Universidade de Aveiro
2021

Pedro
Veloso Teixeira

Percepção de Situações Potencialmente Perigosas para VRUs numa Smart City

Awareness of Potentially Dangerous Situations for VRUs in a Smart City

Universidade de Aveiro
2021

Pedro
Veloso Teixeira

Percepção de Situações Potencialmente Perigosas para VRUs numa Smart City

Awareness of Potentially Dangerous Situations for VRUs in a Smart City

"*Success is not final*
Failure is not fatal
It is the courage to continue that counts"

— Winston Churchill

Universidade de Aveiro
2021

Pedro
Veloso Teixeira

Percepção de Situações Potencialmente Perigosas para VRUs numa Smart City

Awareness of Potentially Dangerous Situations for VRUs in a Smart City

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica da Doutora Susana Isabel Barreto de Miranda Sargento, Professora catedrática do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, do Doutor Pedro Filipe Vieira Rito, Investigador Auxiliar do Instituto de Telecomunicações de Aveiro, e colaboração do Engenheiro Francisco Manuel Malheiro de Castro, V2X Product Owner da Bosch Car Multimédia Portugal, S.A.

Dedico este trabalho ao meu irmão Miguel.

o júri / the jury

presidente / president	Prof. Doutor Paulo Jorge Salvador Serra Ferreira professor associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
vogais / examiners committee	Prof. Doutor Fernando Pedro Lopes Boavida Fernandes professor catedrático do Departamento de Engenharia Informática da Universidade de Coimbra
	Prof. Doutora Susana Isabel Barreto de Miranda Sargento professora catedrática do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos / acknowledgements

Não seria possível começar uma secção de agradecimentos sem agradecer aos meus pais, sem os quais não teria sido possível estar a terminar este capítulo da minha vida e ao meu irmão Miguel, o meu maior orgulho e a quem dedico este trabalho.

À Sara, a minha maior fã, a minha maior inspiração, o meu suporte. Obrigado por tudo e por todos os momentos, ajuda e carinho.

Quero agradecer aos meus colegas, ao Rodrigo (a quem desejo toda a sorte na dissertação que começa agora), à Andreia (a quem desejo muito boa sorte no doutoramento), ao Christian (que me ajudou desde do momento que entrei no NAP e a quem desejo as maiores felicidades), ao Rui (a quem desejo boa sorte no doutoramento e na vida como professor), à Ana (a quem também desejo boa sorte no doutoramento), à Tânia (a quem desejo descanso depois de uma dissertação tão completa), ao Alexandre (a quem desejo boa sorte numa futura carreira de comediante), e aos Pedros, todo o trabalho, dedicação, ajuda, mas também momentos de descontração, almoços e boleias. Sem eles, este trabalho não teria sido possível e este percurso não teria sido tão divertido, interessante e intelectualmente estimulante. Agradeço-lhes muito o quão espectaculares foram comigo e como me integraram na equipa.

Ao Paulo e ao Jorge um enorme agradecimento por tudo o que passamos juntos em tantos projectos. Sem eles este percurso não teria sido possível.

Um agradecimento especial à minha equipa de orientação do lado da Bosch e da Universidade de Aveiro e Instituto de Telecomunicações. Do lado da Bosch, obrigado ao Francisco pela oportunidade de trabalhar numa grande empresa e por confiar em mim e ajudar me em cada passo! Do lado do UA e IT, obrigado ao Pedro Rito pelas muitas conversas sobre como desenhar, implementar e avaliar as diferentes partes do meu trabalho, ao Miguel Luís pelas perguntas incisivas que muito melhoraram a qualidade desta dissertação e ao Duarte pela ajuda naqueles momentos mais críticos de decisões arquitecturais.

Por último, mas não com menos relevância, um muito obrigado à minha orientadora, a professora Susana Sargento, que me acompanhou desde do terceiro ano. Com compreensão mas rigor e exigência, fez de mim um melhor profissional mas também uma melhor pessoa. Muito obrigado pela confiança, pelas oportunidades e por tudo o que me ensinou, sem as quais nada teria sido possível.

Agradecimento especial à Altice Labs e Meo pelo desenvolvimento e disponibilização de uma rede 5G pré-comercial para o desenvolvimento deste trabalho e pela disponibilização do cartão 5G para teste na sua infra-estrutura. Este trabalho contribuiu para a plataforma Aveiro Tech City Living Lab, desenvolvida através da European Urban Innovative Actions programme Aveiro STEAM City (UIA03-084), e é uma base para o caso de uso de Vulnerable Road Users desenvolvido no âmbito do Projeto Mobilizador do Cluster Automóvel de Portugal, Mobinov, POCI-01-0247-FEDER-046095.

Palavras Chave

5G, Arquitecturas Orientadas a Eventos, Cidades Inteligentes, Computação Edge, Prevenção de Colisões, Redes Veiculares, Segurança de Sistemas de Transporte Inteligentes, Sistemas de Transporte Inteligentes, Utilizadores Vulneráveis de Estrada

Resumo

Segundo a Eurostat, em 2019, a maioria - 70 % - dos acidentes mortais em estradas envolveu utentes vulneráveis (VRU), tais como crianças, pessoas com deficiência ou ciclistas. Esta percentagem aumentou em 2020 para pelo menos 77% dos acidentes fatais. Com o aparecimento de Sistemas de Transporte Inteligentes (ITS), e com os veículos a fazer parte de Redes Ad-hoc Veiculares (VANET) e cidades inteligentes a conectar todas as partes de um ambiente urbano, novas soluções podem ser consideradas para combater esses acidentes. Para além disso, com a introdução de soluções de condução autónoma, evitar situações perigosas em estradas e proteger os VRUs é cada vez mais importante.

Esta dissertação propõe uma solução com múltiplos sensores para prevenir potenciais acidentes entre veículos e VRUs. Uma cidade inteligente possui diversos sensores, dispersos nos veículos (e agregados por On-Board Units, OBUs), nos VRUs (por exemplo, smartphones e smartwatches) e na própria estrada (por exemplo, câmaras de vídeo e radares). Todos estes nós e sensores podem comunicar e notificar eventos através de um conjunto de diferentes tecnologias de acesso sem fios, como por exemplo ITS-G5 (com o IEEE 802.11p/WAVE), C-V2X e tecnologias celulares, como LTE, 5G e, no futuro, 6G. Após agregar e processar este conjunto de informações, foi implementado um sistema para prever e notificar potenciais situações de risco que envolvam VRUs e veículos. Este sistema é baseado no cálculo de zonas de risco e pontos de colisão entre VRUs e veículo.

A Infraestrutura existente da Aveiro Smart City (graças ao projeto Aveiro Tech City Living Lab) foi utilizada para desenvolver e avaliar o sistema. Informação dos sensores da cidade e veículos é obtida e agregada à informação dos dispositivos associados aos VRUs numa arquitectura híbrida, com uma implementação edge e cloud.

Os resultados mostram que o sistema é promissor na previsão de potenciais colisões, enquanto apontam para algumas decisões de implementação importantes que devem ser feitas para garantir tempos de notificação adequados, tais como o uso de multi-homing, 5G e 6G, e a exploração do conceito de edge computing. O sistema desenvolvido pode ser considerado como um passo em direção ao nível 5 de automação de veículos autónomos, com a deteção de potenciais colisões a poder fazer parte de um processo de decisão da próxima acção.

Keywords

5G, Collision Avoidance, Edge Computing, Event-Driven Architecture, ITS, ITS Safety, Smart City, Vehicular Ad-hoc Networks, Vulnerable Road Users

Abstract

According to Eurostat, in 2019, most – 70% - fatal road accidents in urban areas involved vulnerable road users (VRUs), such as children, impaired people, cyclists, and animals, with this percentage increasing in 2020 to at least 77% of fatal accidents. With the advent of Intelligent Transportation Systems (ITS), with vehicles being part of a Vehicular Ad-hoc Networks (VANET) and smart cities connecting all parts of an urban environment, new solutions for fighting against these accidents can be considered. Moreover, with the introduction of autonomous driving solutions, avoiding dangerous road situations and protecting the VRUs is more critical than ever.

This dissertation proposes a multi-sensor solution for preventing potential accidents between vehicles and vulnerable road users. A smart city has several sensors, dispersed in the vehicles (and aggregated by the On-Board Units, OBUs), in the VRUs (*e.g.*, smartphones and smartwatches), and in the road itself (*e.g.* cameras, radars). These different nodes communicate with each other through several wireless access technologies, most notable short range standards such as ITS-G5, C-V2X or long range technologies such as LTE, 5G and, in the future, 6G. By aggregating and processing such information, a system was implemented to predict and notify potential hazardous situations involving VRUs and vehicles. This system was based on the computation of risk zones and prediction of collision points between VRUs and vehicles.

The current infrastructure from the Aveiro Smart City (thanks to the Aveiro Tech City Living Lab project) is leveraged to deploy and evaluate the system. Information from the city's sensors and vehicles is gathered and joined to the information from the VRUs own devices in an hybrid architecture, with an edge and cloud deployment.

The obtained results show that the system is promising at predicting potential collisions while pointing at some important deployment decisions that must be made to ensure proper notification timings, such as the usage of multi-homing, 5G, and 6G, and following the concept of edge computing. The developed system can be considered a step towards a level 5 of full automation of autonomous vehicles. The potential collision can be part of the pipeline for the decision of the following action.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

ABS	Anti-lock Braking System
ANACOM	Autoridade Nacional de Comunicações
ASIC	Application-Specific Integrated Circuit
ASN.1	Abstract Syntax Notation One interface description language
ATCLL	Aveiro Tech City Living Lab
Bosch	Bosch Car Multimédia Portugal
BSM	Basic Safety Message
CAM	Cooperative Awareness Message
CAN	Controller Area Network bus
CCU	Central Control Unit
CPM	Collective Perception Message
C-V2X	Cellular Vehicle-to-Everything
DCU	Data Collection Unit
DENM	Decentralized Environmental Notification Message
ELK	Elasticsearch-Logstash-Kibana
ETSI	European Telecommunications Standards Institute
EVA	Emergency Vehicle Alert
GDA	Geographical Destination Area
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ICA	Intersection Collision Announcement
IMU	Inertial Measurement Unit
IoU	Intersection over Union
IT-Aveiro	Institute of Telecommunications of Aveiro
ITS	Intelligent Transport System
ITS-S	Intelligent Transport System Station
JDL	U.S. Joint Directors of Laboratories
Lidar	LIght Detection And Ranging
LoRaWAN	Long Range Wide-Area Network
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MANET	Mobile Ad-hoc Network
MEC	Multi-access Edge Computing
MQTT	Message Queuing Telemetry Transport
MTBF	Mean Time Between Failures
NAP	Network Architectures and Protocols
NFC	Near Field Communication
NGSIv2	Next Generation Service Interfaces version 2
NTP	Network Time Protocol
OBU	On-Board Unit
P2I	Pedestrian to Infrastructure
P2P	Pedestrian to Pedestrian
P2V	Pedestrian to Vehicle
P2X	Pedestrian to Everything
PSM	Personal Safety Message
QoS	Quality of Service
Radar	RAdio Detection And Ranging
ROI	Region of Interest
RSU	Road-Side Unit
SAE	Society of Automotive Engineers
TIM	Traveler Information Message
TTC	Time To Collision
TTL	Time To Live
DETI/UA	Department of Electronics, Telecommunications and Informatics, University of Aveiro
UC	Use Case
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VAM	Vulnerable Road User Awareness Message
VANET	Vehicular Ad-hoc Network
VRU	Vulnerable Road User
WAVE	IEEE 802.11p WAVE
WGS84	EPSG:4326/World Geodetic System 1984
WSGI	Web Server Gateway Interface
YOLO	You Only Look Once real-time object detection system

CHAPTER 1

Introduction

The journey of a thousand miles must begin with a single step.
– Lao Tzu

The safety on roads for vehicles and VRUs is a critical problem. According to the World Health Organization [1], in 2016, 1.35 million people died within road accidents. In cities, at least 70% of fatal accidents involved VRUs, such as children, impaired people, cyclists and animals [2].

With the advent of smart cities, more and more data can be obtained from road users, namely vehicles and VRUs. These data can be used to support the optimization and improvement of urban environments in areas like entertainment, traffic management, urban planning, health, and, more crucially, road safety.

At the core of smart cities are road elements connected between one another and with a common infrastructure. A smart city can contain several Intelligent Transport Systems (ITSs), - such as vehicles with capabilities for sensing the surrounding environment - and VRUs with smartphones and other connected wearable devices, but also other road infrastructure sensors - such as video cameras, RAdio Detection And Ranging (Radar), LIght Detection And Ranging (Lidar) and other sensors - to create a full picture of the current status of the smart city [3] [4] [5].

The amount of information that a smart city can send to vehicles, in addition to the vehicle own sensing capabilities, is now sufficient to support semi-autonomous vehicles, with the capability of helping and substituting the driver in several tasks such as park assist, lane merging assist or keeping stable distance with another car (adaptive cruise control). These capabilities support the end goal of manufacturers to achieve full autonomy within their vehicles [6] [7].

All these tasks must be done while ensuring the safety of passengers and other VRUs outside the vehicle. Current solutions for VRUs safety focus on the vehicle being able to act and react to their surroundings - and for example, break if a VRU is close - based on the information from the vehicle onboard sensors [8]. However, these solutions require the vehicle to be fully equipped with equipment, not supported in older or cheaper vehicles. Moreover, they do not fully explore the full potential of smart city infrastructures and available data.

The *Awareness of Potentially Dangerous Situations for VRUs in a Smart City* is a dissertation project within the area of vehicular networks that focus on addressing these issues by improving the safety of vulnerable road users within a smart city environment by using the vehicles and VRUs communications, and a multi-sensor system to predict potential collisions.

1.1 Goals

The goal of this dissertation is to develop the necessary platforms to increase safety on the city roads, by detecting potentially dangerous situations involving drivers and VRUs, and warning either the driver, the VRU or both. It is also a goal the deployment of these platforms into a real smart city infrastructure, to gather in real time data from vehicles and VRUs through different sources, and notify both vehicles and VRUs on these dangerous situations.

The development is split into two phases. In the first phase, the objective is to develop the parts of the system responsible for gathering and using the information from vehicles, VRUs and infrastructure (*e.g.*, video cameras, Radars, Lidars), and store and provide them to other ITS applications. In the second phase, the objective is to, using the aggregated information resulting from the first phase, create an ITS safety application that can detect and notify vehicles and VRUs about imminent accidents.

This project is not independent of current and past smart city projects; in fact, it is done in cooperation with the Aveiro Tech City Living Lab (ATCLL), the Network Architectures and Protocols (NAP) group from IT-Aveiro, Department of Electronics, Telecommunications and Informatics, University of Aveiro (DETI/UA) and Bosch Car Multimédia Portugal (Bosch).

Figure 1.1 contextualizes the system to be developed with already existing infrastructure both in IT-Aveiro side and Bosch side.

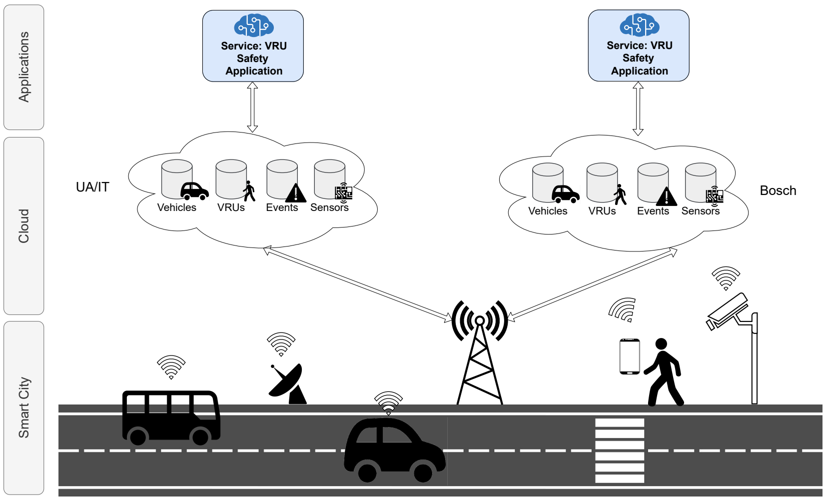


Figure 1.1: Context of the project

Using the hardware and infrastructure from the ATCLL, information from the vehicles On-Board Units (OBUs) and infrastructure in the city of Aveiro (such as video cameras and Radars) can be obtained.

Since this dissertation is being done in cooperation with multiple entities, the developed system must follow a few basic guidelines:

- Be replicable, extensible, and adaptable to make migration to new hardware smooth.
- Respect and use standards defined by entities such as European Telecommunications Standards Institute (ETSI) as much as possible.
- Use existing infrastructure as much as possible.

Technologically speaking, this project is interesting both as a proof-of-concept of the potential of the Vehicle to Everything (V2X) communication paradigm, and a way to showcase the potential of already existing hardware, both sensing and communication equipments.

1.2 CONTRIBUTIONS

The work developed in this dissertation can be summarized as follows:

- Creation of a solution to introduce VRUs in the context of a vehicular network.
- Creation of a solution to prevent potentially dangerous situations on the road.
- Analysis of the edge vs. cloud computing in a real vehicular network scenario.
- Usage of data from a Smart City in a real context.

This work resulted in a set of contributions, that can be summarized as follows:

- Creation and successful submission of an invention report with Bosch, with the end goal of creating a patent.
- Creation of a scientific paper: P. Teixeira, S. Sargento, F. Castro, P. Rito, and M. Luís, *"Awareness of Potentially Dangerous Situations forVRUs in a Smart City."*.
- Presentation and demonstration of the system during the Aveiro Tech Week, from October 11th to 17th, 2021[1].

1.3 DOCUMENT STRUCTURE

The remaining of this dissertation is organized into seven chapters.

Chapter 2 - Safety of Vulnerable Road Users - provides an overview of the problem to be solved, including the definition of potential dangerous situations and VRU characteristics. Use cases to be considered and requirements to be met by the designed solution are also addressed.

Chapter 3 - Background concepts & state-of-the-art - introduces the background of this Dissertation. Previous solutions for guaranteeing the safety of VRUs are presented and analyzed. Some aspects lacking in previous works are listed and proposed as improvements for the solution to be developed. Technological aspects required for the implementation of the solution, such as the relevance of Vehicular Ad-hoc Network (VANET), standard messages, and sensor fusion, are also presented.

---

[1] Available at `https://youtu.be/crWqUD5dClA?t=4269`.

Next, Chapter 4 - VRU Safety Architecture - follows with the architecture of the developed solution. On the VRU side, it presents details of the exchanged messages from the sensors and communication technologies for interaction with the remaining infrastructure. On the vehicle and infrastructure side, it also presents details on their integration in the approach and on the existing infrastructure. Finally, details on the safety application algorithms are presented.

After presenting the architecture, Chapter 5 - Implementation - addresses implementation details, decisions and their rationale.

In Chapter 6 - Evaluation - the evaluation process to validate the work is presented, and the obtained results are discussed. Part of the evaluation was based on simulation, while another part was based on real-life scenarios with vehicles and VRUs.

Finally Chapter 7 - Conclusion - presents a summary of the system, a discussion of its functionalities and results, and discusses ideas for future work.

CHAPTER 2

Safety of Vulnerable Road Users

It is common sense to take a method and try it.
If it fails, admit it frankly and try another.
But above all, try something.
— *Franklin D. Roosevelt*

After presenting the primary motivations in Chapter 1, this chapter presents the problems that VRUs encounter in road scenarios, and that shall be addressed by the proposed system. This chapter starts in Section 2.1, by analyzing potentially dangerous situations on roads between VRUs and vehicles, and the VRUs characteristics. This is the basis for the definition in Section 2.3 of the main use cases the system shall take into consideration. After the definition of the use cases, Section 2.4 presents the main system requirements. Finally, Section 2.5 provides a summary of the chapter.

2.1 Potentially dangerous situations

European Telecommunications Standards Institute (ETSI) defines many traffic situations that possess an imminent safety risk [9]. A Vulnerable Road User is particularly vulnerable in traffic situations where there is a potential conflict with another road user. The traffic conflict point is the intersection of the trajectories of the VRU and the other road user. A conflict, or collision, occurs if both the VRU and the other road user reach the conflict point at about the same time.

When considering a city, and considering pedestrian VRUs, an intersection of trajectories usually happens when a pedestrian is crossing a road:

- VRU crossing the road outside a zebra crossing: always dangerous, by definition.
- VRU crossing the road inside a zebra crossing: can be dangerous in the following situations:
  - Situations with low visibility for a vehicle's driver:
    - Vehicle driving on a curve or a road topology that impairs visibility.

        - Adverse weather conditions (snow, rain) or light conditions (night, fog).
        - VRU height is not enough to be seen by the driver.
        - During overtaking maneuvers, where another vehicle might impair visibility.
        - Areas with a high density of VRUs, where understanding what each one is doing is not doable by the driver.
    - Situations with low visibility for the VRU:
        - Obstacles blocking line of sight of the VRU.
        - Disabled people, including blind people.
        - Children that have a reduced height might not be able to see a vehicle.
    - When either the vehicle's driver and/or the VRU ignores the vertical road markings, priorities, semaphores and/or other road code regulations.
    - When either the vehicle's driver and/or the VRU are distracted.
    - When the vehicle's driver does not follow the speed limits.

Situations involving non pedestrian VRUs could also be considered, but will not be focus of this work:

- Cyclist VRU overtaking a vehicle.
- Cyclist VRU overtaking another VRU (*e.g.* a pedestrian in a cyclist lane).
- VRU animals on the road without human supervision.

Collision avoidance between a vehicle and a VRU can be achieved actively at the level of the vehicle or VRU device, which may have several strategies: slowing down, braking, and others, as described in Section 3.8.

2.2 Users characteristics

Road users will use the system to be developed. There are two types of road users, with different characteristics, that the system will consider:

1. **Drivers**, *i.e.*, road users within a vehicle (*e.g.* personal car or city bus). Drivers, even when practicing safe driving, can get distracted and would benefit from being informed about potentially dangerous situations involving VRUs or obstacles in the road.
2. **Vulnerable Road Users** that are particularly vulnerable to dangerous situations involving vehicles, where they are not as protected as people inside vehicles.

VRUs, in particular, can have different characteristics, with three different profiles of VRUs that can be defined [10]: pedestrians, light vehicles that might have an engine, and motorbikes, all with different characteristics in terms of speed, how predictable is their trajectory, and more. Table 2.1 summarizes the different types of VRUs and their main characteristics.

The system's use cases will focus on adult pedestrians and disabled people. The first provides a good way of validating the entire system, while the latter is a good example of how critical this system is: since these VRUs are more in danger by having hearing and/or visual impairments, a system that can detect and warn them, either visually or through audio, will greatly benefit their overall safety within roads.

Table 2.1: VRU profiles and their characteristics

VRU Profile		**Behavior**	**Speed**	**Reaction capabilities**	**Visibility to drivers**
Pedestrian	*Adult*	Unpredictable	$0-4m/s$	Normal	Normal
	Disabled people		$0-1m/s$	Can be impaired, decreased might not be able to react to visual or audible warnings	Might be poor
	Children		$0-1m/s$	Decreased	Might be poor
	Animals		N/A	Unpredictable reaction	Usually poor
Light vehicles	e.g. *Bicycles*	Somewhat predictable	$0-10m/s$	Normal	Might be poor
Motorbikes		Somewhat predictable	Same as vehicle	Same as a driver	Might be poor

2.3 Use Cases

Keeping in mind all the potentially dangerous situations around VRUs and their characteristics, a few use cases were selected by having as criterion the impact on the safety of all road users; more general use cases, and therefore applicable to more situations, were preferred. Table 2.2 summarizes the use cases and their description and priority in this dissertation. The use cases were prioritized using the following scale:

- Priority 1: The use case is a "must-have" that must be considered and implemented, and is needed for the other use cases.
- Priority 2: The use case is needed to improve the overall quality of the system.
- Priority 3: The use case is a "nice to have" which may include new functionalities, and might or might not be implemented.

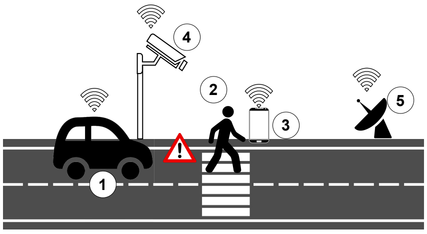
Table 2.2: Summary of use cases to be considered

Use Case		**Description**	**Priority**	**Actors**
UC1: Collision detection with a pedestrian crossing a road	*UC1.1: Collision detection with a pedestrian with ITS-S crossing a road without infrastructure (just pedestrian ITS-S)*	Prevent any accident between a pedestrian and a vehicle	1.A First to be considered	VRU Vehicle
	UC1.2: Collision detection with a pedestrian without ITS-S crossing a road (just infrastructure)		1.B Second to be considered	Vehicle Infrastructure
	UC1.3: Collision detection with a pedestrian with ITS-S crossing a road (infrastructure + pedestrian ITS-S)		1.C Third to be considered	VRU Vehicle Infrastructure
UC2: Collision detection with an animal or obstacle		Prevent any accident between other VRU without smartphone.	3	Vehicle Infrastructure
UC3: Signaling pedestrian hidden by obstacle		Inform other vehicles behind about a pedestrian that is hidden by an obstacle	3	Vehicles
UC4: Present statistics of potential accidents		Show statistics about the potential accidents and the conditions in which would or did occur	2	City Manager

The first Use Case (UC) 1, focuses on the system detecting potential hazardous situations - collisions - using as much information as available in a seamless way. Such use case can be classified by the VRU having or not an Intelligent Transport System Station (ITS-S) and the existence or not of additional smart city infrastructures, such as Radars and video cameras. If one of these is not available - as in UC1.1 and UC1.2 - the system should still work; if both are present - as in UC1.3 - the system should fuse the information provided by both nodes as much as possible to improve the system's overall quality.

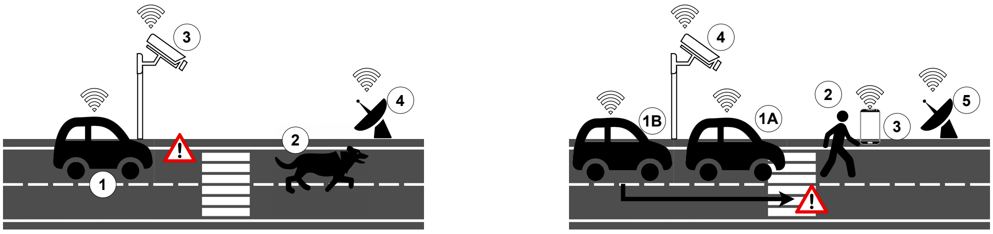
(a) UC1.1: A vehicle (1) and a pedestrian (2) are warned about a potential collision. The pedestrian has an Intelligent Transport System Station (ITS-S), *e.g.* a smartphone (3). Roads have no infrastructure associated.

(b) UC1.2: A vehicle (1) and a pedestrian (2) are warned about a potential collision with a pedestrian **not having** an Intelligent Transport System Station (ITS-S). Roads infrastructure, such as video cameras (3) and Radars (4) at strategic locations, are used to detect the obstacle.



(c) UC1.3: A vehicle (1) and a pedestrian (2) are warned about a potential collision. The pedestrian has an Intelligent Transport System Station (ITS-S), *e.g.* a smartphone (3). Roads are now equipped with infrastructure such as video cameras (4) and Radars (5) at strategic locations, that help the detection of a pedestrian, an animal or other obstacles.

Figure 2.1: Use cases involving collision detection with a pedestrian VRU crossing a road



(a) UC2: A vehicle (1) is warned about an animal/or obstacle (*i.e.* non-human VRU, 2) in its direct trajectory. Infrastructure such as video cameras (3) and Radars (4) are used to detect the obstacle.

(b) UC3: A vehicle (1A) receiving information about a potential collision with a VRU (3) warns vehicles (1B) behind/around. Infrastructure such as video cameras (3) and Radars (4) can also be used to detect the obstacle.

Figure 2.2: Other use cases

The second use case, UC2, extends the system to detect animals or other obstacles on the road. Here, the focus is on improving the safety of the driver and passengers of vehicles by avoiding crashes, either by colliding with the obstacle or by consequence of maneuvers to avoid it.

In addition, the extension provided by UC3 is convenient when the vehicle behind does not receive the information about the potential collision for some reason, when the vehicle in front might brakes suddenly - with the system notifying to avoid a collision and to signal that overtaking the vehicle in front is dangerous.

The last use case, UC4, focuses on a different actor. Other use cases focus on the VRU and vehicle driver as actors. In UC4, the existence of a dashboard for monitoring the city status in terms of potential collisions can support decisions of a city manager actor on optimizing and changing traffic of the city or other rules.

2.3.1 Usage scenarios

It is possible to consider that a VRU, such as an adult pedestrian, might not benefit much from such a system, since they can easily see if an approaching vehicle is going too fast and might collide with them, and thus act to avoid such collision. However, some scenarios, provided here as examples, show the importance of the system to be developed:

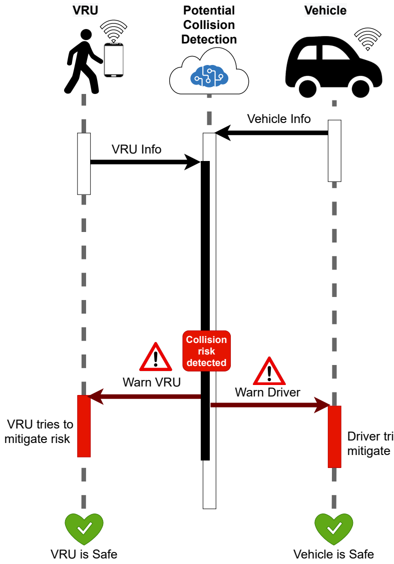
- Crossing a poorly lighted road at night.
- Crossing a road with heavy traffic.
- Crossing a road when the vehicle's visibility is impaired by weather, obstacles, curves, altitude, or other characteristics of the topology of the road.
- Crossing a road with road works poorly signaled.
- Crossing a road for shorter people - *e.g.* children - that have more difficulty in seeing the road.
- Crossing a road for disabled people, including blind people.

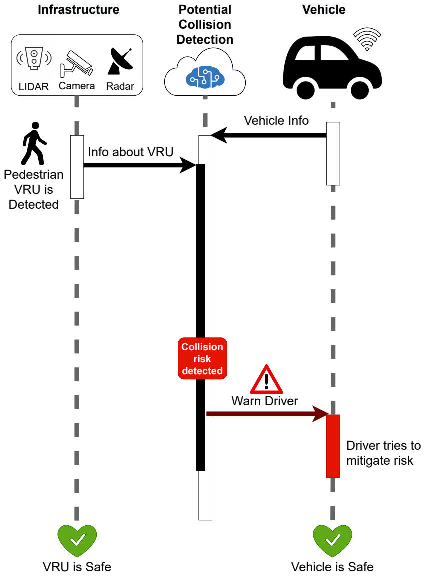
In a different point of view, city managers can also benefit from using such a system to understand trends related to potential accidents, such as times of the day, most common collision types, or, in case of effective collision, trigger communication with the emergency services.

2.3.2 Flow of the use cases

With the use cases defined, it is possible to define how the use cases would be implemented. Figure 2.3, Figure 2.4a and Figure 2.4b show how a potential architecture can comply with the use cases defined in Section 2.3.

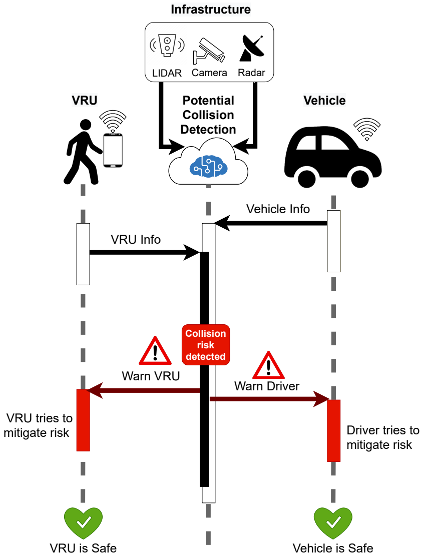
The sequence diagrams show how the system heavily relies on exchanging and processing messages containing information about each node (VRU, vehicle or infrastructure).





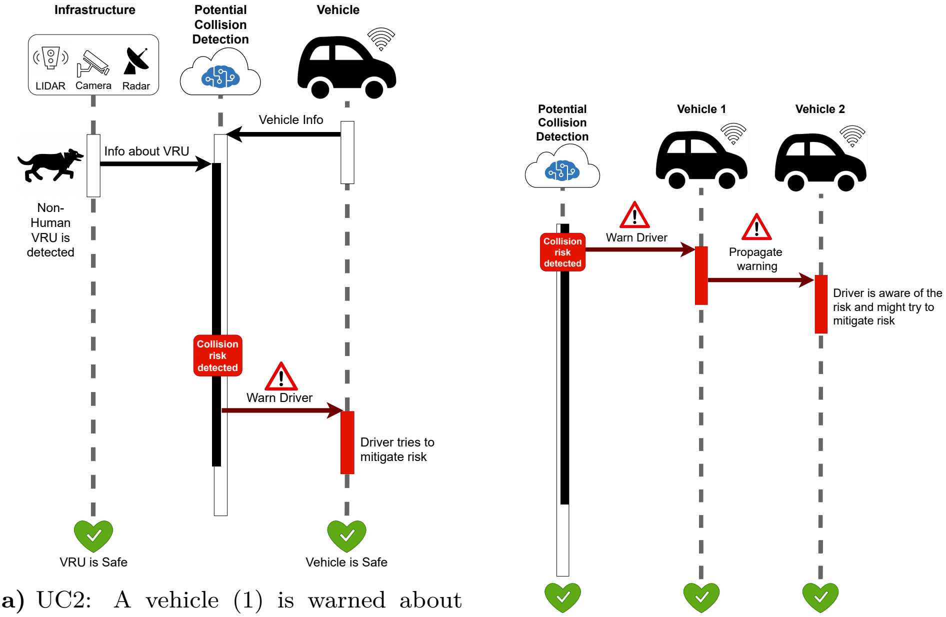
(a) UC1.1: A vehicle and a VRU are warned about a potential collision. The VRU has a ITS-S, *e.g.* a smartphone. Roads have no infrastructure associated.

