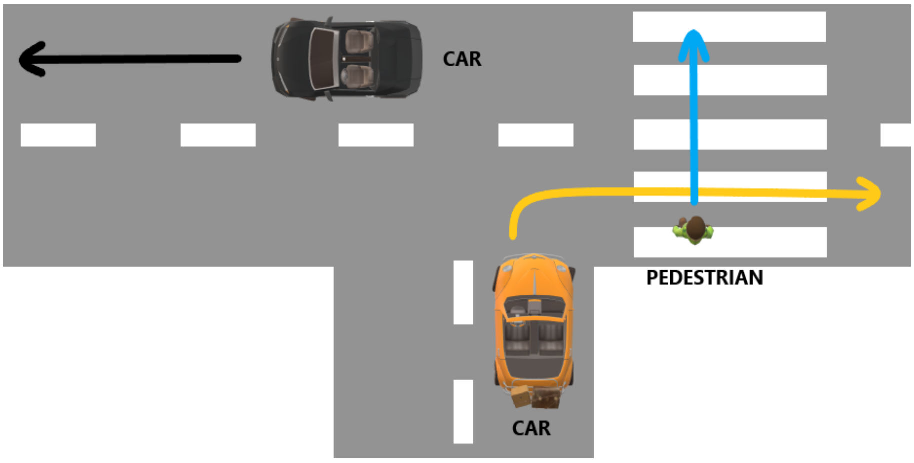
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia Eletrónica e Telecomunicações e de Computadores



A safety perspective for soft mobility in the ITS ecosystem

Mafalda Sofia Compadrinho Gonçalves

Licenciatura em Engenharia Informática, Redes e Telecomunicações

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

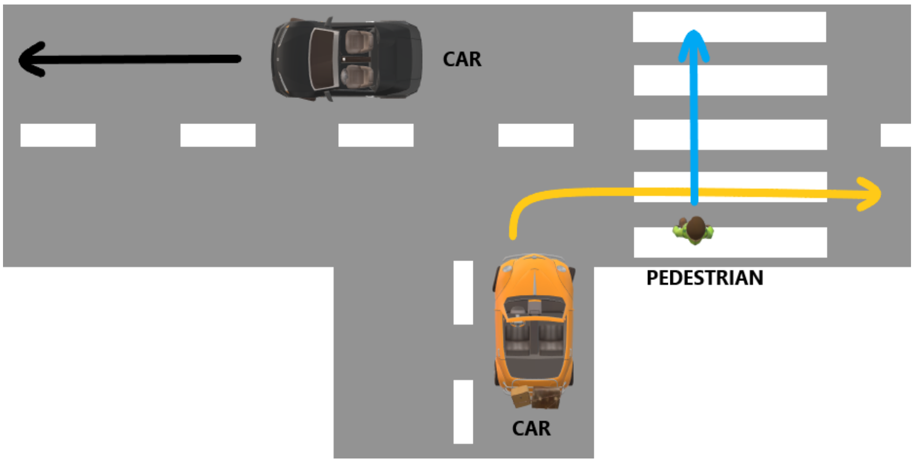
Orientador : Doutor Nuno Miguel Soares Datia

Júri:
Presidente: Doutor Nuno Miguel Machado Cruz
Vogais: Doutor José Manuel de Campos Lages Garcia Simão
Doutor Nuno Miguel Soares Datia

OCTOBER, 2023

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia Eletrónica e Telecomunicações e de Computadores



A safety perspective for soft mobility in the ITS ecosystem

Mafalda Sofia Compadrinho Gonçalves

Licenciatura em Engenharia Informática, Redes e Telecomunicações

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador : Doutor Nuno Miguel Soares Datia

	Júri:
Presidente:	Doutor Nuno Miguel Machado Cruz
Vogais:	Doutor José Manuel de Campos Lages Garcia Simão Doutor Nuno Miguel Soares Datia

OCTOBER, 2023

Acknowledgments

I would like to thank the following people, without whom I would not have been able to complete this research.

My thesis advisor, Doutor Nuno Datia of Instituto Superior de Engenharia de Lisboa, whose knowledge into the subject matter steered me in the right direction throughout this project. Thank you for your encouragement and patience, and also for the insightful conversations and your taste in great music.

I would like to thank all the professors and students involved in the C-Streets project, where I participated as a research investigator, who introduced me to the Intelligent Transport Systems area and who got me to this point of my academic path.

I also want to thank my girlfriend, for all the late nights spent reading this dissertation and listening to me thinking out loud incessantly. Thank you for believing in me and for being always being by my side through this journey.

I am also deeply grateful to my family and friends, who were always there to listen to my doubts and achievements and a special thanks to the ones who assisted me in the tests ran for this project.

Thank you all so much for your unwavering support.

Abstract

All road users have one concern in common, regardless of the mode of transport they use — reaching their destination safely. Users of soft mobility have a high fatality rate, meaning that every time they choose to walk or drive a bicycle, for example, they are ultimately putting their lives at risk. It was found that these entities' safety must be protected, and this motivation is the major reason for designing a tool that will help decrease the number of road accidents and their severity, along with a desire to reduce carbon dioxide emissions by using fuel-efficient vehicles. In this project, a technological system based on information provided by ITS messages, as well as GPS location data, was implemented for users of soft mobility so they can be aware of dangers that show up in their travel path, as well as promoting the use of soft mobility as an alternative mode of transportation. The system has a UI that allows users to receive information without disturbing or diverting their attention from the path ahead for a long time, showing notifications for potentially dangerous road events or other road entities approaching. There is equipment that allows the exchange of messages related to vehicles, roads and traffic, which operates over ITS-G5. To ensure soft mobility users are integrated into the ITS ecosystem, a hybrid network environment that allows communications between cellular and ITS-G5 networks was established. A prototype of this system was designed and tested on multiple use cases, in a real and controlled environment, addressing critical scenarios for the user's safety. According to the results, with the latency involved in E2E communications, this network can support several urban scenarios, where soft mobility users and connected cars share the road. However, in order to guarantee soft mobility user's safety, further testing should be done accounting for the processing time of the application and user's time-to-react.

Keywords: ITS, soft mobility, hybrid network, awareness, safety

Resumo

Independentemente do meio de transporte de preferência, quem partilha vias rodoviárias tem uma preocupação comum - chegar ao seu destino em segurança. Acidentes que envolvem utilizadores de mobilidade suave têm uma taxa de mortalidade mais alta, o que significa que cada vez que estes optam por caminhar ou andar de bicicleta, por exemplo, estão potencialmente a colocar as suas vidas em risco. A necessidade de proteger estas entidades é a principal motivação para o desenvolvimento de uma ferramenta que ajude a diminuir o número de acidentes rodoviários com estas pessoas e a sua gravidade, aliada à redução de emissões de CO_2, utilizando menos veículos de combustível. Este projeto consiste num sistema tecnológico designado a utilizadores de mobilidade suave, que permite informá-los acerca de perigos ao longo do seu trajeto – com base em informação fornecida por mensagens ITS e dados de localização GPS. Este sistema tem uma interface de utilizador que permite que estes recebam informação sem desviar a sua atenção do caminho, apresentando notificações quando surgem potenciais perigos na estrada. Existem equipamentos que permitem a troca de mensagens relacionadas com veículos, estradas e trânsito, que comunicam através da rede ITS-G5. Para integrar os utilizadores de mobilidade suave com o ecossistema ITS, foi criado um ambiente híbrido que permite comunicações entre redes celulares e ITS-G5. O protótipo deste sistema foi testado num ambiente real e controlado, abordando cenários críticos para a segurança do utilizador.Verificou-se que, com a latência envolvida em comunicações E2E, a rede híbrida é viável em cenários onde estes utilizadores partilham a estrada com veículos conectados. No entanto, devem ser realizados mais testes considerando a latência adicional introduzida pelo tempo de processamento da aplicação e pelo tempo de reação dos utilizadores.

Palavras-chave: ITS, mobilidade suave, rede híbrida, alerta, segurança

Contents

List of Figures

Acronyms

ANSR	Autoridade Nacional de Segurança Rodoviária. 7
API	Application Programming Interface. 28, 33, 34, 35, 38
CAM	Cooperative Awareness Message. 2, 14, 15, 18, 22, 23, 24, 26, 30, 33, 35, 36, 42, 43, 46, 47, 49, 50, 51, 52, 53, 54, 55, 57
CCAM	Cooperative, Connected and Automated Mobility. 14
C-ITS	Cooperative Intelligent Transport Systems. 11, 13, 14, 16, 18, 21, 22, 26, 36
C-Roads	Cooperative Roads. 41
C-Streets	Cooperative Streets. 9, 41, 42
CV	Connected Vehicle. 21
DENM	Decentralized Environmental Notification Message. 2, 14, 15, 16, 18, 22, 23, 24, 33, 35, 39, 41, 43, 44, 45, 61
DSRC	Dedicated Short-Range Communication. 17
E2E	End-to-End. 60
ETSI	European Telecommunications Standards Institute. 11, 13, 36
EU	European Union. 5, 7, 8, 14
GDP	Gross Domestic Product. 7
GHG	Greenhouse Gases. 8
GPS	Global Positioning System. 2, 9, 33, 34, 35, 42

HMI	Human-Machine Interface. 3, 9, 14, 19
I2V	Infrastructure-to-Vehicle. 16
ICT	Information and Communications Technology. 14
IDE	Integrated Development Environment. 26, 36
IGRF	International Geomagnetic Reference Field. 30
IP	Internet Protocol. 23
ITS	Intelligent Transport Systems. 2, 3, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 34, 36, 38, 41, 42, 44, 45, 46, 49, 51, 52, 53, 54, 55, 57, 59, 61
ITS-S	ITS-Station. 12, 13, 14, 15, 25, 26, 27, 28, 30, 35, 49, 57, 61
LEZ	Low Emission Zone. 8
MQTT	Message Queuing Telemetry Transport. 22, 23, 24, 25, 26, 33, 35, 36, 44, 46, 47, 51, 52
OBU	On Board Unit. xv, 2, 12, 16, 21, 22, 26, 42, 44, 46, 49, 50, 51, 52, 53, 54, 55, 57
OSM	OpenStreetMap. 33, 38
Protobuf	Protocol Buffers. 36
QoS	Quality of Service. 23, 24
R&D	Research and Development. 9
RSU	Road Side Unit. xv, 2, 12, 16, 21, 22, 23, 24, 25, 42, 43, 44, 46, 51, 57
RTT	Round-Trip Time. 45, 47, 49, 53, 60
RX	Receiver. 12
SDK	Software Development Kit. 33, 37
SM+	Soft Mobility, Electric Scooter and Electric Bicycle Users. 1, 2, 3, 5, 6, 7, 9, 11, 14, 18, 21, 22, 25, 26, 28, 29, 30, 31, 32, 34, 36, 38, 41, 42, 43, 45, 48, 49, 51, 53, 54, 55, 57, 59, 60, 61, 62

TCP	Transmission Control Protocol. 23, 25, 46
TLS	Transport Layer Security. 23
TTC	Time-to-Conflict. 48
TX	Transmitter. 12
UI	User Interface. 5, 9, 26, 30, 37, 38, 39, 52, 54, 60
UTC	Coordinated Universal Time. 46
V2I	Vehicle-to-Infrastructure. 11
V2V	Vehicle-to-Vehicle. 11, 16, 18
V2X	Vehicle-to-Everything. 2, 11, 12, 14, 18, 59
VRU	Vulnerable Road Users. 1, 5, 6, 11, 14, 18, 19
WMM	World Magnetic Model. 30
ZER	Zona de Emissões Reduzidas. 8

1

Introduction

Every single day, there are thousands of people driving vehicles across streets and roads, sharing these spaces with other drivers, such as soft mobility users, electric bicycles and electric scooters trying to get to their destination [34]. Soft mobility includes all non-motorized transports (i.e. human powered mobility), such as pedestrians, bicycles, roller skates and skateboards. It could also be referred to as "zero-impact" — thus, sustainable — means of mobility [40, 45]. Between all of these road users, some are on their way to work, some running errands, some exercising and others simply wandering around. No matter how different their goal is, they all have one thing in common – they want to arrive to their destination safely. The one thing that all road users hope for when they travel from A to B is safety. Considering all the different types of drivers that move around every day, those who drive soft mobility means of transport, are the most at risk in terms of safety [51]. When there is an accident involving these entities, they are the most susceptible to suffering worse injuries [69].

The work presented in this dissertation is aimed for SM+ users — which is the term that will be hereon used to refer to soft mobility users, electric scooter and electric bicycle drivers. Additionally, the term VRU is used to refer to Vulnerable Road Users and it comprises soft mobility users, cyclists, motorcyclists and moped riders. It is important to notice the difference between these two terms, seeing as all SM+ users are VRUs, but not all VRUs are considered SM+ users. To clarify, SM+ users are the same as VRUs, except for motorcycle and moped riders.

For the purpose of this dissertation, the term SM+ will be used to refer to soft mobility users, electric scooter and electric bicycle drivers. The letters "SM" stand for Soft Mobility and the sign "+" refers to the other two types of road users previously mentioned.

With this in mind, a need for a way to protect these members of the road community was found. Hence, an intelligent technological system with the ability to automatically warn drivers of all means of transport about possible dangers along their path, fully integrated with the current Intelligent Transport Systems (ITS) environment, would be a major asset to help prevent these situations and provide additional comfort and safety to everyone by protecting the most vulnerable elements on the road.

ITS [67] is a transportation system which aims to resolve a variety of road traffic issues, such as traffic accidents and congestion, by linking people, roads, and vehicles in an information and communications network via cutting-edge technologies. The three most important components needed for establishing ITS are location (e.g. Global Positioning System (GPS)), mapping (i.e. technology that plots location information on a map) and communications [36].

As stated in "Vehicular ad hoc Networks" [13], nowadays, the ITS ecosystem can identify their users' geolocation —- which makes it possible for application developers to pinpoint the obtained coordinates on a map [59] —- and have vehicles and smart devices connected between each other. There are Road Side Units (RSU) installed in lampposts and traffic lights [46] and On Board Units (OBU) installed in vehicles to exchange ITS messages such as Cooperative Awareness Messages (CAM) [24] and Decentralized Environmental Notification Messages (DENM) [25] between each other. These messages contain information about (i) traffic; (ii) each vehicle's speed, acceleration and direction, as well as some of the vehicle's characteristics (e.g. size, height, weight); (iii) roadworks; (iv) construction works; (v) weather and road conditions. This technology creates an enormous network of interconnected devices that can communicate and collaborate in real-time to improve the safety of the roads. Vehicle-to-Everything (V2X) communications are the kind of communications that happen between a car and other connected elements, such as people. With this type of communication, each element can be used to advertise its location and inform others about its activities. With all this information, if the context of the road users' movement, location, direction and speed is known, it is possible to alert drivers about potentially dangerous situations and help prevent them.

The solution proposed in this dissertation is an intelligent technological system that

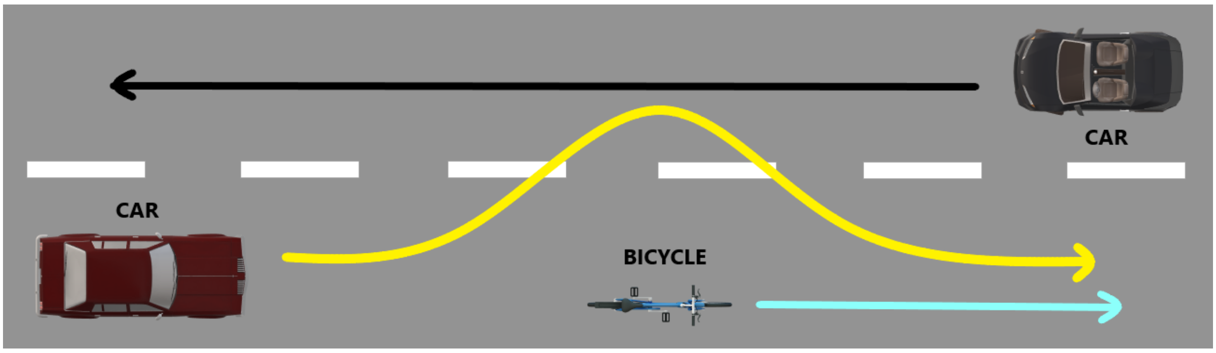


Figure 1.1: Dangerous situation of a car overtaking a cyclist

alerts all users of each other's presence through implicit feedback, exhibiting an informative icon on the device they are running the application on, to notify them of the threat that is approaching, along with the suggestion of an alternative to avoid it. The system should be able to identify different users' geolocation and alert them of possible dangers nearby, – based on information provided by the ITS messages – according to the vehicle they are driving, via an adequate Human-Machine Interface (HMI).

Take for example a two-way straight road where a car driver is approaching a cyclist from behind, with the intent to overtake. On the one hand, the car driver should be notified to slow down due to the presence of the cyclist at a certain distance ahead and, depending on whether there is someone coming the opposite way, they should also be informed if it is safe to overtake the cyclist. On the other hand, the cyclist should also be alerted to the fact that there is a vehicle approaching from behind, so they should pay extra attention to their surroundings. Figure 1.1 illustrates this scenario, where it is dangerous for the car to overtake the cyclist. When the cyclist is notified of the vehicle's presence, they can avoid being caught off guard if the car decides to overtake. From the driver's perspective, they can be notified about what the best time to initiate the maneuver is, instead of rushing to a dangerous situation that can get all the elements involved at risk. The yellow line indicates the path that the red car's driver intends to take; the black line indicates the path that the black car's driver intents to take and finally, the blue line indicates the path that the cyclist intends to take. This is valid to multiple scenarios where SM+ users can be in a vulnerable situation. For instance, a cyclist who wants to enter a roundabout takes a longer time to do so than a vehicle. Vehicle drivers are a lot more likely to miss the presence of two-wheeled vehicles, especially if they come up from the driver's blind spot [65]. In a scenario where there are multiple vehicles going around a busy roundabout, there is a very high chance that vehicles will be wanting to exit the roundabout as the cyclist wants to enter it, which might result in a tragic collision. Figure 1.2 illustrates this example. The yellow line

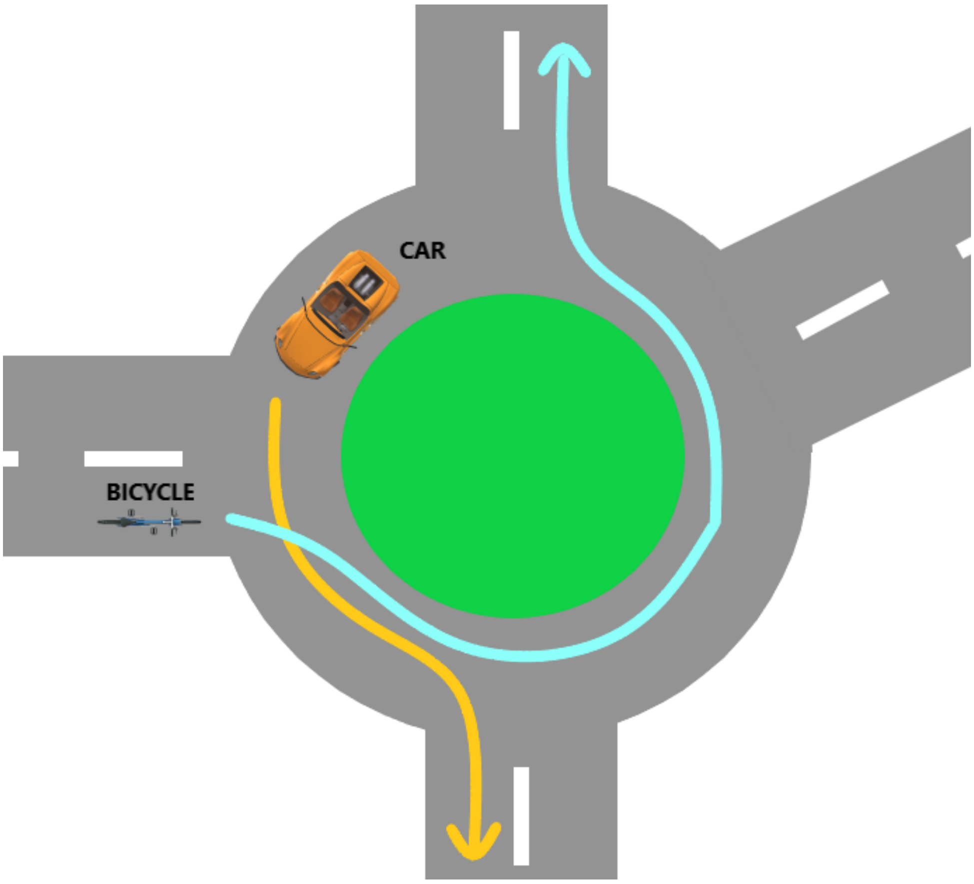


Figure 1.2: Dangerous situation of a cyclist entering a roundabout

indicates the path that the driver intends to take and the blue line indicates the path that the cyclist intends to take. Accidents can be avoided if the system considers all the vehicles on the roundabout and their speed, direction and intent based on context, helping the cyclist to choose the best time to enter the roundabout. Simultaneously, if all the vehicle's drivers on the roundabout are notified about the cyclist's presence, they will be more likely to pay extra attention when exiting, therefore, decreasing the chances of not seeing the bicycle. Another critical scenario would be a pedestrian at a crossing with poor visibility, as illustrated in Figure 1.3. The yellow line represents the path the yellow car's driver intends to take; the black line represents the path that the black car's driver intends to take and lastly, the blue line represents the path that the pedestrian intends to take. If a car had the intent to turn right at a crossing, they would most likely look to the left to make sure there were no vehicles coming. However, if there was a pedestrian crossing the road immediately to the right of the crossing where the driver was, there would be a very high chance that the driver would not see them with enough time to react appropriately – the driver would have to break suddenly, causing a dangerous situation, with no certainty that they would do so in enough time not to harm the pedestrian. This can also be avoided if there is a system that warns

