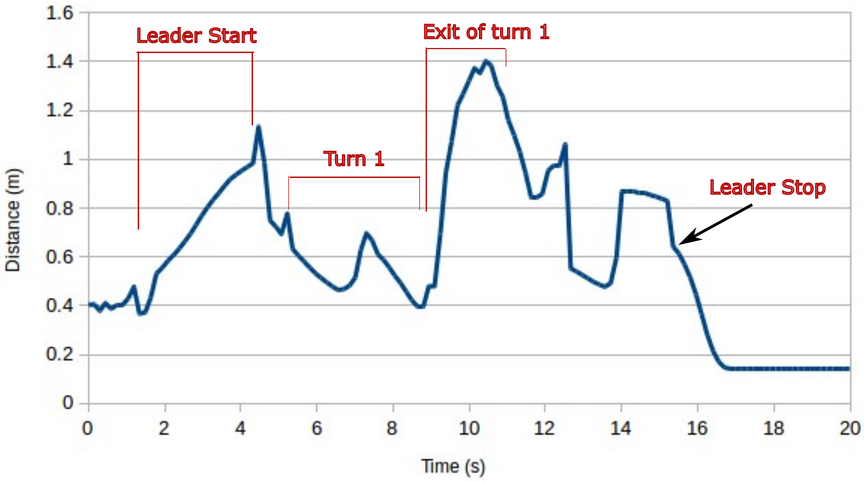


Figure 5.14: Distance from Follower Car to Leader Car

As it can be seen in figure 5.14, initially the cars are separated by a distance of 0.4 meters and when the application starts, the front car starts to move away of the follower car. That part could be seen in the 2 seconds mark of the graph, starting then to get closer after 2 seconds, when the car is almost at 1.2 meters of the Leader Car.

Arriving to a curve, the Leader car starts turning and, because of doing the curve of the wheels with the same PWM, the velocity decreases, and the follower car needs to decrease his velocity too in order to maintain a good distance to the front car. After this is when the following gets tricky. When the follower car decreases his velocity, the leader car starts to move away faster and the distance increases rapidly, making the follower car increase is speed too in a more spontaneous way.

The curve, as it can be seen in figure 5.2, it's a long one and after some back and fourths of distance and velocity, it arrives to a more stable part at 13 seconds, when the car arrives to a straight. After that straight line, the emergency brakes will engage to stop the vehicle, and it's when, closer to the 15 seconds mark, the distance between the vehicles will decrease to a minimal value of 0.2 meters, because the Leader Car stops, and the Follower car will get closer and stop reactively at a distance of 0.4 meters.

5.4.2 Braking Distance Leader

One of the most important metrics in a vehicular emergency braking application scenario, it's the braking distance. For this reason, the braking distance was measured, with the Action Point being the point where the brakes should be activated. These measurements were done with a metric tape, from the Action Point to the zone where the Leader Car stopped.

As is well known, the braking distances measured in these tests are not comparable to a real scenario with a real vehicle, but it's interesting to see how the latency between the edge node and the leader car will affect the distance to stop from the vehicle. Then, in table 5.3 it can be observed the 10 different runs done and the braking distance of the leader car in each one of the tests.

As it can be seen in 5.3, some variation can be observed, with some measures being with a difference of 10 cm. This can happen because of the latency between the Edge Node and the Car, as well as because of the detection time and the time that it takes to engage the brakes.

With this in mind, it has been added the sum of the time, with the detection time, the time between that takes the message to be received in the Leader Car and the time that takes the car to engage the brakes.

These two measurements are deeply related, because if the sum of the time is greater, the braking distance will be greater too. That can be seen in the Runs #7 and #9, where the fact that the times are so much bigger than the others makes the braking distance be a lot bigger too, with the 10 cm deviation.

Runs	Distance to Stop	Time to Stop Leader
Run #1	39,5	176,42
Run #2	37,2	242,221
Run #3	37,1	111,324
Run #4	29,1	129,607
Run #5	33,7	135,997
Run #6	29,3	243,285
Run #7	43,6	351,33
Run #8	33,5	132,101
Run #9	44,4	234,532
Run #10	35,6	174,494
Average	36,3	193,1311
Units	cm	ms

Table 5.3: Braking Distance – Leader Car

5.4.3 Braking Distance of Connected Platoon

For the emergency braking application scenario, besides the follower vehicle stopping reactively, it was used OBUs with 802.11p equipped with Open-C2X communication. In this manner, exists a cooperative platooning scenario that can be tested and compared with the stopping in a reactive form.