(b) UC1.2: A vehicle and a VRU are warned about a potential collision with a VRU **not having** an ITS-S. Roads infrastructure, such as video cameras at strategic locations, are used to detect the obstacle.



(c) UC1.3: A vehicle and a VRU are warned about a potential collision. The VRU has a ITS-S, *e.g.* a smartphone. Roads are now equipped with video cameras at strategic locations, that help the detection of a VRU, an animal or other obstacles.

Figure 2.3: Sequence diagrams for UC1 - collision detection with a VRU crossing a road



(a) UC2: A vehicle (1) is warned about an animal/or obstacle (*i.e.* non-human VRU, 2) in its direct trajectory. Infrastructure such as video cameras (3) are used to detect the obstacle.

(b) UC3: A vehicle (1) receiving information about a potential collision with a VRU (3) warns vehicles (2) behind/around.

Figure 2.4: Sequence diagrams for other use cases

2.4 REQUIREMENTS

From the analysis of the use cases, it is possible to define a set of essential requirements. For now, only functional and user requirements will be defined. The requirements are prioritized using the following scale:

- Priority 1: The requirement is a "must-have" that must be considered and implemented.
- Priority 2: The requirement is needed to improve the system's overall quality but is not a "must-have".
- Priority 3: The requirement is a "nice to have" which may include new functionality and might or might not be implemented.

2.4.1 Functional requirements

The analysis of potentially hazardous situations requires a set of goals to be achieved, which are further detailed in Table 2.3:

- Precise current location of both of vehicle and VRU, with adequate repetition/frequency of the location measurements.
- Being able to predict the trajectory of the vehicle or VRU.
- Provide information about the vertical position (altitude) of both vehicle and VRU. Otherwise, the situation in Figure 2.5 would be considered a potentially dangerous situation.
- Identify that both road users are on a collision course and warn the ITS-S users.

Table 2.3: Functional requirements

Req #	Requirement	Priority
RF_1	The system must be able to obtain the precise location (latitude, longitude and altitude), speed and orientation from each vehicle periodically.	1
RF_2	The system must be able to obtain the precise location (latitude, longitude and altitude), speed and orientation from each VRU periodically.	1
RF_3	The system must be able to update the registered information of a determined vehicle/VRU periodically.	1
RF_4	The system must retrieve all the information for a certain vehicle/VRU periodically.	1
RF_5	The system must be able to predict a potential accident with a probability of at least 51% (*i.e.* more true positives and negatives than false positives and negatives).	1
RF_6	The system must be able to notify both entities of a vehicle-VRU pair of a potential accident periodically.	1

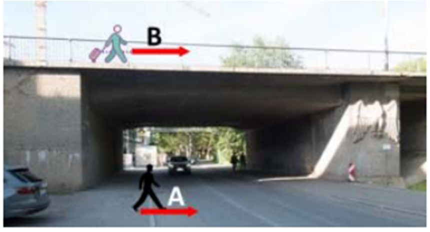


Figure 2.5: Situation where the altitude data is crucial [10]

2.4.2 User interface requirements

This system will interact with road users to warn them about potentially dangerous road situations, with Table 2.4 defining the requirements the user interface must comply with.

Table 2.4: User interface requirements

Req #	Requirement	Priority
RUI_1	The notification to the VRU should say "WARNING, ACCIDENT IMMINENT! Please be careful to your surroundings" or similar, either using a visual or an audio notification.	1
RUI_2	The notification to the vehicle should say "WARNING, PEDESTRIAN AHEAD! Please be careful to your surroundings" or similar, either using a visual or an audio notification.	1
RUI_3	The VRU app should have a Settings page to turn on/off audio notifications.	3

2.4.3 Other non functional requirements

Any requirement and future solution design should have in mind several main constraints, described in Table 2.5.

Table 2.5: Other non functional requirements

Req #	**Requirement**	**Priority**
RP_1	Portability: The system must work with the current Aveiro Tech City Living Lab infrastructure and be adaptable to Bosch Central Control Units (CCUs). The system should, as much as possible, keep the format and information in standard messages.	1
RP_2	Latency: The system should be able to detect a potential accident and warn the vehicle and/or VRU within an appropriated amount of time; otherwise, notifications might be too late to be effective.	1
RP_3	Capacity: The system should be able to handle a peak of vehicles and VRUs.	2
RP_4	Monitoring: All potential accidents should be logged.	3
RP_5	Authentication: The sensor fusion data platform should be open and usable by other applications other than VRU safety applications. The sensor fusion platform could be considered as a potential data marketplace, a trending and growing concept.	3
RP_6	Privacy: Identifying the system's users must not be possible even using all gathered data.	1
RP_6	Resilience: The system's main function is to warn road users on potentially dangerous situations, and it is therefore a critical system, which means that it is time critical and the Mean Time Between Failures (MTBF) should be as small as possible.	1

2.4.4 Assumptions

On the driver's side it is assumed that the vehicle is an ITS-S, *i.e.* it has an On-Board Unit (OBU)/Bosch CCU that can communicate with the rest of the system. In addition, it is assumed that the vehicle has enough sensors to determine its class (*e.g.* light vehicle, bus, heavy truck), speed, position, driving direction and that is capable of presenting information to the driver about the warnings (*e.g.* with a screen or by using the instruments cluster).

On the VRU side it is assumed that either the VRU has a smartphone (or equivalent device) with access to the Internet and sensors such as Global Positioning System (GPS), gyroscope, and accelerometer, or that the VRU can be detected using an object detection technique applied on a camera stream or equivalent.

2.5 SUMMARY

This chapter introduced the main issues around the safety of VRUs on roads, including the requirements of the system to be developed.

In the context of an urban environment, roads can be dangerous, in many different situations, for VRUs, such as children, disabled or impaired people, older people, and other types of pedestrians. In addition, non-human VRUs, such as dogs or wild animals, can pose a danger to vehicles and their occupants. The different profiles of VRUs possess distinct dynamics and ways of reacting and behaving in a road environment. Any system aiming at improving road safety around VRUs shall have in mind the characteristics of each type of VRUs. From these different situations and VRU characteristics, a set of use cases to be considered were listed, and from them, the central system requirements were defined.

The next chapter will present the necessary background concepts and the state-of-the-art in terms of VRUs safety solutions.

CHAPTER 3

Background concepts & state-of-the-art

The greatest enemy of knowledge is not ignorance.
It is the illusion of knowledge.
— *Stephen Hawking*

The last chapter presented the main issues around the safety of VRUs on roads, including the requirements of the system to be developed. This chapter presents the background concepts necessary for the system development, and the state-of-the-art in terms of VRU safety solutions.

This chapter starts by addressing the essential concepts that support the system to be developed. Section 3.2 presents the smart city of Aveiro, powered by Aveiro Tech City Living Lab, a large-scale infrastructure that includes VANETs, sensors and information aggregators. Section 3.3 describes the concept of VANETs, the network access technologies that support them and their challenges. Section 3.4 enumerates the European and United States standards for VANET messages.

Related to the possible sensors and information aggregators systems can use, Section 3.5 provides an overview of the different types of sensors, while Section 3.6 provides the main concepts and advantages on sensor fusion, and Section 3.7 discusses information aggregators.

Afterwards, Section 3.8 discusses the state-of-the-art on the safety of VRUs and of algorithms for detecting potentially dangerous road situations. Finally, Section 3.9 provides a summary of the discussed topics.

3.1 Methodology

To obtain a comprehensive literature review, the following steps were followed:

1. Search about the problem and historical perspective (older papers).

2. Search about the existing solutions and architectures for VRU safety.
    - Inclusion criteria:
        - Papers that are within the vehicular area.
        - Papers published by IEEE, ACM, arXiv, MSc dissertations and PhD theses.
    - Exclusion criteria:
        - Papers not published in English or Portuguese were excluded.
        - Papers lacking full text or that required the purchase of text while searching in the database are excluded.
3. Search about specific points of the proposed solution.
    - Characteristics of VANETs and used network access technologies. Keywords: *vanet, its, 5g, wave, c-v2x.*
    - Standards for messages formats. Keywords: *etsi, sae, cam, vam, denm, cpm, psm, bsm.*
    - Sensors characteristics and sensor fusion concepts. Keywords: *sensor fusion, Radar, lidar, camera, imu.*
    - Existing information aggregators and persistence solutions. Keywords: *broker, event-driven, databases, NoSQL.*
    - Algorithms for detection of hazardous situations involving VRU. Keywords: *vru, road safety, pedestrian, pedestrian safety, collision avoidance, collision prediction.*

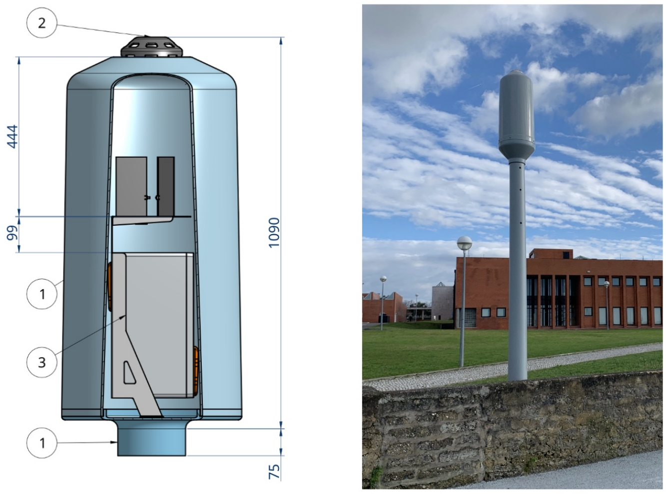
3.2 Aveiro Tech City Living Lab

The Aveiro Tech City Living Lab is an advanced large-scale smart city infrastructure spanned all over the city of Aveiro, Portugal. It can be used by research institutions and industry to develop, test or demonstrate concepts, products, or services within the vehicular and ITS domains. This infrastructure integrates people[1], sensors and vehicles, such as automobiles, bicycles in the city, aerial and aquatic drones and *"moliceiros"* (boats) in the Aveiro Lagoon [11].

The access infrastructure is supported by fiber technology connecting smart lampposts and building facade boxes, shown in Figure 3.1, and spanned throughout the city of Aveiro. These edge points connect to a central cloud where further information gathering and processing can occur, and new services and applications can be developed, tested, and deployed.

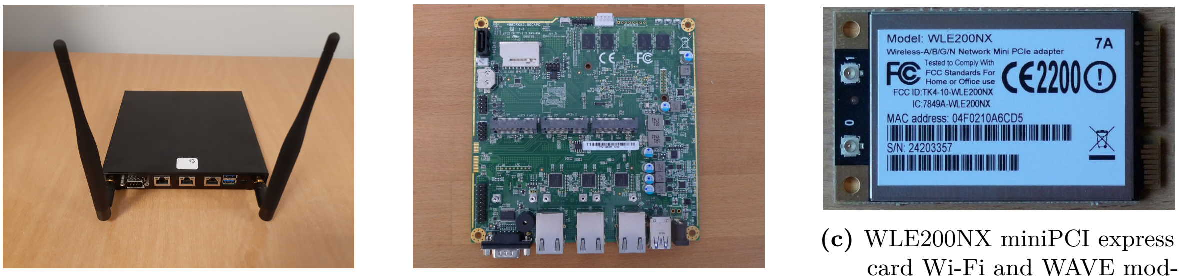
Each smart lamppost can contain Road-Side Units (RSUs) (Figure 3.2a), computing devices such as NVIDIA Jetson Nano (Figure 3.3a), video cameras (Figure 3.3b), passive Radars (Figure 3.3c), Lidars, environmental sensors, reconfigurable radio units, 5G-NR radio and 5G network services, Wi-Fi access points and Long Range Wide-Area Network (LoRaWAN) gateways [11].

Apart from smart lampposts, the project also contemplates the installation and development of several IEEE 802.11p WAVE (WAVE)/ITS-G5 powered OBUs (Figure 3.2a, Figure 3.2b and Figure 3.2c) through the buses of the city of Aveiro, therefore creating a cluster of ITS vehicles that can be used for several experimentations.



(a) Example of smart lamppost that can include a Road-Side Unit (RSU), computing nodes, Radar and video cameras

Figure 3.1: Smart lampposts installed throughout the city of Aveiro



(a) OBU and RSU

(b) OBU and RSU board

(c) WLE200NX miniPCI express card Wi-Fi and WAVE module

Figure 3.2: OBU and RSU equipment details

(a) NVIDIA Jetson Nano used for video processing within RSUs

(b) ReoLink RLC 423 camera

(c) SmartMicro Radar used for the detection of vehicles without OBU

Figure 3.3: Computing and sensing equipment details

The information obtained from the sensors and the communicating vehicles and people is then processed by a centralized cloud, with a set of information aggregators based on the Message Queuing Telemetry Transport (MQTT) protocol (Eclipse Mosquitto broker) for

---

[1] Through the work presented in this dissertation.

internal applications, and the Fiware Orion Context broker for communication with third parties and applications decoupled from the ATCLL infrastructure.

3.3 Vehicular Ad-hoc Networks (VANETs)

This section presents the base concepts of VANETs, their technologies and challenges. Before creating a system, it is important to consider VANETs, since they are the physical and network layers through which the data can be obtained from smart cities roads.

3.3.1 Basic concepts

Vehicular Ad-hoc Networks are a subset of Mobile Ad-hoc Network (MANET), which are networks prepared to manage the high dynamism of nodes - unlike standard IP protocols - where vehicles are capable of wireless communication with the remaining network [12]. Figure 3.4 presents an overview of a VANET.

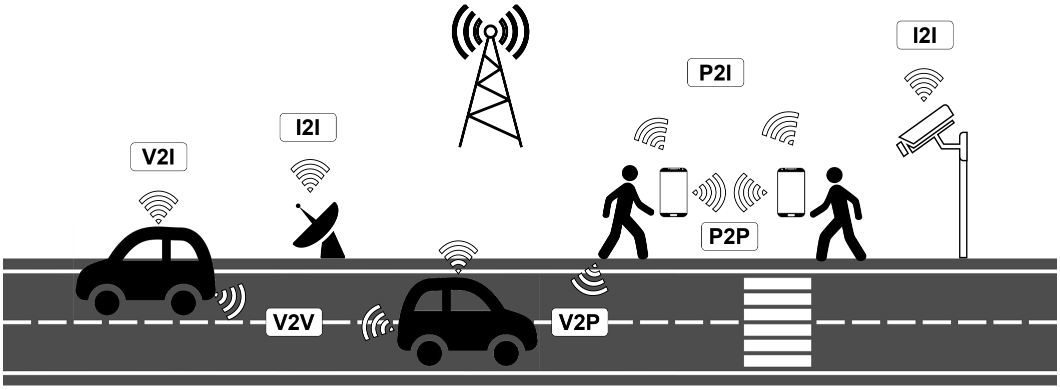


Figure 3.4: Overview of a VANET

In VANETs, the vehicles are part of the network by publishing and receiving information to and from other vehicles (Vehicle to Vehicle (V2V)), infrastructure such as RSUs (Vehicle to Infrastructure (V2I)) and VRUs (Vehicle to Pedestrian (V2P)) - overall called V2X. Vehicles are equipped with ITS equipment, such as OBUs, that contain one or more wireless access technologies and are responsible for processing information from the vehicle sensors, produce messages and transmit them to other nodes of the VANET, such as RSUs.

The infrastructure is another key component of VANETs, with RSUs acting as points of attachment to other nodes (vehicles OBUs, VRUs, and also other infrastructure) connected through wireless access technologies. These RSUs, in addition to video cameras, Radars, Lidars and equivalent sensors, are physically installed within smart cities.

VRUs are another important part of the VANETs. Traditional definitions of VANET exclude the VRUs and their equipment. However, they can easily be integrated through wireless access technologies such as Long Term Evolution (LTE) and 5G, available in their smartphones or equivalent types of equipment. VRUs can also communicate with other VRUs (Pedestrian to Pedestrian (P2P)), infrastructure such as RSUs (Pedestrian to Infrastructure (P2I)) and vehicles (Pedestrian to Vehicle (P2V)) - overall called Pedestrian to Everything (P2X).

3.3.2 Network access technologies

Both vehicular and VRU VANET nodes can use different wireless access technologies to communicate. These technologies can be classified based on their communication range [13]:

- **Short Range** (up to 1 km): ITS-G5 standard (IEEE 802.11p WAVE), 3GPP Cellular Vehicle-to-Everything (C-V2X)/LTE-V2X, Wi-Fi (IEEE 802.11b/g/n/ac/ax), Near Field Communication (NFC) (*e.g.* RFID tags), Bluetooth.
- **Long Range** (more than 1 km): cellular technologies such as LTE, 5G, 6G.

From these, ITS-G5 standard (IEEE 802.11p WAVE), 3GPP C-V2X/LTE-V2X and cellular technologies can be considered as the most pertinent within VANETs [14]:

- **ITS-G5 standard (IEEE 802.11p WAVE)**: Based on WLAN/Wi-Fi Standards (5.9 GHz band), it allows direct device-to-device (ad-hoc) communication with low latency and high reliability, but in short ranges (±1 km); it is low cost/free of cost.
- **3GPP C-V2X/LTE-V2X**: Based on cellular standards (5.9 GHz band). It also allows direct device-to-device (ad-hoc) communication with low latency and high reliability, but in short ranges (±1 km); it has a low cost/free of cost. C-V2X also supports V2N (Device-to-network), enabling cloud services to be part and parcel of the end-to-end solution. C-V2X aims at integrating cellular technologies within V2V scenarios.
- **Cellular technologies**: Including LTE, 5G and 6G. It is based on cellular standards (cellular bands). It allows device-to-other-entities communication through the cellular network, with medium latency and low reliability. It has a longer range at a higher monetary cost.

While both WAVE and C-V2X use the 5.9 GHz band, results from [15] show the C-V2X with potential for smaller user densities (up to 150 users / km^2), with the ITS-G5/WAVE outperforming for higher densities, while results from [16] prove that C-V2X outperforms WAVE in all realistic highway traffic scenarios. However, C-V2X is less mature than WAVE, with first trials only done by the end of 2017.

Cellular technologies like LTE suffer from latency, throughput and responsiveness issues, while 5G tries to address these issues with new radio paradigms, and micro-service oriented core networks [17].

With the future standardization of 6G, with initial pilots expected in 2026 and broad release expected around 2030 [18], new use cases can be built, including time-engineered applications, holographic communications, tactile Internet, and digital twins multi-sense networks [19]. 6G can support new latency requirements and new notifications for both drivers and vehicles, related with different senses - *e.g.* tactile and/or holographic notifications for drivers and VRUs.

Hybrid solutions are also possible by combining multiple access technologies to obtain the best of all technologies at the cost of increased complexity, since a connection manager is required to orchestrate the usage of the different technologies. Examples include Bosch CCUs and OBUs from our group, with both WAVE and 5G.

3.3.3 Challenges

VANETs are characterized by the high mobility of the network nodes, which creates a set of challenges [20]. These challenges shall be taken into consideration in any application or hardware developed to be included within a VANET:

- **High dynamism:** As a consequence of the mobility of vehicular and VRU nodes, the nodes can be connected to a RSU, and seconds later, they can be out of reach of that same RSU. To ensure a good overall Quality of Service (QoS), a seamless handover between RSUs is very important.
- **High need for security:** The usage of privileged information about vehicles and VRUs for complex and critical safety services means not only authenticity, but also that the privacy of the data must be ensured.
- **High need for low latencies:** Low processing and transport latencies are instrumental in keeping the overall QoS for end-users, and also to support critical safety services.
- **Bandwidth limitations:** With the usage of cellular technologies, with significant cost per data, or ITS-G5/C-V2X, with limited bandwidth, ensuring maximum QoS requires that protocols and applications are as efficient as possible in using the communication channels, with minimal overhead.
- **Routing:** It is important to efficiently route the packets from/to vehicular nodes that can change their point of attachment/RSU every few seconds, while keeping overall QoS for critical services.

3.4 STANDARD MESSAGES

For nodes within a VANET, and more globally, to exchange information from or to ITS network nodes (*e.g.* vehicles OBUs, RSUs and other sensors), standardized message formats can be used [21]. These standards allow better interoperability between vendors' applicational stacks and hardware, since they establish a common information set and format.

ITS related standards are defined by entities such as ETSI (within the European Union) or Society of Automotive Engineers (SAE) (within the United States of America). The following sections detail the European message standards for vehicles, events on a road, VRUs and sensors, with the last section presenting details on some non-European standards. Since the system is going to be developed in Europe with cooperation with Bosch, an European business, the European ETSI standards are going to be considered in the system to be developed. This explains the greater detail for each ETSI standard presented in the next sections. However, it is important to emphasize the existence of other standards.

3.4.1 Cooperative Awareness Message (CAM)

Cooperative Awareness Messages (CAMs) are periodic messages exchanged in the ITS network between ITS-Ss to create and maintain awareness of vehicles using the road network or RSUs. A CAM contains status and attribute information of the originating ITS-S [22].

CAMs are used within the context of several VANETs, including in the ATCLL infrastructure and by Bosch own OBUs, for creating awareness about vehicles and city buses.

The content varies depending on the type of ITS-S: for vehicles, the status information includes time, position, motion state, activated systems (*e.g.*, cruise control, pedals, and others), and the attribute information includes data about the dimensions, vehicle type, and role in the road traffic; for RSUs, it contains information, at least, about the station type and location.

Figure 3.5 presents the structure of a CAM. It contains a general ITS header and a basic container, an HF (High-Frequency) container with the fast-changing vehicle data (such as location, heading, or speed), and an LF (Low-Frequency) container with static or slow-changing data (such as the status of the exterior lights or pedals).

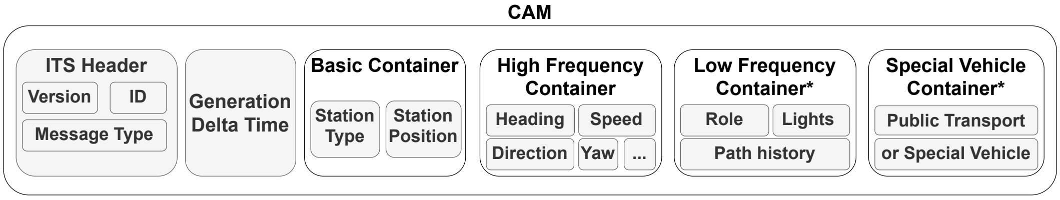


Figure 3.5: CAM structure [22]

CAMs have generation requirements, with the generation frequency between 1 Hz and 10 Hz. The HF container must be in every CAM message, while the low-frequency container can be updated at a maximum of 5 Hz frequency. The generation process must be effective, since the difference between CAM generation time and the time at which the CAM is delivered to networking transport layer shall be less than 50 ms [22].

3.4.2 Decentralized Environmental Notification Message (DENM)

Decentralized Environmental Notification Messages (DENMs) are asynchronous messages exchanged in the ITS network between ITS-Ss to create and maintain awareness about a road event - *e.g.* road hazard or an abnormal traffic condition - such as its type, position, validity, timestamp and the history of the event [23].

Figure 3.6 presents the structure of a DENM. While the content varies depending on the type of event, it is expected that at least the detection time, the position of the event, the type of the related ITS-S and a set of cause codes identifying the type of event are present. These pieces of information are part of the Management Container and Situation Container of a DENM, with other containers to detail about certain types of events - such as the Road Works or the Stationary Vehicle Containers.

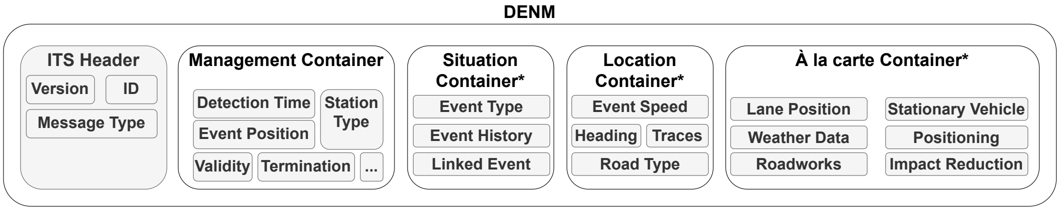


Figure 3.6: DENM structure [23]

DENMs are used within the context of several VANETs, including in the ATCLL infrastructure and by Bosch own OBUs, for creating awareness about road events, including emergency vehicles approaching [24] and adverse weather conditions.

Unlike CAMs, DENMs are generated as events occur, and thus, they are not generated periodically. However, they have a validity period, which after ending, means the DENM can no longer be considered up-to-date.

When an event is no longer occurring, a particular type of DENM, a termination DENM, can be used to signal the end of the event - *e.g.* the end of the road hazard or of adverse weather conditions.

3.4.3 Vulnerable Road User Awareness Message (VAM)

Vulnerable Road User Awareness Messages (VAMs) are periodic messages exchanged in the ITS network between ITS-Ss to create and maintain awareness on VRUs, and support the risk assessment [25]. Just like CAMs, the content varies depending on the type of VRU, but basic status includes information such as time, position, speed, heading, yaw rate and acceleration, orientation, lane position, dimensions and VRU type.

VAMs are used within the context of several VANETs, including in the ATCLL infrastructure - with this dissertation - and by Bosch, for creating awareness about VRUs.

The advantages of a VAM standard message over the usage of a CAM are the added flexibility in terms of fully specifying the VRU type and situation, which is not possible without changing the CAM standard (therefore defeating the purpose of using a standard). VAMs can distinguish between several types of VRU - pedestrian, cyclist, motorcyclist, animal - and within each category, they can distinguish several possible roles (*e.g.* for a cyclist VRU, between bicyclist, a wheelchair user, a horse rider, a roller skater, an e-scooter, and others).

This distinction is crucial: several different VRUs - for example, a child pedestrian or a disabled pedestrian - have different dynamics from a typical pedestrian (as presented in Table 2.1). That information can be used, for example, by safety services to fine-tune an accident prediction algorithm. In addition, using VAMs, several VRUs can be grouped together and represented by a single VAM that will represent a cluster of VRUs, decreasing the network usage. These clusters can be homogeneous (*e.g.*, group of pedestrians) or heterogeneous (*e.g.*, rider on a bicycle). Each cluster is considered as a single entity, and only the cluster head will transmit the VAM.

Figure 3.7 presents the structure of a VAM. The content varies depending on the type of VRU, but it is expected to have, at least, the existence of time and VRU position information. These pieces of information are part of the Basic Container of a VAM, with other containers existing for specifying more detail about certain types of VRUs and their dynamics - such as the High-Frequency container (with the heading, speed, acceleration), a Low-Frequency Container (with information about the type of the VRU), and a set of containers for clustering VRUs.

The usage of VAMs in ITS is still in the early stages. However, the standard is not ignored in the most recent (*i.e.* within 2020) publications and projects. 5G Automotive Association

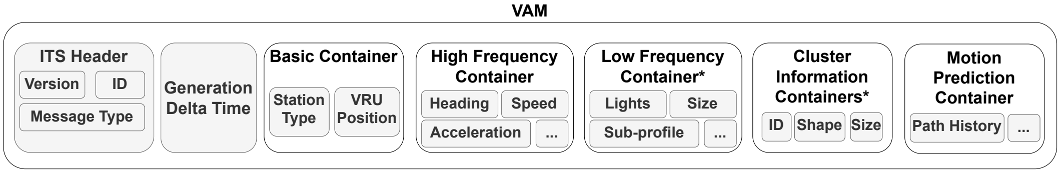


Figure 3.7: VAM structure [25]

defines the VAM as one of the two messages that can be used in the context of communication from/to VRUs (the other message being the U.S. Personal Safety Message (PSM)) [26]. In the project *iMove*, the European Roadmap to Deployment for ITS solutions is presented, with VAMs being referred as one of the message types used for awareness driving [27].

Like CAMs, VAMs have similar generation requirements, with the generation frequency ranging between 1 Hz and 10 Hz. The total end-to-end latency time needs to be lower than 300 milliseconds to maintain a relative age of dynamic data elements not too far from reality [28].

3.4.4 Collective Perception Messages (CPM)

Collective Perception Messages (CPMs) are periodic messages exchanged in the ITS network between ITS-Ss to broadcast information about the current environment as perceived by 1 or more sensors. These messages can therefore be used to share the awareness of one ITS-S with anther ITS-S [29].

Although the work in this dissertation is performed considering CAMs, DENMs and VAMs, CPMs can be used in the future to improve the awareness of both vehicles and VRUs in the road surroundings with standardized sensor information. The choice of not using them in the present work was related to keeping backwards compatibility with other existing services within the ATCLL infrastructure.

Sensors from a vehicle, a VRU and smart city infrastructure can use CPMs to exchange the information obtained from their surroundings, improving the awareness of the situation for all ITS-S and services.

Figure 3.8 presents the structure of a CPM. While the content varies depending on the number of sensor information, it is expected the presence of at least a Management Container. This container provides basic information about the ITS-S that generated the message with an optional Station Data container available to provide more information about it. Apart from these containers, two containers are set for each sensor information - the Sensor Information and the Perceived Object.

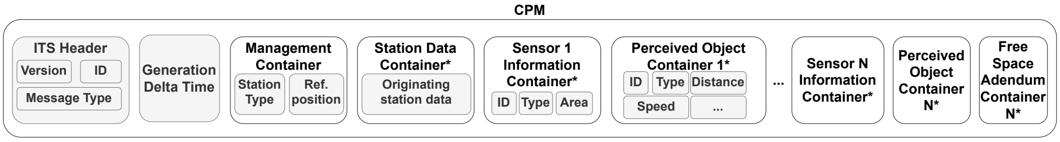


Figure 3.8: CPM structure [29]

The Sensor Information Container describes the sensor type - *e.g.* Radar, Lidar, video cameras or fusion algorithms - and the area the sensor covers. At the same time, the Perceived Object Container identifies the objected perceived by the sensor, the classification, the confidence of the classification, and several data about its dynamics, such as distance, speed, acceleration and angle.

The information from the Perceived Object container means that CPM messages can support the exchange of information from a detection algorithm from smart city infrastructures such as Radar, Lidars and video-cameras.

Like CAMs and VAMs, CPMs have similar generation requirements, with the generation frequency between 1 Hz and 10 Hz.

3.4.5 Non European alternatives

The presented ETSI standards are not an European exclusive. The United States Society of Automotive Engineers J2735 standard [30] defines a set of equivalent messages to the CAMs and VAMs, the Basic Safety Message and Personal Safety Message, respectively.

The DENMs, however, do not possess a direct equivalent, with road events being described by Emergency Vehicle Alert (EVA) to announce the presence of emergency vehicles, the Intersection Collision Announcement (ICA) to inform about a potential intersection collision, and the Traveler Information Message (TIM) to announce about traffic accidents and pre-planned roadwork events. The PSM and Basic Safety Message (BSM) messages are also capable of providing information about the dangerous situations involving VRUs and vehicles, respectively [31].

3.5 Sensor units

One of the main steps of any system to predict potentially dangerous situations in a road is the awareness/perception stage, where information about vehicles and VRUs is obtained.

This set of information can be obtained through a wide range of sensors, such as Inertial Measurement Units (IMUs), Global Navigation Satellite System (GNSS) sensors (*e.g.* GPS, GLONASS), visual sensors (*i.e.* cameras), Radars, Lidars or infra-red sensors.

A short overview of the key sensors in the vehicles is presented in [32] and [33]: GNSS such as GPS, IMU, Camera, Radar and Lidar:

- **GNSS**: Very affordable and simple, but does not possess lane-accuracy precision (at civil implementations), and is not immune to discontinuities of service (*e.g.* within tunnels) [34].
- **IMU**: The usage of accelerometer, gyroscope, and magnetometer can be used to improve the information from other sensors [34].
- **Cameras**: More affordable option, and with appropriated data processing, it can detect and track moving and static objects - *e.g.* through algorithms such as You Only Look Once real-time object detection system (YOLO). Cameras can provide high-resolution color images of the surroundings. However, they are not immune to weather and illumination adverse conditions, nor to the obstruction or non-line-of-sight (NLOS) objects [35].

- **Radar**: Uses the Doppler property of electromagnetic waves to determine the relative speed and position of objects. Radars can be used for measuring velocity, distance and generating maps of the environment, and identifying hard-to-spot obstacles under difficult situations for other sensors, such as weather (*e.g.* rain, fog) and illumination adverse conditions (*e.g.* at night). However, it has more difficulty detecting static, stationary objects, and has coarse resolution compared to other sensors (useful for example in distinguishing type of objects), such as cameras [35]. It has been used in the context of safety of VRUs, for example in [36].
- **LIght Detection And Ranging**: Uses reflection of emitting pulses of light to estimate a distance to an object and generate a 3D representation of the scene in the form of a point cloud. Lidars can be used for measuring velocity, static mapping environment, and identifying objects such as vehicles and pedestrians with reliability and precision at day and night with a broader range, but at a higher cost and without being able to provide information about color [35].

The limits of the usage of camera sensors and computer vision are remembered in [37]. They do not perform well in poor visibility conditions - at night time, in bad weather conditions, or when the pedestrians are not close enough - or in non-line-of-sight (NLOS) position. The authors also remember that, in GPS measurements, 10 m of positioning error should be expected even in good weather conditions.

Tesla is trying to mitigate these issues by using several cameras [38] installed in a car and auto labeling techniques such as NeRF [39]. The alternative is the usage of cameras associated with Radar or Lidar, and the fusion of that information.

Other alternatives exist to detect vehicles, VRUs and other elements of a road. For example, the work in [40] proposes, instead of using complex techniques like image processing to detect VRUs, the usage of passive infra-red sensors to create a temperature model to detect abnormalities in temperature that will correspond to the VRU. Infra-red sensors have more usages, with [41] presenting a cooperative system of sensors, consisting of near-infrared cameras inside the vehicle and wireless personal area network modules for the determination of the relative position of the road users. The Data Fusion algorithm, responsible for detecting potential accidents, is based on a Detection-Classification-Tracking pipeline to ensure that all data of importance is used within a complex environment.

3.6 SENSOR FUSION

While some sensors are more precise than others, all have some sort of error. In addition, not all sensors perform well in all conditions - *e.g.* the GPS discontinuities inside a tunnel. Merging information from several different sensors to complement one another provides a more accurate and robust perception.