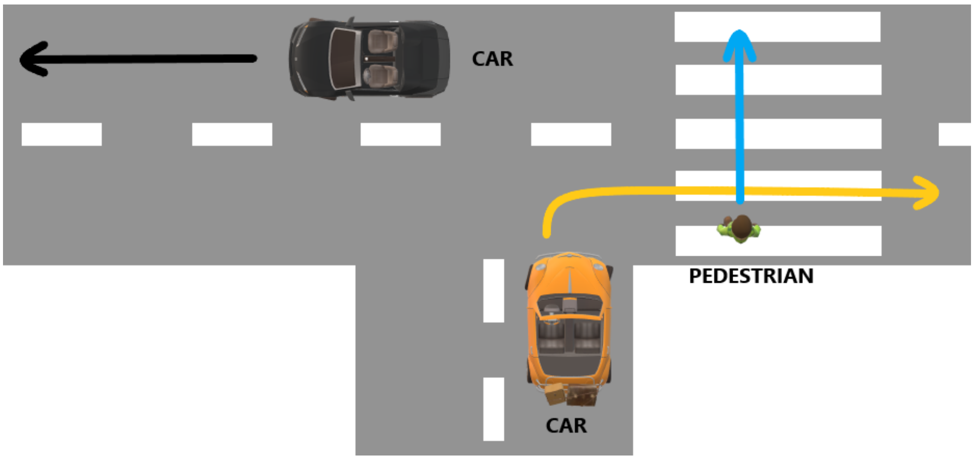


Figure 1.3: Dangerous situation of a pedestrian crossing the road

all users about the presence of other users in or about to enter potentially dangerous situations, based on speed, direction and context. Raising awareness to all users' surroundings can raise the chances of collision avoidance, because it allows all the entities involved to have extra time to react and adapt accordingly to the situation.

The system mentioned should use a User Interface (UI) with a specific design for smartwatches, smartphones and car dashboards, where the former only alerts users of the dangers that come up with a sound and/or vibration notification and an informative icon about the warning. The UI for smartphones and car dashboards should have a map indicating the owner's geolocation, and other vulnerable users' around geolocation as well, when the system deems relevant to do so (i.e. if there's a SM+ user at risk, their location should be visible on the car's map). There can also be shown alternative routes for SM+ drivers, if the path they are choosing places them at risk.

A prototype of this system was developed and tested inside ISEL's campus, where the topology of the road is similar to one found in a municipal area, such as Lisbon (Portugal). Aspects such as the confirmation of which is the most suitable UI, and type of notification to alert users without distracting them in a way that puts them in peril, will also be described.

1.1 Safety issues in soft mobility

According to Olszewski et al. [51], between the years of 2009 and 2013, over half of all road accident victims in the European Union (EU) are Vulnerable Road Users (VRU). In the seven EU countries investigated, 26 670 VRU died and about 10 times more (265,400) were seriously injured. Pedestrians comprised most fatalities (47%), followed

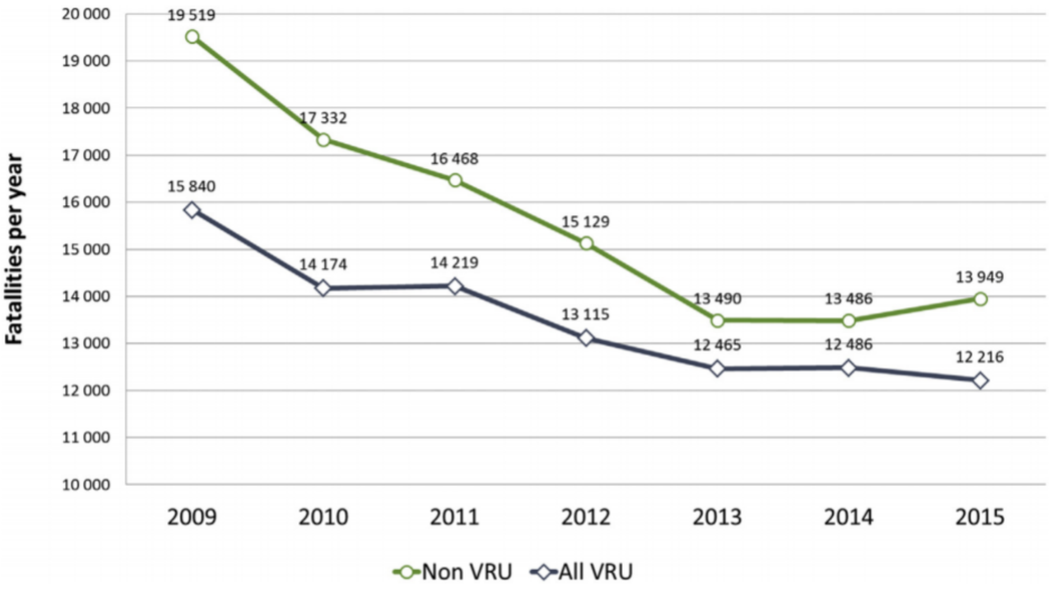


Figure 1.4: Statistics of VRU versus non-VRU fatalities from 2009 to 2015 [51]

by motorcyclists (28%), cyclists (19%), and moped riders (6%). This data regarding vulnerable road users does not only include SM+ users, however, as previously stated, all SM+ users are vulnerable road users. For all the road user groups a declining tendency of fatalities was registered in the period of 2009–2013. However, this tendency was greater in the case of non-VRU (car, bus, truck drivers, and passengers) than in the case of VRU. That means that the efforts into improving traffic safety in Europe gave better results in the case of non-VRU. From 2009 to 2015, the best result was achieved for moped riders (54% reduction), followed by non-VRU (29% reduction), motorcyclists (25% reduction) and pedestrians (22% reduction). Cyclists are the type of road users who faced the worst situation, showing an almost constant number of fatalities during this period (only 11% reduction) - see Figure 1.4.

On the 26th of June 2021, Patrizia Paradiso, a 37 year old pregnant woman, a professor and researcher at Instituto Superior Técnico, died as a result of a road accident. Patrizia was the deadly victim of a collision between a car and her bicycle, on Avenida da Índia, in Lisbon. The alleged reason why the driver did not see the cyclist on the road was due to him being blind sighted by the sun. The four-month pregnant woman was immediately taken to Hospital S. José, but ended up dying on that same day. In a time where sharing road spaces is one of the major topics being discussed, this occurrence generated a wave of indignation — MUBi, Lisbon's Urban Mobility Association, was one of the many entities that manifested their concern about the topic, reminding that Lisbon is the place with most fatal accidents involving bicycles [47]. Américo Silva, a witness of this accident and a cyclist himself, stated that it was the violence of the

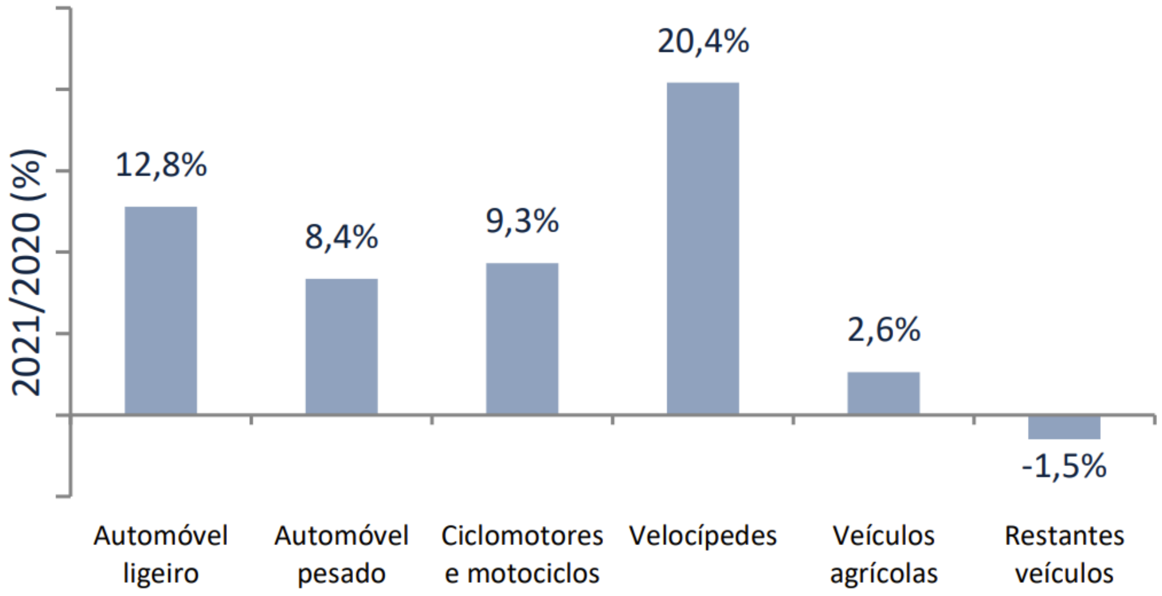


Figure 1.5: Statistics of accidents according to vehicle types, January to July, 2020 vs 2021 [12]

collision that caught his attention. A few months prior to this, in May, Américo had also been victim of an accident in Cascais — he was run over by a car as he was riding his bicycle [56].

Road collisions have a social, personal and economic impact on society. Road traffic injuries are the eighth leading cause of death for all age groups, and the first for children and young adults aged 5-29 years [68].

In mainland Portugal, according to data from Autoridade Nacional de Segurança Rodoviária (ANSR), there has been a 22.8% growth of accidents involving soft mobility comparing the first semester of 2020 to the first semester of 2021 – the most accentuated increase amongst the different means of transport [12] (see Figure 1.5).

Since there is a high rate of fatalities in accidents involving SM+ users, this system aims to decrease this number of accidents, as well as their severity, and help these users feel safe on the road.

1.2 Ecological issues

For many years, there has been an increasing attempt to get society to resort to soft mobility means of transport, in order to strengthen urban sustainability [42]. The transportation sector is responsible for about 5% of the EU's Gross Domestic Product (GDP) and employs over ten million people in Europe - therefore, it is fundamental for the European companies and global supply chains. However, simultaneously, vehicular traffic takes a toll on our society: emissions of Greenhouse Gases (GHG) and atmospheric pollutants, noise, road accident and traffic congestion. Currently, emissions from transport systems represent around 25% of total EU GHG emissions and this number has increased in recent years. The EU has set a goal to be the first climate-neutral continent by 2050 and this requires ambitious changes in this sector. A clear trajectory is needed to achieve a 90% reduction in transport-related GHG emissions by 2050. The European Commission has adopted a set of legislative proposals that aim to make the EU climate, energy, transport and taxation policies able to achieve a reduction of liquid emissions of GHG of, at least, 55% by 2030, when compared to the levels obtained in 1990 [28].

Since the beginning of 2012, Lisbon has had a combination of Low Emission Zone (LEZ) and Access Regulation in place. It is called Zona de Emissões Reduzidas (ZER) and has been frequently updated ever since. This regulation aims to help achieve the environmental goals that Portugal has pledged to meet, through the reduction of CO2 emissions. Since this is a place of high population density, where the geographical topology itself makes it difficult to share space between different vehicles (narrow streets with high slopes, intersections with poor visibility, high traffic flow of various types of vehicles, etc.), there is an urgent need to resort to soft mobility means of transport - for the safety and ecologic reasons previously stated. In 2016, Lisbon was the first capital to subscribe the "Covenant of Mayors for Climate and Energy", assuming the commitment to reduce CO2 emissions in at least 40% by 2030 [71]. From August 2020, the circulation of vehicles built prior to 2000 was prohibited and this constriction was extended to vehicles prior to 2005 by April 2021 [54].

With all the recent growing technological advances in different sectors it is mandatory that mobility does not get left behind. The safety of the population is a critical factor that should be a priority to build a better, safer and ultimately more prosperous society.

To achieve this goal, it is vital that the transport systems meet the requirements for optimization of the transport network, efficient use of land, lower energy and GHG emissions [39]. Soft mobility adherence can be a great step towards meeting this purpose.

1.3 Goals and challenges

The main objective of this project is to raise the awareness of all SM+ users to other road entities that might place them at risk, by alerting them to their presence with enough time for them to react accordingly to the situation or to simply avoid it.

The development of a system such as the one hereby suggested, encounters some challenges at a physical/hardware level as well as at a logical/software level. For the purpose of communications and to have all these devices exchanging messages, almost in real-time, notifying each other of their location and intention (based on context), the signal propagation characteristics and challenges must be taken into consideration. A study on the GPS signal accuracy and its associated error must be conducted. It is also crucial to understand the latency involved in these communications. In this giant web of interconnected elements, some of them move faster than others - in average, fuel-vehicles will move faster than SM+ means of transport - so, when these two types of transportation face the same situation, for example, the need to break due to a pedestrian crossing the road ahead, the fuel-vehicle's driver will have less time to do so, which translates in a need to warn them about this danger earlier. All these aspects play a vital role in the calculation of the time at which each user has to receive the information, so that they have enough time to react. A study on what is the best HMI for each type of road user must be led - a compromise between showcasing enough information to describe the situation but not so much that it will distract the user from their driving must be found (i.e. non-intrusive UI).

1.4 Contributions

This project benefited from the following contributions:

- The research conducted within the scope of the Cooperative Streets (C-Streets) project, as part of ISEL's Research and Development (R&D) team;
- The article of the same title — "A safety perspective for soft mobility in the ITS ecosystem" — published for "INFORUM - Simpósio de Informática", which was written alongside the development of this solution;
- The hybrid network created to allow the communication between devices that operate over cellular networks and ITS equipment that operates over ITS-G5 networks;

- The mobile application developed as a means to process information provided by the ITS equipment and showcase it to the user in a way that is able to alert them of potentially incoming dangerous situations — thus increasing the safety in their travels.

1.5 Organization

The remainder of this document is organized as follows. Chapter 2 presents background knowledge and related works. Chapter 3 describes the solution implemented, detailing the type of equipment used, the network's architecture design, the communication protocols involved, as well as the mobile application's components, workflow, user interface and implementation details. Chapter 4 outlines the tests performed and results obtained. Finally, a discussion about future work and conclusions is presented in Chapter 5.

2

Background knowledge and related work

As stated in Chapter 1, Section 1.3, the goal of this project is to present a solution that improves road safety for SM+ users. The solutions developed with this same objective are still quite limited. Some solutions consider only vehicles, while other focus on a specific type of VRU, and most of them are not fully integrated with the current ITS ecosystem. Some of these approaches will be discussed in Section 2.2. Section 2.1 presents some background knowledge related to the current ITS ecosystem and all that it involves.

2.1 Outlook on the current ITS ecosystem

Cooperative Intelligent Transport Systems (C-ITS) is an application that uses Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) - collectively known as Vehicle-to-Everything (V2X) - communications, at a carrier frequency of 5.9 GHz (5.855 to 5.925 GHz) and 10 MHz bandwidth channels, to increase road traffic safety and efficiency in Europe. Standardization plays an important role in achieving communication interoperability among implementations of different manufacturers. The European Telecommunications Standards Institute (ETSI) Technical Committee on the ITS has developed standards to support C-ITS day-one applications, focusing on protocols supporting applications on the vehicle side. Protocols supporting applications executed on smart infrastructure, such as traffic lights, have also been developed by the European Committee for Standardization working group [63].

A system that implements the ITS protocol stacks and ITS applications is denoted as an ITS-Station (ITS-S). An ITS-S may be integrated in a vehicle (vehicle ITS-S) or at road side (road side ITS-S). A vehicle ITS-S is also named as On Board Unit (OBU), a road side ITS-S is also named as Road Side Unit (RSU) [13]. A detailed description of these two devices will be presented further in the ITS Equipment sub-section.

2.1.1 ITS-G5 Signal Propagation Challenges

Known as ITS-G5 communication system, the communications used in V2X have similar characteristics to regular Wi-Fi communications, with some differences that make it more suitable for the vehicular environment [26].

In any wireless systems, signals undergo different effects when travelling between Transmitter (TX) and Receiver (RX). The different effects that can be seen in a wireless channel are attenuation, reflection, transmission, diffraction, scattering, and wave guiding. The signal strength is decaying as the distance increases between TX and RX, i.e. the signal gets attenuated. Wave guiding is an effect that actually preserves the signal strength due to the fact that the signal is restricted in its expansion. It can occur for example in urban canyons and tunnels. Reflection occurs on smooth surfaces, whereas transmission is when the signal penetrates the object. Scattering spreads the signal in several directions, which occurs on rough surfaces, and diffraction is when the signal is bending around a sharp edge. In Figure 2.1, reflection, transmission, scattering, and diffraction, are illustrated. Smooth, rough, large, and small, are all relative to the wavelength in question. Increased carrier frequency implies smaller wavelength (e.g. 5.9 GHz is equal to a wavelength of 5 cm), more optical propagation, smaller antennas, and higher attenuation (the signal strength is decaying faster with distance) [26].

Urban areas can be particularly challenging for signal propagation, due to the high concentration of various elements that may come between the transmitter and the receiver (e.g. buildings, vegetation, signalling, etc.). These challenges raise implementation issues, since it is necessary to guarantee enough signal coverage to assure that all users send and receive their messages in due time. It is fundamental to map out how many RSUs per Km^2 have to be installed in order to assure the system is reliable.

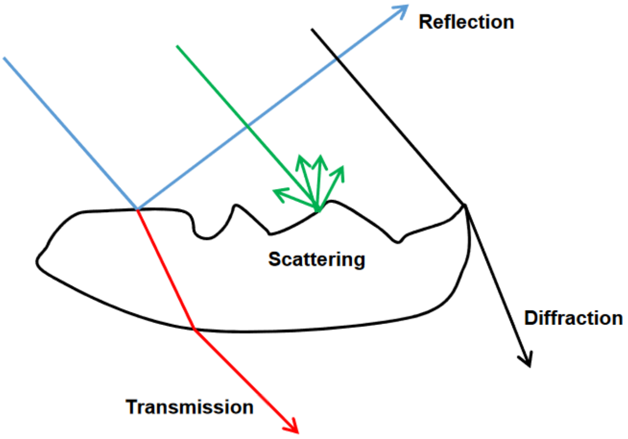


Figure 2.1: Different effects on the signal: transmission, reflection, scattering and diffraction [26]

2.1.2 C-ITS Protocol Stack

As it is declared by Festag, the C-ITS standards follow a general architecture, specified in ETSI EN 302 665 and ISO 21217, with the ITS-S as the core element, representing vehicle, personal (mobile personal devices), roadside (infrastructure), and central (back-end systems and traffic management centers) subsystems. For C-ITS, the ISO OSI reference model was adapted to cover horizontal layers for access technologies, networking and transport, facilities and applications, and vertical entities for management and security. Figure 2.2 illustrates the protocol stack for vehicle and roadside ITS stations, and lists the core standards with their shorthand names for the European C-ITS Release 1.