The best way to compare the emergency braking reactively and using the DENM sent by the OBU it's by measuring the distance between the Follower Car and the Leader Car when the two cars stop. With this in mind, the table 5.4 gives the distances measured when the Follower Car stops reactively and using the OBUs. This distance was measured using the distance given by the LiDAR of the Follower Car, and the uncertainty of these measures were evaluated in the section 5.2.1.

Runs	Distance Leader-Follower React.	Distance Leader-Follower w/OBUs
#1	42.5	83.7
#2	43.4	79.5
#3	52.9	87.1
#4	36.7	81.6
#5	45.4	76.2
#6	48.3	85.3
#7	42.6	72.3
#8	46.7	80.6
#9	43.4	78.4
#10	45.7	77.2
Average	44.76	80.19
Units	cm	cm

Table 5.4: Distance Comparison Reactively and using OBUs

As it can be seen in table 5.4, the distances from the Follower Car to the Leader Car have a big deviation in the two applications. This happens because with the OBUs, the message to stop goes to the Follower Car in a very fast way, while stopping reactively it's different and will just engage the brakes when at a distance of 55 cm. The average distance, it's very different too, being almost the double from the application using the OBUs.

5.5 Results discussion

Overall, the designed system proved to be a reliable and effective solution to an emergency braking scenario, getting consistent results that can be analyzed and discussed the viability of using a multi technology communication. The main objective was accomplished, being done a 5G-capable emergency braking with a platoon [1].

In these tests, many challenges have been raised, with the main problem being the low network caught in CISTER lab that made the 5G-capable module not being able to caught 5G network and just 4G LTE in the best of times. This can affect all the results gathered in this project, because, it will increase the latency and the delays. If it could catch 5G with full network, the delay and latency times would be much better. Furthermore, concerning the robotic test bed, some challenges were also raised, with some problems in the motors and the CISTER floor, with the floor slightly raised that sometimes made the test bed have a strange and different behavior.

Regarding the tests themselves, the emergency braking procedures were done with a relatively low delay, having some inconsistency because of the network caught by the 5G-capable module. Anyway, the total delay time was always under 450 ms, knowing that communications between 4 components needed to be done, it's a good result, having a responsive emergency braking application scenario [2].

However, using a robotic test bed is not like using a real car in a real scenario with factors that can affect all this system. One of the factors is the traffic in the network, if the network is a bit jammed, the latency will be higher and that can cause problems in the scenario. Having this in mind, the conditions under which the tests were carried out are good to understand the behavior in an emergency braking application when the network is not very good.

Regarding the braking distance, the test bed used cannot represent a possible braking distance of a real vehicle, because it is a scale version with reduced mass and its procedure of slowing down is very different to a full-size vehicle weighing 1.5-2 tons, but, it's interesting to measure and compare the braking distance with the latency, to see how the latency can affect the braking procedure.

---

[1] https://youtu.be/ne6JBE-fNnc

[2] https://youtu.be/EWeijo7dQOQ

5.6 Collaborative Publication

The work and foundations of this project culminated in the realization of a publication for the FLOYD project. This publication involved several elements of CISTER and Capgemini.

The main objective of this publication was to perform the same emergency braking application using Capgemini's private 5G network and edge computing infrastructure. This Edge Node hosts a *Docker* container receiving a video stream of a Road-Side Camera and uses an object detection algorithm to detect the approaching of the car to an area of interest. For this purpose, a *Docker* container was created with the object detection software and with the possibility of receiving a video stream via RTSP. When it detects the car in the area of interest, a message is sent via 5G network to stop the vehicle.

The contributions to this publication were all the vehicle related implementations, the platooning, the line follower and the emergency braking procedure. In addition, it was needed to establish the communication between the car and the Edge Node, that in this case was different from the one described in this thesis. To connect to the 5G Private Network at Capgemini headquarters, their SIM Card was used, and a new configuration was done to make the integration possible.

The main changes from the application done in this thesis were:

- **The Edge Node -** It was used an Edge Node running a *Docker* container with object detection and receiving the video stream from the Road-Side Camera to send the message.
- **Communication between Car and Edge Node -** The VPN described in section 4.3.2 did not have to be used. The vehicle now hosted a client, whereas the edge container hosted the server. It was used a 5G Network provided by Capgemini and their SIM Card that was connected to the Edge Node where the container was running.

In order to better understand the architecture used in this publication, a diagram was done, and it can be seen in Figure 5.15.