Sensor fusion allows to overcome imperfections of the data - imprecision, discontinuities, inconsistencies - or create correlations between different sets of data (*e.g.* data from a Lidar and a video camera) [42]. *Sensor fusion* can be defined as "the combination of sensory data or data derived from sensory data, such that the resulting information is in some sense better than it would be possible when these sources were used individually" [43]. The authors briefly review some of the most common approaches:

- **U.S. Joint Directors of Laboratories (JDL) fusion architecture**: abstract, data-centered model, useful for defining the main steps of the fusion process.
- **Waterfall model**: similar to the JDL model, mostly used in defense data fusion processes.
- **Intelligence cycle and Boyd Model**: cyclic models that are effective for giving an overview of the overall task of the system without being able to separate the different sensor fusion tasks.
- **LAAS architecture**: developed for mobile robot systems, very good at guiding a designer in implementing an entire sensor fusion system, from the sensors to the actuation phase.
- **Omnibus model**: unlike the JDL model, the Omnibus model defines the ordering of processes and makes the cyclic nature explicit. It uses general terminology, but is not easily decomposable into modules.
- **Mr. Fusion**: it aims at data at the application level, for example, the output from several network servers. The architecture consists of two main subsystems that are connected through a broker-based architecture.

Considering the databases already existing in the ATCLL infrastructure and the need for an extensible system, the JDL fusion architecture (adapted to automotive applications) and the usage of a broker-based architecture, *i.e.* an event-driven architecture, are promising approaches to the system to be developed.

The work by [44] introduces the U.S. JDL data fusion group model for sensor fusion revised for automotive applications. This model is critical in defining a standardized structure for the ITS safety applications.

The authors in [44] also enumerate three different deployment solutions: centralized (based on a cloud), distributed (based on edge nodes), and hybrid alternatives (based on orchestration of a centralized plus distributed architecture). While central fusion architectures can deliver higher performance at the processing level, they can compromise latencies in the transport times of the data. By opposition, a distributed/edge solution, by processing the information closer from its generation source, will tend to decrease transport times of the data, at the cost of potentially higher processing latencies - since the resources of edge nodes tend to be less than in a centralized cloud.

Sensor fusion is important for semi-autonomous and autonomous vehicles systems, with [32] defining an autonomous vehicle pipeline as *Sensors -> Perception -> Planning -> Control.* Sensor fusion is, in this context, an approach for combining data from disparate – yet complementary – sources, such that coherent information is created for the perception phase. More sensors in a system based on sensor fusion will increase the performance and robustness of the system.

Several sensor fusion techniques exist, including Unsupervised Machine Learning, Bayesian inference, and Kalman Filters [45]:

- **Unsupervised Machine Learning**: Techniques such as clustering or reinforcement learning allow sensor fusion systems to learn in real-time on the best ways to join the data (*e.g.* the best weights for each measurement from each type of sensor).
- **Bayesian inference**: The usage of Bayes rule to determine the probability, for example, of the vehicle or VRUs being in a particular position based on several sensor measurements.
- **Kalman filters**: Optimal state estimator algorithm [46]. A predicted state estimate and a measurement from one or more sensors allow an optimal state estimate computation. The measurements have different weights, with the importance of each one depending on the Gaussian noise of the measurement (the smaller the noise, the more valuable the measurement). Extensions and alternatives to Kalman Filters for nonlinear and non-Gaussian noise, like the Extended Kalman Filter and Particle Filters, can be considered.

Kalman filters, or particle filters, can be used to join several measurements of the same type of GPS sensor - *e.g.* as described by [46] - or data from different types of sensors - *e.g.* Radar and video camera information as described by [47].

3.7 Information aggregators and persistence solutions

The information obtained through sensing and sensor fusion needs to be distributed to applications. Fiware[2] is an example of an open-source platform designed to gather, manage and process data dynamically. The key component is the Fiware-Orion Context Broker[3], responsible for handling data updates, queries, and subscriptions. The data can have different sources/publishers, identified by the Fiware-Service and Fiware-Servicepath attributes, which allows a logical separation between services. The *Mobiwise* project[4] studied the broker as a limiting component, showing that the Orion is more effective in publication times for 20000 edges compared to an alternative, the ETSI M2M.

The data must also be persisted within a set of databases. Traditional databases follow a relational schema. However, to support solutions focused on the persistence of information from standardized messages, relational schemas are very limiting, without the necessary flexibility for changing, adding, and removing fields as standards evolve. NoSQL solutions, such as documented-oriented databases, offer a better solution, ensuring easy change of formats while allowing schema validation if required [48]. Several solutions, including Redis and MongoDB, are available, with MongoDB achieving the best overall performance for scenarios with the high demand of reading and writing operations [49].

In the future, the database to be built could be converted into a semantic database based on ontologies, such as the Semantic Sensor Network ontology (an ontology for describing sensors and their observations), allowing more complex queries. The authors in [50] focus on semantic

---

[2] `https://www.fiware.org/`

[3] `https://fiware-orion.readthedocs.io/en/master/`

[4] Results presented at the final workshop, available at `http://mobiwise.av.it.pt/final_workshop/index.html`.

knowledge concepts in the sensor fusion part, describing OpenIoT, a platform enabling the semantic interoperability of IoT services in the cloud, using semantic web standards such as W3C Semantic Sensor Networks ontology.

3.8 Safety of Vulnerable Road Users

Considering the availability of data, fused and persisted, applications can leverage that information. Within the context of Intelligent Transport Systems, applications can be categorized into two main groups [10]:

- **Safety Applications:** designed to reduce and prevent accidents and collisions between vehicles and/or VRUs, and warn about hazards in the roads. Since they are related to avoid and notify potentially dangerous situations to people's and animals' lives, they are the most crucial ones.
- **Non-Safety Applications:** for infotainment and comfort needs or for traffic management (*e.g.*, to optimize and monitor the traffic flow).

Within safety applications, solutions like emergency braking, optimal speed advisory, overtake support, adaptive cruise control, e-calls, and forward collision avoidance have been developed [15] [51]. From these, the most important with immediate effects are the forward collision avoidance between vehicles and VRUs.

VRU protection falls within ITS safety applications. Achieving VRU protection can be done by passive VRU protection - the reduction of the impact on a VRU when the accident is no longer avoidable - or active VRU protection - actively avoiding the collision [52]. The current technologies and safety methods for VRU active safety can be further divided into three categories [40]:

- **In-vehicle systems**: based on OBUs within vehicles.
- **Carried by-the-pedestrian systems**: *e.g.* based on a smartphone or wearable application.
- **Indirect systems:** based on systems connected to the road infrastructure.

The problem of VRU safety is not new. The work in [53] shows that the issue of VRU safety improvement has been discussed for at least 30 years - since 1991. The authors discuss the possibility of equipping cars with VRU detection devices, or equipping VRUs with devices that provide advice in difficult situations.

By 2004, the work in [54] concluded that a warning system to alert the driver as soon as a VRU is detected and classified by the sensors could be sufficient to reduce the number of accidents, with the next logical step being to act on the car braking system using the collision warnings information.

By 2006, the WATCH-OVER [55] project aimed to design and develop a system for preventing accidents with VRUs, while overcoming the limitations of sensor technologies of not being able to see behind obstacles or having limited view range. Therefore, a system was designed based on a cloud solution that allows communication between OBUs and the user module (*e.g.* a smartphone) through it, without using road infrastructure such as RSUs.

By 2010, the work in [56] defines an architecture for the detection of VRUs based on cooperative sensors (sensors that localize cooperative targets such as vehicles and VRUs). It also presents a wearable tag for sensing the VRU position and other dynamics. Wearables can therefore be considered a future alternative to VRUs carrying smartphones.

More recently, in 2020, within the context of 5GINFIRE [57], the VRU-SAFE project was developed as a system capable of calculating trajectories of moving nodes using real-time context information from OBUs and VRUs and triggering a notification system. The VRU-SAFE project presents a hybrid architecture combining edge and cloud computing. The edge-based processing minimizes the end-to-end transmission delay. In contrast, the Cloud-based solution minimizes the processing delays via powerful virtualized, centralized resources, and the controller that selects between cloud and edge.

The work of [58] emphasizes the relevance of hybrid solutions by defining a Multi-access Edge Computing (MEC) based architecture where VRUs and vehicles communicate with points of access (*e.g.* RSUs), which convey standard messages (in this architecture SAE BSM) to a central cloud that processes the collision risk. The authors also address the addition of smartphones of the VRUs to the overall architecture, with smartphones working as the equivalent to VRUs of OBUs to vehicles. The lack of support for network technologies, such as WAVE, and the limited computational power and energy limitations of smartphones are, however, limitations of such solutions. The defined architecture addresses both issues by shifting most of the computation to the more resourceful cloud.

The work in [59] presents a complete standard-compliant design and implementation of a VRU warning system and its experimental evaluation over another MEC-based and cloud-based architecture, reemphasizing a hybrid architecture solution.

3.8.1 Detection of potential dangerous road situations

The previous section focused on an overview of the leading architectural decisions around systems to predict and detect potentially dangerous situations on the road. However, no focus was given to the main concepts of the algorithms responsible for such detections.

While potentially dangerous road situations can go beyond potential collisions, this section focuses on algorithms to predict such events.

Such algorithms can reach great complexity, since they have to deal with non-rigid moving objects in a permanently changing environment [41]. In addition, those objects - vehicles and VRUs - do not share the same characteristics, with VRUs inertia being much less than vehicles (*e.g.* a pedestrian can do a U-turn in less than one second while a vehicle cannot). As such, their motion dynamic is more challenging to predict [25].

In 2014, the work in [40] proposes a model to detect if a VRU aims to cross a road in a risky zone, and to issue alerts to the vehicles nearby.

A set of equations for a collision warning system between 2 vehicles is defined in [60]. The equations are based on the comparison of a time-to-intersection and a time-to-avoidance to decide if there is a potential collision. A Convolutional Neural Network based solution is defined by [61] to detect a vehicle in front of another, and therefore, to predict a collision.

This solution, while good for vehicle-vehicle collision, could be generalized for VRU and vehicle collision, with one or both of them detecting a potential collision if a vehicle or VRU is detected.

More focused on the dynamics between VRUs and vehicles, [62] presents a set of kinematic equations based on the Newton's Second Law. These equations can be used to determine the distance and time until the vehicle reaches a collision point with a VRU. In addition, [63] defines the concept of Time To Collision (TTC). TTC is frequently used as a measurement of how urgent a situation is. Within their collision risk assessment, [58] presents a trajectory-based collision detection system, based on the Euclidean distance between the vehicle and VRU. Vehicle and VRU position, speed, and acceleration are used to predict future collisions that will happen if the time until the minimum distance between the vehicle and VRU is shorter than the space to collision, s_{col}. This solution, while using some information about the vehicle and VRU, relies too strongly on the correct definition of the s_{col} parameter, and ignores important information such as the heading of the vehicle and VRU.

The work in [37] defines the concept of Geographical Destination Area (GDA). If a vehicle is within this GDA, it will be dangerous for pedestrians. Within the definition of the concept of GDA, a TTC between VRU and vehicle is considered. Such time is computed based on the subsequent positions of a vehicle represented by a bicycle kinematic model. This model considers the heading information, increasing its precision and accuracy compared to previous alternatives. The TTC allows the computation of a point of collision between the VRU and vehicle with a potential collision risk occurring if the pedestrian can reach it, at its current speed, before the vehicle.

While the previous approach considers the heading of the nodes, it fails to consider possible variations in the characteristics of the vehicles that might affect the braking times. It also fails to consider different drivers characteristics that affect their reaction times. As Figure 3.9 depicts, in an actual situation of potential collision between the VRU and the vehicle, it must be considered a reaction time and a braking distance until the vehicle is stopped. The work developed by [64] considers these factors within the computation of three zone lengths: safe, unsafe and potentially unsafe zones. These zones are defined with a center either in the vehicle or VRU.

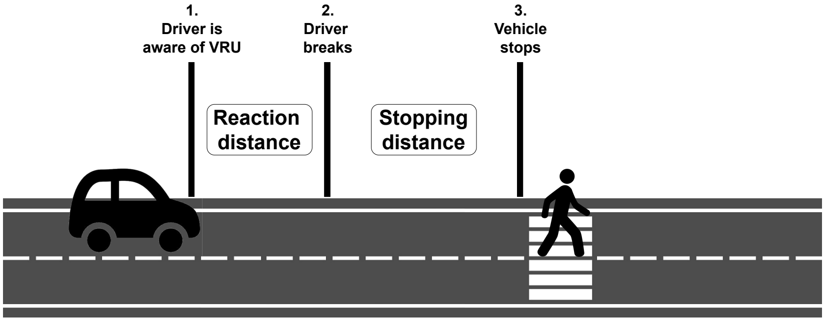


Figure 3.9: Driver reaction times and vehicle deceleration within the context of a potential collision

Figure 3.10 illustrates an example of a situation with the zone center defined in the VRU.

Vehicle a is within the unsafe zone (in red), and therefore a collision is deemed unavoidable. Vehicle b is within the potentially unsafe zone (in orange), where it is not sure that an accident might happen with the current situation. Vehicle c is within the safe zone (in green), where it is sure an accident will not happen with the current situation; however, it is always possible that in a subsequent evaluation of the situation, this changes if, for example, the vehicle suddenly accelerates significantly. This division between zones presents the advantage of allowing the filtering of vehicles and VRUs with a large distance between each other (all nodes that are very far away will, by definition, belong to a safe zone).

As represented, the zones consider vehicles in front or behind the VRU. Such behavior might or might not be wanted in a potential collision system, considering that the system only wants to consider typical forward-collision situations. In that case, the represented zones are only valid for vehicles before (left to) the VRU, with vehicles after (right to) the VRU not being deemed a collision since the vehicle is going forward. However, in situations where the vehicle is reversing, this analysis would not be correct. The best option is to consider the zone length only if the VRU and vehicle are in the same direction.

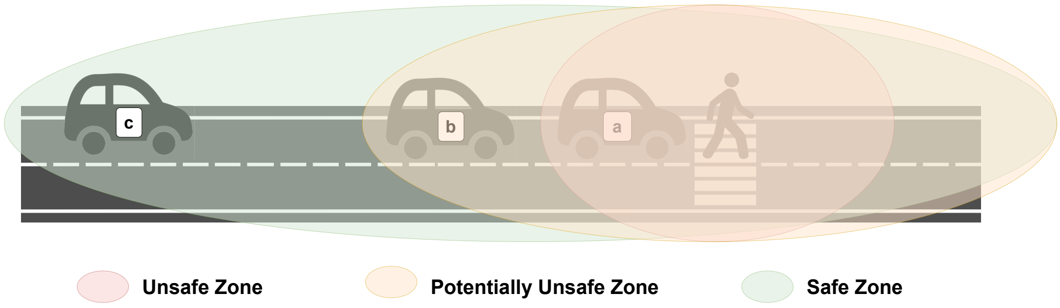


Figure 3.10: Example of collision risk zones (centered in the VRU)

3.8.2 Path and activity prediction

Generally, collision risk indicators require a prediction of the trajectory/path followed by the vehicle or VRU. To both nodes, the predictions can be different. While vehicles trajectories are constrained by the road topography, the traffic, the traffic code, their size, mass, and others, the VRU has much more freedom to move [25].

There are several approaches for predicting the future trajectory of both vehicle and VRU [65]. Probabilistic approaches can be considered, such as Hidden Markov models (already used to predict the driver intent at a four-way intersection) and Markov decision processes (already used to predict the longitudinal speed over time). Machine Learning techniques have also been used, such as Long-term short-term RNNs that have been used for vehicle path prediction [65] or deep learning. While all solutions can be used for the generic trajectory prediction problem, they are too complex for a swift response from the algorithm in the context of a safety service. Machine learning techniques, and especially deep learning techniques, possess an additional problem, with the lack of labeled data, making more difficult the training phase in such techniques.

Apart from these models, predicting the VRUs' movement should also consider its current activity. Understanding the VRU activity - if a VRU is walking, running, or stopped - allows predicting better its future path, *e.g.* if it is stopped, it is unlikely that it will be running in the next second.

The work in [66] presents a solution to recognize simple and complex activities using inertial sensors (accelerometers and gyroscopes). The activity recognition task is approached as a supervised machine learning problem. It is concluded that simple activities can be recognized with very high accuracy using only a single smartphone carried naturally, and presented as future work the combination of a mobile phone with other sensors.

The work in [67] presents a solution that uses the back camera of the mobile phone to detect vehicles approaching the user, alerting the user of a potentially unsafe situation. A machine learning algorithm is implemented on the phone to detect moving vehicles' front views and back views. This solution can be considered as a fail-safe solution to warn VRUs if other sensors fail. The obtained results showed a 77% true positive rate.

Google Activity Recognition API [68] is a solution that, based on information from the smartphone sensors, it can detect the user's current activity - in a vehicle, on a bicycle, running, walking, or still. While incapable of detecting the VRU activity sometimes (mainly because of the API inability to take into account the differences in the smartphone placement, such as carrying the device in hand vs. inside a bag), it proves to be one of the best API in terms of accuracy vs. power consumption, critical in mobile ITS-S such as smartphones [69]. Alternatives for non-Android smartphones, such as Apple *CMMotionActivity*[5] API for iOS devices, are also available.

3.8.3 Timing constraints

Being a safety application means not only that the accuracy of the detection must be high, but also that the timings must allow all the involved nodes to be notified within time. An out-of-time notification will increase the number of false positives and/or false negatives, which are both harmful [58]:

- **False Positives** correspond to pairs of users (vehicles or pedestrians) for which a collision alert is issued, but the users do not collide and will undermine the users' confidence in the system.
- **False Negatives** correspond to pairs of users for which no alert is issued, but the users do collide, possibly resulting in the loss of lives.

The work in [70] relates the end-to-end delay (between VRUs and vehicles awareness and notification of potential accident) with the accuracy of its location. If the VRUs position inaccuracy is 0.5 m, the overall end-to-end latency must not exceed 100 ms. If the position inaccuracy is 1.0 m, the overall end-to-end latency must not exceed 300 ms.

While specific to intersection collision scenarios, ETSI also defines a ceiling of 300 ms for end-to-end latency in safety applications [28].

---

[5] `https://developer.apple.com/documentation/coremotion/cmmotionactivity`

3.9 SUMMARY

This chapter presented the background concepts and state-of-the-art works required to develop the VRU and vehicle detection and accident avoidance.

The smart city of Aveiro, powered by the ATCLL infrastructure, allows the creation of ITS applications that leverage the concepts and benefits of VANETs and multiple sensors. Vehicles and VRUs can be part of a smart city either by using OBUs, ITS-Ss such as smartphones, or by being detected by infrastructure sensors, such as Radars, video cameras and Lidars.

Each sensor possesses several strengths and weaknesses, but none can be used in all circumstances. Therefore, sensor fusion techniques, either powered by Kalman or particle filters or machine learning techniques, are essential to obtain a complete, accurate view of vehicles and VRUs. Such accuracy is especially critical for safety ITS-S applications, including applications that aim to detect potentially dangerous situations.

The most prevalent solutions are the ones based on the usage of VANET infrastructure and information to predict a potential collision between vehicles, and vehicles and VRUs. From more straightforward - and therefore quicker - kinematics-based algorithms to more complex Neural Networks algorithms, several algorithms are possible. While all algorithms, to some extent, fulfill the functional requirement of predicting potentially dangerous situations, not all are compatible with timing requirements.

Since the system to be developed is a critical safety service, timing requirements are exact and critical, and shall be considered in architectural decisions, namely in the decision between edge, centralized cloud, or hybrid solutions, or in the choice of both the aggregator and persistence solutions for the information of sensors and VANETs.

The next chapter will present the overall architecture of the system to be developed, whose decisions in the architecture have been heavily influenced by these chapter’s conclusions.

CHAPTER 4

VRU Safety Architecture

What is a decision? It's a tool to remove confusion.
— Brian Valentine

Chapter 2 defined the safety problem in roads and the main requirements to be fulfilled by the system to be developed, while Chapter 3 defined the related state-of-the-art. This chapter presents the architecture for the VRU safety approach.

First, Section 4.1 presents an overview of the system's architecture, including its main components and roles. Then, Section 4.2 and Section 4.3 provide an overview of the different information sources - vehicles, VRUs and infrastructure sensors such as cameras and Radars - as well as the required information from each source. Afterwards, Section 4.4 provides details on the architecture and algorithms of the safety service to be developed, emphasizing how the information aggregation is performed, and how potentially dangerous situations are detected. Finally, Section 4.5 provides a summary of the significant architectural decisions.

4.1 Proposed architecture

This section designs the architecture of the proposed approach. Figure 4.1 presents an overview of the architecture, including the vehicular and VRU nodes, the information aggregator, the smart city infrastructure and the services that are part of this system, namely a Safety Service.

When designing a solution architecture for the system, two main blocks can be highlighted:

1. **Information Aggregator System**: aggregation of information from multiple sensors into an information aggregator.
2. **VRU Safety Service/Application**: development of services and applications that interact with users to warn them about potentially dangerous and harmful situations in a road involving vulnerable users.

The information aggregator systems - both in a cloud and edge deployment - are responsible for aggregating the information from all nodes – vehicles and VRUs - and other infrastructure

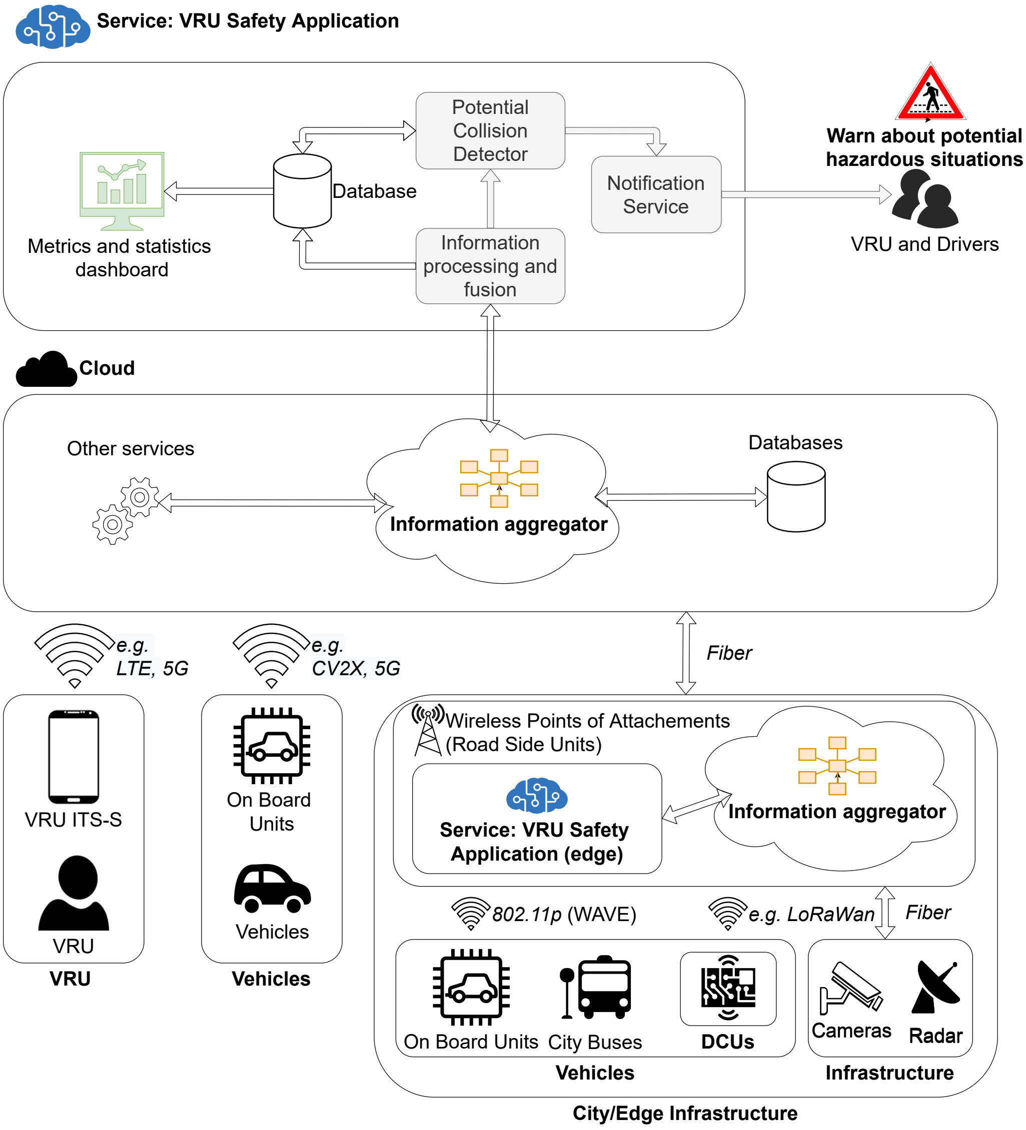


Figure 4.1: Detailed Architecture Overview

- such as Radars, cameras, Lidars – into a single platform accessible by all nodes and potential applications. These applications can then use the aggregated information.

Gathering all this information is a significant challenge, since it means aggregating multiple data sources coming from diverse systems, relayed through different wireless technologies, and managed by different entities (Section 3.7).

This aggregation can be achieved by routing the information to a centralized information aggregator that can store and provide the information to other services. This centralization allows services to obtain the information by accessing a single well-known point without managing how to obtain the information. Since the information is sent asynchronously, an event-driven architecture based on the usage of a broker is ideal as an information aggregator.

An event-driven architecture supported by a broker implements a pattern of publishers/generators and consumers/subscribers of events, events being occurrences in the past of something, such as messages. Unlike patterns such as request-response, they allow publishers and subscribers to be loosely coupled (since both publishers and subscribers only need to know the broker) and communicate asynchronously through the mediation of a message broker.

Vehicles, infrastructure, and VRU nodes can communicate with this information aggregator/broker to publish their information into topics, and applications can subscribe to those topics, without knowing about other nodes. Section 4.2 will further detail possible sources of data for vehicles and VRUs in each node, while Section 4.3 will detail the set of required information.

Information aggregators can be deployed either in the centralized cloud or in an edge point, with both solutions presenting different strengths and challenges. In the edge computing solutions, the processing occurs closer to the nodes (*i.e.*, vehicles/VRU), resulting in potentially lower latencies; however, this is done at the cost of higher deployment complexity, since more equipment is required and orchestration between the different instances is required. By opposition, in a centralized solution no orchestration between edge points is required, but the latencies can be more significant.

RSUs can be considered as edge points with significant computational power to act as an edge alternative to a centralized information aggregator, and can communicate with vehicles but also other city infrastructure, including Data Collection Units (DCUs) in vehicles, cameras, Radars, and other sensors dispersed throughout a smart city. These infrastructure nodes can be used to detect and provide information about vehicles and VRUs, complementing or substituting the primary source of information for both vehicles and VRUs. The nodes can communicate through a set of access technologies, from fiber to cellular (Long Term Evolution Advanced (LTE-A) or 5G), and through a set of standard messages, such as ETSI CPMs, or through proprietary or custom message formats.

The second major part of the architecture is the set of services built on top of the information aggregator/broker, and the processing of events as they happen.

ITS services can be classified as safety or non-safety [9]. Keeping in mind that the problem to be solved is the safety on roads, a Safety Service needs to be developed: the VRU Safety Service is an example of such an ITS Safety Service. This Safety Service uses information from vehicles and VRUs, provided by the information aggregator within the cloud and edge points, to detect and warn about potentially harmful situations for the safety problems both of VRUs and vehicles (*i.e.*, collisions). If the likelihood of such situations is high, the Safety Service also notifies the VRUs and vehicle through standard messages (*e.g.* ETSI DENMs).

The VRU Safety Service, after receiving the relevant information from the subscribed topics of the information aggregator, either in a centralized cloud or edge points, performs a set of operations within its four main modules: the information processing and fusion, the potential collision detector, the notification service, and a metrics and statistics dashboard. Section 4.4 will detail each module, their responsibilities, and their inner workings.

4.2 Information sources

Vehicles and VRUs need to publish information about themselves to the information aggregator/message broker. In order to obtain such information, it requires a set of sensors that obtain, either directly or indirectly, information about the vehicles and VRUs.

4.2.1 Vehicle equipment

As previously described, vehicles can be equipped with OBUs. With several hardware formats, these devices are responsible for periodically reading and processing information about the vehicle. Such information can come from external sensors or from sensors within the vehicle that are accessible through the vehicle internal communication network, *e.g.* Controller Area Network bus (CAN). Communicating with the internal vehicle network also makes it possible to actuate on the vehicle systems (outside the scope of this work).

The information that vehicles can obtain depend on the type of sensors present; a minimum set with basic status data such as position, speed, heading, and vehicle class is expected. This information set is sent in a standardized message format such as ETSI CAMs or the United States SAE BSMs. The vehicles also subscribe to warnings provided by the information aggregator in formats such as ETSI DENMs.

After reading and processing the information, the OBU is responsible for periodically creating a message, preferably in a standard format such as ETSI CAMs or the U.S. SAE BSMs, and transmitting it to other entities.

OBUs communicate with other entities through a set of wireless access technologies, including WAVE, C-V2X and others (as described in Section 3.3.2). Vehicles can also communicate directly with the infrastructure, through cellular technologies such as LTE or 5G, without the need for intermediation from RSUs and information aggregators.

The information flow is not just in the uplink direction. Vehicles can also receive information from other entities, including other vehicles, VRUs and other sensors and applications, to improve the perception of the vehicle systems about the world and to warn vehicles and their drivers about events in the world, including potentially dangerous situations (through standardized messages such as ETSI DENMs).

4.2.2 Vulnerable Road User ITS-S application

The second type of node that can be considered is a VRU. Previously defined in Chapter 2, the VRU is part of the overall system by communicating through a VRU ITS-S. This ITS-S is capable of communicating using wireless access technologies and publish a set of information about the VRU (such as position, heading, speed and profile[1]). This set of information can be published in a standardized format such as ETSI VAM or SAE PSMs.

Unlike vehicles, VRUs do not have a standardized hardware module to obtain basic information about them, and make them part of a vehicular network and available to ITS applications. Several solutions can be considered for the hardware of this module, including any wearable device with processing and networking capabilities. Wearables (*e.g.* smartphones, smartwatches, smartglasses) are the best devices to communicate with VRUs, even non-human VRUs (*e.g.* dogs and other animals), since they mean the least inconvenience possible for the VRU. One of the requirements of the ITS-S application is reading, processing information,

---

[1] As detailed in Section 2.2.

and producing messages periodically and at a fast rate (for example, 10 Hz for ETSI VAMs). Therefore, the wearable device must be able to hold a significant charge, eliminating most smartwatches and smartglasses. Moreover, it is essential to have as much information as possible about the VRU, and this requires a large set of sensors. Finally, it is necessary to have at least one communication technology (but preferable more than one to explore the benefits of multi-homing and avoid complete dependency on just one technology), eliminating most wearables, apart from smartphones.

Smartphones are an ideal hardware platform for inserting VRUs within an ITS system, *i.e.* an ideal ITS-S: they are ubiquitous (most people already possess one), relatively cheap (in comparison to other technologies, including city infrastructure), they possess a myriad of sensors, several network technologies (cellular, Wi-Fi, Bluetooth, NFC) and are highly programmable.

With a smartphone, it is possible to build an ITS-S application that obtains information from sensors and combines the information to produce standard messages - *e.g.*, ETSI VAMs - that are sent to other entities periodically, while also receiving messages from other vehicles, VRUs and events. These events can include warning messages that the ITS-S application can then use to notify visually or even with sound (as specified by requirements in Section 2.4.2).

4.2.3 The infrastructure role

To ensure that the system always works, even if vehicles or VRUs are not equipped with OBUs or a VRU ITS-S (smartphone or equivalent), indirect detection and obtention of information from another set of sensors is crucial.

In the scenario with OBUs and VRU ITS-S available, additional data sources help to improve the quality of the information through data aggregation and data fusion algorithms.

Several sources can be considered, including cameras, Radar, and Lidar, with the first two being considered in this system. Video cameras are cheaper and easier to use sensors by requiring only an optical camera - relatively cheap - and one of several detection algorithms with relatively easy deployment and tuning available. Several state-of-the-art algorithms - *e.g.* YOLO or NVIDIA Deepstream SDK - are capable of detecting vehicles, VRUs and animals and other classes that are not useful in the context of this system. However, even if the accuracy of these algorithms, especially the state-of-the-art algorithms, tend to be high, the set of the given information is still limited, with typical outputs of these algorithms being the class label (vehicle, person, animal), confidence in the classification and the bounding boxes coordinates. Information such as speed, heading, or absolute location coordinates is not directly available, meaning that they need to be inferred. Further discussion on this issue is provided in Section 5.2.2.

Radars are another alternative. Based on the Doppler principle, they can detect multiple objects and define basic proprieties, such as speed and dimensions. They are ideal for situations where the accuracy of the location and speed of objects is critical, and is less susceptible to weather conditions than video cameras. Therefore, they are also considered during the development and evaluation of the system. Further discussion is provided in Section 5.2.3.

4.3 Required Information

Section 4.2 described the possible information sources. This section describes the type of data that needs to be exchanged.

The exchanged information from each node shall follow a well-defined standardized format to support several services producing and subscribing to the information seamlessly. The usage of standardized, well-defined formats creates an uniform way for other ITS applications to use the information, ensuring compatibility at industry-standard with other vendors' equipment and applications, and therefore fulfilling portability requirements RP_1 (as described in Table 2.5). For example, in the context of this dissertation, to allow the system to work with Bosch infrastructure and CCUs, the exchanged information follows standard formats defined by ETSI.

The most basic awareness information required is the location of the node (vehicle or VRU). This information can be codified in a relative (*e.g.*, bounding boxes in a camera detection algorithm or in the Radar classification) or absolute referential (*e.g.*, EPSG:4326/World Geodetic System 1984 (WGS84) coordinates as provided by the GPS module in a OBU or smartphone). Coordinates in a relative referential require a conversion to a global/absolute referential, so that all nodes can understand the real position of the node.

However, and especially for a Safety Service responsible for detecting potentially dangerous situations, the location is insufficient for awareness of the whole situation and dynamics of the node. Information like speed, heading, acceleration, and others is critical.