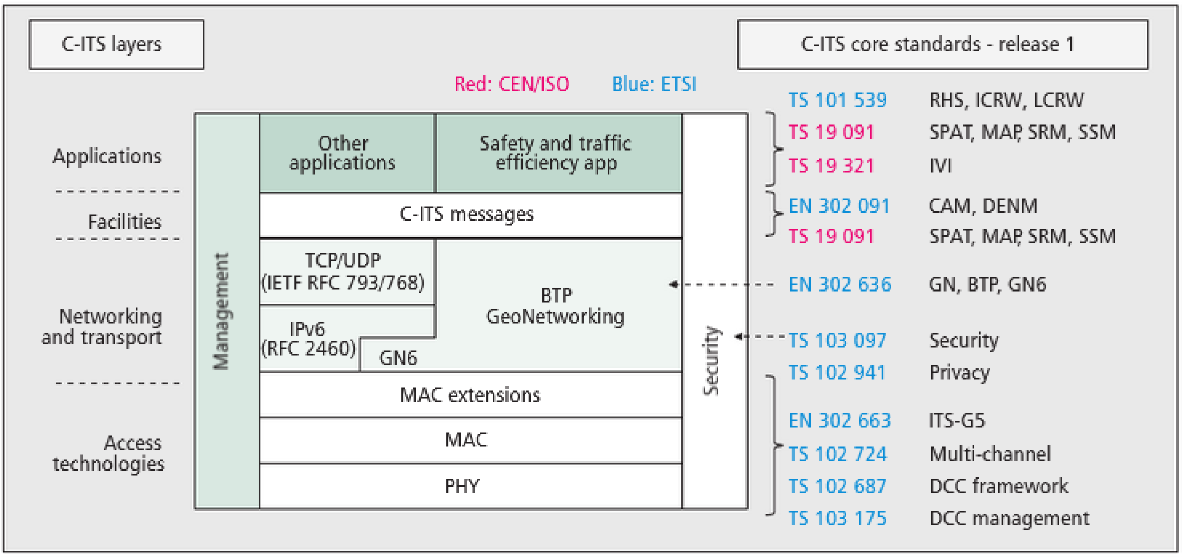


Figure 2.2: Protocol stack and Release 1 core standards for C-ITS in Europe [29]

In 2020, the European Commission published a rolling plan for Information and Communications Technology (ICT) standardization where ITS apply ICT to the mobility sector [22]. Commission explains that the main objective of this policy is to benefit from ITS services and applications in terms of transport efficiency, sustainability, accessibility, safety and security, whilst contributing to the EU's single market and competitiveness objectives and to the Green Deal. The rolling plan's first requested action was to complete the minimum set of standards required for the interoperable deployment of Cooperative, Connected and Automated Mobility (CCAM) services based on V2X communication, connecting all road users and infrastructure, including VRU. There are still on going actions and the second release for C-ITS has not yet been published to this date. However, there have been made many updates to the first release across all C-ITS core standards [27].

2.1.3 ITS Messages

C-ITS messages, defined in the facilities layer of the C-ITS protocol stack, contain important information to help ensure safety for SM+ users.

As asserted in "ETSI EN 302 637-2 V1.4.1", Cooperative Awareness Messages (CAMs) are messages exchanged in the ITS network between ITS-Ss (ITS-Stations) to create and maintain awareness of each other and to support cooperative performance of vehicles using the road network. A CAM contains status and attribute information of the originating ITS-S. The content varies depending on the type of the ITS-S. For a vehicle ITS-S the status information includes time, position, motion state, activated systems, etc. and the attribute information includes data about the dimensions, vehicle type and role in the road traffic, etc. Upon the reception of a CAM, the receiving ITS-S becomes aware of the presence, type, and status of the originating ITS-S. The received information can be used by the receiving ITS-S to support several ITS applications. For example, by comparing the status of the originating ITS-S with its own status, a receiving ITS-S is able to estimate the collision risk with the originating ITS-S and, if necessary, may inform the driver of the vehicle via the HMI.

"ETSI EN 302 637-3 V1.3.1" declares that Decentralized Environmental Notification Messages (DENMs) are mainly used by ITS applications in order to alert road users of a variety of events detected by ITS-Ss. A DENM contains information related to a road hazard or abnormal traffic conditions (e.g. road works, closed lanes, adverse weather conditions, accidents, obstacles on the road, etc.), such as its type, position, speed, acceleration and altitude. Typically for an ITS application, a DENM is disseminated to ITS-Ss that are located in a geographic area through communications among ITS

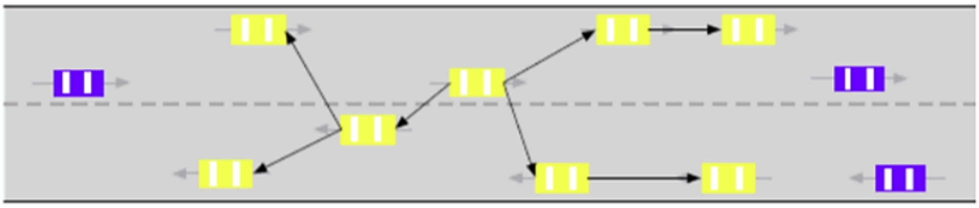


Figure 2.3: CAM dissemination example, n-hop broadcast (n = 2) [23]

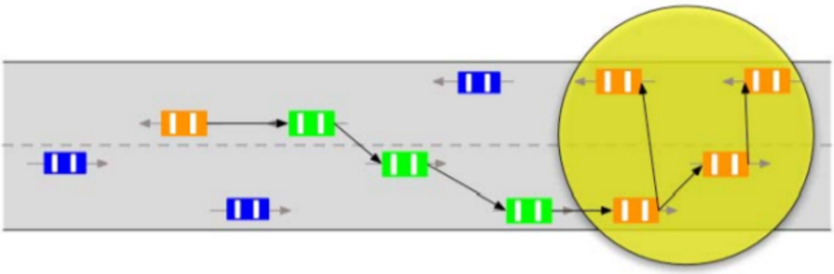


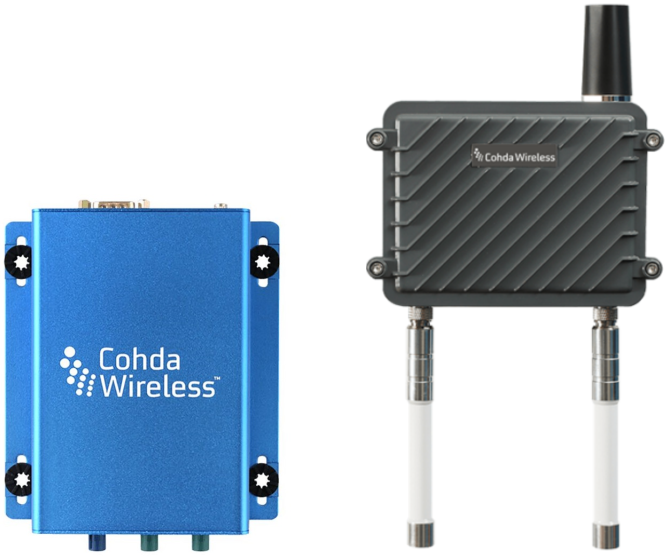
Figure 2.4: DENM dissemination example [23]

stations. At the receiving side, an ITS-S application may present the information to the driver if the information of the road hazard or traffic condition is assessed to be relevant. The driver is then able to take appropriate actions to react to the situation accordingly.

As it is affirmed in “ETSI EN 302 636-2 V1.2.1”, in the Network and Transport layer, there is the GeoNetworking protocol which is is used for ad-hoc communication over ITS-G5 utilizing the geo-addressing. It provides message forwarding services based on the definition of geographical areas and it is used for CAM and DENM dissemination. CAM dissemination involves Single-Hop Broadcast communications (point-to-multipoint). In this scenario, communication starts at a single ITS station and ends at multiple ITS-Ss that are a single hop away from the originating ITS-S. The message is broadcasted until the specified number of hops is reached (n-hop broadcast, n = 1) (see Figure 2.3). DENM dissemination involves GeoBroadcast communications (point-to-multipoint). The communication also starts from a single ITS-S and ends at multiple vehicle ITS-Ss within a geographical area. Stations that receive the DENM but are not in the designated geographical area, should ignore the message (see Figure 2.4).

2.1.4 ITS Equipment

Chang et al. [16], explains that connected vehicle technology has potentials to increase traffic safety, reduce traffic pollution, and ease traffic congestion. In the connected vehicle environment, the information interaction among people, cars, roads, and the



(a) OBU example [17] (b) RSU example [18]

Figure 2.5: Cohda's OBU and RSU

environment is significantly enhanced, and driver behavior will change accordingly due to increased external stimulation. In the architecture of connected vehicle technology, the On Board Unit (OBU) and Road Side Unit (RSU) are key parts of its application in actual transportation services. OBUs are mainly used to provide V2V communications. Placed on top or inside the vehicles, these devices enable the transmission of ITS messages, such as sending new CAM messages or forwarding DENM messages. RSUs are mainly used to provide I2V and V2V communications. These devices are placed in specific places in the infrastructure, such as lamp posts, where signal coverage and accessibility for maintenance are deemed adequate. RSUs also enable the transmission of ITS messages and are usually the devices that originate new DENM messages. Figure 2.5 illustrates an example of an OBU and RSU by Cohda Wireless.

Both OBUs and RSUs can exchange C-ITS messages between each other over the ITS-G5 network, but they cannot communicate directly with other smart devices. To surpass this problem, a way to create a hybrid network with communications over ITS-G5 and cellular networks (e.g. Wi-Fi, 3G, 4G, 5G) was found. The OBUs and RSUs can communicate with an ITS Center over ITS-G5, which will act as an intermediary between the ITS equipment and other smart devices, that will also communicate with the ITS Center, over the available cellular networks. The system architecture will be described in further detail in Chapter 3, Section 3.1.

For the scope of this dissertation, both these devices will be used for testing and prototype development. More detailed information can also be found in Chapter 3.

2.2 Related Work

One of the early attempts to incorporate safety concerns is done by Chandra, in 2014. The work consists on route planning for older drivers and bicyclists, through the use of a multi-objective shortest path algorithm, considering as factors the trip length, traffic, older drivers' reaction time and street design characteristics [15]. In the same year, Chalkia et al. developed a study using historical crash data and road designs to evaluate safety risks, and developed a routing algorithm for school buses, based on Dijkstra's algorithm [14]. In 2015, Kingsbury, Harris, and Durdin combined travel time and distance with safety risks for vehicle routing through the development of two safety models for crash rate prediction: one based on road and traffic characteristics and another based on historical crash records. The project consisted on a geographical information systems algorithm for vehicle routing by considering user preferences based on the users' priorities on trip length, travel time, and safety [37]. In the following year, Li et al. developed an algorithm considering crash data, roadway characteristics, time of day, day of the week and weather conditions as factors to obtain a road risk index and implemented a mixed-integer programming algorithm to find the best route [41]. With a similar approach to the factors considered, in 2017, He and Qin incorporated vehicle type, driver's age and pavement condition to quantify safety in path-finding problems with the factors already considered in the study carried by Li et al.; although this time a multi-objective shortest-path approach was taken [31].

Whilst the safety indexes considered in the studies mentioned above had a variety of different indicators, there are several gaps in the literature. Firstly, most of the data used considers only historical crash data and, whereas using historical data may be enough to predict and avoid some dangerous situations, real-time information should be more accurate and allow road users to avoid even more risks. Secondly, it is known that human error is the leading crash cause [62], and human-behavior and driving style are overlooked in these analysis. Finally, the data considered in these studies is limited [32].

In 2020, Hoseinzadeh et al. formulated a solution that aimed to address precisely these existing gaps, by developing a robust approach to quantify safety in terms of number of crashes, route volatility and driving behavior. Several data sources including historical crash data, real-time connected-vehicles big data and real-time traffic information were integrated and fed into the path-finding problem. The proposed framework was able to work both in real-time and offline modes. This is possible because the algorithm used enables the integration of real-time data and passive data. The real-time data was obtained from vehicles equipped with Dedicated Short-Range Communication (DSRC)

— the standard for the United States' Intelligent Transportation System — devices to collect data in real-world conditions. The data provides a log of vehicle performance factors such as vehicle position, speed and acceleration [32].

The project described in this last article — "Integrating safety and mobility for pathfinding using big data generated by connected vehicles" [32] — shares some similarities with the solution proposed in this dissertation. Although this solution does not suggest alternative routes to users, both aim to identify possible dangerous situations on the road using real-time data. However, the work described in "Integrating safety and mobility for pathfinding using big data generated by connected vehicles" involves only V2V communications and is designed exclusively for vehicles. The solution proposed in this dissertation relies on V2X communications, not only allowing their involvement but focusing on SM+ users and their safety. Using messages exchanged between the road infrastructure and road users, in real-time, it allows SM+ users to be warned about potential upcoming dangers along their path, as well as suggest safer alternative routes. This is done with the information provided by the ITS messages (CAM and DENM) being exchanged between all elements on the road and the infrastructure. Similarly to how the data regarding vehicles' position, speed and acceleration is used in "Integrating safety and mobility for pathfinding using big data generated by connected vehicles", here the information contained in CAM messages is used to achieve the same goal. This introduces an advantage as to the previous solution, because CAM and DENM messages are sent by ITS equipment that will soon be installed on the streets and vehicles, in a standardized way, in Europe. Being SM+ users' safety the main focus of this application, this solution also provides the additional feature of warning users if another road user is approaching in a way that might put them at risk (e.g. vehicle approaching a moped rider from behind over the speed limit for that area). This can be accomplished through the use of hybrid networks (G5 and cellular), which allow ITS equipment to communicate over cellular networks (i.e. Wi-Fi, 3G, 4G or 5G) with other devices, thus allowing ITS equipment to exchange messages with smartphones and smartwatches running the application.

As a more similar solution to what this project aims to accomplish, Dasanayaka et al. [21] identified that, according to data from the World Health Organization, in 2018 more than half of the world's road deaths are recorded amongst VRUs; and that C-ITS is an emerging technology that can highly contribute to improving the safety of this type of road users. The work presented in "Enhancing vulnerable road user safety: a survey of existing practices and consideration for using mobile devices for V2X connections" [21] studies the challenges in integrating VRUs into the C-ITS environment and recognizes that other existing efforts to accomplish this goal are focused on VRU detection and providing awareness warnings to the drivers rather than provide VRUs with any warnings related to potential collision risks of vehicles. This situation may arise due to the fact that the driver bears more responsibility for preventing accidents compared to VRUs, as vehicles traveling at even moderate speeds contribute more energy to a crash compared to VRUs. Nonetheless, as a result of the one-sided nature of communication, VRUs remain unaware of the risk of collision, and the whole agency for avoiding the collision lies entirely with the driver. This paper also presents a way to enable a bi-directional communication between VRUs with a smartphone operating over the available cellular network and the ITS environment, with a publish-subscribe mechanism implemented to act as the "bridge" between both networks. Moreover, it mentions the fact that mobile applications can be developed as a way to inform VRUs of potential dangers through the data received from ITS equipment, but does not provide any solution within this aspect.

Regarding the application's HMI, many studies have used different tools to showcase the results obtained. Rosberg and Ghassemloi, in their study Machine Learning-based path prediction for emergency vehicles, developed a HMI with a framework called Kivy — which is an open-source, flexible, external library for Python, designed to develop multi-touch applications — and it is used to develop user interfaces with Python as its programming language. Nevertheless, in this project, the mobile application was developed using Kotlin and Android Studio to benefit from the tools and features tailored specifically for Android development.

3

C-ITS Application Solution

The solution proposed in this dissertation comprises two distinct components: (i) a hybrid environment that allows SM+ users to communicate with ITS equipment, turning them into connected elements of the road; (ii) a mobile Android application that is the means to inform SM+ users of potential dangers along their travel paths. Both components will be discussed in the following sections.

3.1 Hybrid Environment Architecture

The system's architecture is based on a hybrid network that allows two-way communications over ITS-G5 and cellular networks, as visually depicted in Figure 3.1. Vehicles equipped with OBUs are referred to as "Connected Vehicles" CVs and they utilize said equipment to engage in the exchange ITS messages with other CVs and the infrastructure. OBUs and RSUs intercommunicate within ITS-G5 networks, exchanging C-ITS messages as described in Section 2.1.3. However, their communicative reach is constrained solely to devices that operate within this network domain. In contrast, SM+ users, unlike CVs, do not have OBUs to facilitate the exchange of information with others and their presence goes unnoticed by the connected entities on the road. To overcome this problem, a hybrid ecosystem was created, aiming to turn SM+ users into connected entities too, providing a way to have mobile applications communicating over Wi-Fi or cellular networks, exchanging ITS messages with equipment on the G5 network. In order to allow devices such as smartphones and/or smartwatches to

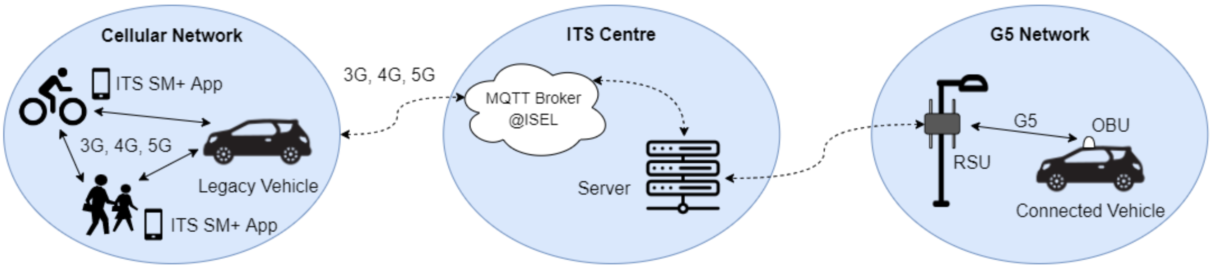


Figure 3.1: Hybrid Environment System Architecture

communicate with this equipment, there has to be an ITS Centre to act as an intermediary between the two networks. Hence, the OBUs and RSUs can communicate between each other over ITS-G5 and with the ITS Centre over Ethernet, and the ITS Centre can also communicate with other smart devices running the application over the available cellular network at that moment — see Figure 3.1.

OBUs, RSUs and smart devices can all be data sources and they all do some level of information processing. OBUs and RSUs send and receive C-ITS messages (e.g. CAM and DENM) and the smart device's user can also send information over to the ITS Centre using some of the application's features. RSUs and OBUs also have storage capacity and will store the C-ITS messages they send and receive in log files over time. The ITS Centre, which acts as a middle-man between the ITS equipment and smart devices, is connected to a Message Queuing Telemetry Transport (MQTT) server and uses the publish/subscribe messaging protocol to send and receive messages between the two ends of the communication — thus establishing the hybrid network.

The hybrid network's architecture is divided into three parts:

- Cellular Network
- ITS Centre
- G5 Network

Figure 3.1 portrays an overview of this architecture. Cellular networks are used by all road users who are not connected to the ITS ecosystem (i.e. legacy vehicles and SM+ users), but become integrated in the connected environment through the use of a mobile application. At the heart of this architecture stands the ITS Centre, serving as the vital intermediary between the mobile application and the connected entities via the G5 network. It encompasses an MQTT broker alongside a server that establishes a direct line of communication with the RSUs in the G5 network. Finally, the G5 network includes all the connected vehicles and infrastructure that support this type of communication, such as OBUs and RSUs.