As it can be seen in Figure 5.15, the main changes to the 3.7 diagram are in the infrastructure part, being used all the Edge Computing Infrastructure of Capgemini that has access to the 5G Private Network and can communicate with the 5G module. The Road Side Camera is just streaming the video to the Docker Container hosted in the Edge Node, with him being responsible to trigger the emergency braking procedure.

With all the integration complete, a publication [58] was then done in order to finalize the FLOYD project, with a video being also made to better explain

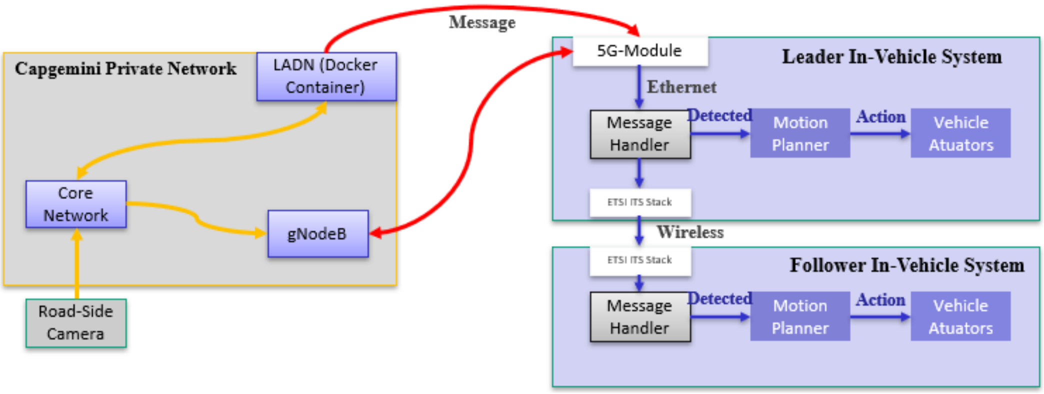


Figure 5.15: Diagram of the Publication System

all the operations done [3]. For a more technical view of the CISTER job and the implementation done in this project, another video was made that demonstrates the work done for this publication [4].

---

[3] https://www.youtube.com/watch?v=R6Q4wi-QrmY

[4] https://www.youtube.com/watch?v=mJanHtjrE_I&list=PLzr8EMXGMN7x8PA-1wsQcXsY5qmVCzmdT

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In conclusion, the research conducted on 5G-enabled autonomous platooning on robotic vehicle test beds has revealed significant advancements and promising prospects for the future of transportation systems. Several objectives were accomplished, like doing a platooning technique based in a sensor information fusion using image processing and LiDAR measurements, maintain a stable communication between all the elements of the application scenario via network and OBUs with 802.11p communication and execute a safety-critical scenario using all the previous implementations done.

This implementation allowed to make a safety-critical application scenario in the form of an emergency braking procedure with infrastructure support, using a camera to detect a vehicle approaching to a zone that has a known accident and sending a message via network that will be received by the Leader Car. On top of this, a platooning technique was also developed, what can be seen has a step forward and a more complex implementation. The main contributions of this work were the platooning technique using sensor information fusion, as well as the full communications between all the elements of the scenario using a 5G-capable module and the OBUs. In addition to that, all the evaluations and validations are also part of the main contributions to this work.

Concerning the results obtained, it is obvious that the delay observed is a little inconsistent because of the low network present in CISTER lab, but it was also clear that this system is very effective, being capable of doing the detection and the emergency braking in a short time, with the bigger delay being under 450 ms. One point that needs to be talked is that this is done using a robotic test bed and in a real world environment many variables needed to be inserted and the results of course would be different.

Nevertheless, this application scenario is a good contribution to increase the safety of the roads around the world, knowing that the future will certainly pass by autonomous vehicles helped by an infrastructure located in strategic places fused by the information gathered by sensors present in the car to ensure that the car has all the information of the surrounded environment.

6.2 Future Work

Regarding the future work, many implementations can be done using the contributions of this project, because of the wide range of vehicular applications that can be done using communications between the several components. For this reason, the possible future projects that can be implemented are, for example:

- More complex scenarios, like cooperative awareness with autonomous driving and waypoints that uses the LiDAR of each car to make a map of the environment and the infrastructure does a fusion of that maps, the cars are controlled via waypoints after having the two maps fused;
- Do this implementation outside to have always the 5G cellular connection, in this way, more tests could be done;
- Make the control of the car being always on the infrastructure, for example, stream the video of the car camera to the infrastructure and send the commands to turn and increase the velocity via network;
- Try to remake this application scenario in a real world environment, with real cars and connected to the vehicle systems.