Table 4.1 lists, in the context of this system, the necessary information to be obtained from vehicles, while Table 4.2 lists the information from VRUs.

Table 4.1: Information necessary about vehicles

Field	**Type**
Vehicle Unique ID	Numeric
Timestamp	Date
Location	WGS84 coordinates (latitude and longitude)
Altitude	Numeric, m
Heading	Numeric, rad
Speed	Numeric, $m.s^{-1}$
Acceleration	Numeric, $m.s^{-2}$
Yaw rate	Numeric, $rad.s^{-1}$
Dimensions	Numeric (length and width)
Class	Enumerate (bicycle, light or heavy vehicle)
Deceleration	Numeric, $m.s^{-2}$ (depends on class)
Driver Reaction	Numeric, s
Pedals state	Boolean List (one Boolean per pedal)
Lights state	Boolean List (one Boolean per light)
Drive Direction	Enumerate (stopped, driving forward or backwards)

Vehicles and VRUs can also be detected by cameras and Radars. In the case of a typical camera detection algorithm, the information described in Table 4.3 can be expected.

The VRU Safety Service warns about potentially dangerous situations. Table 4.4 contains the format of a message containing such a warning.

Table 4.2: Information necessary about VRUs

Field	**Type**
VRU Unique ID	Numeric
Timestamp	Date
Type	Enumerate (VRU, VRU cluster, vehicle[2])
Location	WGS84 coordinates (latitude and longitude)
Altitude	Numeric, m
Heading	Numeric, rad
Speed	Numeric, $m.s^{-1}$
Acceleration	Numeric, $m.s^{-2}$
Yaw rate	Numeric, $rad.s^{-1}$
Class	Enumerate (Child, Adult, Elderly Person, Disabled Person)
Drive Direction	Enumerate (stopped, moving forward or backwards)

Table 4.3: Information necessary about detections of camera, Radar and other smart city sensors

Field	**Type**
Sensor unique ID	Numeric
Timestamp	Date
Detected Person	Boolean
Location of the camera	WGS84 coordinates
Altitude	Numeric, m
Zoom level	Numeric, Percentage
Heading	Numeric, rad
List of detected objects	Label, confidence, bounding boxes, Location
- Label	Label of the detected object (Person, Vehicle)
- Confidence	Value between 0-1 with confidence of detection
- Bounding boxes	Top left coordinates, width and height

Table 4.4: Information necessary about warnings on potential dangerous situations

Field	**Type**
Event Unique ID	Numeric
Timestamp	Date
Location	WGS84 coordinates (latitude and longitude)
Type	Enumerate (*e.g.* as defined in Section 5.4.3)
Event generator ID	Numeric
Event generator type	Numeric (*e.g.* as defined in Section 5.4.3)
Altitude	Numeric, m
Unique ID of involved nodes	Numeric list

4.4 VRU SAFETY SERVICE

The VRU Safety Service is an ITS safety service responsible for detecting and warning about potentially dangerous situations in the road that might affect both of VRUs and vehicles.

The VRU Safety Service contains four main modules:

1. **Information processing and fusion**: Responsible for the perception phase.
2. **Potential collision detector**: Responsible for the decision phase.
3. **Notification service**: Responsible for the actuation phase.
4. **Metrics and statistics dashboard.**

These different modules are supported by a persistence database that can be also used by other services to obtain information about vehicles, VRUs and potentially dangerous situations between both.

The next sections, Section 4.4.1, Section 4.4.2, Section 4.4.3 and Section 4.4.4, will further detail each module.

4.4.1 Information Aggregation

The first module, the information processing and aggregator, is responsible for receiving, parsing, validating, and aggregating information from different data sources, but representing the same vehicle or VRU. This aggregated information is then made available to several ITS applications through persistence in a database.

Information published by the different data sources into the information aggregator can be used directly by services such as a Safety Service. An alternative is the use of coherent information union to obtain a much better perception of the situation. This process is called information aggregation, and it is described in detail in this section.

Algorithms 1, 2 and 3 present how the information processing from the different data sources is performed. The different data sources can be the vehicle equipment (OBUs), the VRU ITS-S application, and the infrastructure sensors (cameras and Radar).

Algorithm 1: Processing of the information from vehicles

```
Input: New message from information aggregator
Parameters: $Frequency_{VehicleMessage}$
Result: New/updated Vehicle Status
switch message source do
    case Vehicle do
        if vehicleMessage format == non standard format then
            convert to standard format;
            vehicleMessage.computeYawRate();
        else
            ▷ If turning (if blinking light is on)
            if vehicleMessage.exteriorLightStatus.turningSignals == ON then
                vehicleMessage.heading = vehicleMessage.headingOfTurning;
                re-run Algorithm 4;
        ▷ Add/Update Vehicle Information with message from vehicle
        oldVehicleInfo = database.get(vehicleMessage.stationID);
        if oldVehicleInfo does not exist or vehicleMessage.timestamp $\geqslant$
          oldVehicleInfo.timestamp then
            vehicleMessage.computeAcceleration();
            ▷ Driver delay reaction parameters, maximum deceleration
              parameters
            vehicleMessage.computeParameters();
            database.addOrReplace(key=vehicleMessage.stationID,
             value=vehicleMessage, Time To Live (TTL)=$Frequency_{VehicleMessage}$);
```

The information processing and fusion module is responsible for receiving vehicle messages from OBUs following the logic from Algorithm 1. Not all messages have the same basic information, such as yaw rate and acceleration. In addition, specific parameters, such as the driver delay and maximum deceleration, have to be inferred from the vehicle information. Therefore, in this step, the message is validated according to a particular standardized format (*e.g.* CAM, BSM), and the necessary parameters are computed.

Similar logic is done for the VRU Information, as described in Algorithm 2, with new information from a VRU triggering the potential collision detection algorithm.

Algorithm 2: Processing of the information from VRUs

```
Input: New message from information aggregator
Parameters: $Frequency_{VehicleMessage}$, $Frequency_{VRUMessage}$
switch message source do
    case VRU ITS-S (e.g. smartphone) do
        ▷ Consider VRU as vehicle if speed is compatible with vehicle speed
        if vruMessage.speed >= speed of a vehicle then
            oldVRUInfo = database.get(vruMessage.stationID);
            if oldVehicleInfo does not exist or vruMessage.timestamp $\geqslant$ oldVehicleInfo.timestamp then
                vruMessage.computeAcceleration();
                ▷ Delay, deceleration
                vruMessage.computeParameters();
                database.addOrReplace(key=vruMessage.stationID, value=vruMessage, Time To Live (TTL)=$Frequency_{VehicleMessage}$);
        ▷ Add/update VRU Information with message from VRU ITS-S
        oldVRUInfo = database.get(vruMessage.stationID);
        if oldVRUInfo does not exist or vruMessage.timestamp $\geqslant$ oldVRUInfo.timestamp then
            database.addOrReplace(key=vruMessage.stationID, value=vruMessage, Time To Live (TTL)=$Frequency_{VRUMessage}$);
            runDetectionAlgorithm(vruMessage, database.get(vehicles);
```

In both situations, new information is stored with a Time To Live (TTL), a time after which the information is automatically deleted from the database. The TTL allows avoiding an inconsistent database with outdated information about the vehicle or VRU. Outdated information can mean detecting false, potentially hazardous situations, increasing the number of false positives and the overall trustworthiness in the system. The chosen TTL is related to the frequency at which messages are produced, and thus, a new message that invalidates the previous message is expected to be received.

Information about vehicles or VRUs can come indirectly from Radars or cameras as well. For both data sources, basic processing is similar as in the previous algorithm, but with slight differences, as described in Algorithm 3.

New information from the Radar is fused with vehicle information from an OBU if it exists. It might happen in a situation where the Radar information is from the same vehicle as the information from OBUs [24]. In this situation, the module is responsible for joining the data from both data sources. The fusion of radar and OBU information means using the size of the vehicle obtained by the Radar and its class in the computation of certain parameters, such as

Algorithm 3: Processing of information from additional nodes

```
Input: New message from information aggregator
Parameters: $Frequency_{VehicleMessage}$ $Frequency_{VRUMessage}$
Result: New/Updated VRU Status or Vehicle Status
switch message source do
    case Radar do
        oldVehicleInfo = database.get(radarMessage.vehicleID);
        ▷ Radar detection corresponds to vehicle with OBU
        if oldVehicleInfo exists and radarMessage.timestamp $\geqslant$
          oldVehicleInfo.timestamp) then
            ▷ Use width, length, vehicle class Information, location from
              Radar
            vehicleInfo = fuseVehicleInformation(oldVehicleInfo,radarMesssage);
            database.addOrReplace(key=radarMessage.vehicleID, value=vehicleInfo,
              TTL=$Frequency_{VehicleMessage}$);
        ▷ Radar detection does not correspond to vehicle with OBU -
          create new Vehicle Information
        else
            vehicleInfo = createVehicleInfo(radarMessage);
            database.addOrReplace(key=radaMessage.vehicleID, value=vehicleInfo,
              TTL=$Frequency_{VehicleMessage}$);
    case Camera do
        for Detection in CameraMessage do
            ▷ Warn VRUs about close vehicles
            if Detection.Label == Vehicle then
                vrus = database.get(vrus);
                for VRUInfo in vrus do
                    distance = haversineDistance(VRUInfo.location, Camera.Location)
                    if distance < threshold then
                        warn(VRUInfo);
            ▷ Use camera to detect VRUs without ITS-S
            else
                vruInfo = createVRUInfo(detection);
                database.addOrReplace(key=camera.ID + ":" + detection.ID,
                  value=vruInfo, TTL=$Frequency_{VRUMessage}$);
                runDetectionAlgorithm(vruInfo, database.get(vehicles);
```

the maximum deceleration value of the vehicle (a parameter used in the collision detection module). The location provided by the Radar and the location provided by the vehicle OBU is fused using, for example, a Kalman Filter. An algorithm like the Kalman Filter is ideal for adjusting the weight given to the positional measurement from the Radar or the vehicle OBU based on the measurements errors (as described in Section 3.6).

New information from the camera needs to be further processed. The initial usage of the camera information would be the usage of the camera to detect VRUs and vehicles and warn if vehicles and VRU are close (without knowing the actual global position of each node). A more complex approach is based on converting the bounding boxes coordinates from a relative coordinate system, where coordinates are defined within the image frame, into a global coordinate system, so it is usable with information from other nodes and sources. A method to make an approximation of the coordinates is presented in Section 5.2.2.

This fusion of information from different sensors allows overcoming the shortcomings of each sensor as described in Section 3.5.

4.4.2 Potential Collision Detector

After the perception phase, where the information aggregation occurs, modules can be built to leverage the full potential of aggregated and fused information. One example of such a module is a potential collision detector.

The potential collision is responsible for deciding if a dangerous situation is likely to occur or not. To do so, it uses the information from vehicles and VRUs, and processes it in a set of stages. Figure 4.2 presents the overall algorithm.

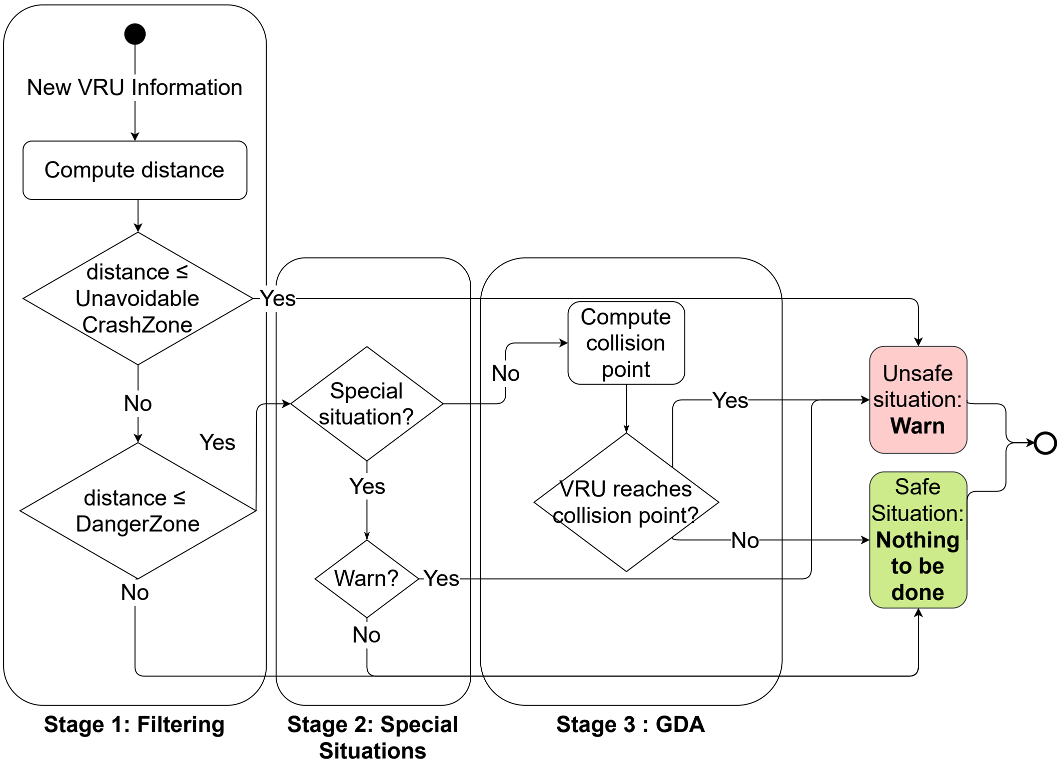


Figure 4.2: Overview of the Potential Collision Detector algorithm

The algorithm is triggered when a new VRU information is obtained, meaning that a VRU is present in the system (with new information from a vehicle being also a possible trigger).

The first step is the filtering stage, described by Algorithm 4. The algorithm computes the distance between the VRU and each vehicle available in the database, and computes the length of three zones: a zone where, if the vehicle and VRU are both inside, an accident is inevitable; a zone where an accident might happen or not if the vehicle and VRU are both inside; and a safe zone if none of those options occur (as described in Section 3.8.1). This step can be considered as a first "naive" approach to filter situations where no further computation is required: safe situations (*e.g.* situations where the VRU and vehicle are extremely far away) or unsafe situations, where collision is inevitable.

Since the zone lengths are defined from the vehicle perspective, a vehicle and VRU are both within the same zone if their distance is smaller or equal to the zone length.

In this step, the main challenge is the definition of the zone lengths, which have to be accurate enough, by using the most information about the vehicle and its driver (including the time of reaction of a person or driver), while keeping the computation simple enough for the algorithm to be fast. The algorithm must be as fast as possible to detect potential accidents as soon as possible.

The defined zone lengths have in mind the dynamics of the vehicle, such as acceleration, speed, position, and also its characteristics, defined by the vehicle's braking capability, $d_{deceleration}$, that will be greater the better the vehicle can brake. Driver characteristics, like age, visibility conditions, tiredness, that influence a $t_{reaction}$ [64], are also taken into consideration.

Algorithm 4: Potential Collision Detection - Stage 1 (Filtering)

```
Trigger: New VRU Information
Input: New VRU Information, Vehicles Information
Parameters: $t_{reaction}$ driver reaction delay
$d_{deceleration}$ maximum braking deceleration
$t_{guard}$ guard time, i.e. level of the conservatism of the detection
Result: Collision: true/false, Vehicle WarningMessage, VRU WarningMessage
for vehicleInfo in Vehicles Information do
    distance = haversineDistance(vruInfo.location, vehicleInfo.location) ;
    ▷ Speed of vehicle after driver reaction delay
    $v_{brake} = a_{vehicle} \times t_{reaction} + v_{vehicle}$ ;
    ▷ Travelled distance during driver reaction delay
    $d_{reaction} = \frac{1}{2} \times a_{vehicle} \times {t_{reaction}}^2 + v_{vehicle} \times t_{reaction}$ ;
    ▷ Safety Distances
    thresholdUnavoidableCrashZone = $d_{reaction} + \left(-\frac{{v_{brake}}^2}{2 \times d_{deceleration}}\right)$ ;
    thresholdDangerZone = thresholdUnavoidableCrashZone $+ v_{vehicle} * t_{guard}$ ;
    ▷ A - Imminent, unavoidable crash - notify immediately
    if distance $\leq$ thresholdUnavoidableCrashZone then
        return True, WarningMessage(collision Risk, human presence on the Road),
         WarningMessage(collision Risk, human presence on the Road);
    ▷ B - Danger/risk zone - further assessment is needed
    else if distance $<$ thresholdDangerZone then
        continue to stage 2;
    ▷ C - Safe zone - no further assessment is needed
    else
        return False, None, None;
```

The choice of the correct distance formula is a challenge. Several distances, including Euclidean, Manhattan, and Haversine, are possible. Section 5.4.1 further details which distance formula is considered and the rationale for the choice.

The second step is the phase of detection of the exceptional situations. To decrease the miss rate and the false positive rate of the algorithm, certain exceptional situations and scenarios where an accident is more likely, less likely, or certain can be considered, as described in Algorithm 5.

Algorithm 5: Potential Collision Detection - Stage 2 (Special Situations)

```
Input: VRU Information, Vehicle Information
Parameters: Speed limit, Altitude threshold, Heading threshold
Result: Collision: true/false, Vehicle WarningMessage, VRU WarningMessage
▷ B. Notify both VRU and vehicle
▷ Vehicle going too fast
if $vehicleInfo.speed > speed\ limit$ then
    return True, WarningMessage(collision Risk, vehicle too fast),
      WarningMessage(collision Risk, vehicle too fast);
▷ Weight of VRU < weight_child (children, etc.), which can be 40
if $vruInfo.weight < weight\_child$ then
    return True, WarningMessage(collision Risk, children presence on the road),
      WarningMessage(collision Risk, children presence on the road);
▷ C. Don't notify driver
▷ If driver is already braking
if $vehicleInfo.accelerationControl.brakePedal == ENGAGED$ then
    return True, None, WarningMessage(collision Risk, vehicle is braking);
▷ D. No notification for both VRU and vehicle
▷ Same heading:  VRU parallel to vehicle
if $difference(vehicleInfo.heading, vruInfo.heading) < heading\ threshold$ then
    return False, None, None;
▷ Different altitudes
if $difference(vehicle.altitude, vru.altitude) < altitude\ threshold$ then
    return False, None, None;
▷ E. Else, continue to stage 3
continue to stage 3;
```

In this stage, certain situations that can be extremely dangerous - if a vehicle is going too fast or if the VRU is a child - are immediately considered as potential accidents. By opposition, some situations are ignored to avoid false positives: situations where the driver is already braking, or where there is a significant difference in altitude (*e.g.* a VRU on top of a bridge and vehicle is under the bridge) or heading (*i.e.* VRU and the vehicle cannot collide because they are moving in opposite directions).

If no situation from stage 2 is verifiable, stage 3 is executed, as described in Algorithm 6. In this stage, the goal is to predict the future trajectory of the vehicle and VRU based on the current dynamic. By predicting the trajectory, it is possible to analyze if it is likely that the vehicle and VRU are going to crash in the near future. The prediction follows a kinematic approach, as described in Section 3.8.2.

```
Algorithm 6: Potential Collision Detection - Stage 3 (Compute GDA)
  Input: $v, W, L, \omega$: Velocity, width, length and yaw rate of vehicle
  $x, y$: VRU position in the vehicle coordinate system
  Parameters: $v_{pMax}$: Maximum velocity of a VRU
  Result: Collision: true/false, Vehicle WarningMessage, VRU WarningMessage
  ▷ Bicycle Kinematics Model (with 2 tires)
  ▷ Radius of gravity center
  $r_S = {}^{v}/_{\omega}$;
  ▷ Ackerman angle
  $\epsilon_r = {}^{L}/_{r_S}$;
  ▷ Instantaneous Center of Rotation
  $x_S = 0$ ;
  $y_S = r_S \times \cos({}^{\epsilon_r}/_{2})$ ;
  ▷ Angle between vehicle and VRU
  $\theta = atan2(\frac{y-y_S}{r_S}, \frac{x-x_S}{r_S})$ ;
  ▷ Time to collision
  $t_{collision} = \frac{\theta-({}^{\epsilon_r}/_{2})+({}^{\pi}/_{2})}{\omega}$ ;
  ▷ Location of point of collision
  $x_{collision} = r_S \times \cos(\theta + \frac{\epsilon_r}{2} - \frac{\pi}{2}) + x_S$ ;
  $y_{collision} = r_S \times \sin(\theta + \frac{\epsilon_r}{2} - \frac{\pi}{2}) + y_S$ ;
  ▷ Distance between VRU and potential collision
  $d = \sqrt{(x - x_{collision})^2 + (y - y_{collision})^2}$ ;
  ▷ Potential collision detected
  if $d < v_{pMax} \times t_{collision}$ then
      return True, WarningMessage(collision Risk, human presence on the road),
        WarningMessage(collision Risk, human presence on the road);
  return False, None, None;
```

4.4.3 Notification Service

If a collision is detected in the decision phase by the potential collision detector, a third module is responsible for the actuation phase - the notification service. If a dangerous situation is detected in this module, the relevant nodes are notified using a predetermined message format (*e.g.* ETSI DENM).

Since the architecture is based on brokers as information aggregators, this service is responsible for publishing the message into determined topics.

4.4.4 Metrics and statistics dashboard

To observe and monitor the current situation of the system and detect any trends and patterns in the detected accidents, a metrics and statistics dashboard is useful. Such dashboard allows the system to fulfill the monitoring requirements (RP_4, All potential accidents should be logged, described in Table 2.5).

This dashboard is responsible for processing any logging from the VRU Safety Service and present in a graphical way for managers of the system. The dashboard should be flexible, *i.e.* allow future additions and configurations in an easy-to-use way by the managers.

4.4.5 Persistence layer

Supporting all these modules is the persistence layer, *i.e.* a database (with 1 or more instances) responsible for storing and retrieving the information used by the VRU Safety Service.

It is crucial that the execution time is reduced as much as possible. The persistence layer must, therefore, be as fast as possible, rendering impossible solutions with a centralized database outside the local machine where the VRU Safety Service is deployed, since the query times would be prohibitively high.

In addition, the database must have enough flexibility to accommodate new message formats or revisions of message standards, where fields can appear, disappear, change the name, or type. The need for flexibility thwarts the usage of relational databases such as Microsoft SQL Server, MySQL, and PostgreSQL. The usage of other non-relational databases, such as document-oriented databases (*e.g.* MongoDB, Redis, CrateDB), is therefore more recommended.

4.5 SUMMARY

This chapter presented the overall architecture of the VRU safey approach.

Vehicles and VRUs can be part of the overall architecture, either by using OBUs, ITS-Ss such as smartphones, or by being detected by infrastructure elements, such as Radars, cameras, and Lidars.

The architecture follows an event-driven architecture, where each message from vehicles and VRUs is an event published into an information aggregator/broker. Applications, such as the VRU Safety Service, can subscribe and publish into this broker in order to obtain information about vehicles and VRUs, and warn them about potentially dangerous situations.

The VRU Safety Service is responsible for detecting and warning about potentially dangerous situations. The potential collision detector module of the service follows a three stage algorithm that employs kinematic equations, and a set of information from vehicle's and VRUs' messages, to compute if an accident is likely or not to occur, and warn them if it is the case.

The next chapter will provide details on the actual deployment of this architecture.

CHAPTER 5

Implementation

We shall never surrender.
— Winston Churchill

Chapter 4 defined the architecture to address the safety problem for VRUs on roads. This chapter presents how this architecture is deployed, while pointing to relevant aspects of the implementation.

First, Section 5.1 provides an overview of the deployment of the system within the context of the ATCLL project, including the main implementation and technological decisions and their rationale. Within this deployment, Section 5.2 provides details on how each information source is used. Afterwards, Section 5.3 introduces details on the VRU side ITS-S application, while Section 5.4 provides details on the Safety Service, responsible for detecting potentially dangerous situations, such as the rationale for the hyper-parameters choice. Finally, Section 5.5 summarizes the most important implementation decisions.

5.1 Deployment overview

Figure 5.1 shows the system architecture as it was deployed in the ATCLL infrastructure. The elements that were developed from scratch or partially adapted during the development are shown in gray.

Three main blocks can be discerned in the deployment of the system: the VRU ITS-S system, an Android-based smartphone with an application; the edge infrastructure, through which information from vehicles, cameras, and Radars are obtained; and the central ATCLL infrastructure, where the two main modules of the solution architecture, the Information Aggregator System - now implemented in a set of 2 brokers, the FIWARE Orion Context Broker and an internal MQTT Bridge Broker - and VRU Safety Service are deployed.

The architecture follows an event-driven design, with an hybrid approach (edge and cloud) to the processing and gathering of information:

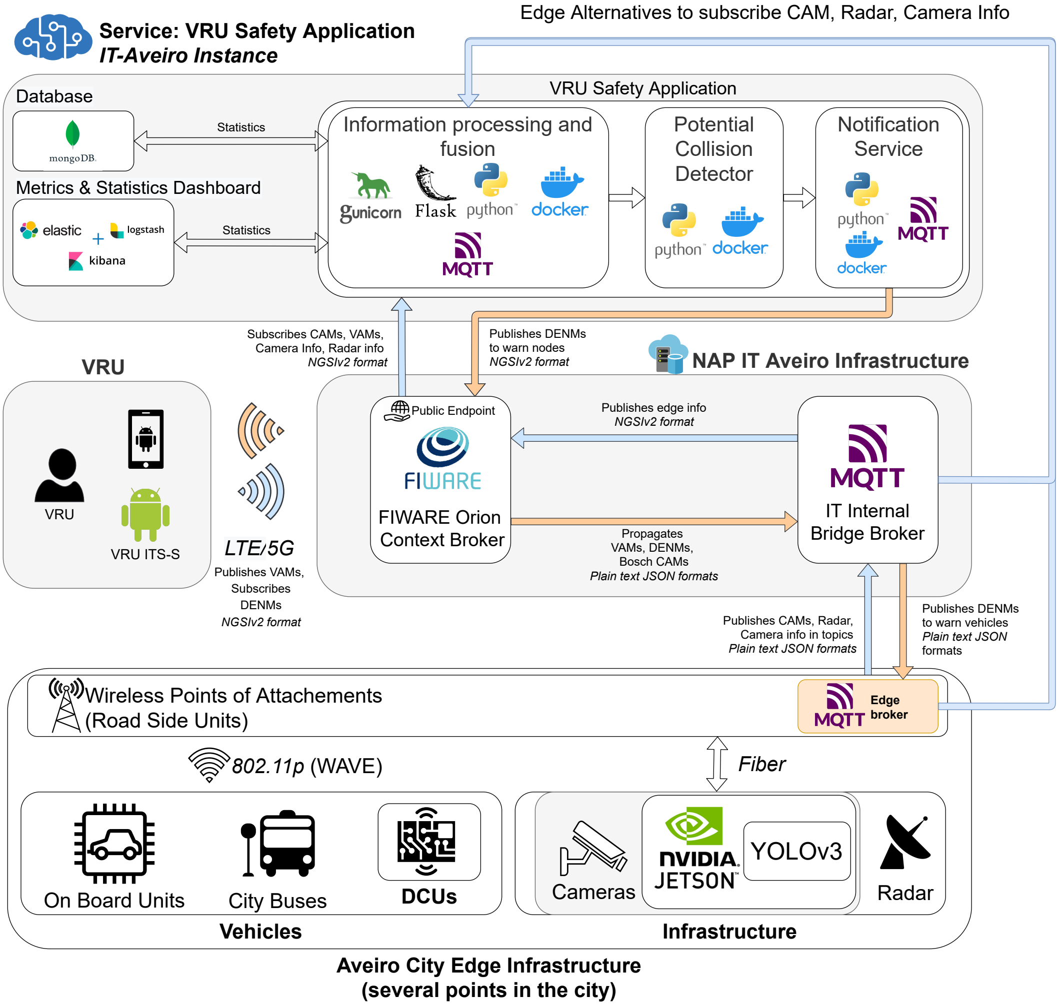


Figure 5.1: Detailed Deployment Overview

- **Edge**: Responsible for the gathering and processing of information from vehicles, cameras and Radars. Edge information can be directly used through the subscription of their brokers. While no instance of the VRU Safety Application was deployed in the edge, the service is prepared to subscribe to information from the edge (more details on Section 6.2.2).
- **Cloud**: Responsible for gathering and processing information from all edge nodes and VRUs ITS-S. The VRU Safety Application and its auxiliary components were deployed in the cloud infrastructure and subscribes both the edge and cloud brokers.

The rationale of the choice of the technology associated with each block is summarized in Table 5.1, which provides an overview of the developed parts of the architecture, the technological decisions, and their rationale.

The ATCLL has two information aggregator/brokers: the Fiware Orion Context Broker, and a public broker supported by an internal Eclipse Mosquitto MQTT broker. Both brokers follow an architectural pattern of publisher-subscriber, assuring the transportation of data from any source/producer of information to any consumer, and allowing the creation of

Table 5.1: Main technological decisions

Element	**Technology**	**Rationale for choice**
Information aggregator	Fiware Orion Context Broker	Broker appropriated for managing the entire lifecycle of context information including updates, queries, registrations and subscriptions[1].
VRU ITS-S	Smartphone	Ubiquitous, easy to be used by VRUs, highly programmable and contains a myriad of sensors.
- Application	Android Application	Allows access to low-level APIs of Android-OS based smartphones.
- Sensors for location	Google Fused Location Provider API	Allows a battery-friendly usage of several sensors (Wi-Fi, Bluetooth, GPS) to maximize the precision of the location.
- Additional sensors and APIs	Accelerometer, gyroscope, compass, Google Activity Detection API	Provides additional information about the VRU (heading, if it is still or walking).
Safety Service	Dockerized Web Server Gateway Interface (WSGI) Python application	Usage of Python to leverage the Machine Learning libraries, of WSGI to improve performance while replying to several requests from several VRUs and vehicles, Docker to allow easy replication of the application in other contexts.
- Database	MongoDB document-based database	Allows flexible structure (non-relational) while keeping high performance and a JSON-like schema.
Statistics Dashboard	Elasticsearch-Logstash-Kibana (ELK) Stack	Allows monitoring, logging, and advanced statistical analysis with a plug-and-play solution based on logging from an application.
Messages formats	ETSI formats	Allows interoperability with other systems. Used European standards since this system has a European deployment.

entities and subscriptions. However, the Fiware Orion Context Broker is more sophisticated by logically separating data by different services, having each service its different types of data, or entities, each with its attributes and subscriptions. In addition, the Fiware Orion enforces a standardized data format, the Next Generation Service Interfaces version 2 (NGSIv2), unlike the Eclipse Mosquitto MQTT, where formats are not enforced.

During the development of this system, and from the VRU side, the processing, parsing, and conversion of information from VRU, camera, and VRU Safety Application was added to both information aggregators[2]. This allowed other services, directly dependent on the internal MQTT browser, to fully utilize all the information produced by the VRU safety application. However, the VRU Safety Application was designed for only communicating directly with the Fiware Orion Context broker. Such decision was made to ensure the decoupling from existing infrastructure by making the safety application only depend on a public endpoint of the ATCLL infrastructure[3].

Both information aggregators are also populated with information from the Aveiro VANET: vehicles and Aveiro City Buses equipped with OBUs and other city infrastructure, namely Radars and cameras, throughout the city of Aveiro. This set of information is aggregated and

---

[2] The radar information and processing was also added but it was not done by the author of this dissertation [24].

[3] For evaluation purposes (Section 6.2.2), the application is also capable of subscribing to the internal MQTT broker, although that is the preferred deployment solution.

sent to the central information aggregators in the ATCLL infrastructure from RSUs scattered through the city of Aveiro. OBUs are connected through ITS-G5 to the RSUs (and cellular to a base station), while cameras and Radars are directly connected through Ethernet, and the RSUs themselves are connected to the central infrastructure through fiber.

The only node that does not use the RSUs to communicate with the infrastructure is the VRU ITS-S, an Android-based smartphone. It publishes and subscribes directly to the Fiware Orion Context Broker, since it is the public information aggregator, through Wi-Fi or cellular technologies (such as LTE or, more recently, 5G). The choice of the smartphone is motivated by its ubiquity (since most VRUs will already possess one), its high programmability, and its reduced cost compared to other solutions. However, alternatives, such as a custom-designed Application-Specific Integrated Circuit (ASIC) could be more powerful and cost-effective, while keeping the same set of functionalities as a smartphone.

The usage of European ETSI standards for message formats - for Vehicles ETSI CAMs, for VRUs ITS-Ss applications ETSI VAMs, for Safety Services notifications ETSI DENMs and for cameras and Radars messages with predefined NGSIv2 format - allow an easy interoperability with other infrastructures, for example from Bosch. Figure 5.2 details this cooperation with Bosch side.