Data flows bidirectionally, so both entities within cellular networks and those existing within the G5 network can exchange information. The ITS Centre plays a major role in making this possible. It has an MQTT broker and a server that bear the responsibility of forwarding ITS messages from the mobile applications to the ITS infrastructure situated within the G5 network, and vice-versa. The MQTT broker has a topic allocated for each type of ITS message (e.g. CAM, DENM). The server's role is defined according to the direction of the information flow. When the data is being sent from the G5 network over to the mobile application, the server takes on the role of receiving messages from the RSUs to which it is connected to and forwarding them to the MQTT broker — publishing them on their designated topics — via Ethernet. The mobile applications become subscribers to the relevant ITS messages' MQTT topics and are able to receive this information, process it and present it to the user through the Human-Computer Interface (HCI), auditory cues or vibration alerts. In contrast, when data is being sent from the mobile application to the G5 network, the server's role changes. Here, the mobile application uses Wi-Fi or a cellular network (i.e. 3G, 4G or 5G) to send data to the MQTT broker, which subsequently publishes it on the designated topic. CAM messages are automatically sent at regular, periodical intervals by the application, to advertise the user's position, speed, acceleration and direction. This data allows a comprehensive understanding of the user's current and projected trajectory. The server, aligned with this process, subscribes to the ITS messages' MQTT topics, which allows it to not only capture this information, but also to transmit it to the RSUs to which is it connected, via Ethernet and instructing them to forward the messages to other ITS equipment within the G5 network.

3.1.1 MQTT Protocol

MQTT is a standard, lightweight, publish-subscribe network messaging protocol. It usually runs over TCP/IP, although any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. Its design is ideal for connecting remote devices with a small code footprint and minimal network bandwidth requirements. It is highly scalable, being able to connect with millions of devices, and has security enabled, making it easy to encrypt messages using TLS and authenticating clients using modern authentication protocols (e.g. OAuth). Regarding reliability of message delivery, MQTT offers three defined Quality of Service (QoS) levels [50]:

- 0 — at most once: messages are delivered according to the best efforts of the operating environment. Message loss can occur. This could be used in a scenario where message loss is not a problem, because messages are being published at a high frequency, meaning another one will be published soon after.

- 1 — at least once: messages are assured to arrive, but duplicates can occur.
- 2 — exactly once: messages are assured to arrive exactly once. This could be used in a scenario where duplicate or lost messages could lead to problems in the data.

In this system, the MQTT client will use the QoS level 0, since it does not need to guarantee that all the messages being sent are delivered. Message loss is not critical, since messages will be sent at a rate of one message per second. If a message is lost, another one is sure to arrive soon after. The latency involved in using a higher QoS level would be greater than the time it takes for another message to arrive.

Publish-Subscribe Messaging Protocol

The MQTT protocol defines two types of entities in the network: a message broker and a number of clients. The broker is the server that receives all messages from the clients and then routes those messages to relevant destination clients. A client is anything that can interact with the broker to send and receive messages. When a client connects to a broker, it can subscribe any message topic in the broker. When it does so, it will start listening for messages published by other clients on that same topic. The client can also publish messages to the broker under any message topic by specifying the topic and message to the broker. Upon reception of the message published by the client, the broker will then forward it to all the clients subscribed on the specified topic [70].

In this architecture, the smart devices as well as the ITS Centre are all clients of the same MQTT broker. Two topics were created in order to facilitate the communication between the smart devices and the ITS Centre:

1. "smplusApp/cam"
2. "smplusApp/denm"

As soon as the ITS Centre connects to the MQTT broker, it subscribes the first topic to start listening for incoming messages sent by smart devices running the application. It also starts publishing the CAM and DENM messages received from the RSU to the first and second topics, respectively. On the other hand, when the smart devices connect to the MQTT broker, they subscribe both topics to start receiving the CAM and DENM messages that are forwarded by the ITS Centre and come originally from the RSU.

They also publish messages on the first topic to be forwarded to the ITS Centre. Since all elements are publishing ITS messages in topics that they subscribe, and in order to give the ability to each element to recognize if the received messages were self-generated, both the mobile application and the ITS Centre verify the ITS-S ID in the messages and discard them if they were themselves the message's original source.

3.1.2 ITS Equipment

Currently, ITS equipment from different manufacturers requires different methods to access and get the information logged in their storage. For the solution hereby presented, a Siemens' RSU was used. These RSUs provide an "XFER" interface [44] that operates with web sockets and provides the ability, for an authenticated user, to get all the messages logged in its storage. The XFER interface is based on the WebSocket protocol as described in RFC 6455 [30], ergo it provides full-duplex communication channels over a single TCP connection — meaning that, after the handshake, the exchange of the data is bi-directional with a minimum framing. This framing grants message-based data communication on top of the stream-based TCP protocol [44].

Observing the architecture's design (see Figure 3.1), the part of the communication that occurs via this interface is the data exchange between the ITS Server and the RSU. Thus, it is ultimately what allows the ITS Centre to communicate with the G5 network. The ITS Centre can use the XFER interface to get all messages being sent and received by the RSU, process them and publish them to the "smplusApp/cam" or "smplusApp/denm" MQTT topic, according to the type of message being handled. On the other side, there will be smart devices subscribed to these MQTT topics through the application and they will start receiving alerts related to the information received.

3.2 Mobile Application

Having established the hybrid network's environment, the ITS SM+ application was developed, aiming to warn SM+ users about incoming dangers by showing notifications on the mobile phone's screen indicating the direction from where the danger is approaching. The application advertises the user's geographical location periodically and receives information regarding the location of all other connected entities on the road. Upon processing all this data, it is able to determine whether one of these other vehicles is approaching in a way that could place the user in harm's way and, if so, it alerts the user to said danger, allowing them to react accordingly.

This chapter describes the application's information workflow, detailing how data related to the user's geographical location and orientation is obtained and processed and the logic behind the evaluation of whether the approach of another vehicle is considered dangerous for the user or not) (see Section 3.2.1) The characteristics of the alerts mentioned, as well as the description of how they are represented on the UI is available on Section 3.2.2.

3.2.1 Components and workflow description

The following sub-sections consist on a description of how the distance between the other elements on the road and the user is calculated and processed (see Section 3.2.1), followed by an explanation of how the calculation of the direction from which the danger is approaching is managed (see Section 3.2.1). The following sub-section details the events that occur when the application is initialized, and when it is running, along with the main components that are set-up upon the start of its execution (see Section 3.2.1). Then, a sub-section portraying how the C-ITS messages are originally generated before being sent by the application, and how they are handled when they are received from other devices is presented (see Section 3.2.1), followed by an explanation of how these messages are serialized and deserialized in order to be able to be sent and received via MQTT by the application (see Section 3.2.1). Finally, a sub-section describing the programming language and Integrated Development Environment (IDE) chosen to develop this application is displayed (see Section 3.2.1).

Distance processing

One of the main processing tasks preformed by the ITS SM+ application is correlating the user's geographical location with all the other connected elements' on the nearby roads. The application is aware of the other ITS-Ss on the road by the reception of the CAM messages that their OBUs send periodically, advertising their location. When a message is received from a different ITS-S' device, the first step that needs to be taken is to verify if this ITS-S is approaching the user or not. The application can obtain the other device's location through the CAM message received — seeing as it has fields that contain the latitude and longitude of the sender. In order to check if this ITS-S is approaching or moving away from the user, the application calculates the current distance between the user and the other device for each message received. Then, it compares the most recent distance value with the previous one: if it is lower, it means that the device is approaching.

This calculation of a distance between two sets of coordinates is done according to the Haversine formula [33]. The Haversine formula is a mathematical formula (see Equation 3.1) used in navigation and geography to calculate the geographic distance between two points on the surface of a sphere, such as the Earth. It inherited its name from the Haversine function, which is defined as: $\sin^2\left(\frac{\theta}{2}\right)$, where θ is the central angle between two points on the sphere. In the context of the Haversine formula 3.1, ϕ_1, ϕ_2 represent the latitude of two points and λ_1, λ_2 represent the longitude of two points, respectively.

$$\text{haversin}\left(\frac{d}{r}\right) = \text{haversin}\left(\phi_2 - \phi_1\right) + \cos\left(\phi_1\right)\cos\left(\phi_2\right)\text{haversin}\left(\lambda_2 - \lambda_1\right) \tag{3.1}$$

One can derive the Haversine formula to calculate the distance between two sets of coordinates, as illustrated on Equation 3.2.

$$\begin{aligned} \text{a} &= \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\left(\phi_1\right) * \cos\left(\phi_2\right) * \sin^2\left(\frac{\Delta\lambda}{2}\right) \\ \text{c} &= 2 * \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \\ \text{d} &= R * c \end{aligned} \tag{3.2}$$

where:

- d: distance between the two points along the surface of the sphere (i.e. Earth);
- R: radius of the sphere — Earth's radius in this case (i.e. 6,371 km);
- $\Delta\phi$: difference in latitude between the two points;
- $\Delta\lambda$: difference in longitude between the two points;
- ϕ_1, ϕ_2: latitudes of each of the points.

Although this provides the distance between two points and, as previously described in this section, makes it possible to verify if other ITS-Ss are approaching the user, in a real-life scenario, this information would not be sufficient to arrive to this conclusion. It would be necessary to include context information of the underlying map, to filter out situations where the ITS-S is approaching, but is in a different street — thus not being a threat to the user, because it might not even cross their path. At the same time, with this context information, a situation where another ITS-S is approaching would only be considered dangerous if this ITS-S was sharing the same road as the user; or if it was in a road where there was the possibility that they would meet at an intersection point. Another way in which this information would be valuable, would be for considering different road types. If the user is riding a bicycle and there is a car approaching in a parallel road, this should not be considered dangerous, as they would truly be in two different roads, and it would be unlikely that the car would steer towards the cyclist. On the other hand, if speed was a metric considered for this evaluation, the car going over the speed limit for that road could pose as a dangerous situation for two reasons: the probability of the driver losing control of the car and moving towards the cyclist would be higher, and the air friction between the moving car and the cyclist could be the cause of the cyclist losing balance and end up injuring themselves.

Orientation processing

The following step after knowing a danger is coming towards the user is to understand the direction from which is it approaching. To accomplish this, the application calculates the direction in which the other ITS-S is moving and presents a warning in the relative orientation to the user. Data from the accelerometer and magnetometer is used in the ITS SM+ application, and it is the basis of how the orientation processing is done.

The magnetometer is a sensor used for measuring magnetic fields and, if not influenced by another strong nearby one, it measures the Earth's magnetic field. The data is provides corresponds to the magnetic field strength along the X, Y and Z axis in micro Tesla (μT).

In light of the fact that Earth's magnetic field is parallel to its surface, when a device is aligned parallelly to the Earth's surface, the measurement of its heading can be accurately determined using the X and Y components of the magnetometer. Nevertheless, if the device is inclined, the accuracy of the heading value will decrease. For this reason, it becomes necessary to implement tilt compensation techniques through the incorporation of data from the accelerometer.

The accelerometer is a sensor used for measuring acceleration along the three axis X, Y and Z in meters per second (m/s^2) and its readings can be used for magnetometer correction.

Sensor readings are obtained through the SensorManager Application Programming Interface (API) and a global SensorManager object is created and initialized when the application starts, in the activity's onCreate() method (see Section 3.2.1). Then, in the onResume() method, listeners for the accelerometer and magnetometer sensors are registered, while the opposite happens in the onPause() method, to unregister them in case the application is closed. The SensorEventListener interface provides the onSensorChanged() method that will be called whenever new sensor values are received [9]. The ITS SM+ application overrides this method to apply a low-pass filter to the updated sensor values received, and then call a function to handle the heading updates.

The need for the use of this low-pass filter comes from the fact that using sensor data directly results in jittery heading values and compass rotation. As explained by Christiansen and Shalamov [66], applying a low-pass filter smooths out uncommon reading values, which are most often a result of noise, and keeps the common values.

As mentioned above, to calculate the magnetic heading, it is necessary to implement tilt compensation to guarantee accurate heading values when the mobile phone is not in a straight position. In this application, the technique chosen to accomplish this was to obtain a tilt compensated heading angle (azimuth). The determination of the magnetic heading involves the following steps [66]:

1. Calculate the cross product between the magnetic field vector (X, Y and Z readings from the magnetometer) and the gravity vector (X, Y and Z readings from the accelerometer), resulting in a new vector, denoted as 'H';
2. Normalize the gravity vector and the new 'H' vector to ensure consistent values;
3. Calculate the cross product between the gravity vector and the vector 'H', which yields a new vector 'M', pointing towards the Earth's magnetic field;
4. Apply the arc-tangent function to obtain the heading in radians;
5. Convert the angle obtained from radians to degrees;
6. Remap the heading angle from the original [-180, 180] range to the range of [0, 360].

It is important to note that there are two different "Norths" to consider: the true North and the magnetic North. The true North, also known as geographical North, refers to the direction towards the North Pole, which is the point where the Earth's axis of rotation intersects the surface. It remains relatively constant over time and is used as a reference for navigation, mapping and astronomical observations — it is the "fixed" reference point for direction [57]. The magnetic North refers to the direction pointed to by a magnetic compass needle, which is influenced by Earth's magnetic field. The magnetic North pole is the point on the Earth's surface where the planet's magnetic field lines are oriented vertically and it can vary in position over time, in response to changes in the Earth's magnetic core [48]. To navigate accurately, especially over longer distances, it is essential to account for the difference between true and magnetic North — which is known as magnetic declination (i.e. the angular difference between the direction of magnetic north and true north) [49]. When referring to the declination, it is denoted as positive when on the East side and negative when West side of the true North [48].

Changes in the Earth's magnetic field have been observed over the course of several years and two Geomagnetic models were established based on the information obtained: the International Geomagnetic Reference Field (IGRF) and the World Magnetic Model (WMM). In the implementation of the ITS SM+ application, the WMM was the model chosen to determine the magnetic inclination. This choice was due to the fact that there is an open-source Java implementation of it, created by Los Alamos National Laboratory [43] that uses coefficients that have been updated in 2020 and are valid until 2025.

The true heading of the user (i.e. the heading based on the true North) is calculated by adding the magnetic declination to the magnetic heading (i.e. the heading based on the magnetic North) and, when the result surpasses 360 degrees, a subtraction of 360 is made from the result, to keep it between the desired range of [0, 360] (e.g. if the result is 365 degrees, it is mapped to 5 degrees).

According to the acquired value for the user's true heading, the app's map orientation is also updated on the UI, rotating along with the user to match their orientation. Whenever the onResume() callback is called, the user's current screen rotation is obtained using the "defaultDisplay.rotation" property, provided by Android's WindowManager class [11]. Then, the difference between the user's true heading and the device's orientation is calculated and, if the result is lower than 0 or higher than 360, the value of 360 is added or subtracted from it, respectively, so that the result stays in the range of 0 to 360 degrees.

In view of this, in order to evaluate the direction from which another ITS-S is approaching the user, the absolute difference between the other ITS-S' heading — that is read from the CAM message received from it — and the user's true heading is calculated. The approaching direction will be the result of this absolute difference minus 180. Finally, if this result is negative, the direction on which the notification warning should appear, will be obtained by subtracting the absolute value of this result from 360. Figure 3.2 illustrates this situation — the green arrow and lines represent the user's heading and the red ones represent the incoming ITS-S' heading.

In the situation presented in this image, the ITS SM+ application's user has a 90° true

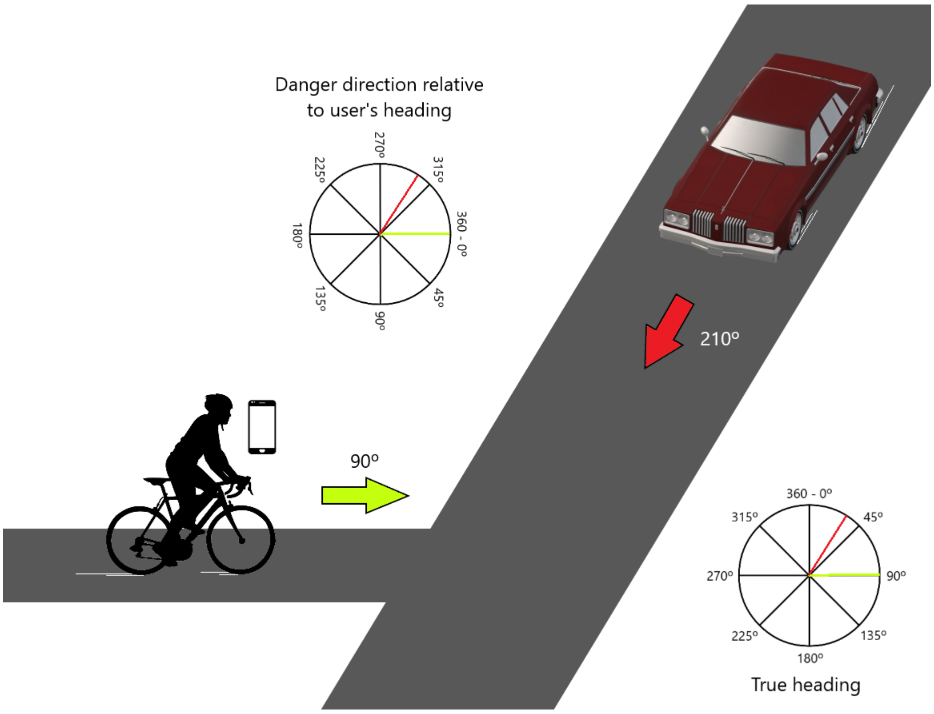


Figure 3.2: Danger direction calculation support illustration

heading value and the approaching car has a 210º true heading, which implies that if they meet at the road's intersection, the collision would happen around the 30º heading — which means that, from the user's perspective (i.e. considering that the user's heading is the true North and calculating the car's heading relatively to the user's movement direction), the danger direction would be of 300º and the warning should appear on the upper left corner of the screen, corresponding to the Northwest cardinal point (see Equation 3.3).

$$\begin{aligned} |90 - 210| &= 120 \\ 120 - 180 &= -60 \\ 360 - 60 &= 300 \end{aligned} \tag{3.3}$$

Application activities, setting up MQTT connection, map and location services

As it is the standard for Android applications, the ITS SM+ application is based on activities. This concept is of high importance to understand the information flow and the actions that are performed upon the application's initialization, during its execution and upon its interruption. The Activity class plays an indispensable role within the framework of an Android application, and the manner in which activities are initiated and structured constitutes a fundamental aspect of the platform's application model. In contrast to programming paradigms wherein applications are launched with a main() method, the Android system initiates code in an Activity instance by triggering specific callback methods that correspond to the specific stages of its life-cycle. In order to transition between these different stages, the Activity class offers a suite of six callbacks, namely: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). These callbacks are invoked by the system as the activity progresses into distinct states [1]. Each of this callbacks has the following purpose:

- onCreate(): it is called when the activity is first created. It handles the one-time initialization actions, such as setting up components and the UI, and binding data;
- onStart(): it is called just before the activity becomes visible to the user. It initiates the start of all processes and/or operations that should begin when the activity is about to start;
- onResume(): it is called when the activity is about to come into the foreground and become interactive. It is often used to resume any activities that were paused in onPause() and to obtain resources that should only be held when the activity is in the foreground.
- onPause(): it is called when the user is about to leave the activity, which means that it is no longer in the foreground, but it is still visible if the user is in multi-window mode. It is the indicated place to save user data, release resources and/or perform any cleanup that needs to happen before the activity is paused.
- onStop(): it is called when the activity is no longer visible to the user, which can occur when a newly launched activity covers the entire screen. It is a good place to release resources that are no longer needed and to stop any background services or processes that are specific to this activity.
- onDestroy(): it is called just before the activity is destroyed, either due to the user finishing it or system resources being reclaimed. This is where resources that weren't previously released in onStop(), should be released, and any final cleanup should be performed.

When initialized, thus in the onCreate() callback, these are the main actions performed in the SM+ mobile application:

- Set up the MQTT connection with the broker and subscribe the topic "smplusApp/cam" to receive CAM messages and the topic "smplusApp/denm" to receive DENM messages.
- Load the map from OpenStreetMap as the background layer of the main activity presented to the user.
- Set up SensorManager class properties to access accelerometer and magnetic field data from the Android phone.
- Set up the Location and LocationManager class properties to access GPS data.

OpenStreetMap (OSM)) is a free, open geographic database updated and maintained by a community of volunteers via open collaboration. Contributors collect data from surveys, trace it from aerial imagery, and also from other freely licensed geodata sources [52]. The Android application was integrated with OSM data through the use of OSMDroid — a widely used open-source library that provides Android-compatible OSM map views and tools — to be able to have the map as the application's background and display locations. This library was added to the Android's project dependencies and the map view it provides was initialized in the Android XML file, to become visible in the main application layout. Some map settings were configured, such as the tile source for OSM data, initial zoom level and the center coordinates on which the map should appear, as soon as the application is launched. The tile source chosen was "MAPNIK", which is an OSMDroid out-of-the-box, pre-configured map source containing the "standard" OSM map style [53]. Some extra functionalities were also implemented to allow different user interactions such as panning, zooming and tapping on the map markers or features. Lastly, an icon was added to it as an overlay to represent the user's current location on top of the map.