With this possible implementations, the value of the work will be increased, and more tests could be done to make more advances on the autonomous driving area.

References

[1] "Platooningbackfinalnotype.jpg." `https://upload.wikimedia.org/wikipedia/commons/3/3a/Platooning_Back_022414_Final_noTpye.jpg`. [Cited on pages ix and 3]

[2] G. A. Association, "White paper: C-v2x use cases volume ii: Examples and service level requirements." [Cited on pages 3, 9, and 21]

[3] L. M. Castiglione, P. Falcone, A. Petrillo, S. P. Romano, and S. Santini, "Cooperative intersection crossing over 5g," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 303–317, 2021. [Cited on page 3]

[4] M. Pinheiro, *Implementing OpenSource Vehicular Networking Testbed.* PhD thesis, FEUP, Portugal, 2022. [Cited on pages 4, 5, 37, 40, and 50]

[5] "Floyd 5g/sdn intelligent systems for low latency v2x communications in cross-domain mobility applications." `https://www.cmuportugal.org/large-scale-collaborative-research-projects/floyd/`. Accessed: 2023-03-30. [Cited on page 4]

[6] D. Almeida, *Implementing and Tuning an Autonomous Racing Car Testbed.* PhD thesis, ISEP, Portugal, 2019. [Cited on pages 5 and 37]

[7] "Vehicularcom.jpg." `https://group-media.mercedes-benz.com/marsMediaSite/en/instance/ko/More-traffic-safety-with-Car-to-X-communication-Daimler-is-starting-a-Europe-wide-cooperation-project.xhtml?oid=43505926`. [Cited on pages ix and 10]

[8] I. Soto, M. Calderon, O. Amador, and M. Urueña, "A survey on road safety and traffic efficiency vehicular applications based on c-v2x technologies," *Vehicular Communications*, vol. 33, p. 100428, 2022. [Cited on page 10]

[9] "Intersection.jpg." `https://www.semiconductorforu.com/automotive-electronics-vehicle-vehicle-v2v-comms-whats-tech-available/`. [Cited on pages ix and 11]

[10] A. Balador, A. Bazzi, U. Hernandez-Jayo, I. de la Iglesia, and H. Ahmadvand, “A survey on vehicular communication for cooperative truck platooning application,” *Vehicular Communications*, vol. 35, p. 100460, 2022. [Cited on page 11]

[11] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, “A survey of autonomous vehicles: Enabling communication technologies and challenges,” *Sensors*, vol. 21, no. 3, 2021. [Cited on page 11]

[12] L. Carabulea, C. Pozna, C. Antonya, C. Husar, and A. Băicoianu, “The influence of the advanced emergency braking system in critical scenarios for autonomous vehicles,” *IOP Conference Series: Materials Science and Engineering*, vol. 1220, p. 012045, jan 2022. [Cited on page 11]

[13] W. Nawapom, F. Bai, and M. Priyantha, “Broadcasting in vanet,” (Anchorage, AK), pp. 7–12, 2007. [Cited on page 12]

[14] M. Conti and S. Giordano, “Mobile ad hoc networking: milestones, challenges, and new research directions,” *IEEE Communications Magazine*, vol. 52, no. 1, pp. 85–96, 2014. [Cited on page 12]

[15] “Integrity and authentications for service security in vehicular ad hoc networks (vanets): A review - scientific figure on researchgate.” `https://www.researchgate.net/figure/Vehicular-ad-hoc-network-VANETS_fig1_352787389`. [Cited on pages ix and 12]

[16] B. Vieira, “A simulation approach for increased safety in advanced c-its scenarios,” Master’s thesis, Porto, Portugal, 2019. [Cited on pages ix and 13]

[17] “Comparing limeric and dcc approaches for vanet channel congestion control.” `https://www.researchgate.net/figure/Comparison-of-the-protocol-stacks-in-the-US-called-WAVE-and-the-EU-stack-developed-by_fig1_286562334`. [Cited on pages ix and 14]

[18] M. Arndt, “Etsi - automotive intelligent transport | intelligent transport systems (its).” `https://www.etsi.org/technologies/automotive-intelligent-transport`. [Cited on page 14]

[19] “Ieee 1609 - family of standards for wireless access in vehicular environments (wave).” `https://www.standards.its.dot.gov/Factsheets/Factsheet/80`. [Cited on page 14]