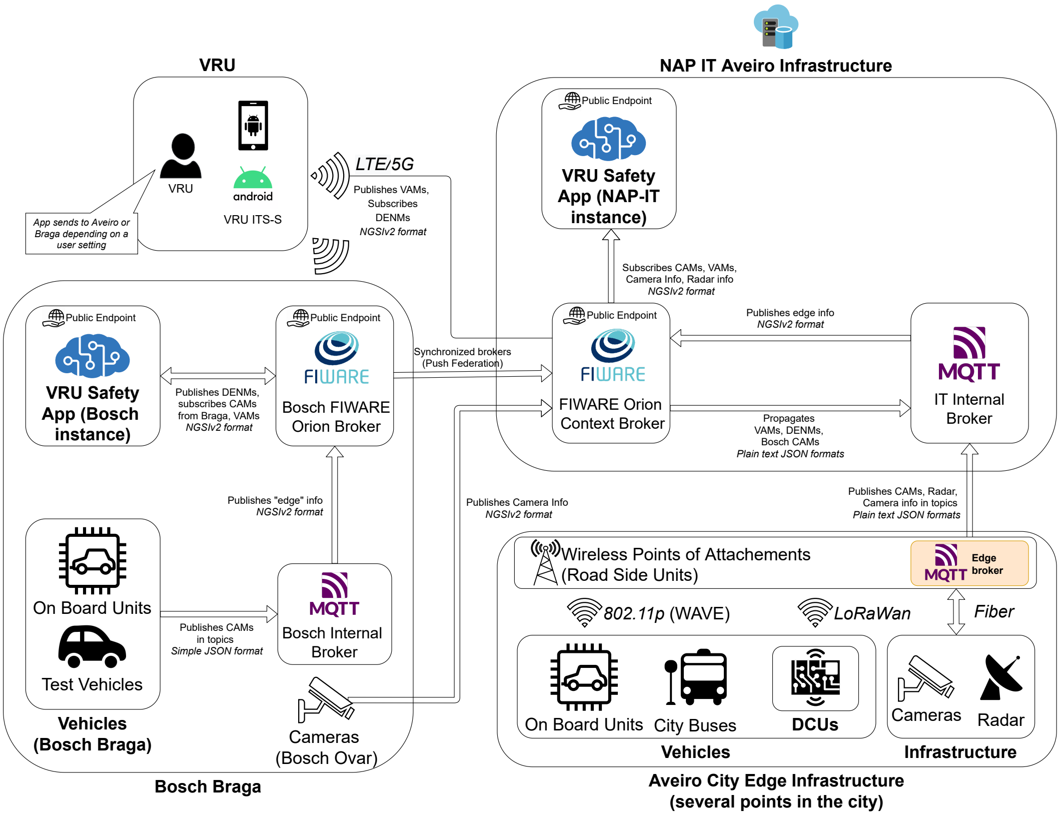


Figure 5.2: Detailed Deployment Overview - Interaction with Bosch

The Bosch infrastructure, or any other vendor's infrastructure, can reuse part of the full system deployment by communicating with the public information aggregator, the Fiware Orion Context broker. In the example of Bosch, by being a manufacturer of OBUs - denominated

CCUs by the company - it is also capable of sending information from vehicles. In this situation, additional processing is required and developed by using the ETSI CAM specification, described by the Abstract Syntax Notation One interface description language (ASN.1), to decode the received message from Bosch CCUs. The decoded CAMs are then published in the public ATCLL endpoint, the Fiware Orion broker. Once again, to use the information, it is only needed to subscribe to the appropriate service.

The following sections further discuss details of implementation for the several components of the deployment architecture.

5.2 Usage of existing information and infrastructure

Most of the data sources described in Section 4.2 either use the existing hardware or software infrastructure already existing from the ATCLL project. However, in all cases, some modifications were needed to support the information aggregation and sensor aggregator, which are the basis for the safety service: messages from vehicles OBUs, Radars, and camera detections were converted from the original format to an unique standard format, allowing the Safety Service to use an unique format for vehicle and VRU information.

It should be noted that all the different data sources - vehicles messages from their OBUs (CAMs), radar and camera detections, and VRUs messages from their ITS-S (VAMs) or camera detections - were used and deployed in the system.

5.2.1 Vehicle information

The connected vehicles propagate information, coded in ETSI CAM format, into the information aggregator/Fiware Orion broker. In addition, Bosch CCUs also publish CAMs into the broker.

To use the information, it is only needed to subscribe to the appropriate service in the Fiware Orion to obtain CAMs. Appendix A.1 presents an example of such a message.

5.2.2 Video Cameras

The video cameras are integrated within Smart LampPosts connected by fiber and 5G to the central ATCLL infrastructure. These video cameras are connected to a set of NVIDIA Jetson Nano, responsible for processing the video frames and running a Deep Machine Learning algorithm. This Deep Learning algorithm, a pre-trained YOLOv3 model, is optimized to the hardware using NVIDIA TensorRT, and is capable of detecting pedestrians and vehicles. The detection information is then propagated into the ATCLL internal broker, and then propagated and published into the Fiware Orion public broker.

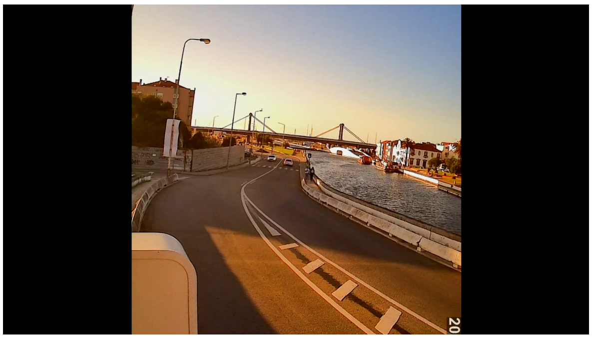
The image detection information is sent using a pre-defined NGSIv2 message for camera detection Information. Appendix A.3 presents an example of such a message, that includes the coordinates from the bounding boxes of each detected object. This information can be used to compute the location of the object - vehicle, VRU - in global coordinates through a conversion process.

Considering a frame obtained by a camera, as represented in Figure 5.3a, the coordinates of the bounding box representing an object are within a local coordinate system, *i.e.* the camera frame. These relative coordinates can be converted into a global coordinate system, *i.e.* WGS84 coordinates through a set of transformational and rotation matrices, as described by several authors such as [71]- [72]. Intrinsic camera parameters are required to define the set of matrices, such as the focal length and the correlation between distance in the frame and the actual distance. While the focal length is usually part of the specification of the cameras, the correlation is not and requires additional fine-tuning.

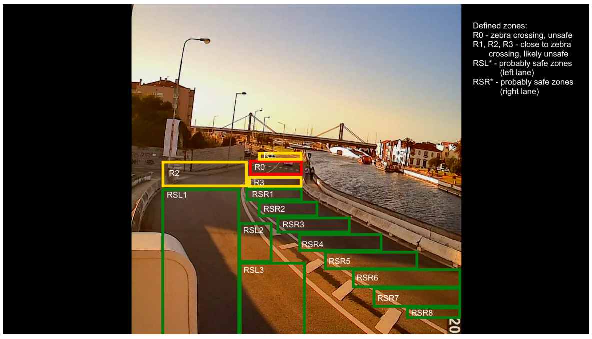
Since the fine-tuning needs to be redone for each position the camera can be pointed at, and both cameras from ATCLL and Bosch Ovar are capable of movement, adequate accuracy would be significantly affected. Therefore, a more straightforward solution was considered. First, a frame is divided into a set of squares - that are effectively Region of Interests (ROIs) - each one with a known fixed global position. An object with a determined bounding box will have a global position corresponding to the square that is within. Since a bounding box of an object may overlap more than 1 square, the square to be considered is the one with which the Intersection over Union (IoU) with the bounding box is larger.

Figure 5.3b presents an example of such a division into ROIs. Having in mind the topology of the roads, 3 types of zones can be distinguished: far away from the zebra crossing (green potentially safe zones), at the zebra crossing (unsafe red zones), and close to the zebra crossing (yellow, potentially unsafe zone). Based on this division, detected objects (vehicle C1 to C4 and person P1) were then associated with the ROIs, which overlaps the most with (with the highest IoU), as presented in Figure 5.3c.

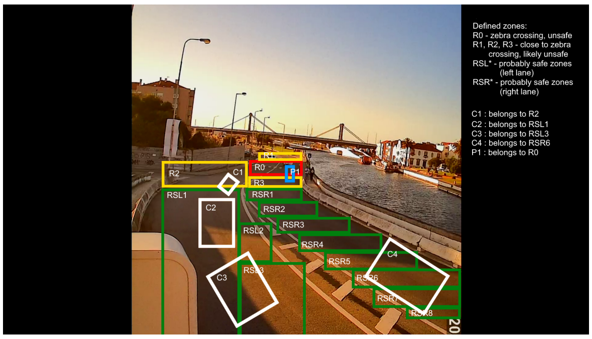
With this algorithm, an approximation of the location of objects could be obtained by defining the object's location with the coordinates of the center of the associated bounding box. This information was then used to detect vehicles and VRUs that could not be detected by any other source. While the camera, with this algorithm, can only provide an approximated position and not much more information about VRU or vehicle (such as speed and heading), it can be used as a fall-through sensor, to detect with more error potential dangerous situations.



(a) Original frame as obtained by a camera



(b) Example of manual division of the original frame by a set of squares/ROIs



(c) Association of bounding boxes with squares/ROIs

Figure 5.3: Camera bounding box processing algorithm

5.2.3 Radar

The Radars are capable of tracking several objects in several lanes using the Doppler effect and returning an object list with the following parameters [73]: object location, velocity, heading angle, length of the object, and a random unique ID.

Similar to cameras, the object location is given in relative coordinates, and a similar conversion process needs to happen. The work by [24] developed and integrated this conversion within the ATCLL by applying a set of mathematical operations to the original coordinates - rotating the coordinate system and applying a transformational matrix. This set of operations results in the existence of messages as described in appendix A.4.

5.3 VULNERABLE ROAD USER ITS-S APPLICATION

5.3.1 Localization issue

The first main task of the VRU ITS-S smartphone-based application is to obtain information from pertinent sensors to create an accurate awareness of the VRU. From the possible set of information, the most important is the VRU location.

Section 3.5 discussed the different sensors, while Section 3.6 discussed how joining information from different sensors improves both accuracy and resilience against the failure of a single sensor. To obtain the location of the VRU, the approach is, therefore, the joining of the data from the GNSS sensor with data from an accelerometer, gyroscope, magnetometer, and orientation sensors using a Kalman Filter. The errors from each sensor were given by the Google Sensors API[4].

An alternative that was deployed in the final version of the system consisted in using the Fused Sensor Provider API from Google[5]. This API fuses information, not only from the accelerometer, gyroscope, magnetometer, and orientation sensors and the GNSS sensor, but also Wi-Fi, Bluetooth, and cellular signal strength. At Google I/O 2019[6], the implementation of the API was given further details, with its inner workings being similar to a Kalman filter by having two phases, a prediction phase and an update phase. This API uses models to obtain coordinates based on the sensors' readings in the predict phase. Then, the best measurements will be given more weight in the update phase, where all predictions are fused, similar to a typical Kalman filter.

While the inner workings are similar to the original developed Kalman filter, the fine-tuning done by Google allowed better results. A showcase at Google I/O 2019 illustrates that the API keeps the accuracy of the location and speed even within a building - something unlikely or even impossible with the sole usage of a GNSS GPS sensor, accelerometer gyroscope and magnetometer.

It should be noted that, if the VRU ITS-S application is not available, the information from the camera detections can be used to detect, albeit with less accuracy, the VRU.

5.3.2 Generation of P2X messages

After obtaining information from the sensors, messages to communicate the state of the VRU had to be generated periodically and sent to the central information aggregator.

At the time of the beginning of the development of this dissertation, and until November 2020, no stabilized standard existed on VAMs (example described in appendix A.2). Since the standard was in draft status [74], it was decided to use the stabilized Bosch internal format, similar to ETSI standard in terms of information and fields formats it contains.

The high dynamism of potentially dangerous situations is a significant challenge since, for example, a pedestrian can easily and quickly walk forward or backward in the space of a second, and create or diffuse a potentially dangerous situation. The developed system has this in mind by triggering the analysis of potentially dangerous situations very frequently. Since the system collision detection is triggered every time a VAM is sent, the maximum ETSI defined frequency, 10 Hz, is chosen for the frequency at which VAMs are sent. This decision means that the situation around the VRU is re-evaluated every 100 ms.

Analyzing the situation too quickly or too slowly could lead either to false warnings/false positives or to missed warnings/false negatives - both not recommended in safety applications. More importantly, false negatives can have catastrophic consequences for the lives of VRUs

---

[4] By defining the *onAccuracyChanged* callback as described in `hbtps://developer.android.com/guide/topics/sensors/sensors_position`.

[5] Available at `hbtps://developers.google.com/location-context/fused-location-provider`

[6] Available at `hbtps://www.youtube.com/watch?v=MEjFW_tLrFQ`

and occupants of vehicles. False positives are also problematic since they will induce the system users to ignore the warnings or, even worse, disable the warnings and the system.

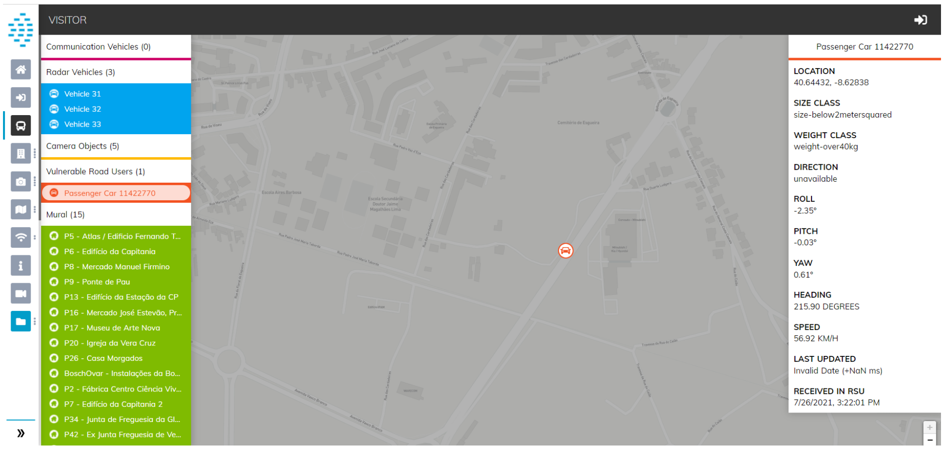
5.3.3 Usage as a low-cost OBU

The developed application, associated with the sensors, makes the smartphone a VRU ITS-S capable of similar basic functionalities of an OBU. By obtaining and processing the necessary information to generate periodic messages to make the vehicular network aware of the vehicle and its essential information - such as location - the VRU ITS-S acts as an OBU.

A smartphone cannot fully comply with all the VANET standards - *e.g.* there is only experimental support for ITS-G5 smartphones (as described by [75]), although connecting a ITS-G5 module is also possible as described by [76]. However, it can use cellular access technologies to send information about the vehicle to a point of the VANET infrastructure.

The smartphone - or any equivalent device - can therefore act as a low-cost OBU. The developed VRU ITS-S predicts this situation by changing the station type of the produced VAM message to a vehicle if it detects that the speed of the smartphone is not compatible with a VRU walking (*i.e.* if greater than a threshold defined in Section 5.4.2). This information can then be used to consider VAMs with such a station type as a vehicle and not a VRU, as considered in Algorithm 2.

The ATCLL real-time dashboard is, apart from the Safety Application, another example of a service that has in consideration such information, as it can be seen in Figure 5.4.



Passenger Car 11422770

LOCATION
40.64432, -8.62838

SIZE CLASS
size-below2metersquared

WEIGHT CLASS
weight-over40kg

DIRECTION
unavailable

ROLL
-2.35°

PITCH
-0.03°

YAW
0.61°

HEADING
215.90 DEGREES

SPEED
56.92 KM/H

LAST UPDATED
Invalid Date (+NaN ms)

RECEIVED IN RSU
7/26/2021, 3:22:01 PM

Figure 5.4: Example of a vehicle detected by the VRU ITS-S (smartphone) as seen in the ATCLL real time dashboard

5.4 SAFETY SERVICE

After obtaining all the required information, a Safety Service can be built to predict potential collisions following the algorithms described in Section 4.4.

The Fiware Orion Context Broker is the public endpoint within the ATCLL infrastructure and, therefore, the best choice to obtain the information required by the Safety Service, while decreasing decoupling from a private infrastructure. This broker notifies subscribers of information through HTTP requests that embed the information within the request body. Therefore, the Safety Service is built as an HTTP application that can process those requests[7]. This application is then responsible for communicating with the collision detector and notification service modules.

The following sections present some details on the essential aspects of those modules.

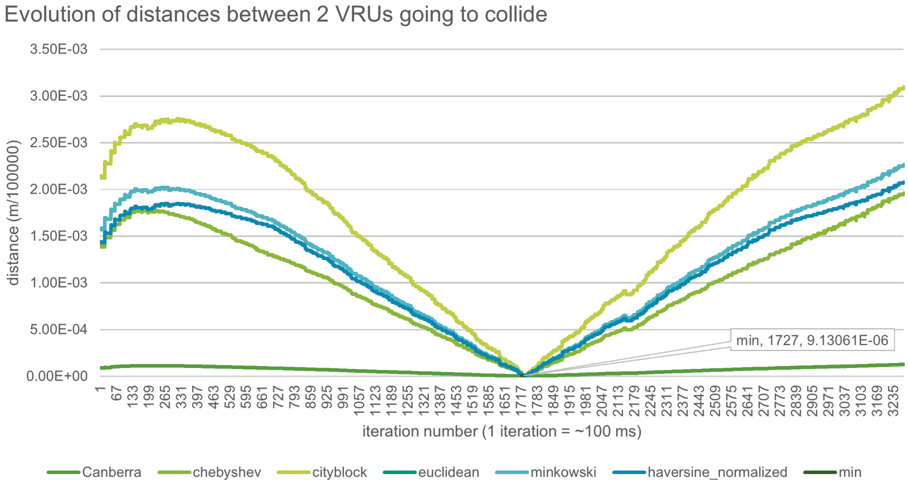
5.4.1 Distances

A critical component of the algorithms described in Section 4.4 to determine the likelihood of a potential collision is the computation of the distance between the vehicle and VRU.

To compute the distance between elements, several alternatives can be considered - Manhattan (L^1 or Cityblock distance), Euclidean (L^2 distance), Canberra, Chebyshev, and Haversine. The Haversine formula is the most precise one for representing a distance between two coordinates on Earth, since it represents the angular distance between two points on the surface of a sphere [77].

An appropriate distance formula for this safety application must accurately estimate the distance between nodes, or, if not possible, can underestimate the value of distance - *i.e.* given two nodes, return a distance smaller than the real - but cannot overestimate - *i.e.* given two nodes, return a distance greater than the real. Underestimations can increase the number of false positives, while overestimations increase the number of false negatives, and therefore increase the danger for VRUs and vehicles. Thus, understanding which formula could be used is essential.

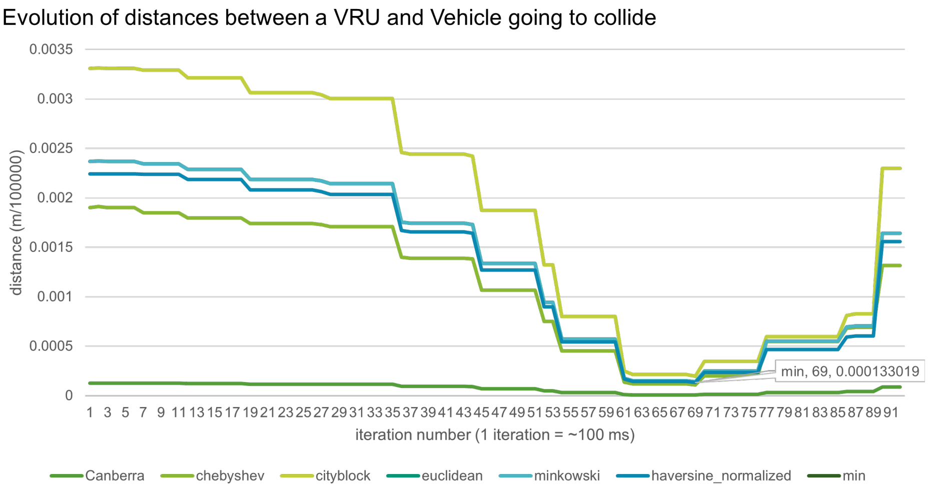
To that effect, a study to understand the impact of several distances was executed and can be visualized in Figure 5.5. This study observed the tendency of all distances, with the Haversine distance being the most precise formula to compute distance within Earth by definition.



(a) Results for a test between 2 VRUs

---

[7] However, it is also capable of subscribing a MQTT broker to support the study of Section 6.2.2.



(b) Results for a test between 1 VRUs and a vehicle

Figure 5.5: Analysis of several distance formulas

The results show that Canberra and Chebyshev distances could be used, since they are more pessimistic than the Haversine distance. However, both, particularly the Canberra distance, would be too pessimistic, significantly increasing the number of false positives and, therefore, the system's trustworthiness.

By opposition, the results also allow to eliminate the Manhattan and Euclidean distances for being too optimistic (*i.e.* estimate the nodes as further away than they are).

5.4.2 Hyper-parameters

The algorithms defined in Section 4.4 use a set of parameters that need to be defined. The current implementation considers a set of static parameters, with values described in Table 5.2.

Each parameter has a default value that is used by the system if the information about the VRU or vehicle is available. Depending on the sensor that detected the vehicle or VRU - *e.g.* OBU, Radar, camera - not all parameters are necessary.

With future iterations of this system and more labeled data of potential accidents, the parameters can be fine-tuned using Machine Learning techniques.

5.4.3 Notification service and extension of the DENM standard

If a potential collision is detected, vehicle, VRU and infrastructure are warned of such incident through an ETSI DENM. The notification service receives the information about the potential collision, creates the DENM and publishes it to the appropriated nodes, which are used by the nodes' applications to receive notifications about events.

Figure 5.6a presents an example of visual notification of a DENM arriving at a vehicle OBU[8], while Figure 5.6b presents a notification as seen by the VRU. In the case of the VRU, a sound notification saying *"Warning! Please verify your surroundings"* was also implemented

---

[8] The author did not develop the UI of this application.

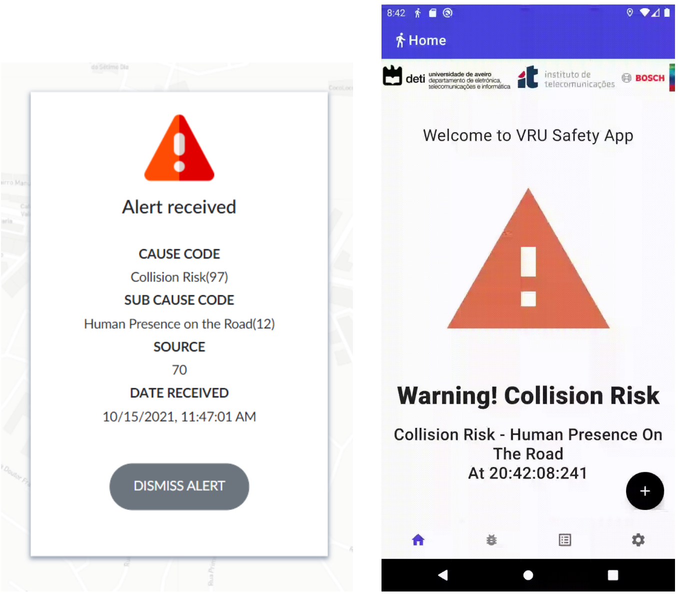
Table 5.2: Considered hyper-parameters for the Safety Service

Parameter	**Value**	**Rationale**	**Used in**
Frequency CAM	1 Hz	Already implemented as such and within ETSI constraints.	TTL in database
Frequency VAM	10 Hz	Fastest possible frequency within ETSI constraints.	TTL in database
Vehicle minimum speed	10 $m.s^{-1}$	The maximum speed of a bicycle (VRU profile 2). Follows ETSI TR 103 300-1.	Threshold to consider a VAM as coming from a vehicle
$d_{deceleration}$, maximum deceleration when braking	4.00 $m.s^{-2}$ (heavy vehicles) 8.00 $m.s^{-2}$ (light vehicles)	Deceleration of vehicle with Anti-lock Braking System (ABS)[a]when the speed is 40$km.h^{-1}$.	Computing the length of unavoidable crash zone
t_{guard}, guard time/level of the conservatism	1 s	Follows previous work [64].	Computing the length of danger zone
Driver delay time	1.3 s (at night) 1.0 s (at daylight)	Follows previous work [78].	Computing the speed and travelled distance during driver reaction delay
Vehicle speed limit	30 $m.s^{-1}$ = 110 $km.h^{-1}$	Start of illegal speed outside highways in Portugal [b]	Threshold to decide to notify that vehicle is going too fast
Altitude threshold	1 m	4 m = Maximum altitude allowed by European Union and national law; 1 m = 1 / 4 of the said altitude	Avoid false positives related to noise in altitude and heading
Heading threshold	+/- 5 degrees	Filter small variations of the heading.	Same as previously
Dimensions of	(Length, Height, Width)		Compute GDA to determine a potential collision
- Light vehicle	4.966 m, 1.55 m, 1.895 m	Considered top values for Large Cars, (conservative consideration) [79].	
- Heavy vehicle	16.5 m, 4 m, 2.6 m	Considered UE Weights and dimensions' Directive [80].	
- Bicycle	2m, 1m, 0.50 m	Considered UE Weights and dimensions' Directive [80].	

[a] Conservative speed consideration, considering ABS is in most cars). Considered values by [81], [82].

[b] From Portuguese law, 90 $km.h^{-1}$ is maximum speed outside highways and 20 $km.h^{-1}$ over the limit (110 $km.h^{-1}$) is the start of a serious crime.

to make it possible for blind people to be able to benefit from the warnings (and fulfilling the UI requirements described in Table 2.4).



Alert received

CAUSE CODE

Collision Risk(97)

SUB CAUSE CODE

Human Presence on the Road(12)

SOURCE

70

DATE RECEIVED

10/15/2021, 11:47:01 AM

Welcome to VRU Safety App

Warning! Collision Risk

Collision Risk - Human Presence On The Road
At 20:42:08:241

(a) Notification received by a vehicle OBU application **(b)** Notification received by the VRU

Figure 5.6: Example of visual notifications of a potential collision for vehicle and VRU

If the vehicle that is part of the potential collision has been detected through an OBU, the notification, however, is more complex. Vehicles communicating with an OBU with ITS-G5 are not always available, with their availability being dependent upon its connection to a RSU. Moreover, the DENM cannot simply be sent to the original RSU; in the meantime a handover - *i.e.* a change in the RSU or RSUs to which the OBU is connected to - might have happened. The system currently considers this scenario unlikely since a situation is re-evaluated at the VAM frequency (10 Hz). During 100 ms, it is unlikely to happen a handover with loss of connection to the original RSU. However, to ensure the warning always reaches the OBU, an integration with a mobility manager would be required.

The generated warning message, an ETSI DENM, has a set of fields for defining the type and sub-type of event, with the sub-type allowing more information on the incident being reported. While some existing codes can be reused - namely *dangerousSituation (99)* as type and *hazardousLocationAnimalOnTheRoad (11), humanPresenceOnTheRoad (12) or collisionRisk (97) as subtype* - the standard required an extension of the possible cause codes for certain situations:

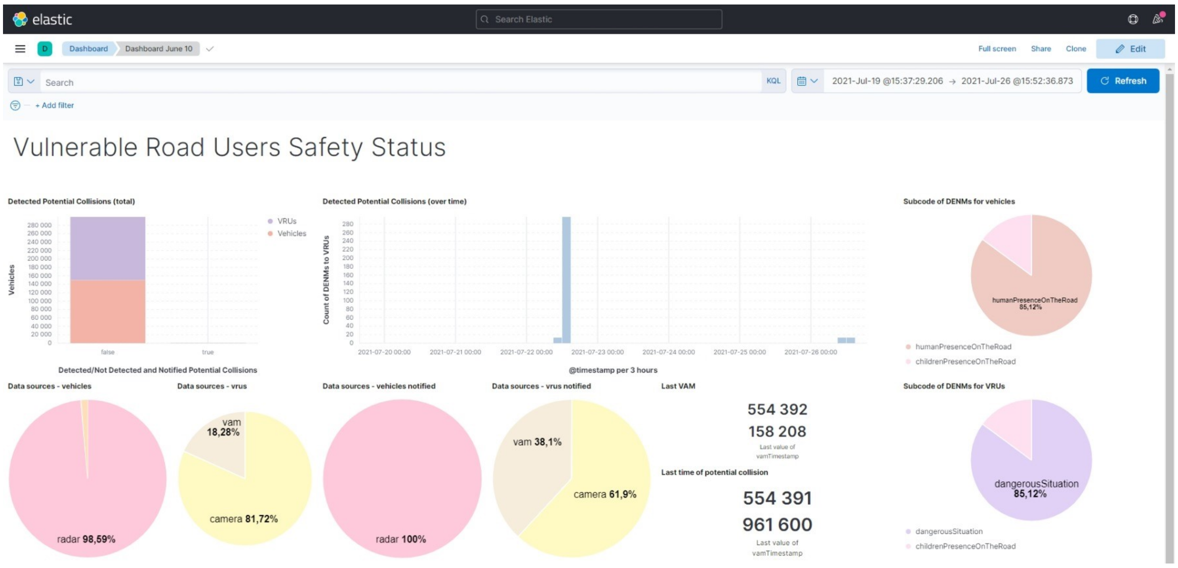
- *fastVehicle (28)*: to warn VRUs if the vehicle is going too fast;
- *childrenPresenceOnTheRoad (29)*: to warn vehicles about a children VRU. Children require more caution from drivers than typical VRUs, since they tend to be more fragile and harder to see. Therefore, this distinction is important for drivers.

By notifying vehicles and VRUs, the system is also capable of providing the expected end-to-end functionality, being capable of fulfilling the usage scenarios defined in Section 2.3. Use Case 1, *Collision detection with a pedestrian crossing a road*, and its variants with and without infrastructure or VRU ITS-S, are fully fulfilled with the notification of the potential collision.

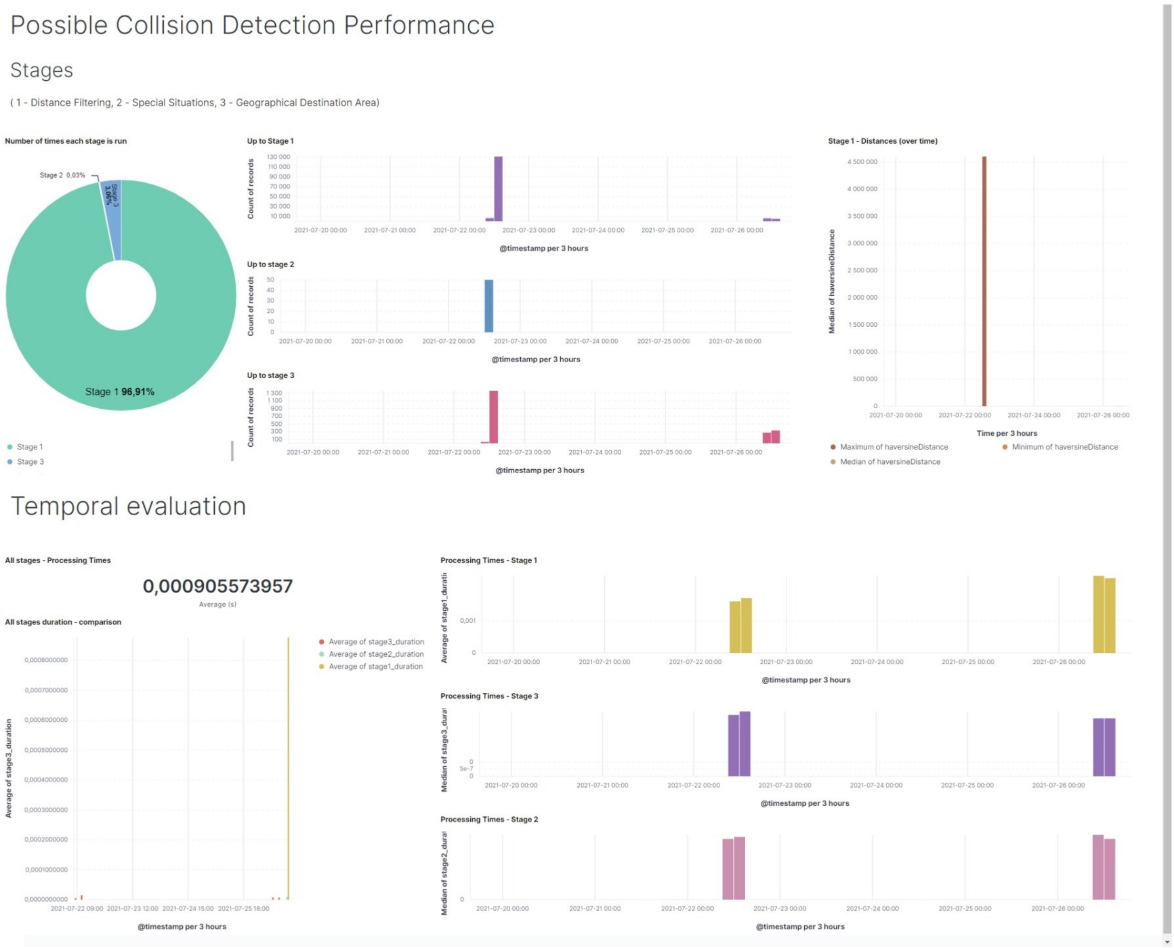
5.4.4 Metrics and statistics dashboard

To observe and monitor the system and observe any trends and patterns in the detected accidents, the logging from the Safety Service is connected to an Elasticsearch-Logstash-Kibana (ELK) stack. In the ELK stack, logs from an application are processed by the Logstash, stored by the ElasticSearch, and visualized in a Kibana visualization tool.

Figure 5.7 presents an example of a dashboard used during the development of the system to understand its overall status. Other variants are possible, including adding more graphics to query other information from the logging of the Safety Service.



(a) Visualization of the overview of the Safety Service



(b) Visualization of details of the algorithm executions

Figure 5.7: Dashboard visualization on Kibana based on logging data from the Safety Service

5.5 SUMMARY

This chapter enumerated the main decisions for implementing the previously defined architecture of the VRU safety approach.

The usage of existing infrastructure from the ATCLL project meant the usage of existing information aggregators based on the publish/subscribe paradigm - the Eclipse Mosquitto MQTT broker and the Fiware Orion Context Broker. These brokers were used in the context of this system, since they suited the requirements of the defined system, and it would be, therefore, redundant to implement a different information aggregator.

The usage of standardized information formats - ETSI CAMs, VAMs and DENMs, both by existing ATCLL infrastructure and by the new VRU ITS-S and Safety Services, allowed an easier integration with the current infrastructure, while rendering future developments of the service or other services easier. The usage of a smartphone as a VRU ITS-S allows an easy entry of VRUs - and even vehicles - on the overall system. Finally, the usage of several types of information from different sensors increases the robustness of the overall system.

The next chapter will present an evaluation of the deployed solution.

CHAPTER 6

Evaluation

Comparison is the thief of joy.
— Theodore Roosevelt

Chapter 4 and Chapter 5 presented the architecture of the VRU safety system, and the implementation details considered during the development of the system. This chapter presents the evaluation results obtained from a set of laboratory and real-life tests.