The SensorManager class is part of the Android Software Development Kit (SDK) and it is used for working with sensors on Android devices. It allows the retrieval of sensor data and registration of listeners to receive updates from these sensors, such as the accelerometer, gyroscope and magnetometer [61].

The Android operating system provides a Location API which consists on a set of software components and services for handling location-related functionality within Android applications. It allows developers to access location data, track device movements and build location-aware features into their apps [4]. The Android Location API provides a unified framework for working with various location providers [7], such as:

- GPS provider: uses the device's GPS hardware to provide accurate location information;

- Network provider: uses Wi-Fi and cellular network data to estimate the device's location, typically faster, but less accurate than GPS.
- Passive provider: listens for location updates generated by other apps or system components and can be used without directly requesting location updates.

The Location API supplies the Location data class, which is used to retrieve and work with location data in Android apps. It encapsulates and represents geographic location information, including latitude, longitude, altitude, accuracy, bearing and velocity [5]. The LocationManager class is also supplied by the Android's Location API and it provides access to the system's location services. It acts as a bridge between the application and the underlying location providers (e.g. GPS, network, passive), allowing the request of location updates, acquiring the last known location and specifying location criteria. Overall, the "LocationManager" is responsible for interacting with the location providers, managing and giving access to location services and location data sources on the device and the "Location" data class is the data object that gets passed around when location updates are received [6].

Upon start, the ITS SM+ application creates an instance of "LocationRequest" configured with the following parameters:

- Priority.PRIORITY_BALANCED_POWER_ACCURACY: sets the location request to have a balanced accuracy and power consumption priority. It tries to provide a reasonably accurate location without consuming too much power.
- DEFAULT_UPDATE_INTERVAL: default update interval for location updates, which was set to be of 1 second.

Then it defines an instance of "LocationCallback", which is used to receive location updates when the specified update interval is met. It overrides the "onLocationResult" method to handle location updates and, if the last location obtained is not null, it performs the following tasks:

1. Creates a "MyLocationOverlay" to display the user's current location on the map;
2. Enables location and follows the location on the overlay;
3. Adds the overlay to the map's overlays;
4. Calculates the magnetic declination based on the last received location;

5. Updates the "currentKnownLocation" variable with the new location.

Afterwords, it obtains an instance of "FusedLocationProviderClient" to interact with the Fused Location Provider, which is also a component of the Android Location API. It provides a high-level and efficient way to retrieve the user's location information and it is designed to combine data from various sources (e.g. GPS, Wi-Fi, cell towers and sensors), to provide accurate and up-to-date location information to Android applications. This component is optimized for battery efficiency and it integrates with the Android permissions system [4]. For this reason, permission checks were implemented, to verify if the application has been granted the "ACCESS_FINE_LOCATION" permission — which will allow it to access precise location. If the permission is granted, it requests location updates with the "LocationRequest" and "LocationCallback" formerly described; otherwise, it checks if the Android version is at least Android Q (API level 29) and prompts the permission request one more time. This double-check is necessary because from Android Q onward, Android introduced changes to the permission model, particularly related to location permissions. One significant change is the introduction of "background location access" permissions. On Android 10 and later, apps need separate permissions for foreground and background location access. Foreground access allows an app to access the device's location while it is in the foreground (visible to the user), whilst background access allows an app to access location data even when it's running in the background, and it requires additional user consent. The double check uses different methods to request this permission, seeing as in the previous Android versions there was just one method to request location access, while on Android Q and later, the permission request process for background location access is separate from foreground access, creating the need to use a different method for each.

MQTT C-ITS messages: message generation and handling

As mentioned in the previous sub-section, upon the application's initialization, it creates an MQTT client, connects to the MQTT broker. When it succeeds doing so, the application subscribes the defined topics to receive CAM and DENM messages and sets a task to be repeated every 3 seconds that generates CAM proto messages. This task fills the CAM messages fields with default values for this type of message, except for the ones that indicate the user's location (i.e. latitude, longitude and altitude). The station ID was also defined as the number "43858", which was chosen at random for testing purposes. This is not a problem in a testing environment, as long as it is guaranteed that no other ITS-Ss involved have the same station ID. After having the CAM protobuf message generated, it proceeds to serializing it, by converting it to a byte array. Finally, every time one of these messages is generated, it is immediately published in the "smplusApp/cam" MQTT topic. Because the application has also subscribed the same topic where it is publishing the generated CAM message, whenever it receives a message, it deserializes it and checks if the station ID contains a different value than its own. If not, it discards the message, seeing as that would be its own message.

C-ITS messages serialization

The mobile application is hereby presented as a solution for raising SM+ users' awareness for potential situations that could endanger them and it relies on the exchange of C-ITS messages with ITS equipment in the infrastructure to do so. As a means to allow a mobile phone running this application to send and receive C-ITS messages through MQTT, Protobuf messages were used as the data serialization format. Protocol Buffers (Protobuf) are a language-neutral, platform-neutral extensible mechanism for serializing structured data, developed by Google [55]. Through the use of Protobuf messages, it was possible to create a language-agnostic version of the C-ITS messages — the necessary data structures were defined using a language-neutral schema definition language (i.e. ".proto" file), mirroring the official data structures defined by ETSI — for the communications between the mobile application and the ITS Server. The Protobuf compiler (i.e. "protoc") was used to generate Java and Kotlin code, which simplified the process of parsing and encoding/decoding messages, reducing the susceptibility to human error. Protobuf's high efficiency is also an advantage worth mentioning in the choice of this solution, seeing as it produces smaller message sizes compared to many other serialization formats, which is crucial for optimizing network and storage utilization.

Programming Language and IDE

The ITS SM+ Application was developed in Android Studio, using the Kotlin programming language. Some of the reasons amongst why Android Studio was chosen as the IDE for this solution are the following [10]:

- It is the official IDE for Android application development, developed and maintained by Google — the creator of the Android platform —, which ensures that it is well-aligned with the latest Android features, updates and best practices.

- It offers a rich set of tools and features tailored specifically for Android development. It includes a user-friendly layout editor, code analysis tools, debugging capabilities, and integration with Android SDKs, emulators, and devices.
- It provides performance analysis tools to help identify and resolve performance bottlenecks, memory leaks and other issues that could impact the application's speed and efficiency.
- It has a range of built-in emulators and supports running apps on physical devices.
- It supports plugins and extensions, allowing the use of third-party tools, libraries and frameworks.
- There is extensive official documentation, tutorials and resources specific to Android Studio provided by Google.
- It has a large user community, which translates in the availability of many online forums, discussion boards and resources where it is possible to seek help, share knowledge and find solutions for potential challenges.

Regarding the reasons why Kotlin was chosen as the programming language to be used in the development of this solution, these are the main ones (refer to [3], [38]):

- It offers a more concise syntax compared to Java, which renders the code more readable and expressive.
- Its type system includes null safety features, reducing the likelihood of null pointer exceptions, a common source of crashes in Android applications.
- It supports functional programming concepts, such as automatically casting a variable to the appropriate type when certain conditions are met, reducing the need for explicit casting.
- It has built-in co-routines that enable efficient and concise asynchronous programming, making it easier to manage background tasks and maintains a responsive UI.
- It is officially supported by Android Studio, providing a smooth development experience with features like code completion, debugging and profiling.
- It is also officially supported by Google, which means that there are ample resources, libraries and community support available.

- It is a modern programming language and Google endorsed it as a first-class language for Android development, indicating its commitment and support for Kotlin in the Android ecosystem.

3.2.2 User Interface

The UI of the ITS SM+ application, consists of a set of different views and buttons defined over a ConstraintLayout. Android applications have a layout file, that is conventionally named "activity_main.xml", which defines the UI of the primary screen of an Android app. This layout file specifies how various UI elements (e.g. views, buttons, text fields, and images) are organized and displayed on the screen. A ConstraintLayout is a flexible and powerful layout manager in Android Studio that allows the assignment of constraints for every child view/widget relative to other views present, and it was the one chosen for this application. To define a view/widget's position in ConstraintLayout, it is necessary to add at least one horizontal and one vertical constraint for it. Each constraint represents a connection or alignment to another view/widget, the parent layout, or an invisible guideline [2]. The first view added to the application was the OSM map and it was set to cover the entirety of the screen. Over it, eight different views were added to show the warning of an incoming danger towards the user — one for each cardinal point (i.e. North, Northeast, East, Southeast, South, Southwest, West and Northwest). Each of these views was set with a specific shape, size and position to match the cardinal point where the danger is approaching from. The North and South warnings are rectangles positioned on the top and south center of the screen, respectively; while the East and West warnings are rectangles positioned on the right and left sides, vertically aligned with the middle of the screen (see Figure 3.3a).

The other warnings for the rest of the cardinal points are smaller rectangles positioned in the corners of the screen, according to the cardinal point they represent (see Figure 3.3b).

The background of each warning was defined in one of the XML files in the "drawable" resources folder, as it is the Android Studio's conventional folder to store graphics that can be drawn to the screen and retrieved with APIs [8]. The warnings are also animated with pulsating motion animations. These are defined in XML files and stored in the Android Studio's conventional resources folder to store animations that modify properties of the target object, denominated "anim".

At a functional level, when the application is first executed, the views are set to be invisible. When a warning is received, the view corresponding to the cardinal point

(a) North, South, East and West warnings on the UI

(b) Northeast, Northwest, Southeast, and Southwest warnings on the UI

Figure 3.3: Warnings for dangers incoming from all cardinal points on the UI

in the same direction from which the danger is approaching becomes visible, until a transparent button that is over it is pressed. At this point, the animated views become invisible once again.

Along all of these views, eight buttons were also added to the ConstraintLayout. Each one of these buttons has the same shape and size, and is positioned in the same place as the eight views described above; however, they are transparent. They are meant to be tapped by the user to acknowledge that they understood the danger notification received.

The result is that, when a road user approaches in a way that places the app user at risk, the view containing the warning that matches the direction from which the danger is approaching becomes visible and starts blinking, until the user taps it on the screen. Once the user touches the area corresponding to the warning, since there is a transparent button over it, it will trigger the actions that were defined to happen when the button is clicked.

Alongside this functionality, whenever a DENM message is received in the "smplusApp/denm" topic, a widget is added to the map on the location corresponding to the event received. This widget is essentially an icon that is indicative of the type of event that occurred. Section 4.2 of the next chapter contains screenshots taken of the application in use that demonstrate this functionality.

4

Tests and Results

To test the implementation of the mobile application developed and previously described on Chapter 3, three main scenarios were outlined. The first one had the primary objective of verifying if the hybrid network was able to guarantee that all elements involved in the communications were aware of each other's presence — thus, rendering SM+ users connected elements of the road. The second one aimed to test the functionality of showing the event warnings generated from the reception of DENM messages. Finally, the third one aimed to test if the solution found to warn SM+ users about potential incoming dangerous situations is valid, taking into account the latency involved. For this last scenario, two different sets of tests were performed: (i) measurements of the latency introduced by the hybrid network communications; (ii) measurements of the total latency corresponding to the hybrid network's communications, the ITS SM+ application's processing time and the user's reaction time.

4.1 First scenario: hybrid environment prototype demonstration

On the 29th of April, 2022, the "C-Roads Final Event - From C-Roads Portugal to C-Streets: C-ITS Handover" took place at ISEL's campus. For this event, a demonstration of the developed prototype working over the hybrid ITS-G5/cellular environment was made, within the scope of the Cooperative Roads (C-Roads) [19] and C-Streets [20] projects. Many partners of the project's consortium attended the event. During this occasion, a user test was performed by two ISEL members of the C-Streets project, alongside two members of the audience. By conducting these tests, the verification of the successful exchange of messages between the two networks was achieved. These tests involved the following equipment:

- Siemens' ESCoS RSU - installed in the demonstration room near a window, in order to have GPS signal
- Cohda's MK5 OBU - placed on top of a car, turning it into a connected vehicle
- Two smartphones running the ITS SM+ application
- Laptop running the host application (simulating the ITS Centre server)

The RSU and OBU were both engaged in running their proprietary applications, which enabled the initiation of sending CAM messages advertising the coordinates of their respective positions. The RSU was strategically located near a window inside the demonstration room, to acquire GPS signal. Within this same room, the laptop was set up running the host application, effectively operating as the ITS Centre's server. This application showed (see Figure 4.1) the positions of all vehicles on a map, regardless of whether they were communicating through G5 or through the ITS SM+ application. Concurrently, the OBU was set up atop a car's roof, thus turning it into a connected vehicle. This car was parked right outside the room housing the RSU. One of the smartphones was located inside another car, previously considered a legacy vehicle, due to its inability to communicate with the G5 network. Nonetheless, through the utilization of the ITS SM+ application, this smartphone was also granted the connected vehicle status. The application was configured to send CAM messages at regular intervals, with a frequency of one message per second. The second smartphone was placed in a bicycle and operated in a similar way to the one in the legacy vehicle. The sole difference between the two laid in the message transmission rate. For the smartphone on the bicycle, this frequency was adjusted to one message every three seconds. This adaptation accounted for the relatively slower pace of movement associated with bicycles, rendering frequent updates of their location less imperative.

Upon the completion of the setup process, every device's position could be observed in the host application. The smartphone inside the previously legacy vehicle was shown as a car icon and the smartphone on the bicycle was shown as a bicycle on the map. At first, all of these devices remained stationary. However, subsequent to a concise introduction about how the hybrid network was designed, a series of tests was carried

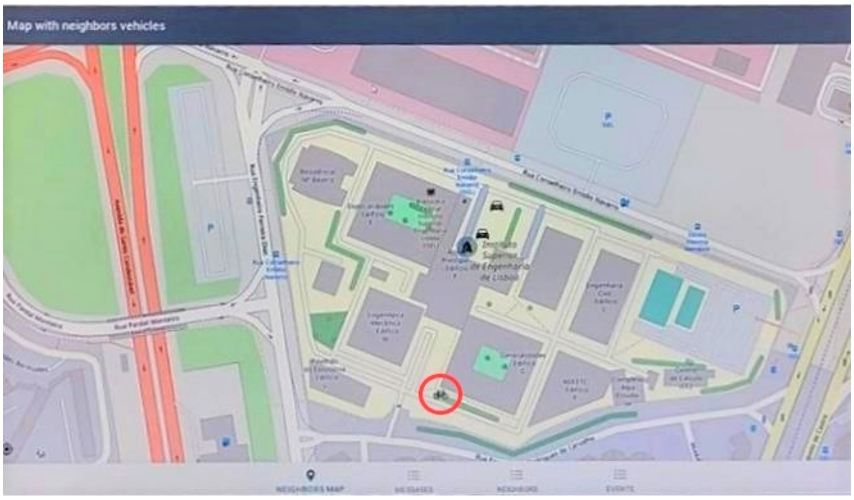


Figure 4.1: Host Application Demonstration (bicycle icon inside the red circle)

out. During this phase, a volunteer from the audience partook in the trials to evaluate the network and application's behavior.

The tests consisted on the volunteer embarking on a bicycle ride around the campus, while all the equipment was actively exchanging ITS messages. A map with an icon representing the cyclist — in this particular case, the SM+ user — moving around the campus was visually accessible to the rest of the audience, as depicted on Figure 4.1. Simultaneously, the cyclist possessed the ability to observe the positions of other vehicles and the stationary RSU, through the CAM messages they were sending and see icons for DENM messages on the event's position, if the RSU was set to send them.

During the final phase of the test, a total of four repetitions were completed with two volunteers. The conclusion was that the application needed to fix the message storage mechanism by limiting it and/or periodically clear them. For DENM messages, a magic number should be found to set a maximum limit to how many messages can be locally stored in the application. Once this limit is reached, whenever a new DENM message is received, the application should delete the oldest one, to accommodate the new entry. For CAM messages, a similar strategy should be adopted, with the difference that once the limit is reached, the message to be deleted should be the oldest message with the same station ID (i.e. a unique number that identifies the device that is sending the messages) as the one of the new message received.

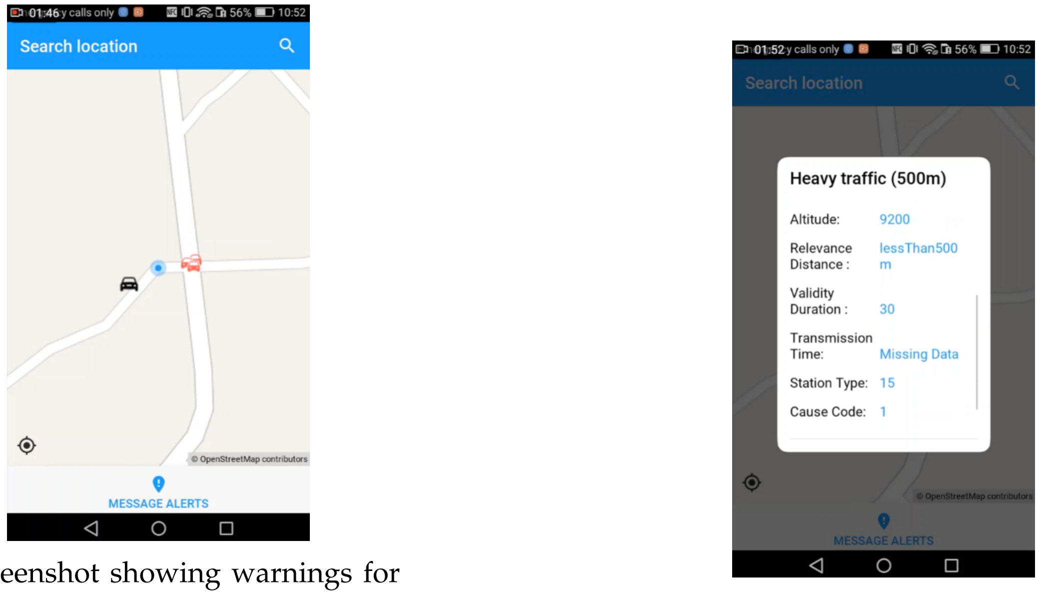
In spite of the need for further improvement, the tests were deemed successful in attaining their primary objective: the establishment of mutual awareness among all elements involved.

4.2 Second scenario: show warnings for traffic events with DENM messages

For this scenario, a test was developed where a DENM message of heavy-traffic was sent by the Siemens' RSU and a warning should have been presented to the user with this information, in the form of an icon on the map, on the event's location. In order to do so, a DENM message was created with this information (i.e. the cause code and subcause code fields were set to match the heavy-traffic event type, the event location was set to the desired coordinates and a relevance area of 5 meters was set for the purpose of this test) in the ITS Centre's server.

The use case consists on a user walking down a road and clicking on the event warnings as they appear, to see details about them. These warnings should only be presented as the user enters the relevance area set in the DENM message sent by the RSU. After creating the each DENM message, the server sent them to the Siemens' RSU through the XFER interface, which then broadcasted them to be received by the OBU. The OBU was running a script that made it echo the received messages, so the RSU received them again from the OBU and, when that happened, it sent them back to the ITS Centre's server using the same communication method. At this point, the server captured the messages and forwarded them onto the "smplusApp/denm" MQTT topic. As soon as the smartphone started running the application, it automatically set itself as an MQTT client and subscribed the topic (see Chapter 3, Section 3.2.1), so it received the messages and calculated the moment when the user entered each relevance area, by comparing their location to the events' and checking whether they were inside the radius set by this property or not. To test if the application was operating correctly, the smartphone started running the application outside of the relevance area of the events and, in the moment the user entered the relevance area, the alerts were shown on the smartphone's screen, as expected.

Figure 4.2 portrays two screenshots taken of the application in use during this test, on a Huawei P8 Lite mobile phone. The one on the left shows two events on the map: one for a car stopped on the road and another representing heavy traffic ahead (see Figure 4.2a). The one on the right shows some information regarding the heavy traffic event on the pop-up window that appears when the user clicks on the event icon (see Figure 4.2b). In this particular example, the heavy traffic message comes from one of the DENM messages received, with the field that indicates the type of event corresponding to an intense accumulation of vehicles at a 500 meter distance from the user.