[20] “A framework for supporting network continuity in vehicular ipv6 communications - scientific figure on researchgate.” `https://www.researchgate.net/figure/ISO-ETSI-reference-communication-stack_fig1_260536157`. [Cited on pages ix and 15]

[21] "Etsi en 302 637-2 v1.3.1." `https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.01_30/en_30263702v010301v.pdf`. [Cited on pages ix, 16, and 17]

[22] J. Larmouth, *ASN.1 Complete*. Morgan Kaufmann, 2000. [Cited on page 17]

[23] "Etsi en 302 637-3 v1.2.1." `https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.02.01_30/en_30263703v010201v.pdf`. [Cited on pages ix and 18]

[24] D. Jiang and L. Delgrossi, "Ieee 802.11p: Towards an international standard for wireless access in vehicular environments," in *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pp. 2036–2040, 2008. [Cited on page 19]

[25] ETSI, "Its-g5 access layer specification for intelligent transport systems operating in the 5 ghz frequency band." [Cited on page 19]

[26] Y. Li, W. Chen, K. Zhang, T. Zheng, and H. Feng, "Dsrc based vehicular platoon control considering the realistic v2v/v2i communications," (Chongqing, China), pp. 7585–7590, 2017. [Cited on page 19]

[27] A. M. S. Abdelgader and W. Lenan, "The physical layer of the ieee 802.11p wave communication standard: The specifications and challenges," p. 8, 2014. [Cited on page 19]

[28] J. A., A.-K. S., and D. A., "Performance evaluation of ieee 1609 wave and ieee 802.11p for vehicular communications," (Poznan, Poland), pp. 1–5, 2012. [Cited on page 19]

[29] S. Grafling, P. Mahonen, and J. Riihijarvi, "Performance evaluation of ieee 1609 wave and ieee 802.11p for vehicular communications," (Jeju Island, Korea (South)), pp. 344–348, 2010. [Cited on page 19]

[30] S. Eichler, "Performance evaluation of the ieee 802.11p wave communication standard," (Baltimore, MD, USA), pp. 2199–2203, 2007. [Cited on page 19]

[31] A. Bazzi, B. M. Masini, A. Zanella, and I. Thibault, "On the performance of ieee 802.11p and lte-v2v for the cooperative awareness of connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10419–10432, 2017. [Cited on page 19]

[32] A. Bazzi, G. Cecchini, M. Menarini, B. M. Masini, and A. Zanella, "Survey and perspectives of vehicular wi-fi versus sidelink cellular-v2x in the 5g era," *Future Internet*, vol. 11, no. 6, 2019. [Cited on page 20]

[33] "C-v2x architecture.jpg." `https://www.youtube.com/watch?v=rAePhaTgP5U`. [Cited on pages ix and 20]

[34] GSMA, "Cellular vehicle-to-everything (c-v2x)." [Cited on page 21]

[35] T. Renzler, M. Stolz, and D. Watzenig, *Decentralized Dynamic Platooning Architecture with V2V Communication Tested in Omnet++*. PhD thesis, IEEE, Austria, 2019. [Cited on page 21]

[36] A. Petrillo, A. Pescapé, and S. Santini, "A collaborative approach for improving the security of vehicular scenarios: The case of platooning," vol. 122, pp. 59–75, 2018. [Cited on page 21]

[37] L. Xu, L. Y. Wang, G. Yin, and H. Zhang, "Communication information structures and contents for enhanced safety of highway vehicle platoons," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206–4220, 2014. [Cited on page 21]

[38] Z. Mohamed-Kassim and A. Filippone, "Fuel savings on a heavy vehicle via aerodynamic drag reduction," *Transportation Research Part D: Transport and Environment*, vol. 15, no. 5, pp. 275–284, 2010. [Cited on page 22]

[39] M. Goli and A. Eskandarian, "Merging strategies, trajectory planning and controls for platoon of connected, and autonomous vehicles," *International Journal of Intelligent Transportation Systems Research*, vol. Free Access at:, 07 2019. [Cited on pages ix and 22]

[40] L. Konstantinopoulou, A. Coda, and F. Schmidt, "Specifications for multi-brand truck platooning," 2019. [Cited on page 22]

[41] S. Ellwanger and E. Wohlfarth, "Truck platooning application," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 966–971, 2017. [Cited on page 22]