First, Section 6.1 enumerates the characteristics of the equipment used during the evaluation of the system. Then, Section 6.2 presents and discusses the results from a temporal analysis, necessary to understand if the system fulfills the requirement specifications defined in Section 3.8.3. Section 6.3 supplements these results by providing a scalability analysis of the system. Afterwards, Section 6.4 provides an analysis on the detection performance and accuracy of the detection of potentially dangerous situation, as well as the effect of several situations and hyper-parameters values in the system's overall behavior. Finally, Section 6.5 summarizes of the most relevant results is presented.

6.1 EQUIPMENT

All the presented evaluation results used real hardware from the ATCLL infrastructure. The usage of simulators such as Eclipse SUMO (Simulation of Urban MObility)[1] could allow the validation of the system. However, using real hardware in real contexts allows testing the system performance without any simplification that the simulators might introduce to road situations. For example, simulators might minimize or not consider the noise of the measurements, the wireless conditions with obstacles, or the impact of the performance of each hardware device in the system overall performance.

Table 6.1 lists the equipment used for all the evaluations of the system.

---

[1] `https://www.eclipse.org/sumo/`

Table 6.1: Equipment characteristics

Node	Equipment	Model	Characteristics
VRU	LTE ITS-S	Samsung Galaxy A50	LTE-A, Android 11
	5G ITS-S	Xiaomi Mi MIX 3 5G	5G, Android 9
Vehicle	OBU	PC Engines APU3C4	AMD Embedded G series GX-412TC, 1 GHz quad-core, 4GB RAM, 30GB SSD, WLE200NX miniPCI express card WAVE module, GPS Module and 3 Gigabit Ethernet channels
	RSU	PC Engines APU2E4	Same as APU3C4 but with 2 WLE200NX miniPCI express card Wi-Fi and WAVE module
Radar		Smartmicro UMRR-11 Type 4 Traffic Management Sensor	24GHz Radar sensor with multi-lane and multi-object tracking with 4D Doppler based radial motion detection [73]
Camera	Camera	Reolink RLC-423	CMOS 5MP Sensor, day/night mode, pan and tilt capability, 2560x1920@30 fps
	Computer	NVIDIA Jetson Nano	Developer Kit Version Jetpack 4.4.1, Tegra 210

Part of this equipment - namely the Radar, camera, and RSUs - were installed in the city of Aveiro in the context of the ATCLL project, within Smart LampPosts, as described in Section 3.2.

Applications such as the brokers or the Safety Application were deployed within the ATCLL infrastructure, with the specifications described in Table 6.2.

Table 6.2: System specification of machines used in the infrastructure

Central NAP Infrastructure (cloud brokers)	
Processor	Intel Xeon Silver 4215 (12 of 16 threads) @ 2.494GHz
Memory	16 GB
Storage	338 GB
Operative System	Ubuntu 18.04.4 LTS 64 bits
Safety Application Virtual Machine	
Processor	Intel Xeon Silver 4215 (4 of 16 threads) @ 2.494GHz
Memory	4 GB
Storage	62 GB
Operative System	Ubuntu 20.04.2 LTS 64 bits

6.2 Temporal analysis

Being a safety ITS application, timing is critical. In the case of a safety application responsible for detecting potentially dangerous situations between vehicles and VRUs, the need for highly responsive analysis of the situation and notification of the vehicle and VRU is absolute.

The concept of highly responsive can either be seen subjectively or objectively. The system can work fast enough to the human eyes (subjective), but that is not enough to guarantee that the system can work, under load, in all situations, within an appropriate time constraint - as referred in Section 3.8.3. Therefore, it is essential to analyze the timing of all the steps of the system to determine the necessary time to receive information, process it (also considering stress tests), and notify about a potentially dangerous situation.

In this context, and as described in Section 3.8.3, and considering a positioning inaccuracy of 1.0 m (conservative decision), the upper bound for the end-to-end

time is 300 ms.

6.2.1 Evaluation methodology

To analyze the timing of all the steps of the system, several timestamps were implemented within the critical steps of the system flow, used to compute the time between steps. Figure 6.1 and Figure 6.2 show the system flow when receiving information and notification of a potential accident, respectively. In both figures, the timestamps are marked - C for Car/Vehicle-related timestamps, S for Safety App-related timestamps, V for VRU ITS-S application, and with * for timestamps for the notification flow.

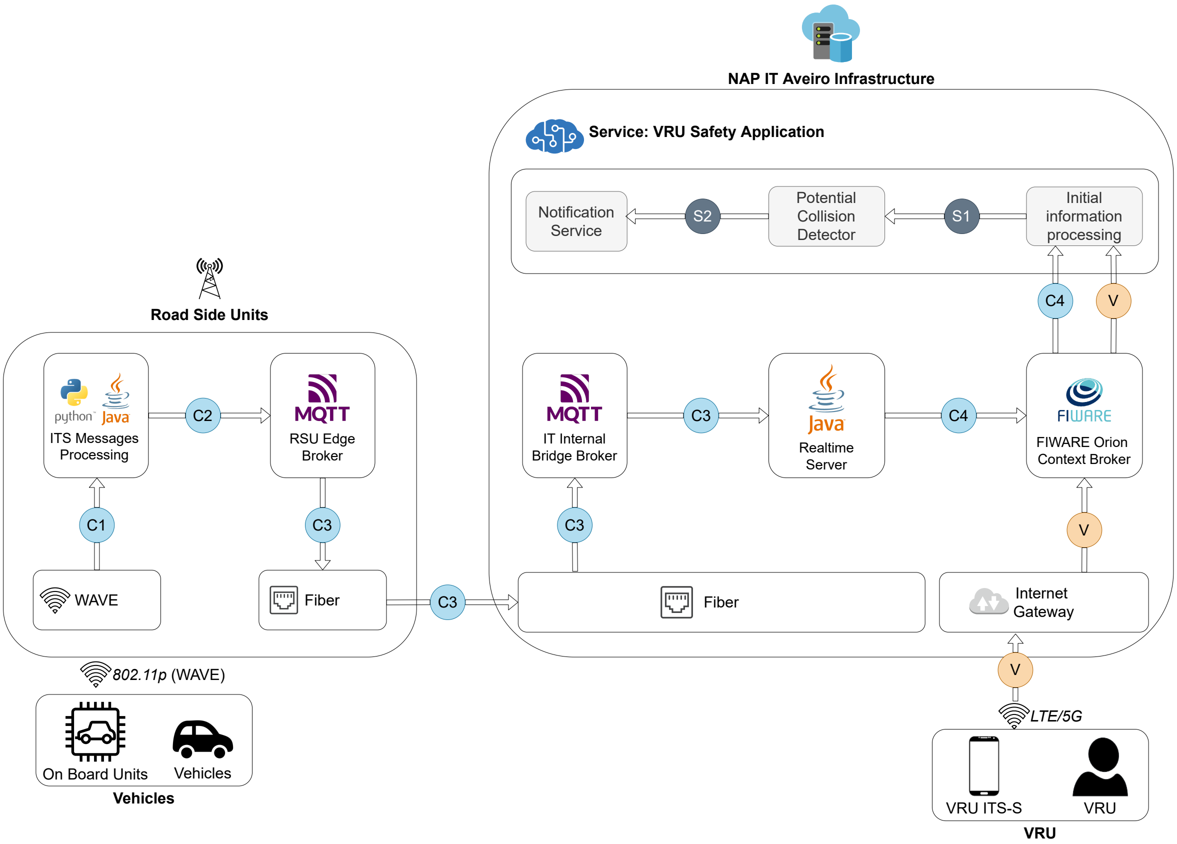


Figure 6.1: Timing Analysis Setup in the flow of receiving information

Since the timestamps were deployed in different applications and in different machines, it was crucial to consider a typical distributed computing problem, the synchronization of the clocks of each machine. To ensure all clocks were synchronized, and therefore the obtained results were consistent, the clocks were synchronized with a common Network Time Protocol (NTP) server. To avoid increasing *drifts* affecting the value of the clock, all applications and related machines were synchronized with the same NTP server before each experiment. This included the Android smartphones used as VRUs ITS-Ss.

After ensuring that all timestamps were consistent, experiments were done to obtain values for each timestamp and compute the time between each step.

It should be noted that other sources of information exist - namely cameras and Radars. However, since they publish their information to the same infrastructure as the vehicle, a

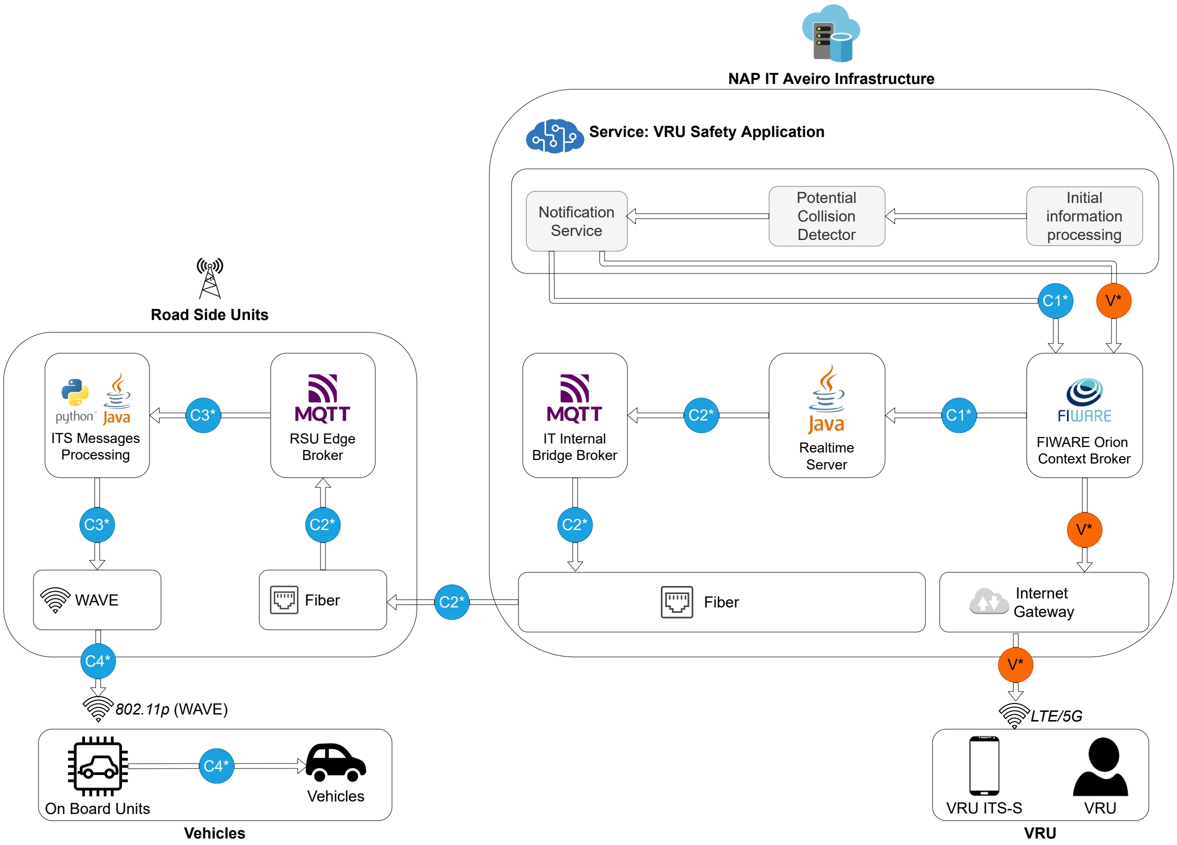


Figure 6.2: Timing Analysis Setup in the flow of notification of a potential accident

timing analysis of the vehicle information flow also allows understanding the temporal behavior of Radar and camera messages.

The following sections detail the obtained timings results for vehicles, VRUs and the Safety Application. The results contain the mean of 10000 experiments, and the confidence interval of 95%.

6.2.2 Centralized vs edge computing for vehicles

Table 6.3 presents, for vehicles, the times computed based on the testing setup described in Section 6.2.

An analysis of the results shows that, under normal load conditions, (*i.e.*, during the day, with several city buses) with a city with the most significant points of delay are the two brokers required for the information to reach the public endpoint of the Safety App service. The first conclusion is to remove one of the brokers, decreasing the total time associated with vehicles by 50-60 ms. However, this is not feasible since the system has been developed on top of existing infrastructure that cannot be changed without consequences for other services. Notice that we consider that, during the night, when no buses are in the roads of Aveiro, the infrastructure is in light load, and while scalability tests are performed, the infrastructure would be in high load.

An alternative would be to shift the developed system towards an edge solution, with the deployment of the safety application as closely as possible to vehicles, *i.e.* in the RSUs.

Table 6.3: Stages evaluated for the vehicle

ID	Name	Formula	Value (ms)	Percentage
Number of tests = 10000				
Car Sending Time = C1 + C2 + C3 + C4 = 123.56 ms				
C1	OBU to RSU	Time between CAM being produced and being received by RSU	3.77 ±0.00486	3.05%
C2	RSU Processing	Time between receiving a CAM and publishing it in the local MQTT broker	3.75 ±0.127	3.03%
C3	RSU to Fiware Orion Realtime server	Time between publishing a CAM in the local MQTT broker and publishing in the Fiware Orion realtime server for parsing to NGSIv2	59.27 ±0.000886	47.97%
C4	FIWARE ORION Transport Time	Time between publishing a CAM in the central broker and subscribing it in the safety application	56.77 ±0.0111	45.95%
Car Notification Time = C1* + C2* + C3* + C4* = 69.2 ms				
C1*	Safety App to MQTT Bridge	Time between creating a DENM in the safety application and publishing it in the central broker	30.17 ±0.626	43.60%
C2*	MQTT Bridge to RSU	Time between publishing a DENM in the central broker and it arriving at the RSU	27.26 ±0.247	39.39%
C3*	RSU Parsing time	Time between the DENM arriving at the RSU and being ready to be sent to the OBU	4.78 ±0.0899	6.91%
C4*	OBU Processing time	Time between the DENM being sent to the OBU and being available to other OBU applications	6.99 ±0.215	10.10%
Car End-to-end Time = C + C* = 192.76 ms				

However, this is not recommended since the edge equipment does not have as much processing capacity as the cloud solutions. Scalability is also an issue, with horizontal (adding more machines is time-consuming, and the space in the city renders the solution unfeasible) and vertical (the equipment is not easily upgradable) scalability not easily achievable. By opposition, cloud infrastructure tends to be easier to scale, either by adding more machines or upgrading the existing ones to a central data center. Having as many edge points as possible throughout the city can mitigate this issue with an automated deployment of the safety application with a tool such as Ansible, but at the expense of more monetary costs.

Moreover, changing for a total edge solution would require some relay service between the VRU ITS-S and the edge, which adds another layer of complexity to the system. This service would ensure that the vehicle and smartphone are redirected to communicate with the best edge point(s). Otherwise, the information might be processed far away from the vehicle and VRU, increasing delays and, thus, defeating the purpose of the deployment of an edge solution.

An intermediate solution would be to implement the safety application such that it is capable of receiving information from multiple sources and multiple brokers, both from a centralized cloud (more complex brokers such as the Fiware Orion) or from edge points. Lightweight brokers like the Eclipse Mosquitto are more suitable for low power, low computing capabilities pieces of equipment - typical of edge computing systems.

Such solution was implemented, with the safety app capable of obtaining information of vehicles, not only from the cloud Fiware Orion Context Broker, but also the central Eclipse Mosquitto MQTT Bridge broker responsible for aggregating the information of the brokers present in each RSU/edge point. Table 6.4 presents the obtained results for this solution.

Table 6.4: Stages evaluated for the vehicle (subscribing edge information from the central broker)

ID	Name	Description	Value (ms)	Percentage
Number of tests = 10000				
Car Sending Time = C1 + C2 + C3 = 17.21 ms (before was 123.56 ms)				
C1	OBU to RSU	Time between a CAM being produced and being received by RSU	3.77 ±0.00486	21.91%
C2	RSU Processing	Time between receiving a CAM and publishing it in the local MQTT broker	3.75 ±0.127	21.79%
C3	RSU to MQTT Bridge	Time between publishing a CAM in the local MQTT broker and publishing in the central broker	9.69 ±0.022	56.30%
Car Notification Time = C1* + C2* + C3* + C4* = 59.15 ms (from table 6.3)				
Car End-to-end Time = C + C* = 76.36 ms (before was 192.76 ms)				

In this solution, the need for the information to reach the Fiware Orion is eliminated, meaning that the information can be consumed after being published into the RSU/edge broker and bridged into the central MQTT broker. This optimization means that the C_4 time, Fiware ORION Transport Time, is eliminated, meaning that at least 56.77 ms can be saved. The temporal improvement is further improved because the time between the central MQTT broker and the Fiware Orion (corresponding to parsing to the NGSIv2 format done by a real-time server) is also eliminated. This optimization meant that C_3 time - now time between the RSU and the MQTT Bridge - decreased to just 9.69 ms (and not 59.27 ms).

To further analyze how much the obtained times could be improved, the safety application was changed such as to obtain information from vehicles even closer from their generation sources, *i.e.* from the Mosquitto broker within the edge equipment (*i.e.* the RSUs). Table 6.5 presents the results of this implementation.

Results are further improved since the C_3 time decreases from the previous 9.69 ms to 5.14 ms. These results stem from the fact that it is no longer required to wait for the synchronization of information between the edge broker and the central broker. However, this is possible at the expense of adding complexity to the implementation of the safety application, since all RSUs have to be manually subscribed - instead of just one central broker.

Figure 6.3 summarises the results obtained for the 3 alternatives.

Both edge implementations make possible to conclude that edge computing is essential to decrease the overall end-to-end latencies to values more compatible with timing requirements of safety ITS applications. However, it also requires more complexity in the edge equipment - the RSUs need to have a message broker to support the solution described in Table 6.5, and that might not always be possible.

Table 6.5: Stages evaluated for the vehicle (subscribing edge information from the edge broker)

ID	Name	Description	Value (ms)	Percentage
Number of tests = 10000				
Car Sending Time = C1 + C2 + C3 = 12.66 ms (before was 17.21 ms)				
C1	OBU to RSU	Time between a CAM being produced and being received by RSU	3.77 ±0.00486	29.78%
C2	RSU Processing	Time between receiving a CAM and publishing it in the local MQTT broker	3.75 ±0.127	29.62%
C3	RSU to MQTT Bridge	Time between publishing a CAM in the local MQTT broker and subscribing it in the safety application	5.14 ±0.0216	40.60%
Car Notification Time = C1* + C2* + C3* + C4* = 59.15 ms (from table 6.3)				
Car End-to-end Time = C + C* = 71.81 ms (before was 76.36 ms)				

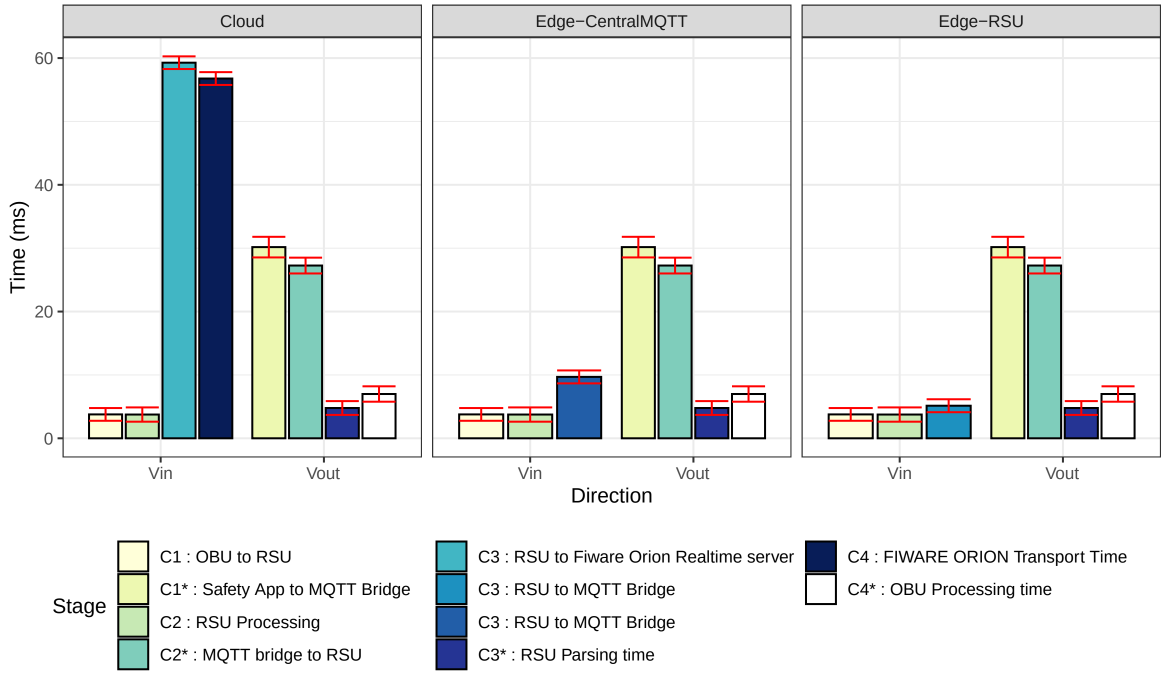


Figure 6.3: Centralized vs edge computing approaches for vehicles and their impact in the timing of the subscription of information

6.2.3 LTE vs 5G performance for VRU ITS-Ss

Figure 6.4 presents, for VRU ITS-Ss, the times computed based on the testing setup described in Section 6.2. Results include measurements both for a LTE-A smartphone and 5G smartphone using a pre-commercial 5G network[2] for two places in the city of Aveiro. All measurements were done with the smartphones in the same place, at the same time. The results consider the mean of 8000 experiments and a confidence interval of 95%.

Results show a clear advantage of 5G over LTE-A, both in up-link (V_{in} times) and down-

---

[2] By October 1st 2021, the Autoridade Nacional de Comunicações (ANACOM) auction for allocation of 5G related frequency was not finished and therefore, the usage of 5G commercial network was not possible.

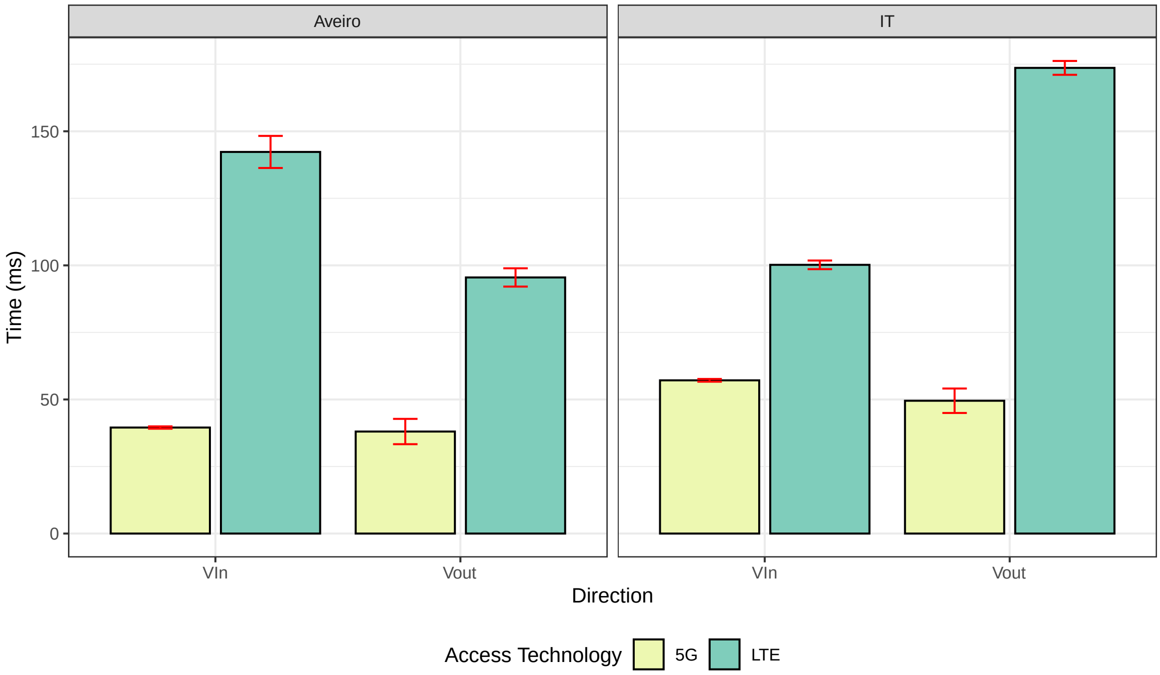


Figure 6.4: Stages evaluated for the VRU ITS-S application (set of tests 1, next to a 5G cell tower in Aveiro and in IT-Aveiro)

link/notification time (V_{out} times), with the difference between LTE-A and 5G being very significant - 120 ms and 100 ms, for up-link and down-link, respectively.

The times are prohibitively high in LTE-A; only 5G can fully support latencies compatible with an emergency service, such as the one that was developed - 300 ms for an average precision (1.0 m) as described in Section 3.8.3. 5G would be an even clear advantage for migrating to an edge computing approach with more processing in the VRU ITS-Ss nodes.

However, the usage of 5G should be taken with caution. The smartphone reporting that it is using 5G is not a synonym of entirely using the capabilities of the 5G generation. It is possible to visualize that the obtained timings in a non-ideal location (inside of IT-Aveiro building) are far from the ideal. Only in an ideal location - location with direct line of sight to a 5G antenna close to *Cais da Fonte Nova*, in the city of Aveiro, Portugal) - the results are within the expected by the operator in terms of bandwidth (upload of 100 Mbits/sec, download of 1200 Mbits/sec)[3].

Another important conclusion is that the 5G front-haul and back-haul must be performing correctly. The first measurements obtained for 5G were performed during a day where the network core had issues related to the antennas[4]. These issues severely impacted the performance of the 5G network, rendering its performance to essentially LTE-A values.

While issues with the network are expected not to happen very often, it is nonetheless vital to ensure that the impact of any potential issue in the overall end-to-end latency is minimized - especially in a safety ITS application. Once again, this shows the importance of

---

[3] Source: Experimental test of Non-Standalone 5G Architecture from Altice Labs Aveiro, July 2019.

[4] Since an experimental pre-production 5G network was used, connectivity issues can be expected.

multi-homing as a way to minimize possible loss of service or degradation of the performance of just one access technology by making it possible to use several network access technologies (5G or LTE-A, C-V2X, WAVE) simultaneously, therefore reducing or eliminating the impact of performance issues in one of those technologies.

Also important to have in mind is that the temporal values include some processing time associated with processing the messages by the Fiware ORION broker. To better understand the impact of the communication part only, a *iperf* based test was conducted to measure the bandwidth achievable between the smartphone and the server where the ORION broker is deployed. Figure 6.5 presents the results of 10 tests of 1 minute each, with a confidence interval of 95%.

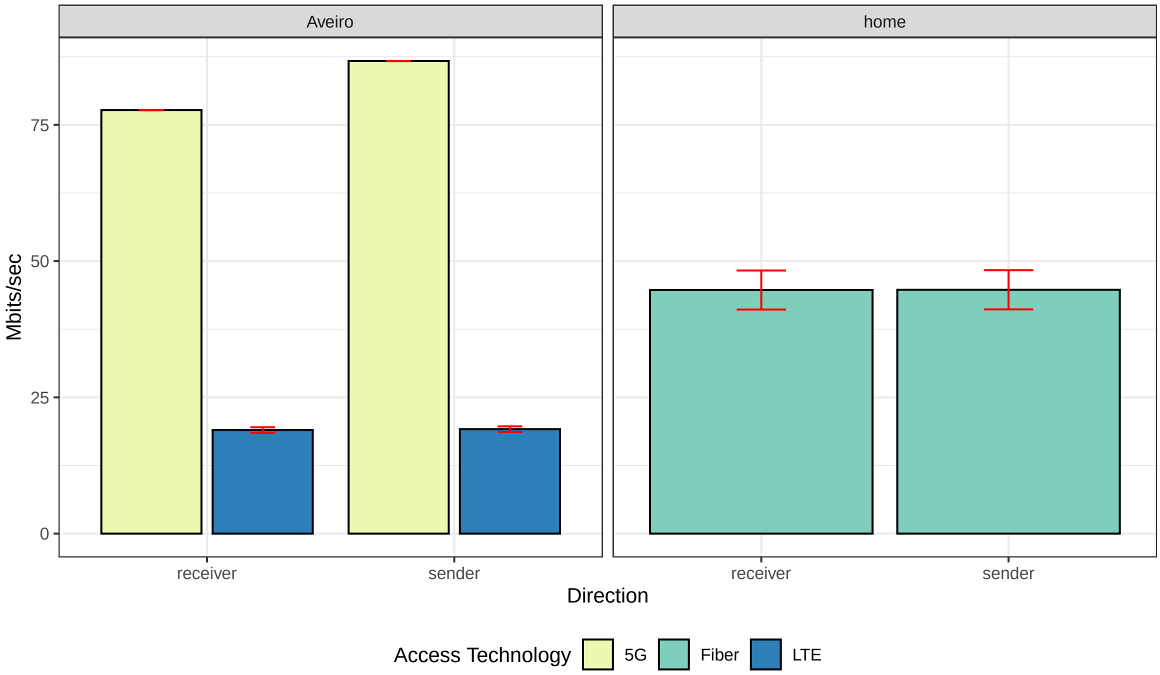


Figure 6.5: Bandwidth test for 2 smartphones in Aveiro and comparison with a cabled home network

The tests in Aveiro were performed next to a 5G cell tower with two smartphones, one supporting 5G and one supporting just LTE-A. During the execution of these *iperf* tests, the Speedtest tool was used to understand the loss of performance for having to use a VPN to connect to the ORION broker. Results given by the Speedtest tool - 1184.6 $Mbits/sec$ of download and 83.892 $Mbits/sec$ of upload for 5G and 34.062 $Mbits/sec$ and 12.396 $Mbits/sec$ of download and upload in LTE-A - are very penalizing for the 5G download bandwidth capability.

To understand if this can be a limitation of the IT-Aveiro infrastructure - and in particular with the IT-Aveiro VPN server , a controlled *iperf* test between the ORION broker server and a computer connected via 1 Gigabit Ethernet to a router connected to a fiber network, was conducted. Results at home - less than 50 Mbits/second of download and upload - suggest that the bandwidth seems limited by the IT-Aveiro infrastructure and not by the network characteristics, supporting the conjecture that the 5G results could be significantly improved.

6.2.4 Safety application

Table 6.6 presents the times computed based on each timestamp described in Figure 6.1 and Figure 6.2 for the safety application.

Table 6.6: Stages evaluated for the application

ID	Name	Description	Value (ms)	Percentage	Number of tests
S1	Message processing time	Difference between time VAM message entered application and algorithm starts running	3.12 ±0.369	59.7%	8175
S2	Potential Collision Algorithm time	Execution time of the algorithm	0.0548 ±0.398	40.3%	508344
Safety App Time = S1 + S2 = 13.6 ms					

Results show that the focus on having a computationally simple algorithm, being triggered more frequently instead of a complex algorithm, has the advantage of ensuring the times of the safety application to not significantly impact the overall time of the system execution.

It should be noted that the time S_2, the algorithm time, presents a much higher number of tests since the algorithm runs, for each combination of VRU and vehicle information (from Radar and OBUs). The confidence interval is 95%.

Section 6.3.3 will present further details on how well the application scales with more nodes.

6.2.5 Overall end-to-end time

In Table 2.5, the RP_2 latency requirement was defined, stating that the system should be able to detect a potential accident and warn the vehicle and VRU within an appropriated amount of time, with Section 3.8.3 defining an end-to-end latency ceiling of 300 ms. This value shall be respected. Otherwise, notifications might be too late to be effective, therefore increasing potentially fatal false negatives.

Considering a worst case scenario defined in Figure 6.6, where a new VRU information is sent after the vehicle information (instead of simultaneously), and considering a central edge approach (with lower end-to-end latencies than the centralized cloud solution) and 5G VRU ITS-S (since the central approach for vehicle info and LTE for VRU presented greater latencies), the end-to-end time for vehicle and VRU is given by the following formulas:

for the vehicle:

$$\begin{aligned} t_{Vehicle} &= C_1 + C_2 + C_3 + V_{in}^{5G} + S_1 + S_2 + C_1^* + C_2^* + C_3^* + C_4^* \\ &= 3.77 + 3.75 + 9.69 + 57.14 + 3.12 \\ &+ 0.0548 + 30.17 + 27.26 + 4.78 + 6.99 \\ &= 146.72\,ms \leq 300\,ms \end{aligned} \tag{6.1}$$

and for the VRU :

$$\begin{aligned} t_{VRU} &= C_1 + C_2 + C_3 + V_{in}^{5}G + S_1 + S_2 + V_{out}^{5G} \\ &= 3.77 + 3.75 + 9.69 + 57.14 + 3.12 + 0.0548 + 49.52 \\ &= 127.04\,ms \leq 300\,ms \end{aligned} \tag{6.2}$$

Results prove, therefore, that the temporal requirement was fulfilled, with both end-to-end times below the 300 ms ceiling value.

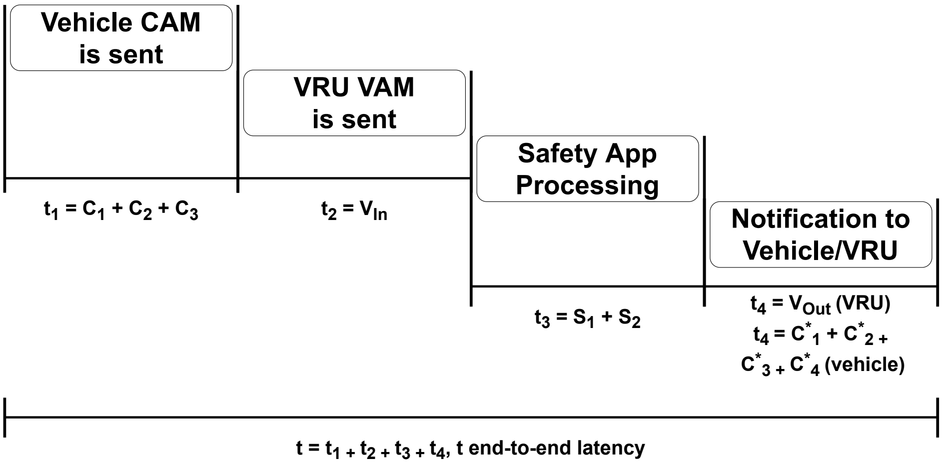


Figure 6.6: Worst case situation considered

6.3 PERFORMANCE AND SCALABILITY OF THE SYSTEM

The results so far considered a reduced number of vehicular and VRU nodes. The temporal analysis performed in Section 6.2 considered that the network and vehicular equipment are under normal conditions and usage, *i.e.* they were performed during the day, with typical traffic from Aveiro. However, this is not enough to fully prove that the system can perform under stress. It is therefore essential to study the effect of having N vehicles and M VRUs ITS-Ss, with various values for both N and M, in the critical steps of the infrastructure, both for the vehicular and VRU information and notifications. This set of tests allow proving the scalability of the system.