Heavy traffic (500m)

Altitude: 9200
Relevance Distance : lessThan500 m
Validity Duration : 30
Transmission Time: Missing Data
Station Type: 15
Cause Code: 1

(a) Screenshot showing warnings for a car stopped on the road and heavy-traffic

(b) Screenshot showing details for the heavy-traffic event warning

Figure 4.2: Warnings for events on the road received through DENM messages

4.3 Third scenario: solution's viability assessment with latency measurements

It is crucial to study the latency involved in these communications, as it is vital to ensure that SM+ users have enough time to react effectively to potential dangerous situations. Considering all types of road users, some of them move faster than others: generally, vehicles outpace SM+ means of transportation. When these two categories of road users encounter a similar scenario, such as the need to break suddenly due to a pedestrian crossing the street, the driver of the vehicle will most likely be travelling at a higher speed, thus having less time to react, which translates in a need to alert them about this danger at an earlier stage. Ensuring that, when a danger approaches an SM+ user, the time it takes for the information to be sent out by the ITS equipment, processed at the ITS Centre, and transmitted to the user's smartphone is shorter than the time it takes for the situation to unfold — along with the user's response time to the warning — becomes a real necessity.

4.3.1 RTT latency assessment

The first set of tests aimed to calculate the latency involved in communications over the hybrid network through the evaluation of the Round-Trip Time (RTT) — which

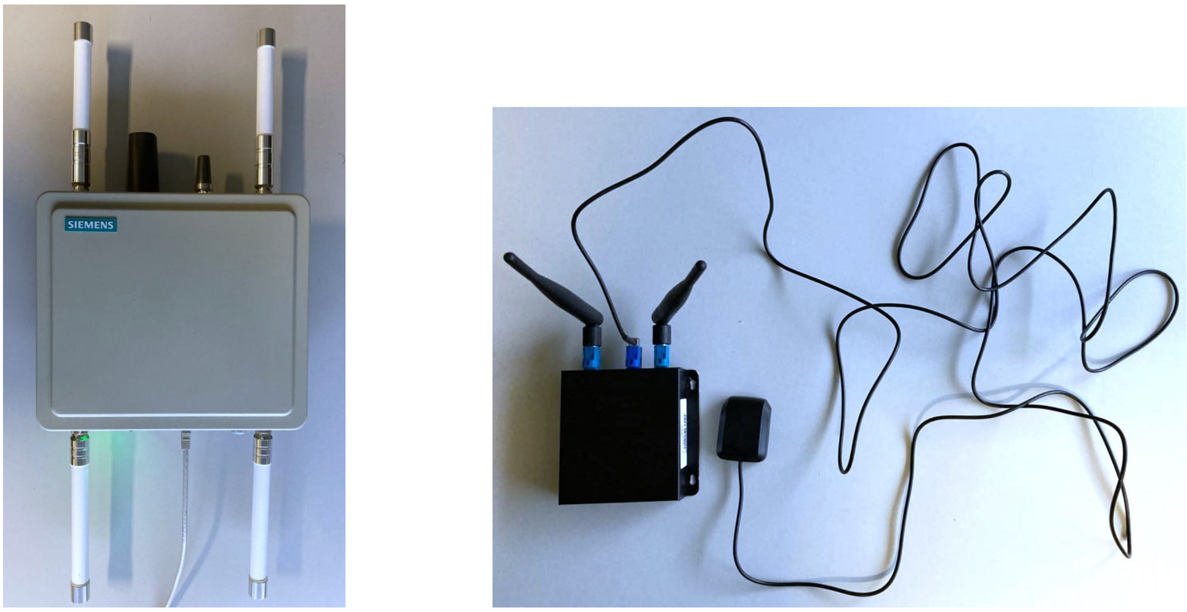

(a) Siemens' RSU (b) Unex's OBU

Figure 4.3: ITS Equipment

is the performance metric chosen to calculate the time it takes for a message to travel from the source (device operating over a cellular network) to the destination (OBU operating over the ITS-G5 network) and back again. To evaluate the latency in the hybrid network's communications, a series of measurements were performed using ISEL's MQTT broker, a Huawei Y6 smartphone, the Siemens' RSU previously mentioned in Section 4.1 (depicted in Figure 4.3a), and an Unex's OBU (depicted in Figure 4.3b).

ITS equipment from different manufacturers usually requires different methods to access and get the information logged in its storage. In this particular case, the Siemens' RSU was accessed through its proprietary XFER interface [44], which operates with web sockets. Authenticated users can utilize this interface to retrieve all stored messages. The XFER interface relies on the WebSocket protocol, outlined in RFC 6455 [30], providing full-duplex communication channels over a single TCP connection.

The ITS Centre can then use this interface to retrieve all the messages being sent and received by the RSU, process them and publish them on the MQTT broker. To avoid problems regarding device time synchronization during the latency measurements, the mobile application was configured to generate CAM messages, encode and send them. These CAM messages had an additional field indicating the current local time of the smartphone (UTC+1). The messages were sent from the smartphone, to the ITS Centre, which then forwarded them to the RSU and gave it the instruction to disseminate them.

Finally, the OBU received the messages and sent them back to the smartphone, using

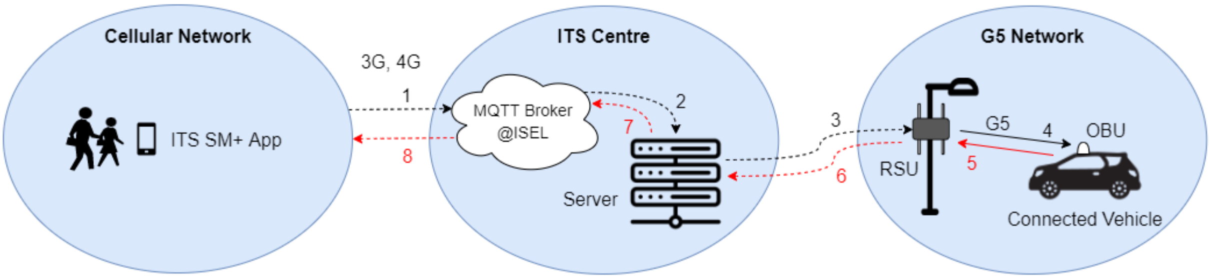


Figure 4.4: Latency Tests with ISEL's MQTT Broker Architecture

the same methodology. Upon the reception of each message, the mobile application made a comparison between the timestamp previously added and its new acquired local time. Through this process, the difference between the two time values was computed, facilitating an approximate determination of the RTT. The arrows shown in Figure 4.4 illustrate the path through which the CAM messages were sent, including the additional field containing the device's timestamp.

Three tests were performed with a duration of 10 minutes each. Messages were sent at a rate of one message per second, culminating in approximately 600 messages for each test, over both 3G and 4G networks.

Results

Two box plots, showcasing the obtained results, are visible in Figure 4.5. With 3G, about 99.3% of the latencies fell within the range of 64 to 159 ms, with half of them spanning from 90 to 117 ms and a median of approximately 100 ms. In the case of the 4G network, approximately 99.3% of the latencies were confined between 49 and 115 ms, with 50% of them ranging between 66 and 86 ms and a median of about 76 ms. The recorded minimum and maximum values were approximately 1.76 ms and 983.86 ms, respectively, both observed over the 3G network.

Furthermore, two Student T-Tests were performed to verify the existence of a statistical difference between the means of the measurements across the 3G and 4G networks. These calculations were made with the support of RStudio, an integrated development environment for the statistical computing programming language, R. In the initial test, the null hypothesis, denoted as $H_0 : \overline{x}_{3G} = \overline{x}_{4G}$, was evaluated, while its complementary assertion was the alternative hypothesis, expressed as $H_1 : \overline{x}_{3G} \neq \overline{x}_{4G}$. This entailed a two-sample, two-sided T-Test, assuming equal variances between the two populations. The mean of the latency recorded over the 3G network (M = 126.3244, SD = 109.5994, n = 1929) was hypothesized to be equivalent to the mean of the latency measured over the 4G network (M = 78.5733, SD = 23.4089, n = 1929). At a 95% confidence

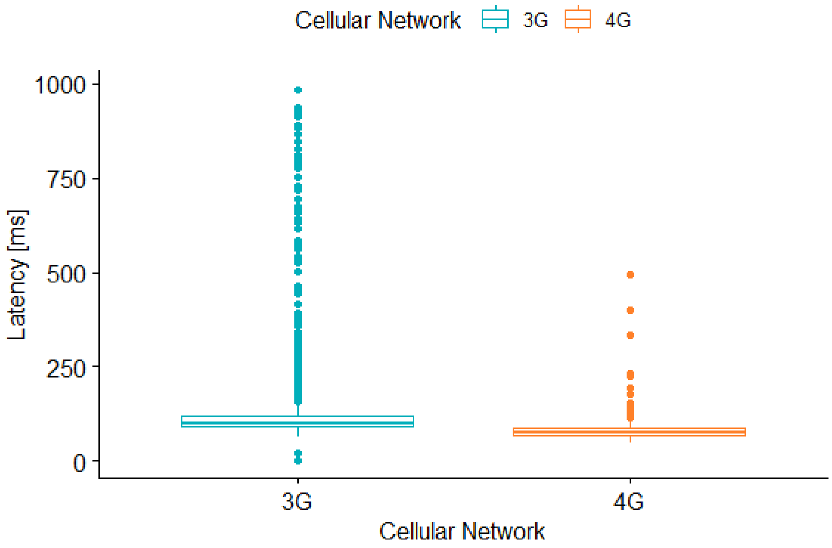


Figure 4.5: Latency Test Results over 3G and 4G cellular networks

interval ($\alpha = 5\%$), the null hypothesis was rejected, as evidenced by the test statistic $t(3856) = 18.709$ and a p-value of less than 2.2×10^{-16}.

Hence, a definitive inference can be drawn that a statistically significant distinction exists between the two means. The second test was aimed at determining if one mean was greater than the other. Therefore, the null hypothesis was formulated as $H_0 : \overline{x}_{3G} \geqslant \overline{x}_{4G}$, while the alternative hypothesis stood as $H_1 : \overline{x}_{3G} < \overline{x}_{4G}$. This constituted a two-sample, one-sided T-Test, assuming equal variances between the two populations. The mean of the latency observed within the 3G network (M = 126.3244, SD = 109.5994, n = 1929) was hypothesized to be greater than or equal to the mean of the latency measured over the 4G network (M = 78.5733, SD = 23.4089, n = 1929). At a 95% confidence interval ($\alpha = 5\%$), the null hypothesis was supported, as indicated by the test statistic $t(3856) = 18.709$, $p = 1$. Given the prior determination of a statistically significant difference between the two means, it can now be concluded that the mean latency observed within 3G communications does indeed surpass that within 4G.

According to Scholliers et al. [60], the exchange of messages at a Time-to-Conflict (TTC) of 5 seconds is deemed sufficient for validating the hybrid network in this type of applications. This means that, to ensure SM+ users' safety, they should be able to receive information regarding potential dangers approaching at a distance of 100 meters in urban settings (where vehicle speeds reach a maximum of 50 km/h), and 160 meters in extra-urban scenarios (where vehicle speeds peak at 90 km/h).

The test results obtained reveal that the latency measurements can fluctuate based on the current cellular network type, with 4G consistently exhibiting lower latency values.

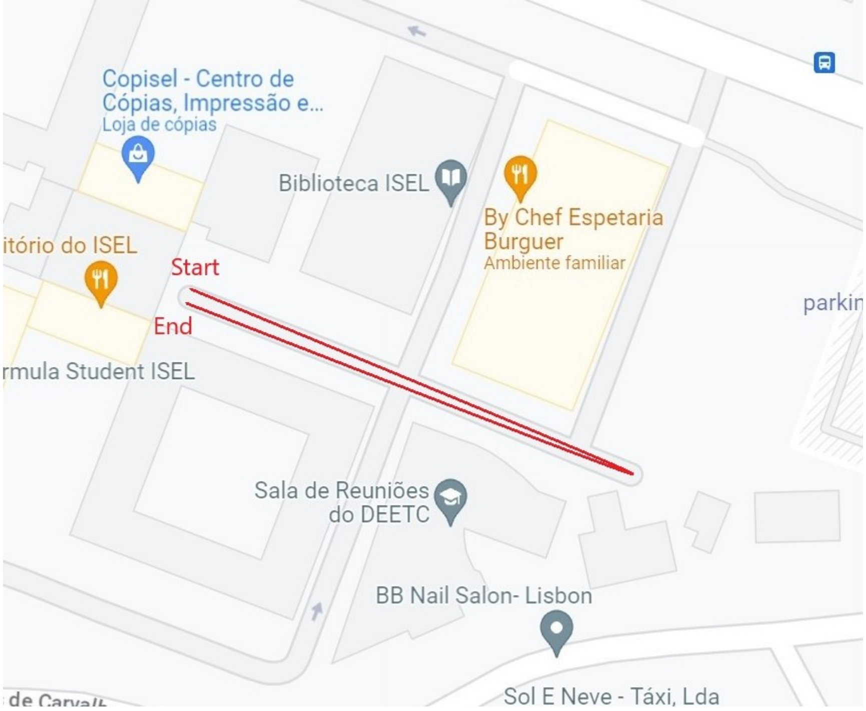


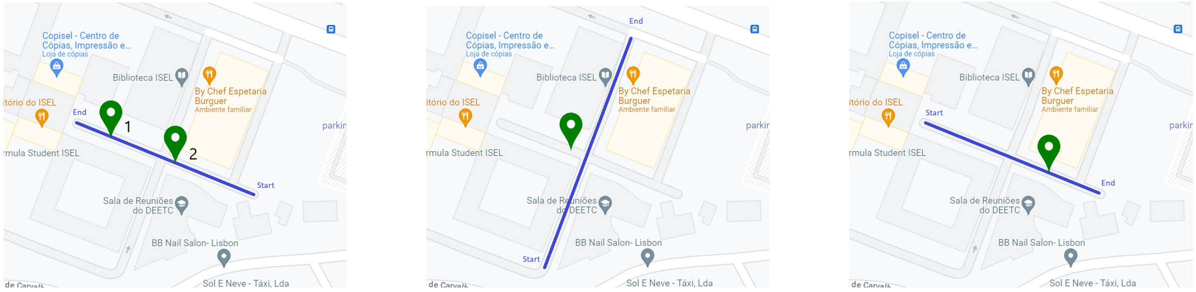
Figure 4.6: Path walked by the user running the ITS SM+ app for the reaction time tests

Seeing as all latency values found lie below the 5 second maximum RTT found by Scholliers et al. [60], this architecture can be deemed reliable for its intended purpose of raising SM+ users' awareness to incoming dangers.

4.3.2 Latency tests involving application processing and users' reaction time

This test was performed considering a use case of a user walking down a road while multiple other ITS-Ss approached them in a way that put them at risk. For this use case, the dangerous situation was the simulation of a vehicle with an OBU approaching the user. The aim was to verify if the time it took from the moment the CAM message was sent from a vehicle's OBU to the ITS Centre's server and then to the user's smartphone, along with the time it took for user to click on the warning button area to acknowledge the understanding of the warning, was lower than the time it would take for the vehicle to reach them. This test was done inside of ISEL's campus and the path walked by the user is illustrated on Figure 4.6.

Three sets of CAM messages were defined to simulate five different vehicles approaching from different directions. The first approached the user from the front, and the path it travelled is represented on Figure 4.7a. The set of CAM messages defined for



(a) Path traveled by the first and fifth vehicles

(b) Path traveled by the second and fourth vehicles

(c) Path traveled by the third vehicle

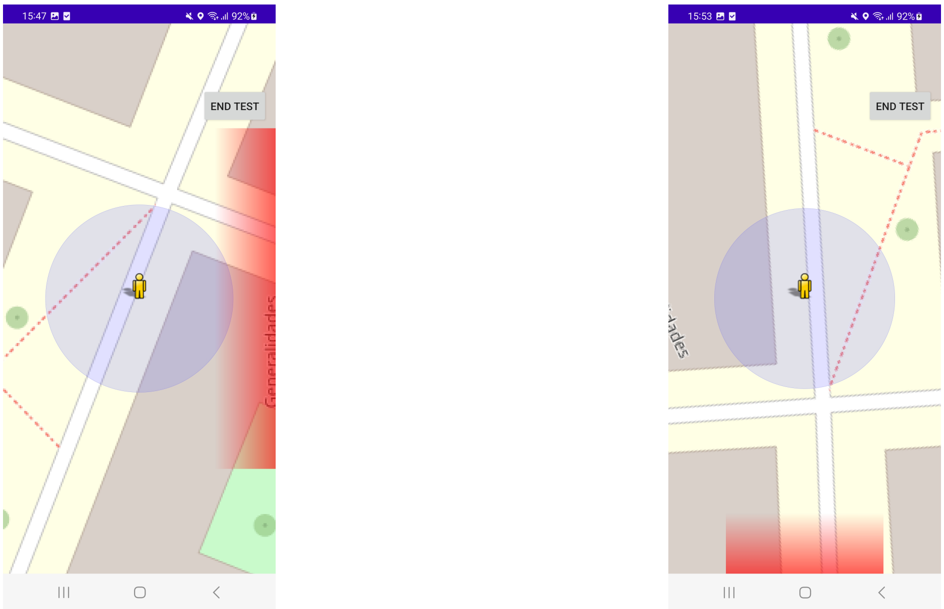
Figure 4.7: Paths travelled by the all simulated vehicles

this vehicle's travel path contained ten CAMs, each one with a set of coordinates corresponding to this path, ordered according to the "Start" and "End" locations illustrated in the figure. The same goes for the other vehicles' travel paths. The second vehicle approached the user from their right-side (i.e. East), according to the path shown on Figure 4.7b; the third vehicle approached the user from behind, according to the path illustrated on Figure 4.7c; finally, the fourth and fifth vehicles travelled the same paths as the second and first ones, respectively. However, the publication of the sets of CAM messages that simulate these vehicles was only triggered after the user turned back at the end of the road and was walking back the same way, thus the warnings appeared on the opposite sides as they previously did (i.e. the fourth vehicle appeared from the user's left-side and the fifth vehicle appeared from behind them this time).

The first CAM message related to each vehicle was sent when the user was at a 5 meter distance from a specific geographical point, which is represented by the green markers on the figures. Regarding Figure 4.7a, the green marker next to the number "1" represents the point considered for the simulation of the first vehicle and the one next to the number "2" represents the point considered for the simulation of the fourth vehicle. Each CAM message belonging to this vehicle's travel simulation was sent with an interval of 1 second, as it is the default defined in "ETSI EN 302 637-2 V1.4.1" [24].

The two images present on Figure 4.8 show screenshots that were taken during the tests, with the application running on a Samsung Galaxy S21 Ultra smartphone. Figure 4.8a corresponds to the simulation of the first vehicle, that approaches the user from their right side, and Figure 4.8b corresponds to the fifth vehicle, that approaches the user from behind.

For the purpose of simulating the car approaching the user, the information flow is more complex than in a real-life scenario, where the OBU would be set up on a car's roof and send CAM messages advertising its geographical location. Seeing as these



(a) Warning of a car approaching from the right side of the user

(b) Warning of a car approaching from the back

Figure 4.8: Screenshots taken during the tests for the second and fifth vehicles

tests did not involve a real vehicle, the messages containing the information to simulate these vehicles had to be pre-configured in the ITS Centre's server. As previously explained in Chapter 3, Section 3.2.1, the ITS SM+ application sends periodical CAM messages advertising the user's location. Upon the reception of this information, the server checks if the user is in a distance shorter than 5 meters to the location point chosen to trigger the sending of the CAM messages by the OBU. When this condition is verified, the server sends the messages with the predefined data to simulate the corresponding vehicle to the RSU. Then, the RSU broadcasts this information, which is received and echoed by the OBU. When the RSU receives the echo of the message from the OBU, returns it back to the ITS Centre's server, which finally publishes it by MQTT to be received by the mobile application.

The aim of this test scenario was to verify if the communication's inherent latency plus the user's reaction-time to interact with the warning presented in the app was lower than the time it would take for the simulated vehicle to reach them. The vehicle's speed considered was of 50 km/h — which is the maximum velocity allowed inside municipal areas. This was the value chosen for the speed limit, because it is the type of area commonly shared by SM+ users.