[42] C. Wang, S. Gong, A. Zhou, T. Li, and S. Peeta, "Cooperative adaptive cruise control for connected autonomous vehicles by factoring communication-related constraints," *Transportation Research Part C: Emerging Technologies*, vol. 113, pp. 124–145, 2020. ISTTT 23 TR_C-23rd International Symposium on Transportation and Traffic Theory (ISTTT 23). [Cited on page 23]

[43] H. Cao, S. Gangakhedkar, A. R. Ali, M. Gharba, and J. Eichinger, "A testbed for experimenting 5g-v2x requiring ultra reliability and low-latency," in *WSA 2017; 21th International ITG Workshop on Smart Antennas*, pp. 1–4, 2017. [Cited on page 23]

[44] D. Lu, Z. Li, D. Huang, X. Lu, Y. Deng, A. Chowdhary, and B. Li, "Vc-bots: A vehicular cloud computing testbed with mobile robots," in *Proceedings of*

the First International Workshop on Internet of Vehicles and Vehicles of Internet, IoV-VoI '16, (New York, NY, USA), p. 31–36, Association for Computing Machinery, 2016. [Cited on page 23]

[45] G. Soós, D. Ficzere, P. Varga, and Z. Szalay, "Practical 5g kpi measurement results on a non-standalone architecture," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5, 2020. [Cited on page 23]

[46] A. Zafeiropoulos, P. Gouvas, E. Fotopoulou, G. Tsiolis, T. Xirofotos, J. Bonnet, G. Carrozzo, S. Rizou, A. Gavras, M. J. Barros, X. Costa-Perez, A. Prasad, M. Gramaglia, A. Tzanakaki, D. Simeonidou, J. Cosmas, M. Fallgren, R. Muñoz, and R. Vilalta, "Enabling vertical industries adoption of 5g technologies: A cartography of evolving solutions," in *2018 European Conference on Networks and Communications (EuCNC)*, pp. 1–9, 2018. [Cited on page 23]

[47] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, "Connected roads of the future: Use cases, requirements, and design considerations for vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 110–123, 2018. [Cited on pages ix and 23]

[48] P. Varga, J. Peto, A. Franko, D. Balla, D. Haja, F. Janky, G. Soos, D. Ficzere, M. Maliosz, and L. Toka, "5g support for industrial iot applications— challenges, solutions, and research gaps," *Sensors*, vol. 20, no. 3, 2020. [Cited on page 24]

[49] H. Ullah, N. Gopalakrishnan Nair, A. Moore, C. Nugent, P. Muschamp, and M. Cuevas, "5g communication: An overview of vehicle-to-everything, drones, and healthcare use-cases," *IEEE Access*, vol. 7, pp. 37251–37268, 2019. [Cited on page 24]

[50] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, "Connected roads of the future: Use cases, requirements, and design considerations for vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. PP, pp. 1–1, 07 2018. [Cited on page 24]

[51] "5gcar final project report." `https://5gcar.eu/wp-content/uploads/2019/08/5GCAR_D1.3_v1.0.pdf`. [Cited on page 24]

[52] Q. Qiu, S. Liu, S. Xu, and S. Yu, "Study on security and privacy in 5g-enabled applications," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–15, 12 2020. [Cited on page 24]

[53] M. Gerlach, "Trust for vehicular applications," in *Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07)*, pp. 295–304, 2007. [Cited on page 24]

[54] M. O'Kelly, V. Sukhil, H. Abbas, J. Harkins, C. Kao, Y. V. Pant, R. Mangharam, D. Agarwal, M. Behl, P. Burgio, and M. Bertogna, "F1/10: An open-source autonomous cyber-physical platform," 2019. [Cited on pages ix, 28, and 29]

[55] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. cite arxiv:1804.02767Comment: Tech Report. [Cited on page 31]

[56] "Rplidar a1 slamtec." `https://www.adafruit.com/product/4010`. [Cited on pages ix and 36]

[57] "Sim8200ea-m2 5g hat for raspberry pi." `https://www.waveshare.com/sim8200ea-m2-5g-hat.htm`. [Cited on page 47]

[58] M. Araújo, J. Silva, P. M. Santos, H. Singh, D. Gunjal, J. Fonseca, P. Duarte, B. Mendes, R. Barbosa, P. Steenkiste, S. Sabamoniri, L. Lam, J. Pereira, and H. Kurunathan, "Demo object detection under 5g-edge mobility," in *Proceedings of the 24th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, (Boston, Massachusetts, USA), June 2023. [Cited on page 72]