6.3.1 Vehicle RSUs and OBUs

To understand the impact of N vehicles and M VRUs ITS-Ss with N, $m > 1$ on the times associated with the vehicle information (Table 6.3), CAMs were injected into the infrastructure by a threaded program, responsible for publishing N CAMs on a RSU present in a laboratory environment. The RSU being deployed in a laboratory allowed a controlled environment, where the number of CAMs were exactly the number being injected by the thread program, *i.e.* without the noise of OBUs from the cars or buses.

Then, times were re-obtained using the same procedure described in Section 6.2. Results are shown in Figure 6.7, which represent a mean of 10000 tests.

Results do not show any tendency with up to $N = 1000$ [5]). The only conclusion that can be considered is that values C_1 to C_2 tend to be stable and within the system specifications, and values related to the Eclipse Mosquitto MQTT and Fiware Orion brokers do not have

---

[5] Considered reasonable for RSU coverage in an urban environment, in a road with two lanes for each direction, considering the coverage of a RSU of around 300 m

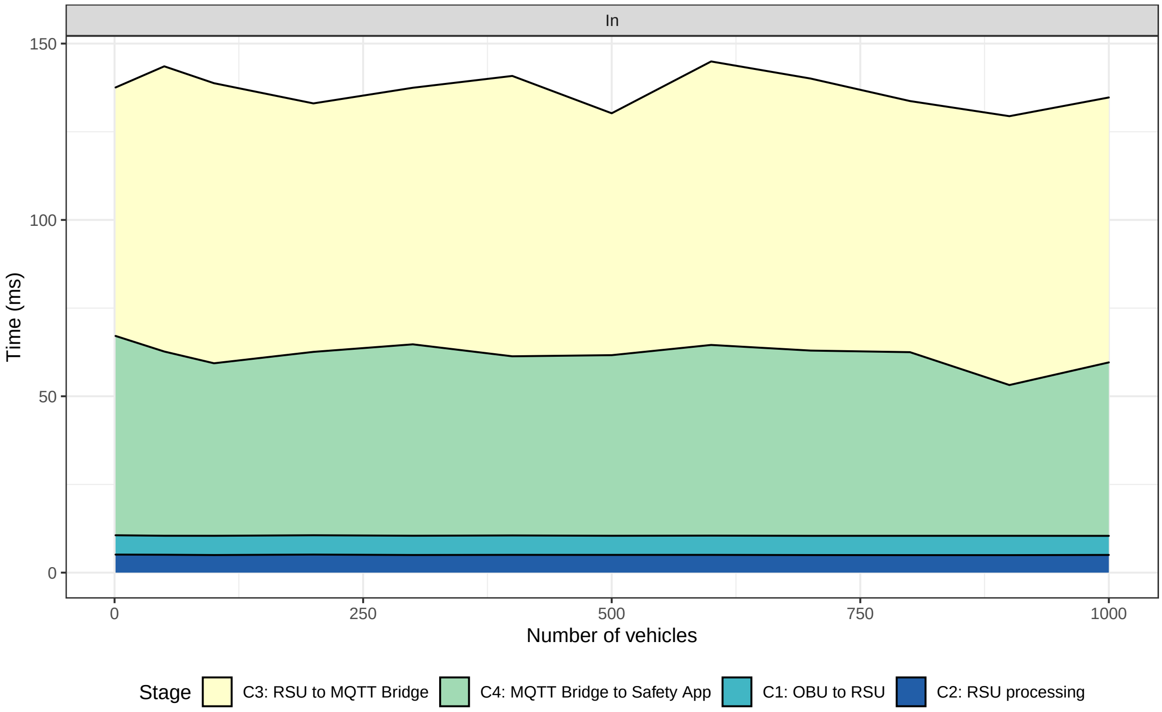


Figure 6.7: Results of the scalability evaluation for the vehicle

any relation with the variation of the value of N. This conclusion can be explained by the fact that both brokers are a shared infrastructure, with load differences during the execution of the several tests that stem from the normal variations of information within a city.

6.3.2 VRU ITS-S Application

Analyzing the scalability or the impact of M vehicles and N VRUs ITS-Ss in the times of VRUs ITS-Ss, however, is not as easy to achieve. Since in this system, the VRU ITS-S is a smartphone communicating to the rest of the system through cellular (either LTE-A or 5G), fully analyzing the scalability would include considering the impact in the network core. To get results with statistical significance, thousands of mobile nodes would be required. Moreover, even if that equipment was obtainable, either directly or through simulation, it could not realistically be tested since the cellular network (Vodafone for LTE-A and pre-commercial Altice Meo for 5G) is used in production.

However, it is also important to consider that the VRU ITS application running in the smartphone runs within a limited computing power. Considering a scalability analysis scenario, that implies that the application needs to be able to process hundreds or thousands of DENMs per second without crashing or blocking. Therefore, to understand if the VRU ITS application can scale, DENMs were injected into the infrastructure to simulate the VRU being notified about multiple potential collisions.

Figure 6.8 presents the results, which represent a mean of 50000 tests. Results show that the VRU ITS-S application is susceptible to receiving more DENMs per second, with the notification time varying proportionally with the number of DENMs.

However, it should be noted that the tested numbers of DENMs per second are unrealistic. The DENMs are used, in the context of the VRU ITS-S application, to provide a visual notification to the VRU that there is a dangerous situation. Since a VRU only wants to know if it is in danger or not to act accordingly, there is no need of sending all the DENMs representing all the potential dangerous situations with all vehicles, to the VRU ITS-S application - only the DENMs to notify the VRU about danger, avoiding overloading the network and VRU ITS-S application. It should also be noted that the VRU ITS-S application can only receive, by design, notifications directed at them - filtering any broadcasted DENMs that may be sent by other vehicles or VRUs. This decision was made to simplify the testing phase, avoiding noise from other projects within the same infrastructure also publishing DENMs.

In addition, the notification times - that measure the time between DENM generation and the VRU being visually and audibly notified - will, naturally, tend to infinite since the application itself can only process a finite number of UI operations per second. Therefore, in a situation where thousands of DENMs are sent to the application, the application will naturally start to queue the UI operations, making the notification times cumulatively higher (as it was verified for a number of DENMs superior to 2500).

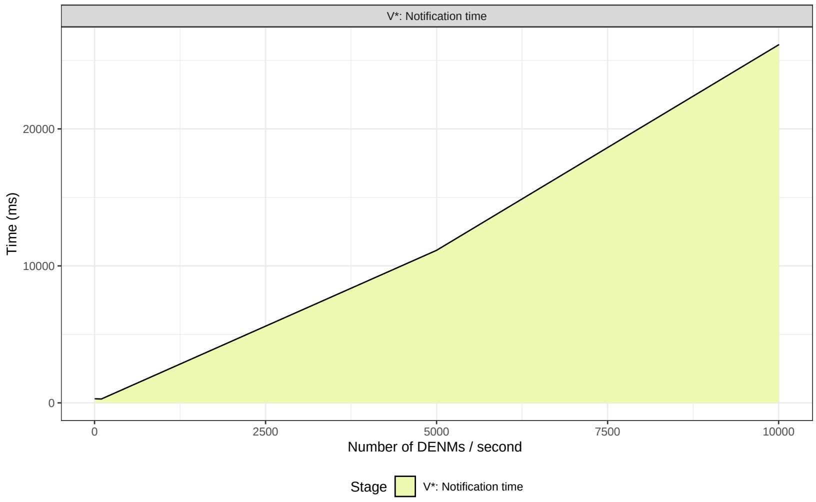


Figure 6.8: Results for scalability tests of VRU ITS-S

It should be noted that, understanding the impact of sending more VAMs per second is not critical since it is not supposed to have that scalability issue: the VRU ITS application is supposed to be exclusive to 1 VRU. Even in a situation of a VRU ITS application representing more than 1 VRU or a cluster of VRUs, just 1 VAM would be sent representing the whole cluster.

6.3.3 Safety application

The results presented so far focused on the scalability of the nodes. However, to fully understand how the system scales, the Safety Application needs to be tested. Figure 6.9 presents the composition of a testbed used for this evaluation - 3 OBUs, 1 RSU and 6 VRU

ITS-Ss smartphones[6], and Figure 6.10 presents the results of a mean of 50000 tests and confidence interval of 95%.

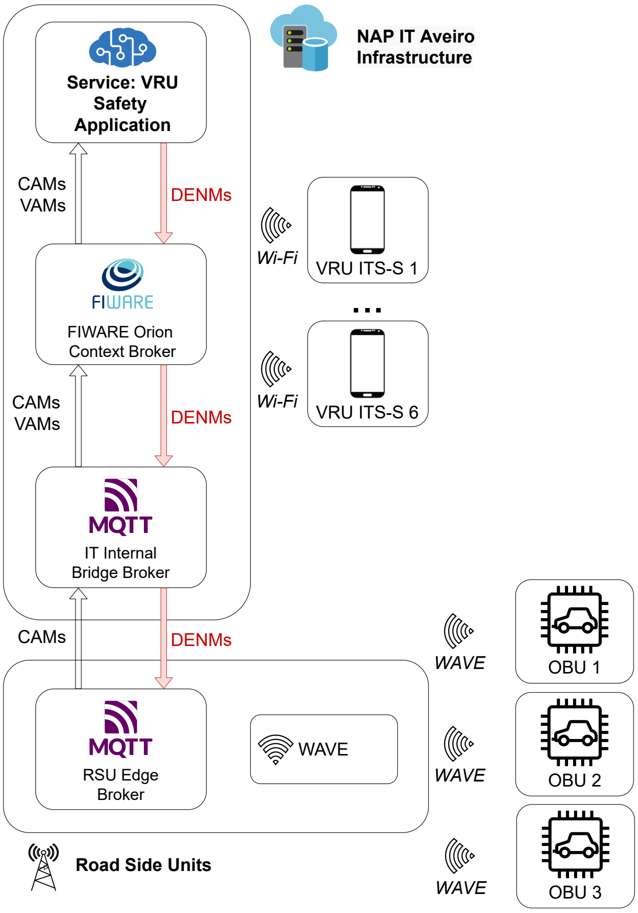


Figure 6.9: Experimental setup for scalability tests of the Safety App

Results show no visible tendency with increasing values of vehicles (simulated by different frequencies of generation of CAMs). However, results also show that the upper bound of the value of both times S_1 and S_2 is within the values presented in Table 6.6, which indicates that the safety application scales for a number of vehicles expected for one or a few RSUs.

However, it is not possible to conclude that one instance of the Safety App can scale enough for processing information about the whole city. An edge approach, with several nodes deployed in each RSU, is a more scalable solution, with the correct operation of the App for that situation, as shown in this test.

---

[6] 1 Xiaomi Redmi Note 9T, 2 Motorola Moto G 5G, 1 OPPO Reno 4Z 5G, 1 Samsung Galaxy A50, 1 Xiaomi Mi MIX 3 5G - all Android smartphones with different hardware to simulate different VRUs

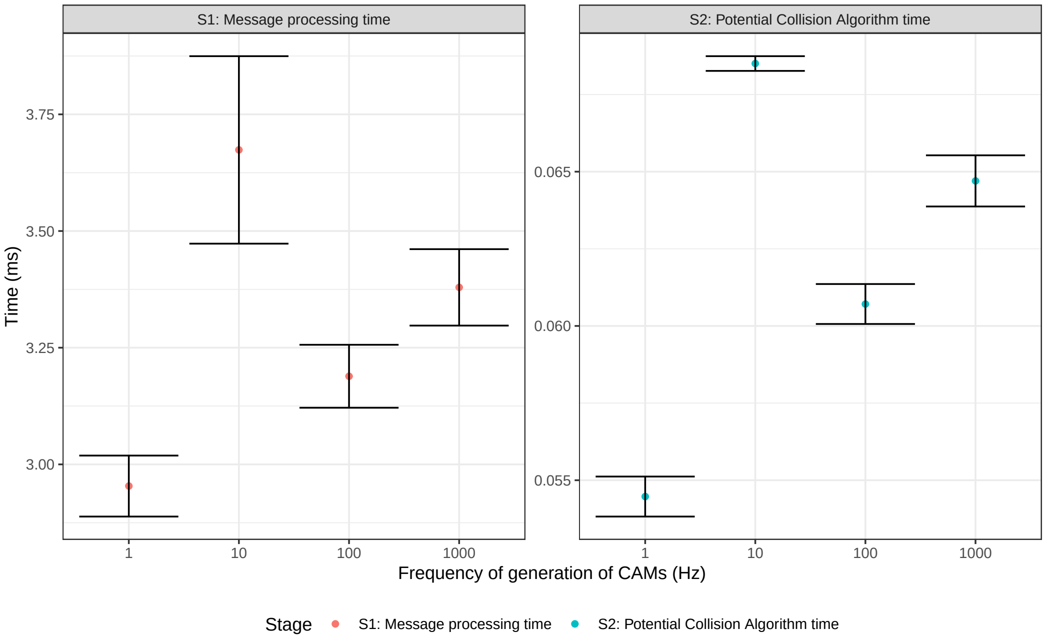


Figure 6.10: Results for scalability tests of Safety App

6.4 ACCURACY EVALUATION

6.4.1 Evaluation methodology

Apart from a timing and scalability analysis, the safety application must detect potentially dangerous situations and warn the nodes about them. To understand if the system works as required, two sets of tests are considered: a proof of concept test (henceforth denominated as test 1) and an accuracy test (henceforth denominated test 2). These tests allow the validation of results obtained during simulations done in laboratory conditions, between real vehicles and VRUs in simulated positions.

For each set of tests, the scenarios are performed online and in real-time, and the messages associated with the sensing of the vehicle and VRU from the different possible sensing sources are saved. Then, offline, the messages are injected into the developed system to obtain the information required to understand the accuracy of the system, including the accuracy of the location providers for the VRU and vehicle, the length of potential and unavoidable crash zones, and the effect of specific parameters of the collision detection algorithm.

It should be noted that, while the tests considered just 1 VRU and 1 vehicle, the system is prepared to process several VRUs and vehicles. The choice of just 1 VRUs and 1 vehicle during these tests was done to simplify the logistics of the tests.

Figure 6.11 presents the experimental setup for the proof of concept test. In this test, the experimental setup is at a crossroad where a VRU was walking through a straight road while a vehicle was approaching it, and if no action was taken, an accident would happen.

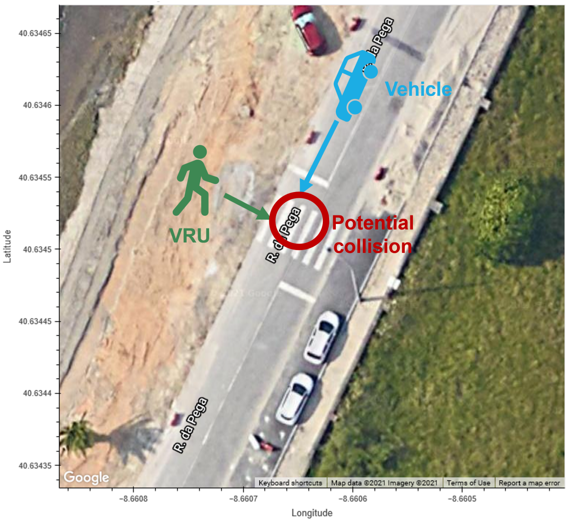
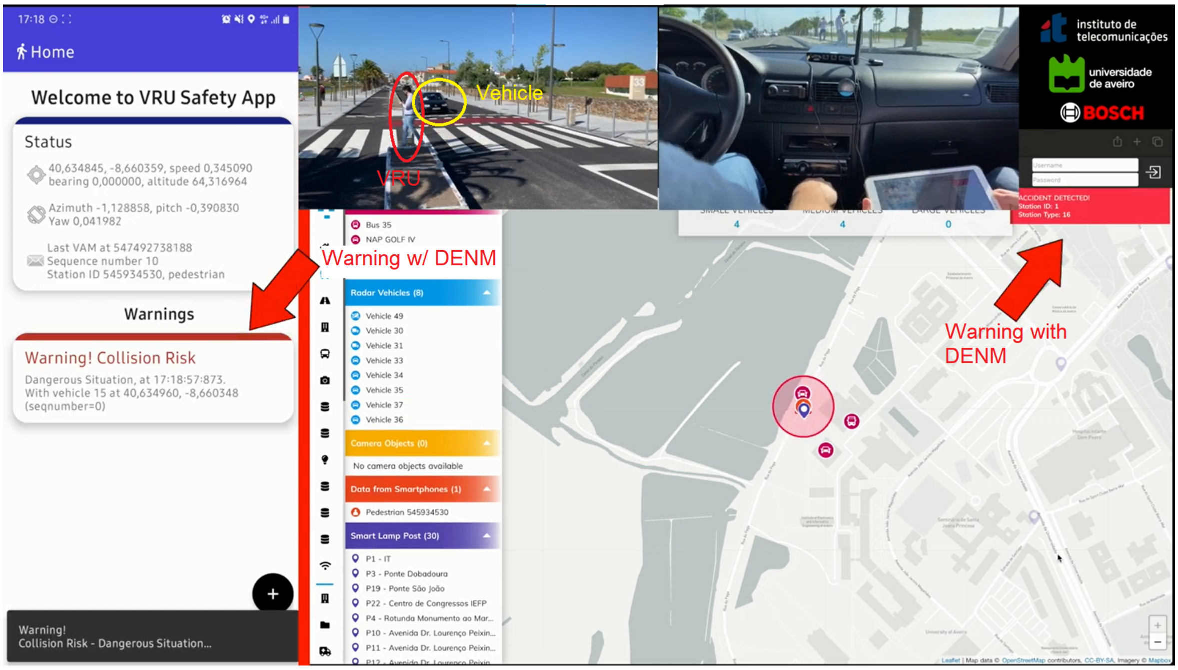


Figure 6.11: Test 1 Experimental Setup

The goal of this test is to understand if the system worked-end-to-end, therefore full-filling the use cases. It used an initial version of the safety application with a simple distance comparison algorithm to detect potentially dangerous situations.

Figure 6.12 presents the operation of the system. The test showed that the system was capable of detecting the potential collisions, and warned both the VRU and vehicle through the VRU ITS-S smartphone application and a City Manager dashboard, respectively[7].



Welcome to VRU Safety App

Status

40,634845, -8,660359, speed 0,345090
bearing 0,000000, altitude 64,316964

Azimuth -1,128858, pitch -0,390830
Yaw 0,041982

Last VAM at 547492738188
Sequence number 10
Station ID 545934530, pedestrian

Warnings

Warning! Collision Risk
Dangerous Situation, at 17:18:57:873.
With vehicle 15 at 40,634960, -8,660348
(seqnumber=0)

Warning!
Collision Risk - Dangerous Situation...

Bus 35
NAP GOLF IV

Radar Vehicles (8)
Vehicle 49
Vehicle 30
Vehicle 31
Vehicle 33
Vehicle 34
Vehicle 35
Vehicle 37
Vehicle 36

Camera Objects (0)
No camera objects available

Data from Smartphones (1)
Pedestrian 545934530

Smart Lamp Post (30)
P1 - IT
P3 - Ponte Dobadoura
P19 - Ponte São João
P22 - Centro de Congressos IEFP
P4 - Rotunda Monumento ao Mar...
P10 - Avenida Dr. Lourenço Peixin...
P11 - Avenida Dr. Lourenço Peixin...

ACCIDENT DETECTED!
Station ID: 1
Station Type: 16

Figure 6.12: Test 1 Results[8]

---

[7] Part of the ATCLL infrastructure was not developed by the author.

[8] Full video version available at `https://www.youtube.com/watch?v=m9PBEtlsQeA`.

This test considered just 1 VRU and 1 vehicle, and therefore just 1 warning possibility. However, while with more vehicles and VRUs, the system would be able to detect the potential dangerous situations between all vehicles and all VRUs, the notification process of the vehicle would have to be improved. Further discussion is provided in Section 7.2.

With a proof-of-concept test executed, the main focus was on improving and testing the potential collision algorithms to have in consideration other information on the dynamics of the nodes (*e.g.* speed, heading, acceleration), and also the fusion of information - in the case of vehicles, from OBUs and Radars.

Figure 6.13 presents the experimental setup for test 2, which is a crossroad in a semi-curved road in Aveiro (road of *Ponte Dobadoura*, a place where a RSU, a Radar and a camera were installed). During the execution of the test, the vehicle and VRU potentially collided twice (left and right situations in the figure) while the VRU was crossing the road.

Figure 6.13: Test 2 Experimental Setup. From left to right, run 1 and 2

The goal of this test was to understand if the system worked with an improved potential collision detection algorithm and how well it would work. The test allowed to understand the precision of the location of the nodes and if the new algorithm could detect true positives (actual potential collisions) and negatives (non potentially collisions).

The following sections focus on the detailed analysis of the results obtained in test 2.

6.4.2 Location of each node

To detect potentially dangerous situations between elements of a road, basic awareness of their position is required. Even if the developed algorithms have in mind additional information to decide if a collision is imminent or not, an accurate position is always essential and required.

Both in the case of the vehicle and the VRU, different sources of location were used. In the case of the VRU, the sources were video cameras and the Fused Location Provider API (as described in Section 5.3.1). In the case of the vehicle, the sources were video cameras, Radar and the GPS location provided by the cellular module of the OBUs, which was sent in CAMs through ITS-G5. All these sources present different characteristics, summarized in Table 6.7.

Table 6.7: Equipment characteristics concerning geolocalization

Equipment	**Model**	**Detectable Nodes**	**Accuracy**	**Frequency**
Radar	Smartmicro UMRR-11 Type 4 Traffic Management	Vehicles	0.25 m [73]	17.2 Hz to 13.(3) Hz (period between 58 ms and 75 ms)
OBU and RSU GPS	SIMCom SIM7600	Vehicles	2.5 m (Open Sky) [83]	1 Hz
Camera	Reolink RLC-423 with YOLO	Vehicles, VRU	10 m (either inside or outside the zebra crossing)	5 Hz (20 ms)

Figure 6.14 presents the overall situation as perceived by the several different location providers, and a detailed view of the crossroad where potential accidents might happen, while Figure 6.15 and Figure 6.16 provides a separated view for VRU and vehicle, respectively. The real path/ground truth of the vehicle and VRU location is also presented for comparison.

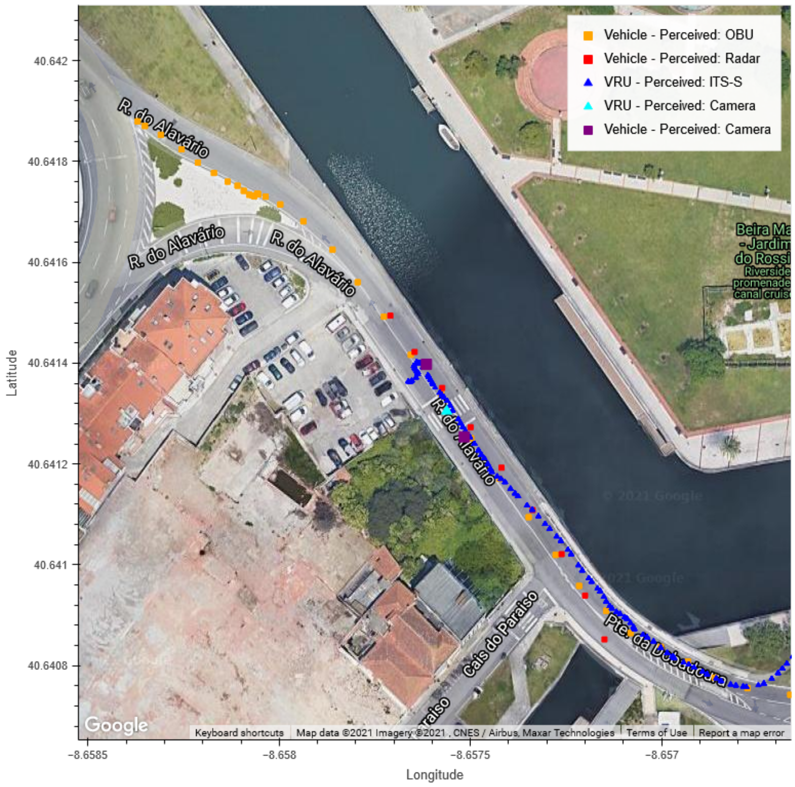
An initial view of the overall results shows that the camera information is not yet precise enough, but can detect and give an approximation of the position - of VRUs and vehicles - and if they are or not in the zebra crossing. Therefore, the camera sensor can be considered a support for a fall-through situation when no other sensors are available. In addition, results seem to indicate that, for a vehicle, both the OBU and Radar have similar accuracy.

To further study the differences in accuracy between the different location providers, the distance between each perceived location and the real position was computed. Figure 6.17 provides the variation of this distance over the time of the test, while Table 6.8 provides the average distance and introduces information on the number of different locations each location provider detected. The considered accuracy parameter - distance between the real location and the location perceived by each source - was computed for each possible location source. In the case of the vehicle, further analysis was done in order to understand the difference in precision when the vehicle is within the field-of-view of the radar.

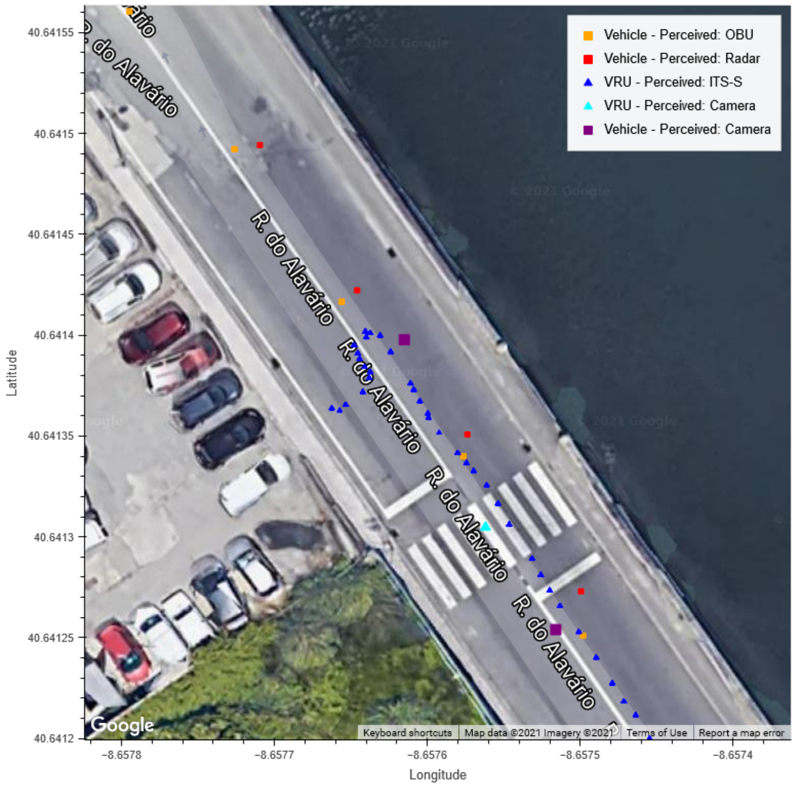
Table 6.8: Accuracy for all location providers of vehicle and VRU

Node	**Data source**	**Average distance to real path**	**Different values**[a]
Vehicle	OBU GPS	0.00271936192202737 m	102
Vehicle	Radar	0.01885715148424999 m	45
Vehicle	Camera	0.04309101130585277 m	2
OBU GPS is 7.7 times more precise than radar			
Vehicle (*)	OBU GPS	0.003148733828667454 m	33
Vehicle (*)	Radar	0.002789311334078819 m	11
Vehicle (*)	Camera	0.012649264281028283 m	2
Radar is 88.6 times more precise than OBU GPS			
VRU	ITS-S	0.002507307896086674 m	837
VRU	Camera	0.045448513053427314 m	1
ITS-S is 5.5 times more precise			

[a] Number of different messages, *i.e.* ignoring messages with repeated information about location.



(a) Overview



(b) Detail of the crossroad to be crossed

Figure 6.14: VRU and vehicle path in test 2

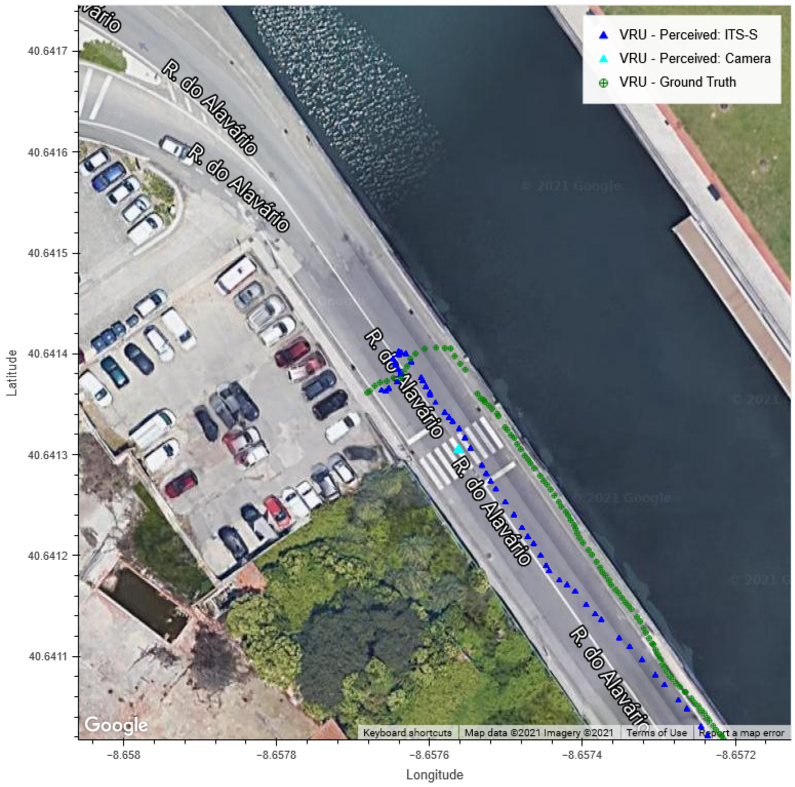


Figure 6.15: Perceived VRU path by each location provider (smartphone and camera)

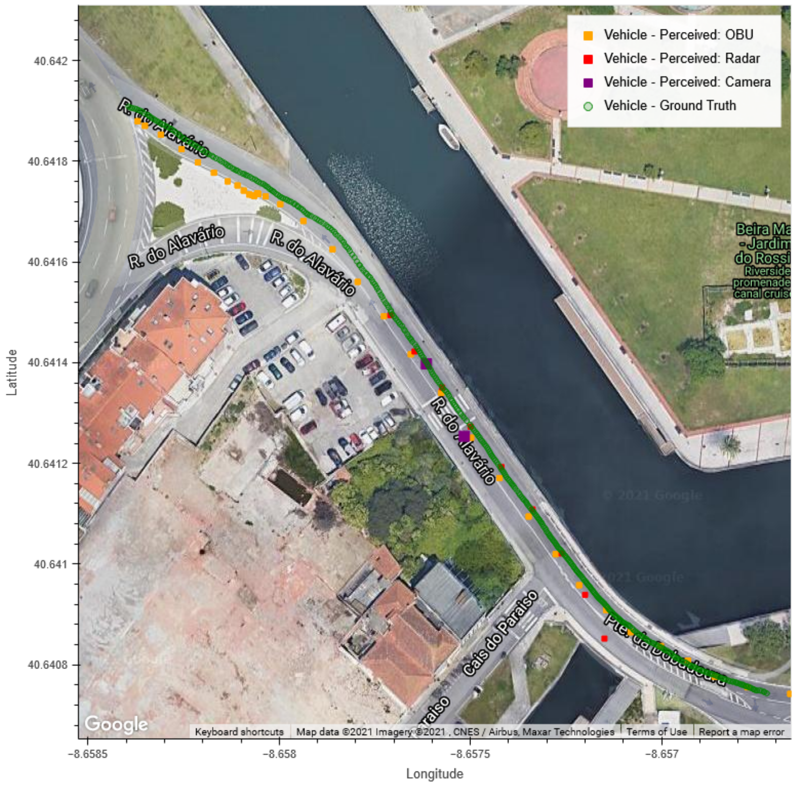


Figure 6.16: Perceived vehicle path by each location provider (OBU, Radar and camera)

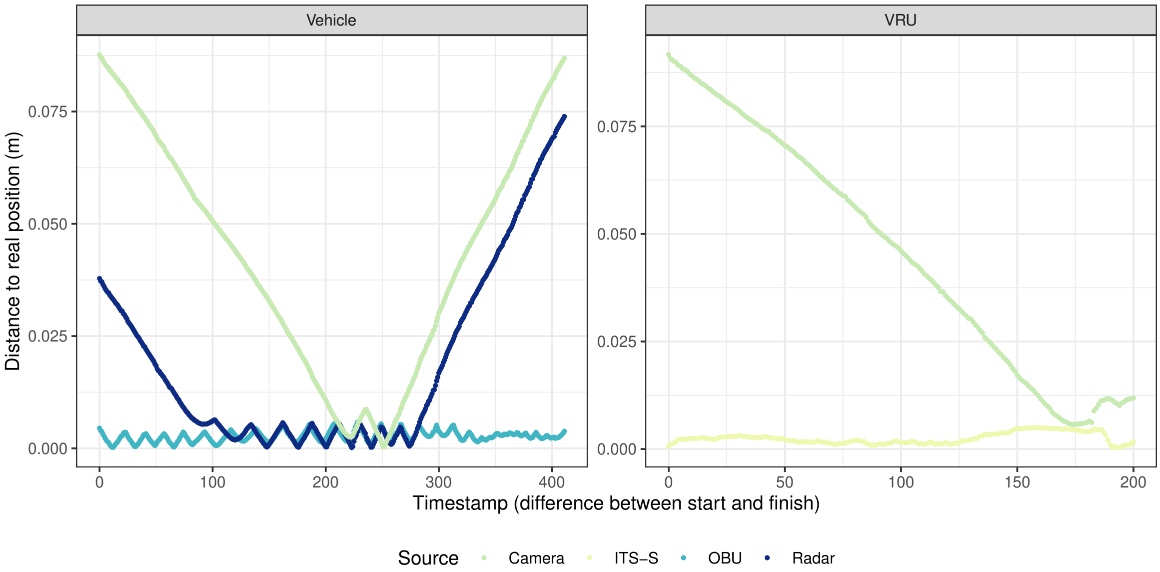


Figure 6.17: Comparison of the difference between perceived and real location (distance) over time, for vehicle and VRU

For the vehicle, the results show that, for the overall situation (*Vehicle*) the OBU location tends to be more precise than the Radar - approximately 7.7 times. However, the Radar is greatly affected by the vehicle being out of its line-of-sight. In these situations, the position given by the radar is a bad estimation.