Figure 4.9 illustrates the workflow for this use case. When a message is received from the mobile application and the ITS Centre's server verifies that the user is within the 5 meter distance from the predefined point to start the vehicle's travel simulation, the

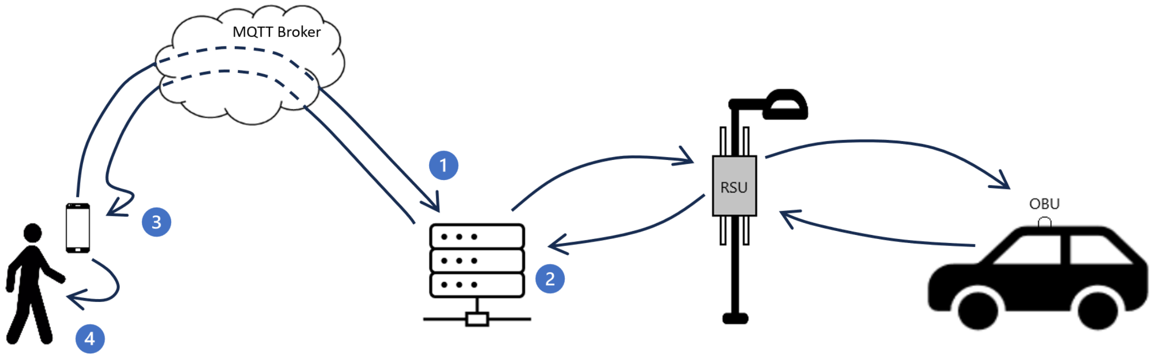


Figure 4.9: Information flow for the user's reaction-time tests

timestamp of that moment [35] is registered and added to a list. This list will have five entries: one for each instant when a vehicle's travel simulation was triggered. Each number on Figure 4.9 represents a stage at which instants were registered to be analyzed afterwards. The number "1" corresponds to the moment when a vehicle's approach simulation was triggered. When the ITS Centre receives the messages back from the ITS equipment, it records the timestamp once more and saves it on a different list (represented by the number "2" in the figure). If a real vehicle with an OBU had been used for this scenario, the test would be simpler: the timestamp at which the CAM messages had been sent would be generated by the OBU itself — seeing as CAM messages have a specific field for this purpose [24] — and it would be sufficient to compare that instant with the instant at which the user clicked on the warning buttons. However, because the vehicles were simulated for the tests, the generation time of the messages was set by the ITS Centre's server when it triggered the sending of each one of them. When the test is finished, before the server's code execution is stopped, the lists are written to a text file for posterior analysis.

Upon the ITS Centre's reception of the messages from the ITS equipment, it publishes them on the "smplusApp/cam" MQTT topic and they are then received by the mobile phone, which presents the corresponding warnings to the user. In the mobile application, two different moments are also recorded in lists. The first is the moment when the message was received by the application (represented by the number "3" in Figure 4.9) and the second is the instant when the user clicks on the warning button (represented by the number "4" in the figure). For these tests, another button was added to the UI (see Chapter 3, Section 3.2.2), responsible to send all the registered timestamps — corresponding to the instants when the user clicked on the screen after seeing the notification, and a "true" or "false" value, indicating whether the user pressed the correct button (i.e. the button on top of the warning that appeared on the screen) or a wrong one.

In order to determine if the architecture and solution proposed in this document is viable, it is necessary to measure the time it takes for the messages to come from a vehicle's OBU to the moment when the user clicks on the warning button to verify that they were successfully notified of the incoming danger — and check if that time guarantees that the user was warned before the danger reached them. To accomplish this objective, there is a need to discard some of the times that were measured in the test, namely the time it took from the moment the vehicle's simulation was triggered (see icon with the number "1" on Figure 4.9) until the messages were received by the OBU. Since the time it took for the messages to be triggered by the ITS Centre's server until they were received by it again is known — which corresponds to the RTT for the CAM messages from the ITS Centre to the OBU and back —, dividing this total in half results in the an approximate value for the time the CAM message took to come from the OBU to the server. With this data, it is possible to calculate the total time spent in a communication between a vehicle and an ITS SM+ application user, in a simulation close to a real-life scenario.

These tests counted with the participation of four voluntaries, ranging from 23 to 29 years of age, and all smartphone users. Each test was done under the same conditions: all participants travelled the same path, the coordinates used for the simulation of the vehicles were the same, as well as the location where each vehicle's movement was triggered. Nevertheless, it is important to note that not all participants walked at the same speed, and one participant in particular reported barely looking away from the smartphone's screen, while the others reported having used the application in the same manner as they use other navigation applications — frequently looking at the screen, but still paying attention to their surroundings. None of the participants was previously aware of the moment when the warnings would appear, in order to keep the experiment as close to a real-life scenario as possible. Moreover, the tests were performed twice: once with the smartphone connected to 3G and a second time with the smartphone connected to 4G. To test with 4G, the order in which the simulations for each vehicle was triggered had to be changed, to guarantee that the element of surprise was still at play, and that the volunteers did not know from where they would be coming next.

Results

Initially, the latency associated with the OBU sending CAM messages to the ITS SM+ application was analyzed. This latency is a result of the sum of the times measured

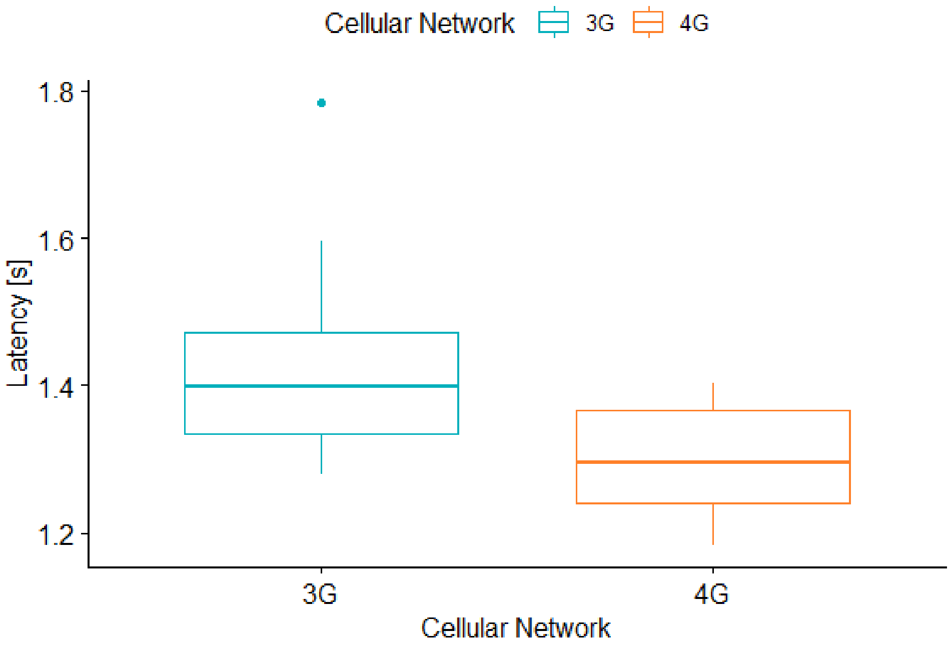


Figure 4.10: Latency test results for the OBU sending CAMs to the ITS SM+ App

between the OBU and the ITS Centre's server and between the latter and the smartphone. It was found that, using the 3G cellular network, 99.3% of the latencies lie between 1.279 and 1.595 seconds [s], with 50% of them ranging between 1.335 and 1.472 s, and with a median of 1.399 s. With 4G, 99.3% of the latencies lie between 1.182 and 1.403 s, with 50% of the ranging between 1.241 and 1.367 s, and with a median of 1.296 s (see Figure 4.10).

Regarding the users' reaction-times recorded, Figures 4.11 and 4.12 effectively show that participant "3" had a significantly lower reaction-time compared to the others. This participant reported having performed the test continuously looking at the phone's screen, while the others were more aware of their surroundings and glanced at the phone to check for notifications periodically. Regardless, all users stated that the visual notification was sufficient to make them notice the incoming danger, although it was also commonly agreed between all participants that an audio warning or a vibration would be a great improvement as ways to provide these alerts. The three other users' reaction-times were a lot more similar, around the 3 second mark. It is important to notice that these values do not stand only for the user's reaction-times (i.e. the time it takes for the user to press the button once the warning appears on the screen), but it also includes all the processing time the application needs — to calculate the distance, orientation, and to handle sensor and UI updates —, from the moment it receives the messages from the server to the moment the user presses the button. Independently of

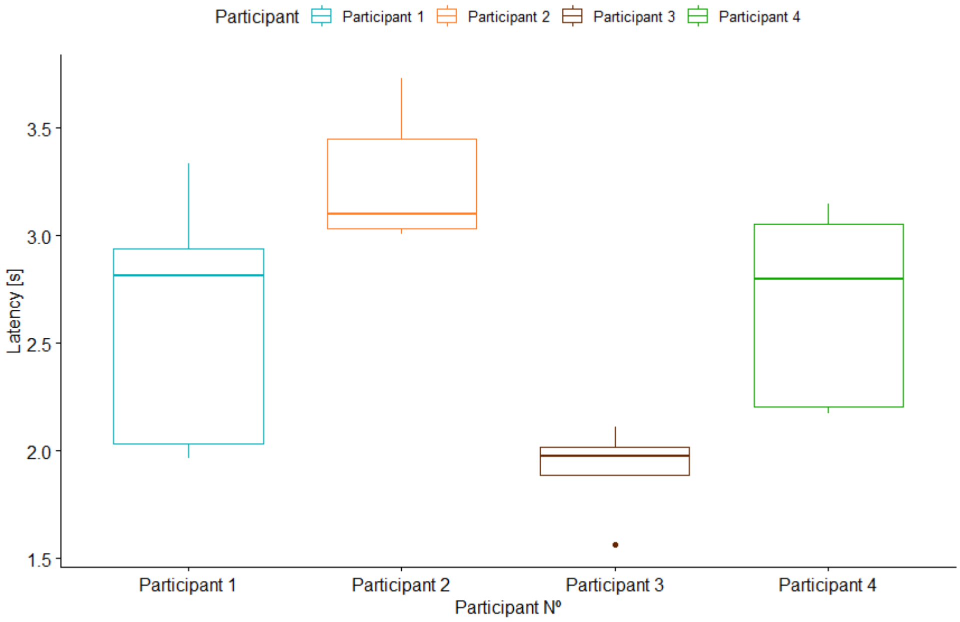


Figure 4.11: Participants' reaction-time measurements (over 3G)

each participant's individual measurements, comparing the results obtained with the smartphone connected to 3G or 4G, it was verified that, over 3G, 50% of the latencies recorded were between 3.349 and 4.351 s, with a median of 4.079 s; over 4G, 50% of the latencies were found between 3.051 and 4.109 s, with a median of 3.499 s.

Finally, the results obtained for the total latency involved in these exchange of messages were based on the sum of the previous results (i.e. the time it took for the CAM messages to be sent by the OBU and received by the ITS SM+ application and the time it took for the information to be processed, presented to the user and for the user to react accordingly). Figures 4.13 and 4.14 aim to show these results for each participant involved in the tests. Once more, it is clearly visible that the fact that participant 3's lower reaction-time greatly influenced the results. Nonetheless, considering the other participants who registered higher measurements, there was approximately a total latency of 4 seconds for this scenario over 4G and slightly higher ones over 3G. Additionally, it was also verified that none of the participants ever pressed the wrong button, which is indicative that this type of visual warning is a good basis to notify the user's of potential risks — which does not rule out the possibility of improving the user experience by providing other forms of notifications.

Furthermore, two Student T-Tests were performed in order to verify if there is statistical difference between the means of the measurements in 3G and 4G networks. For the first T-Test, the null hypothesis considered was $H_0 : \overline{x}_{3G} = \overline{x}_{4G}$, while the alternative hypothesis was its complement, $H_1 : \overline{x}_{3G} \neq \overline{x}_{4G}$. This was a two sample, two-sided

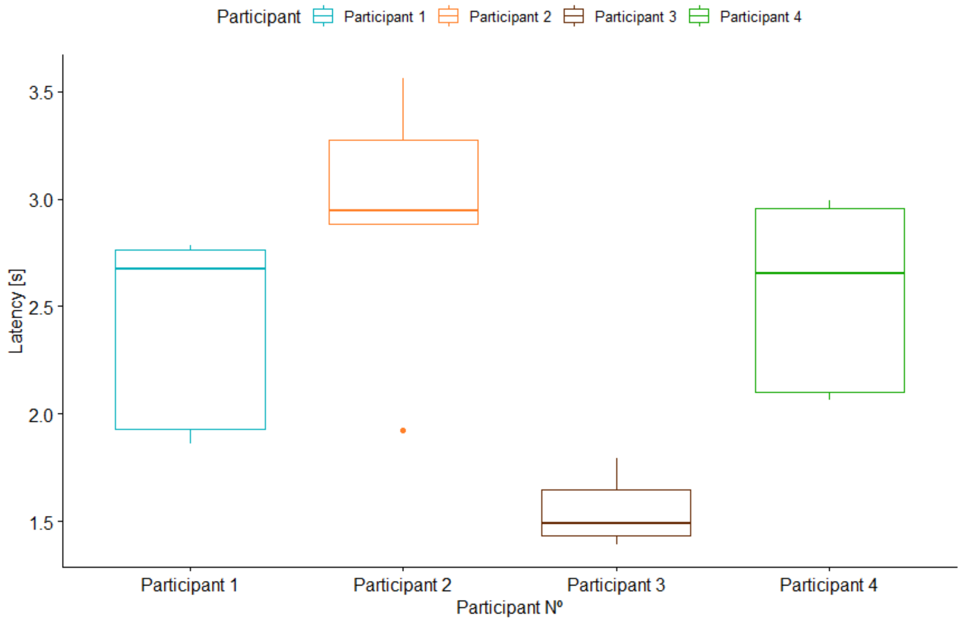


Figure 4.12: Participants' reaction-time measurements (over 4G)

T-Test, assuming the variances between the two populations were equal. The mean of the latency measured over 3G (M = 3.904, SD = 0.6524233, n = 20) was hypothesized to be equal to the mean of the latency measured over 4G (M = 3.577, SD = 0.6829665, n = 20). At a 95% confidence interval ($\alpha = 5\%$), the null hypothesis cannot be rejected, $p = 0.1307; p > \alpha$, which means that there is not enough evidence to conclude that the means of the 3G and 4G measurements are significantly different. Although the sample means suggest that the 3G measures have a slightly higher mean than the 4G ones, this difference is not statistically significant based on the T-Test results. In addition, the confidence interval includes zero, which also supports the conclusion that the difference in means is not significantly different from zero. The second test aimed to verify if one of them was greater than the other, therefore the null hypothesis considered was $H_0 : \overline{x}_{3G} \geqslant \overline{x}_{4G}$ and the alternative hypothesis was $H_1 : \overline{x}_{3G} < \overline{x}_{4G}$. This was a two sample, one-sided T-Test, assuming the variances between the two populations were equal. The mean of the latency measured over 3G (M = 3.904, SD = 0.6524233, n = 20) was hypothesized to be greater or equal to the mean of the latency measured over 4G (M = 3.577, SD = 0.6829665, n = 20). At a 95% confidence interval ($\alpha = 5\%$), the null hypothesis cannot be rejected either, $p = 0.9346$. In conclusion, there is no statistically significant difference between the means of the 3G and 4G measurements. This could be due to the fact that the sample size is small. In order to properly evaluate these results and draw conclusions by comparing them to the ones described in Section 4.3.1, a similar sample size should be used here, as to the one used for the simpler latency

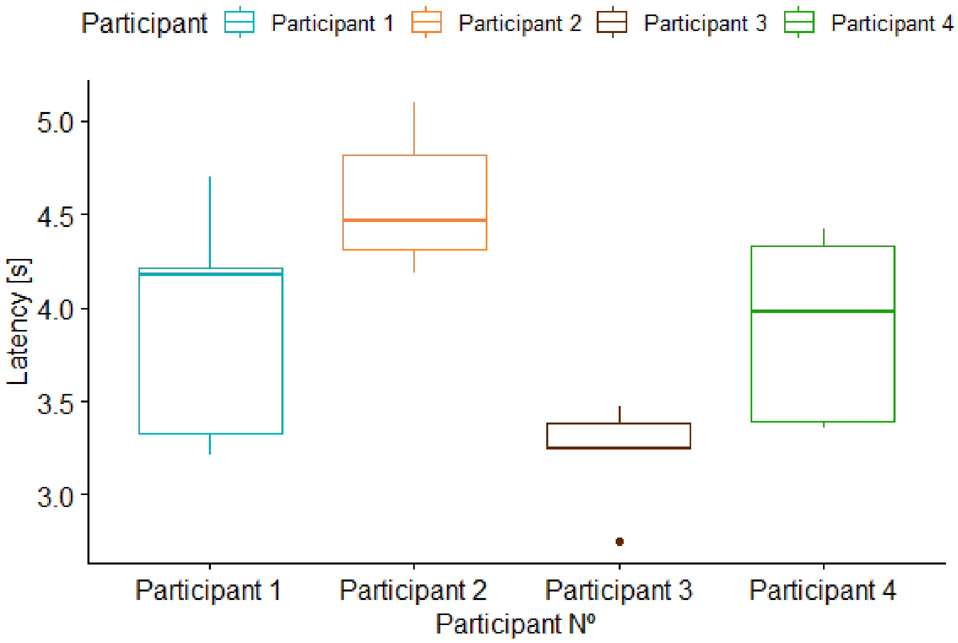


Figure 4.13: Overall time measurements for each participant (over 3G)

tests.

Altogether, considering a maximum of 5 seconds to guarantee that the user is alerted about potential dangers, in a municipal area where the speed limit is of 50km/h, this solution would only be viable if the user was warned about the danger around 69.4 meters away from the predicted point of collision. In a municipal or urban area this may not possible, because these areas generally contain many intersections and it might not be possible to predict when another ITS-S nearby can actually become a danger for an SM+ user or not. However, it is important to note that most of the latency introduced by this solution was due to the application's processing time. Comparing these results to the ones verified in Section 4.3.1, the latency verified in this chapter is significantly higher, even when analyzing only the latency involved in the OBU sending a message to the ITS Center. This could be a result of the fact that the ITS Centre was under a higher processing demand, due to the fact that it was running multiple threads: one to handle the reception of CAM messages from the ITS SM+ application advertising the user's location, another to verify if the user was in distance equal to or lower than 5 meters to the predefined point from whereon the user should start receiving alerts for the incoming vehicle, another to trigger the sending of CAM messages to the RSU when the previous conditions was verified, and another to handle message replies from the RSU. It is important to note that the code running in the ITS Centre and in the ITS SM+ application did not undergo an optimization process to guarantee the its best performance. For this reason, this solution should not be disregarded yet, but these results

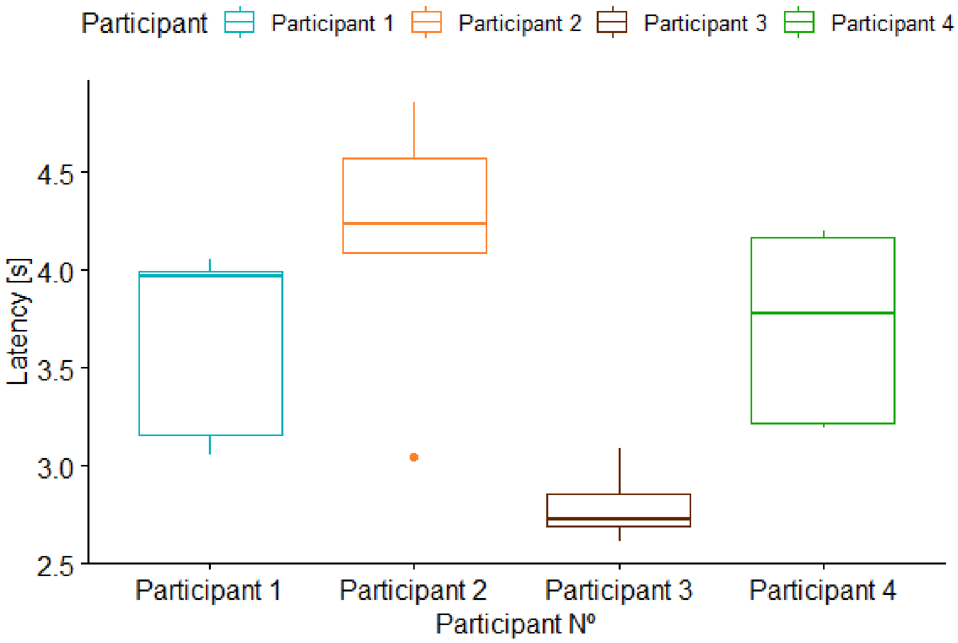


Figure 4.14: Overall time measurements for each participant (over 4G)

should be taken into account for future developments in this area.

5

Conclusions and Future Work

The aim of the development of the solution presented in this document was to create a tool to help protect SM+ users, who are the most vulnerable type of road users, from potential dangers along their travels. A hybrid network was developed to allow communication between cellular networks or Wi-Fi and ITS equipment that exchanges data over the standard for V2X communications — the G5 network. Furthermore, an Android mobile application was developed as a means to present alerts to SM+ users in regard to possible hazards that might come along their travel paths.

As a result of this research, the hybrid network can be deemed viable for this type of solution, seeing as it successfully allows different devices that operate over 3G, 4G, 5G or Wi-Fi to intercommunicate with equipment operating over G5. In future applications similar to this one, it is expected that the 5G network will be more commonly used, due to its better performance. Taking into account other related works that involve communications over cellular networks and the exchange of data with ITS equipment, it is important to stress that the hybrid network developed in this project is a state-of-the-art approach to connect today's solutions to the future's.

The mobile application produced for this purpose can also be considered a viable way to alert SM+ users of potential risks, however, certain aspects would need to be improved to guarantee their safety. The software developed as described in this document is not sufficiently reliable at this stage, although it presents as a launching pad for future improvements. The algorithm behind the Android application that determines whether the approach of another person or vehicle will put the user at risk is prone to generate false positives. Within this context, a false positive would be the application evaluating a situation as dangerous when in reality it wasn't. Take for example a scenario where a pedestrian is walking on the sidewalk and a car passes by, under the speed limit. This is a normal scenario that will happen constantly in an urban scenario and should not be considered dangerous for the user. However, in this prototype, the algorithm could deem it so. In conclusion, context information should be taken into account and integrated into the algorithm, so it could predict these potentially harmful situations more accurately and filter out a lot of cases where other entities are approaching the user, but are in separate roads that will not intersect the one the SM+ user is on. From the user perspective, to be constantly receiving warnings for false positives, will result in a negative experience. It is likely that after some time, the user will start to ignore the warnings. Bearing in mind that this is a critical application, it it crucial to guarantee that the user is effectively warned when a danger approaches to guarantee their safety. The algorithm must be trustworthy and dependable — thus, the warnings should be accurate as to the determination of when a situation is dangerous or not. Through the tests executed, it was found that the UI should include the implementation of other ways to notify the user of the potential dangers, such as sound and/or vibration alerts. The prototype has a non-intrusive UI in the way that it is based on visual warnings and does not distract the user from the path ahead. When a warning appears on the screen, the user's attention is pulled to it, due to the colour and shape that pop up to represent the incoming danger. However, if the user does not notice any warnings, it is natural that they would take a peek at the screen, to make sure that no warnings were in fact received. Adding a sound and/or vibration to the warning would be beneficial, seeing as it would grab the user's attention to the screen more effectively, so they wouldn't feel the need to look at it periodically to make sure no warning had appeared yet. It also became apparent that the mobile application should take into account user privacy aspects. The SM+ application needs the user's accurate geographical location constantly during its use and, on top of that, it broadcasts the user's location to all listening devices. In light of this, the advertisement of the locations should be done in a way that would prevent the tracing back of geographical points to a specific user, guaranteeing their anonymity.

The hybrid network's architecture viability was tested performing latency trials, aiming to calculate the End-to-End (E2E) RTTs between the two networks. Because the RTTs measured during these tests, over both 3G and 4G networks, were always under 1 second, thus respecting the 5 second maximum RTT limit found, the architecture was deemed viable. It also became apparent that using 4G networks, the latency values are consistently lower than using 3G. This means that each user's time to react will be dependent on the cellular network available at a certain moment. The latency assessment shows that, depending on the cellular network, the values range from 1 to 984 ms, demonstrating that the hybrid network is a viable solution for integrating SM+ into the ITS ecosystem. When combining this with all the necessary processing to compute positioning, distance and orientation algorithms asynchronously, as well as human reaction times, the latency recorded greatly increases and invalidates the certainty of the solution's viability, especially considering that it would be used as means of improving safety — thus, it could not be unstable, as it is meant to be used for a critical application. However, these findings cannot be disregarded either, since the results obtained in the tests with the participants using the mobile application and reacting to the warnings presented to them were inconclusive.

Future Work

As future work, the latter tests should be repeated with a higher sample size and it would also be important to do them with the smartphone operating over 5G — seeing as it is the type of cellular network that is becoming more commonly used and it introduces lower latency when compared to 3G and 4G. The software developed for the Android application could also be optimized with special emphasis on the information processing time and memory management to avoid introducing unnecessary overhead and increasing communication latency. The user interface could also be enhanced by adding extra features and by improving the ones already implemented. The algorithm applied to manage the map's rotation along with the smartphone's orientation changes should be smoother and well as the refresh rate of the warnings an incoming ITS-S passes by the user.

There are two main features that would greatly add to this solution's applicability: context recognition and alternative route suggestion. The former would help prevent disseminating unnecessary alerts. This would allow the filtering out of situations where other ITS-Ss might be approaching the user, but when they are in different roads that do not intersect with the one the SM+ user is on, they do not pose as a threat to them — therefore no notification would be processed or shown. Furthermore, it could also allow the recognition of different types of roads and not show warnings for cyclists who are on bicycle lanes or pedestrians on sidewalks, when a vehicle is simply passing alongside them under the speed limit, for example. The latter would be helpful considering the presentation of event warnings. In case there was an accident ahead, based on DENM messages, a cyclist could be advised to take a different, safer route to their destination. It would also be beneficial to implement a "heat-map" screen, based on a definition of what is a low, medium or high rate of message transmission, to show the user which areas are potentially more dangerous and should, consequently, be avoided.

Other aspects that were not specifically considered in this implementation are SM+ user's security and privacy. Especially considering that this application requires accurate location data to always be transmitted during its use, these aspects should be studied and mechanisms should be implemented to guarantee, not only SM+ user's physical safety, but their online safety as well.

References

[1] Google Google for Developers. “The activity lifecycle”. (Apr. 2023), [Online]. Available: https://developer.android.com/guide/components/activities/activity-lifecycle.

[2] Google Google for Developers. “Build a responsive UI with ConstraintLayout”. (Aug. 2023), [Online]. Available: https://developer.android.com/develop/ui/views/layout/constraint-layout.

[3] Google Google for Developers. “Android’s Kotlin-first approach”. (May 2023), [Online]. Available: https://developer.android.com/kotlin/first.

[4] Google Google for Developers. “Build location-aware apps”. (Jun. 2023), [Online]. Available: https://developer.android.com/training/location.

[5] Google Google for Developers. “Location”. (Aug. 2023), [Online]. Available: https://developer.android.com/reference/android/location/Location.

[6] Google Google for Developers. “LocationManager”. (Jun. 2023), [Online]. Available: https://developer.android.com/reference/android/location/LocationManager.

[7] Google Google for Developers. “LocationProvider”. (Feb. 2023), [Online]. Available: https://developer.android.com/reference/android/location/LocationProvider.

[8] Google Google for Developers. “Drawable resources”. (May 2023), [Online]. Available: https://developer.android.com/guide/topics/resources/drawable-resource.

[9] Google Google for Developers. “SensorEventListener”. (Feb. 2022), [Online]. Available: `https://developer.android.com/reference/android/hardware/SensorEventListener`.

[10] Google Google for Developers. “Meet Android Studio”. (Aug. 2023), [Online]. Available: `https://developer.android.com/studio/intro`.

[11] Google Google for Developers. “WindowManager”. (Sep. 2023), [Online]. Available: `https://developer.android.com/reference/android/view/WindowManager`.

[12] Autoridade Nacional de Segurança Rodoviária, “Relatório de sinistralidade de julho de 2021”, *www.ansr.pt*, page 32, Jul. 2021.

[13] Claudia Campolo, Antonella Molinaro, and Riccardo Scopigno, “Vehicular ad hoc networks”,

[14] Eleni Chalkia, Josep Maria Salanova Grau, Evangelos Bekiaris, Georgia Ayfandopoulou, Chiara Ferarini, and Evangelos Mitsakis, “Routing algorithms for the safe transportation of pupils to school using school buses”, in *Transport Research Arena (TRA) 5th Conference: Transport Solutions from Research to Deployment, Paris, France*, 2014, pages 1–10.

[15] Shailesh Chandra, “Safety-based path finding in urban areas for older drivers and bicyclists”, *Transportation research part C: emerging technologies*, vol. 48, pages 143–157, 2014.

[16] Xin Chang, Haijian Li, Jian Rong, Zhufei Huang, Xiaoxuan Chen, and Yunlong Zhang, “Effects of on-board unit on driving behavior in connected vehicle traffic flow”, *Journal of Advanced Transportation*, vol. 2019, 2019.

[17] Cohda Wireless, “Mk5 obu”, 2020. [Online]. Available: `https://www.cohdawireless.com/solutions/hardware/mk5-obu/`.

[18] Cohda Wireless, “Mk5 rsu”, 2020. [Online]. Available: `https://www.cohdawireless.com/solutions/hardware/mk5-rsu/`.

[19] C-Roads Platform. “C-Roads - The Platform of Harmonised C-ITS Deployment in Europe”. (2021), [Online]. Available: `https://www.c-roads.eu/`.

[20] C-Roads Platform. “C-Roads Cooperative Streets”. (2021).

[21] Nishanthi Dasanayaka, Khondokar Fida Hasan, Charles Wang, and Yanming Feng, “Enhancing vulnerable road user safety: A survey of existing practices and consideration for using mobile devices for v2x connections”, *arXiv preprint arXiv:2010.15502*, 2020.

[22] European Commission, "Rolling plan for ict standardisation", 2020. [Online]. Available: https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/intelligent-transport-systems-cooperative-connected-and-automated-mobility-its-ccam-and.

[23] European Telecommunications Standards Institute, "Etsi en 302 636-2 v1.2.1", Nov. 2013. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263602/01.02.01_60/en_30263602v010201p.pdf.

[24] European Telecommunications Standards Institute, "Etsi en 302 637-2 v1.4.1", Apr. 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf.

[25] European Telecommunications Standards Institute, "Etsi en 302 637-3 v1.3.1", Apr. 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.03.01_60/en_30263703v010301p.pdf.

[26] European Telecommunications Standards Institute, "Etsi tr 103 257-1 v1.1.1", May 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/103200_103299/10325701/01.01.01_60/tr_10325701v010101p.pdf.

[27] European Telecommunications Standards Institute, "Etsi tr 101 607 - v.1.2.1", Feb. 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/101600_101699/101607/01.02.01_60/tr_101607v010201p.pdf.

[28] European Commission, "Mobility and transport factsheet", *Sustainable and Smart Mobility Strategy*, 2020. DOI: 0.2775/82355NA-06-20-172-EN-C10.2775/932NA-06-20-172-EN-N. [Online]. Available: https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal/transport-and-green-deal_pt.

[29] Andreas Festag, "Cooperative intelligent transport systems standards in europe", *Communications Magazine, IEEE*, vol. 52, pages 166–172, Dec. 2014. DOI: 10.1109/MCOM.2014.6979970.

[30] A. Melnikov I. Fette, "Rfc 6455 - the websocket protocol", Dec. 2011.

[31] Zhaoxiang He and Xiao Qin, "Incorporating a safety index into pathfinding", *Transportation Research Record*, vol. 2659, no. 1, pages 63–70, 2017.

[32] Nima Hoseinzadeh, Ramin Arvin, Asad J Khattak, and Lee D Han, “Integrating safety and mobility for pathfinding using big data generated by connected vehicles”, *Journal of Intelligent Transportation Systems*, vol. 24, no. 4, pages 404–420, 2020.

[33] Akshay Upadhyay. “Haversine Formula – Calculate geographic distance on earth”. (Jun. 2018), [Online]. Available: https://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/.

[34] I.P. Instituto da Mobilidade e dos Transportes, “Relatório de tráfego na rede nacional de autoestradas 2º trimestre de 2021”, *www.imt-ip.pt*, page 98, 2021. [Online]. Available: https://www.imt-ip.pt/sites/IMTT/Portugues/InfraestruturasRodoviarias/RedeRodoviaria/Relatrios/RelatÃşriodeTrÃąfego-2ÂžTrimestrede2021.pdf.

[35] Google Google for Developers. “Convert Epoch/Unix Timestamp”. (Jun. 2023), [Online]. Available: https://developer.android.com/reference/java/time/Instant.html#now().

[36] Japan Society of Civil Engineers (JSCE), “Intelligent transport systems (its) introduction guide”, Aug. 2016. [Online]. Available: https://www.jsce-int.org/system/files/ITS_Introduction_Guide_2.pdf.

[37] H Kingsbury, D Harris, and P Durdin, “Journey optimisation by safest route”, in *Australasian Road Safety Conference, 1st, 2015, Gold Coast, Queensland, Australia*, 2015.

[38] JetBrains. “Kotlin Programming Language”. (2023), [Online]. Available: https://developer.android.com/kotlin/first.

[39] Anna Kramers, “Designing next generation multimodal traveler information systems to support sustainability-oriented decisions”, *Environmental Modelling & Software*, vol. 56, pages 83–93, 2014.

[40] Rosa Anna La Rocca, “Soft mobility and urban transformation”, *TeMA-Journal of Land Use, Mobility and Environment*, vol. 2, 2009.

[41] Zhaojian Li, Ilya Kolmanovsky, Ella Atkins, Jianbo Lu, Dimitar P Filev, and John Michelini, “Road risk modeling and cloud-aided safety-based route planning”, *IEEE transactions on cybernetics*, vol. 46, no. 11, pages 2473–2483, 2015.

[42] Françoise Lieberherr-Gardiol, “Urban sustainability and governance: Issues for the twenty-first century”, *International Social Science Journal*, vol. 59, no. 193-194, pages 331–342, 2008.

[43] National Oceanic and Atmospheric Administration (NOAA). "World Magnetic Model (WMM)". (Aug. 2023), [Online]. Available: https://www.ncei.noaa.gov/products/world-magnetic-model.

[44] Filip Machovec, "Its xfer gateway interface specification", *ESCoS Roadside Unit*, Jun. 2019.

[45] Mai Wahba Madkour and Alaa A ELMANZLAWI, "The role of soft mobility towards strengthening urban transformation and sustainable well-being: Case study: Mansoura city, egypt", *Journal of Engineering Research*, vol. 2, no. August, pages 145–159, 2018.

[46] Ioannis Mavromatis, Andrea Tassi, Robert J Piechocki, and Andrew Nix, "Efficient millimeter-wave infrastructure placement for city-scale its", in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, IEEE, 2019, pages 1–5.

[47] Sofia Cristino, "Morte de ciclista grávida gera onda de indignação", *Jornal de Notícias*, Jun. 2021. [Online]. Available: https://www.jn.pt/local/noticias/lisboa/lisboa/morte-de-ciclista-gravida-gera-onda-de-indignacao-13886682.html.

[48] National Oceanic and Atmospheric Administration (NOAA). "Magnetic Declination". (Aug. 2023), [Online]. Available: https://www.ncei.noaa.gov/products/geomagnetism-frequently-asked-questions.

[49] National Oceanic and Atmospheric Administration (NOAA). "Magnetic Declination". (Nov. 2014), [Online]. Available: https://www.ngdc.noaa.gov/geomag/declination.shtml.

[50] OASIS, "Mqtt version 5.0", *OASIS Standard*, 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf.

[51] Piotr Olszewski, Piotr Szagała, Daniel Rabczenko, and Anna Zielińska, "Investigating safety of vulnerable road users in selected eu countries", *Journal of safety research*, vol. 68, pages 49–57, 2019.

[52] Anil Kr. Mourya. "How to Integrate or Work with Open Street Map (OSM) in an Android App (Kotlin)". (Jun. 2023), [Online]. Available: https://medium.com/@mr.appbuilder/how-to-integrate-and-work-with-open-street-map-osm-in-an-android-app-kotlin-564b38590bfe.

[53] OSMDroid Contributors. "OSMDroid | MapSources". (Sep. 2019), [Online]. Available: https://osmdroid.github.io/osmdroid/Map-Sources.html.

[54] Redação do Jornal Idealista, "Baixa de lisboa: Carros anteriores a 2005 proibidos de circular a partir de 2021", *Idealista*, Feb. 2020. [Online]. Available: `https://www.idealista.pt/news/imobiliario/construcao/2020/02/06/42345-baixa-de-lisboa-carros-anteriores-a-2005-proibidos-de-circular-a-partir-de-2021`.

[55] Google LLC. "Overview | Protocol Buffers Documentation". (2023), [Online]. Available: `https://protobuf.dev/overview/`.

[56] Mariana Marques Tiago, "Ciclista grávida atropelada mortalmente na avenida da índia, em lisboa", *Público*, Jun. 2021. [Online]. Available: `https://www.publico.pt/2021/06/28/local/noticia/ciclista-atropelada-mortalmente-avenida-india-lisboa-1968221`.

[57] "True north and magnetic north: what's the difference?" (Sep. 2023), [Online]. Available: `https://www.rmg.co.uk/stories/topics/true-north-magnetic-north-whats-difference`.

[58] Felix Rosberg and Aidin Ghassemloi, *Machine learning-based path prediction for emergency vehicles*, 2018.

[59] José Santa, Fernando Pereñíguez, Antonio Moragón, and Antonio F Skarmeta, "Vehicle-to-infrastructure messaging proposal based on cam/denm specifications", in *2013 IFIP Wireless Days (WD)*, IEEE, 2013, pages 1–7.

[60] Johan Scholliers, Mirjami Jutila, Mikko Valta, Kimmo Kauvo, Ari Virtanen, and Pasi Pyykönen, "Co-operative Traffic Solutions for Hybrid Communication Environments", *Transportation Research Procedia*, vol. 14, pages 4542–4551, 2016.

[61] Google Google for Developers. "SensorManager | Android Developers". (Apr. 2023), [Online]. Available: `https://developer.android.com/reference/android/hardware/SensorManager`.

[62] David Shinar, "Crash causes, countermeasures, and safety policy implications", *Accident Analysis & Prevention*, vol. 125, pages 224–231, 2019.

[63] Katrin Sjoberg, Peter Andres, Teodor Buburuzan, and Achim Brakemeier, "Cooperative intelligent transport systems in europe: Current deployment status and outlook", *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pages 89–97, 2017.

[64] Mafalda Gonçalves, "A safety perspective for soft mobility in the its ecosystem", *INFORUM - Simpósio de Informática*, pages 1–12, 2022.

[65] Sabine Franke Vincent; Timpf, “Agent-based line-of-sight simulation for safer crossings”, *UC Santa BarbaraGIScience 2021 Short Paper Proceedings*, Sep. 2021. DOI: `10.25436/E2MW24`. [Online]. Available: `https://escholarship.org/content/qt0x82c21d/qt0x82c21d.pdf`.

[66] Kenneth R. Christiansen and Alexander Shalamov. “Motion Sensors Explainer”. (Aug. 2017), [Online]. Available: `https://www.w3.org/TR/motion-sensors/`.

[67] Bob Williams, *Intelligent transport systems standards*. Artech House, 2008.

[68] World Health Organization *et al.*, “Global status report on road safety 2018: Summary”, World Health Organization, Tech. Rep., 2018.

[69] World Health Organization *et al.*, “Road safety”, 2020.

[70] Michael Yuan, “Getting to know mqtt”, *IBM Developer Articles*, 2021. [Online]. Available: `https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/`.

[71] Câmara Municipal de Lisboa, “Zona de emissões reduzidas”, Jan. 2020. [Online]. Available: `https://zer.lisboa.pt/Relatorio_ZER.pdf`.