Situation *Vehicle (*)* describe the overall accuracy without considering the location detections provided by the Radar when the vehicle is out of the direct line of sight. In this situation, it is possible to see the radar is considerably - aproximately 88 times - more accurate than the OBU GPS.

can show a very close approximation with a much higher frequency (the OBU GPS can only measure one time per second).

This means that, the higher precision of the Radar is also only possible within the range of detection of the Radar. By opposition, the OBU can be used in any place where the GPS is available.

The frequency of the data should also be considered. While the OBU GPS provides more distinct locations, the location update frequency is more reduced than the one of the Radar - 1 Hz vs 13.(3) to 17.2 Hz. This is critical in situations - such as collision avoidance - when it is crucial to obtain as much samples as fast as possible of location.

In order to maximize the accuracy of the location of the vehicle, a potential solution is the fusion of data - in the case of the vehicle, between the OBU GPS position and the Radar given position, since it can improve the precision and number of points (more locations per second).

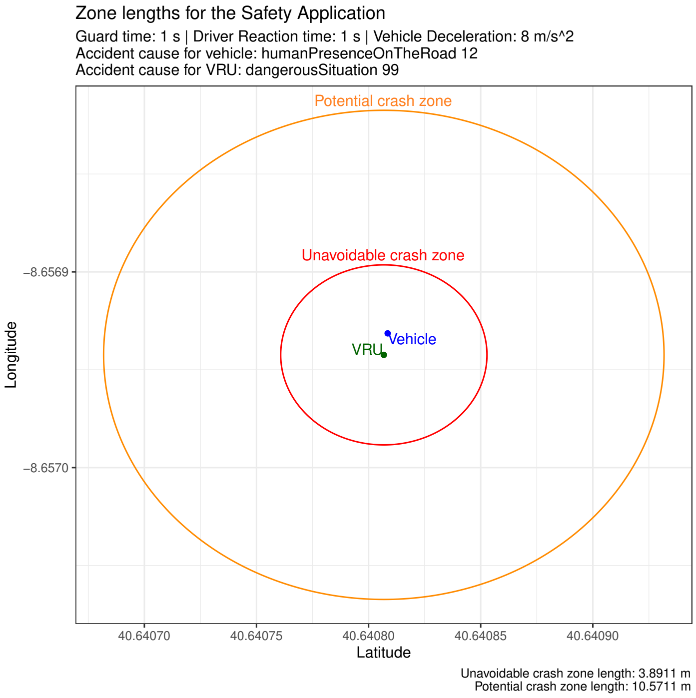
For the VRU, results show that the VRU perceived location is very close to reality, thanks to the usage of the fusion of data - the Fused Location Provider API from Google. This API showcases the potential of sensor fusion by fusing the information from several smartphone sensors, GPS, Wi-Fi, and Bluetooth signal information, improving the accuracy from a simple GPS reading. By opposition, and as expected, the camera detection is very inaccurate.

6.4.3 Zone length and collision points

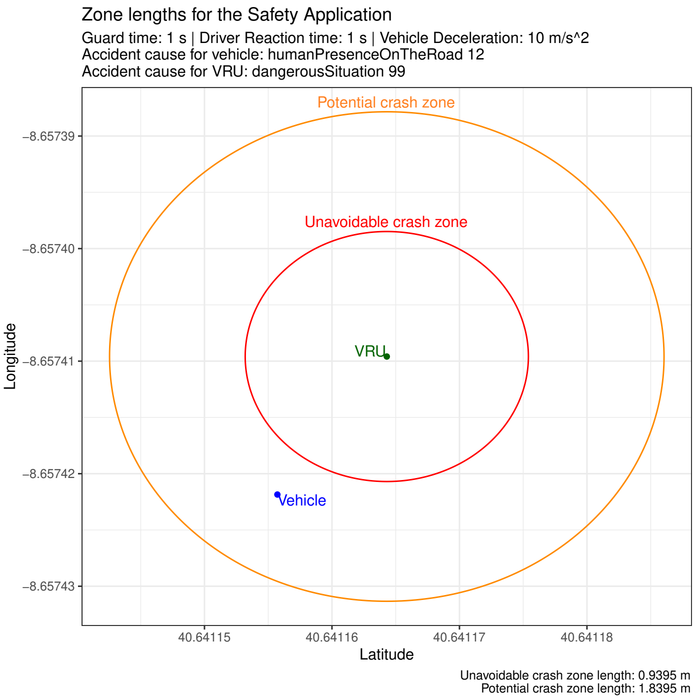
A critical step of the safety algorithm is, based on the dynamics of the vehicle and VRU, the computation of the length of 2 zones - the imminent, unavoidable crash zone and the risk/danger zone. In the imminent, unavoidable crash zone, an accident is unavoidable, and both VRU and the vehicle must be notified. In the danger zone, further analysis is required to determine if a collision is likely to happen or not.

These zones are defined by their length and, by convention, are centered on the VRU location. Centering in the vehicle location would be possible with similar results.

Figure 6.18 shows the zone lengths in examples of real potential collision situations, *i.e.* situations that are very likely to end in collision.



(a) Vehicle is within the unavoidable crash zone



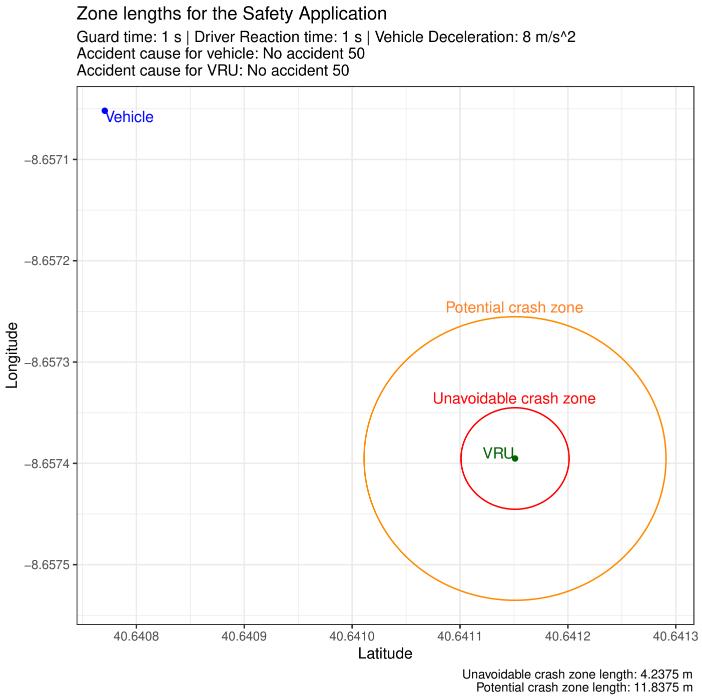
(b) Vehicle is not within the unavoidable crash zone

Figure 6.18: Potential crash and unavoidable crash zones for 2 accident situations in test 2

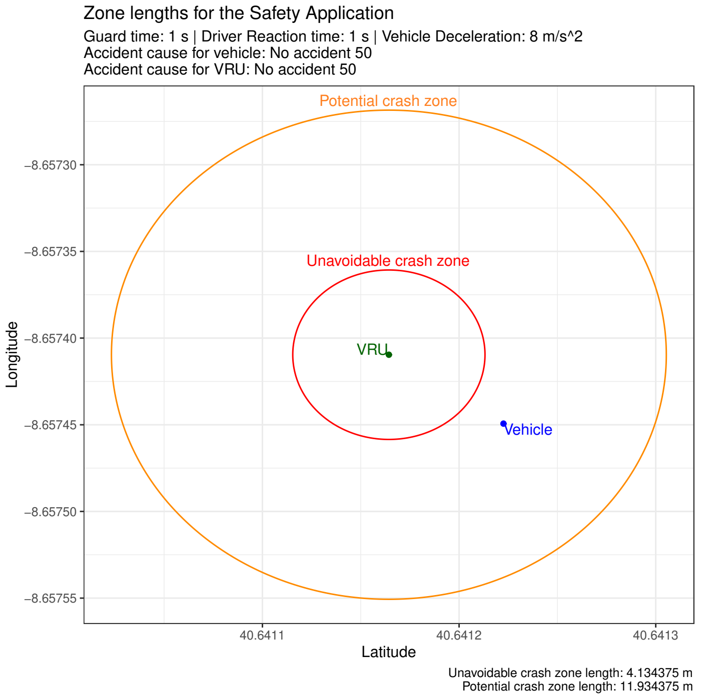
For a potential collision to be detected, one of two disjoint situations must happen. Either the vehicle is within the VRU unavoidable crash zone - as in Figure 6.18a; or the vehicle is not within the unavoidable crash zone, but it is within a potential crash zone, and the further required analysis concludes a collision is likely to happen - as in Figure 6.18b.

By opposition to these unsafe situations, Figure 6.19 shows the zone lengths for safe situations (*i.e.* a situation that is very unlikely to end in collision). If the vehicle is outside the potential crash zone and unavoidable crash zone - as in Figure 6.19a - the situation is safe, *i.e.* no further analysis is required, and a potential collision is not likely to happen.

However, it is also possible to consider a situation safe if the vehicle is within the potential crash zone, but the additional analysis concludes that a collision is not likely to happen - such as in the situation in Figure 6.19b.



(a) Vehicle is outside the potential crash zone

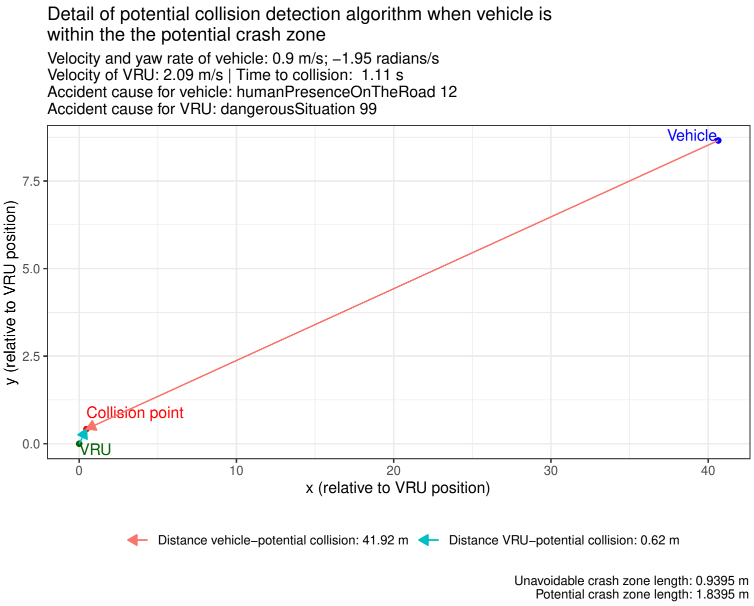


(b) Vehicle is within the potential crash zone

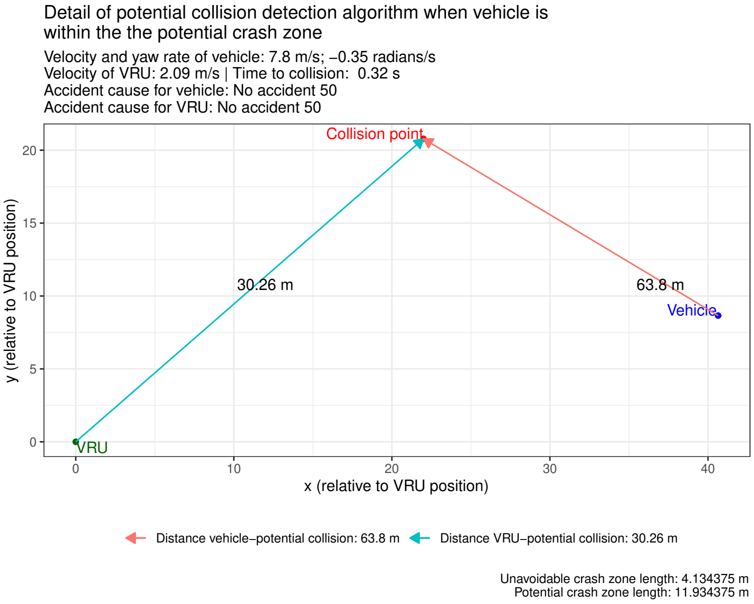
Figure 6.19: Potential crash and unavoidable crash zones for 2 safe situations in test 2

While in both situations in Figure 6.18a and Figure 6.19b, the vehicle is within the potential crash zone, the expected result of the situations is different, with the former being a potential collision and the latter representing a safe situation. This differentiation of outcome when the vehicle is within the potential crash zone is supported by further analysis required to determine if a collision is likely to happen or not.

Figure 6.20 presents an overview of the results of that analysis for safe and unsafe situations. This analysis allows a more thorough examination of the potentially dangerous situations. It starts by computing the point of collision and time to a collision between the nodes based on their dynamics - location and speed of both nodes, dimensions, and yaw rate of the vehicle. Then, based on the VRU current speed and location, it is examined if the VRU is going to achieve the point of collision before the time to collision.



(a) Detailed view of the unsafe situation represented in Figure 6.18a



(b) Detailed view of the safe situation represented in Figure 6.19b

Figure 6.20: Analysis of 2 situations of test 2 where the vehicle is within the potential crash zone

In Figure 6.20a, the obtained point of collision is extremely close to the VRU - 0.62 m - which, having in mind the speed of VRU of 2.09 $m.s^{-1}$ and a time until collision of 1.11 s, it makes extremely likely an accident to happen. Therefore, the algorithm considered this to be a dangerous situation for the VRU, while notifying the vehicle about a dangerous human presence on the road (with codes 99 and 12 on a DENM ETSI standard message).

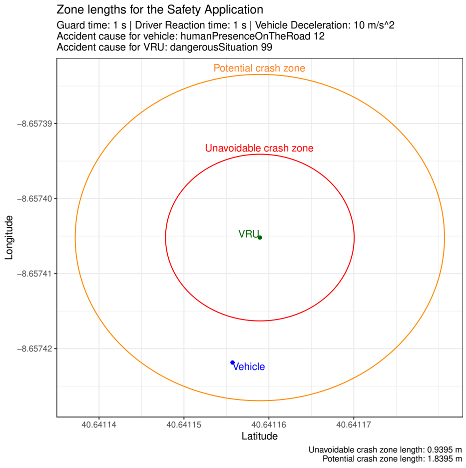
By opposition, in Figure 6.20b, the obtained point of collision is far away from the VRU - 30.26 m - which, having in mind the speed of VRU of 2.09 $m{\cdot}s{-}1$ and a time until collision of 0.32 s, it makes extremely unlikely an accident. Therefore the algorithm considered this to be a safe VRU.

6.4.4 Effect of VRU Actions

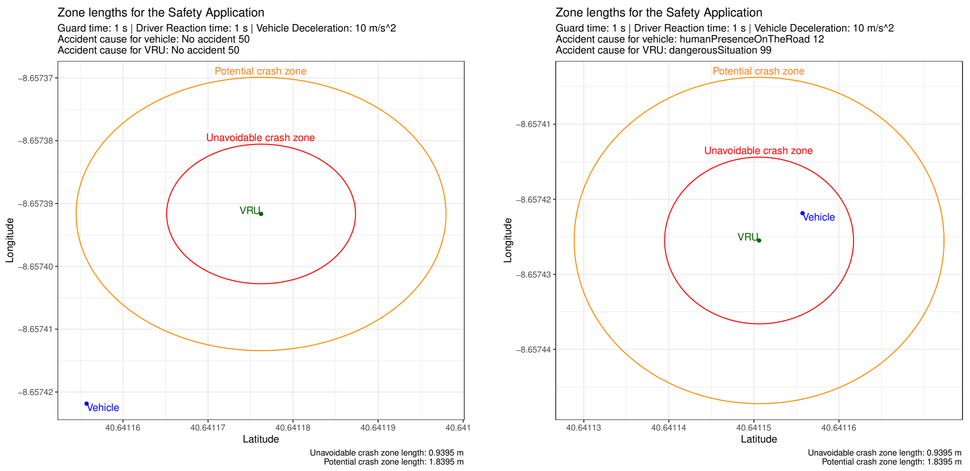
As the algorithm runs, it will warn both VRU and the vehicle about the potential collision. After that notification of the nodes, it is expected that at least one of them will react to avoid the collision.

The most effective reaction tends to come from the VRU. Since it is smaller and more agile than a vehicle, while also requiring less force to change direction or accelerate/ than a vehicle, it can easily avoid the accident by either going forward or backward or eventually even stopping.

Figure 6.21 presents an original situation and the effect that several actions from the VRU in the next interval of time can have in mitigating or not a potential collision with a vehicle.



(a) Original situation



(b) Situation if VRU runs 2 m in the opposite direction

(c) Situation if VRU runs 2 m in the current direction

Figure 6.21: Analysis of the original situation and the effect it has several actions from the VRU in the next time interval

Results show that a proper decision from the VRU can mitigate the risk of collision - as in Figure 6.21b where an accident is no longer detected, or, in opposition, worsen the situation - as in Figure 6.21c where the vehicle is now in the unavoidable crash zone.

6.4.5 Effect of hyper-parameters tuning

The computation of the expected zone lengths and, if necessary, of the point of collision between VRU and vehicle have in mind, not only basic information about the nodes - *i.e.* the location - but also other dynamics of the nodes.

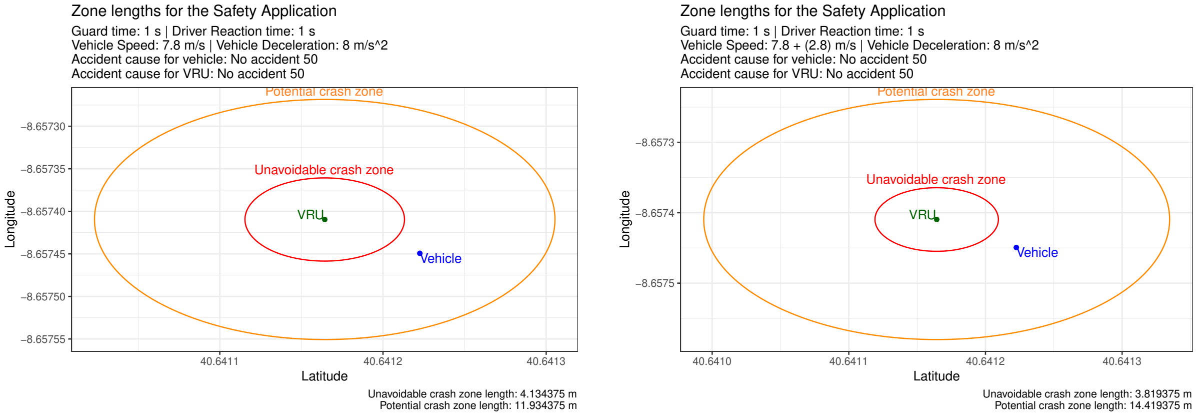
When analyzing the zones lengths and the potential point of collision, it is required to consider, in a vehicle, not only the location, but also the speed and acceleration. Moreover, a vehicle, since it is a heavy body when a dangerous situation is detected, cannot immediately stop. Its maximum deceleration value should be considered. This value depends on the vehicle profile (*e.g.* a light vehicle will tend to have a greater deceleration value than a loaded heavy truck). In addition, the braking process itself does not start immediately after the danger is perceived, with the driver reaction delay time also having to be taken into consideration by the potential collision detection algorithm.

Both the location and the speed information are obtained through the vehicle messages, while the maximum deceleration value and driver reaction delay are parameters statically defined based on the vehicle class and described in Section 5.4.2.

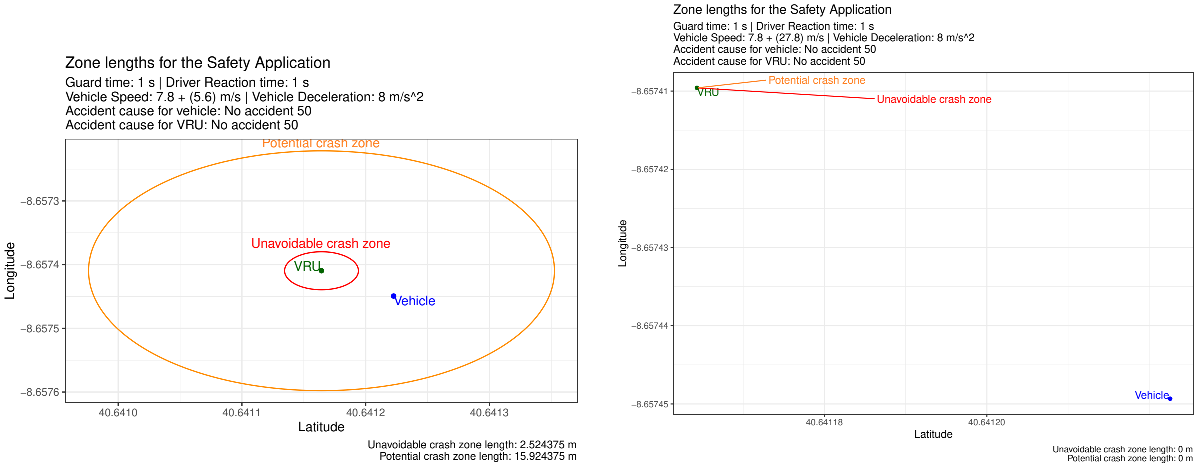
The next sections focus on the impact of varying the vehicle speed and of two hyper-parameters - the driver reaction delay, $t_{reaction}$ and the level of the conservatism of the algorithm, t_{guard}.

Speed

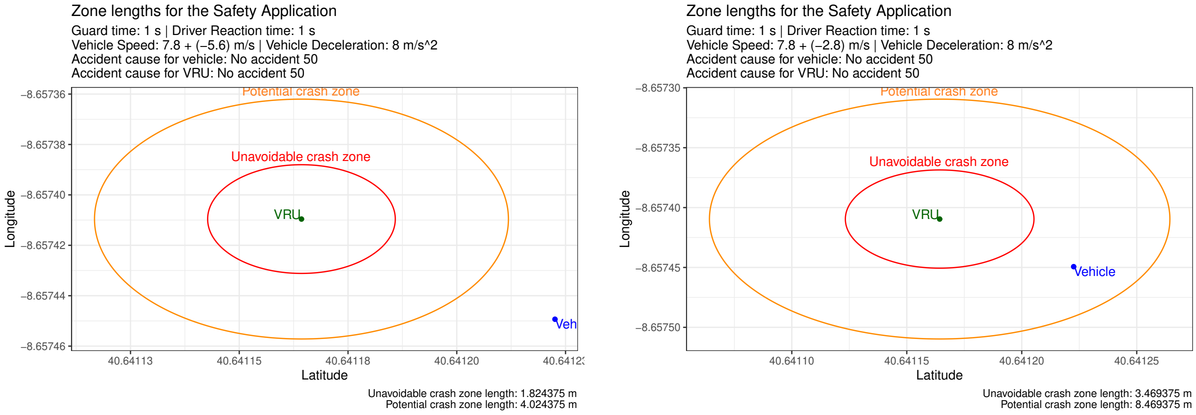
Figure 6.22a presents an initial situation, while Figure 6.22 provides a view of impact of the vehicle speed decreasing or increasing for the exact same situation.



(a) With original vehicle speed $v = 7.8\,m/s$ $(\pm 28.1km/h)$ **(b)** Original speed $+2.8\,m/s$ $(\pm 10km/h)$



(c) Original speed $+5.6\,m/s$ $(\pm 20km/h)$ **(d)** Original speed $+27.8\,m/s$ $(\pm 100km/h)$



(e) Original speed $-5.6\,m/s$ $(\pm 20km/h)$ **(f)** Original speed $-2.8\,m/s$ $(\pm 10km/h)$

Figure 6.22: Analysis of the effect of the variation of speed in the overall situation

In the same situation, the obvious conclusion would be that the higher the vehicle's speed, the unsafer the situation tends to be. However, this is not always the case: increasing the speed can avoid the collision between VRU and vehicle. In Figure 6.22b and Figure 6.22c, the unavoidable crash zone length decreases, *i.e.* it is less likely to happen an unavoidable crash. However, the potential crash zone length increases, which means that the algorithm is also more uncertain about whether a potential collision might happen or not. For an extreme situation, such as the one in Figure 6.22d, a collision is completely impossible: the vehicle's drastic increase in speed means it will quickly pass by the potential collision point with the VRU, therefore rendering impossible a collision. The algorithm expresses this, since the zone lengths are 0 (*i.e.* collision with the VRU is impossible unless the vehicle is in the same position).

Nevertheless, this does not mean that the safer solution is to increase speed. The safer solution seems to decrease the speed, *i.e.* brake a bit, creating the only certainly safe solution (*i.e.* where the vehicle is outside the potentially or certain crash zones), as presented in Figure 6.22e. Reducing just a bit the speed, such as the situation in Figure 6.22f, is not enough to make the situation completely safe, without uncertainty, even if the zone lengths decrease, therefore decreasing the potential for a crash.

Therefore, from these examples, it is not possible to conclude a direct relationship between speed and being in a safe or unsafe situation where a crash is more likely or less likely - this depends on other factors. However, it is possible to understand that extreme situations - either fully braking (pronounced speed reduction) or fully accelerating (pronounced speed increase) tend to produce safer situations in the context of the algorithm. Such conclusion does not mean that either solution is entirely safe; that depends on the full context of the situation (*e.g.* if the road is wet and another vehicle is right in front or behind the vehicle). In addition, and by definition, a pronounced speed increase is never safe.

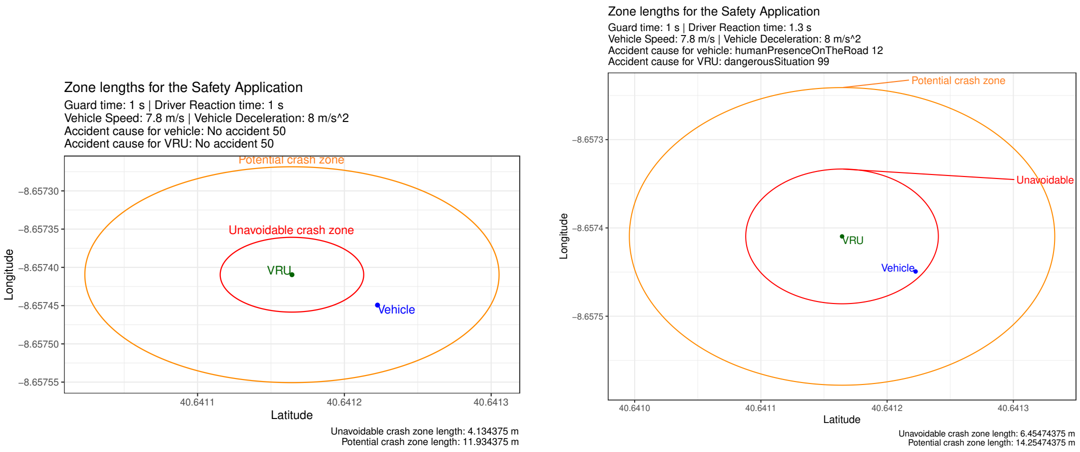
Driver Reaction Delay

Figure 6.23 illustrates the impact for several values of driver reaction delay, $t_{reaction}$.

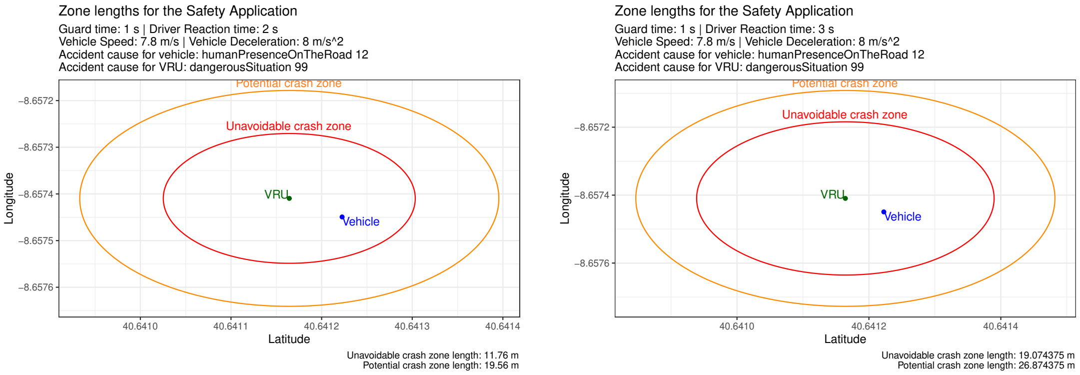
The results show that, even with a smaller increase in the $t_{reaction}$, such as the one presented in Figure 6.23b, it can transform a safe situation into a certain crash. The results are more pronounced in the situations where the $t_{reaction}$ increases to 2, 3 and 5 seconds (as presented in Figure 6.23c, Figure 6.23d and Figure 6.23e), where the situation is more and more unsafe, with an unavoidable crash zone increasing in length and getting closer and closer to the potential crash zone.

By opposition, a decrease in the $t_{reaction}$ to 0.5 s, as presented in Figure 6.23f, has the reverse effect in the situation, decreasing the likelihood of a potential accident, and significantly reducing the likelihood of inevitable accident - with an unavoidable crash zone decreasing from 4.1 to 0.2 m.

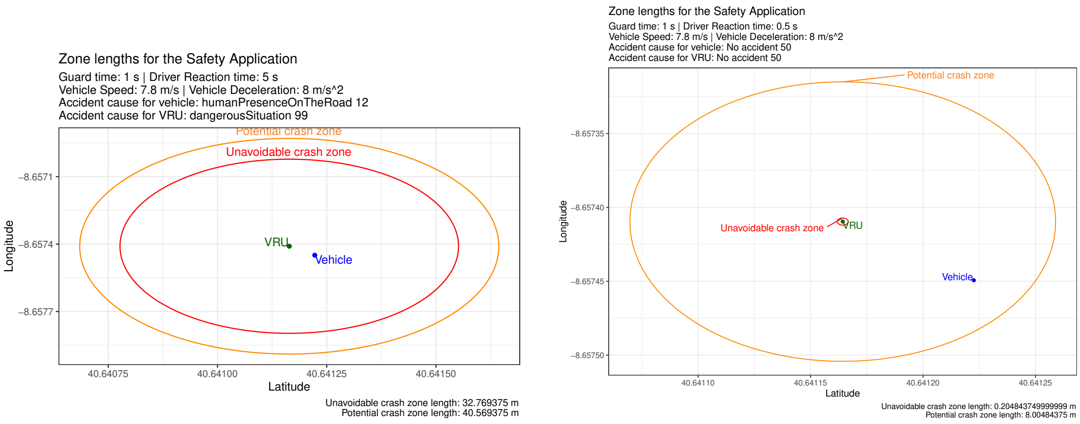
As expected, the $t_{reaction}$ has an inversely proportional relation to the situation being safe or not, with drivers with impaired attention being more likely to provoke an accident. In particular, the results from Figure 6.23b show the importance of knowing if it is night or not to fine-tune the algorithm. Information about being night or not can be supported by the camera information in the current deployment of the system.



(a) With original $t_{reaction} = 1.0s$ **(b)** $t_{reaction} = 1.3s$ (at night)



(c) $t_{reaction} = 2s$ **(d)** $t_{reaction} = 3s$



(e) $t_{reaction} = 5s$ **(f)** $t_{reaction} = 0.5s$

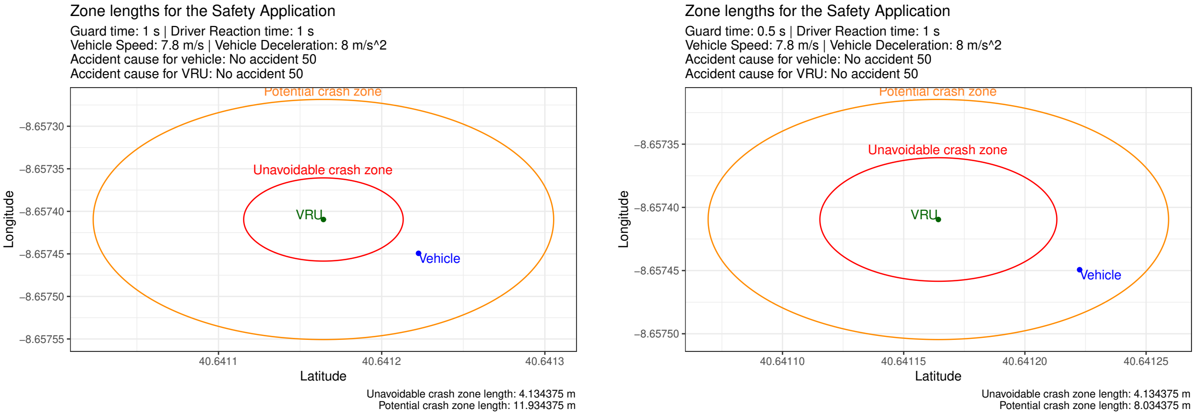
Figure 6.23: Analysis of the effect that $t_{reaction}$ has in the overall situation

Time of guard

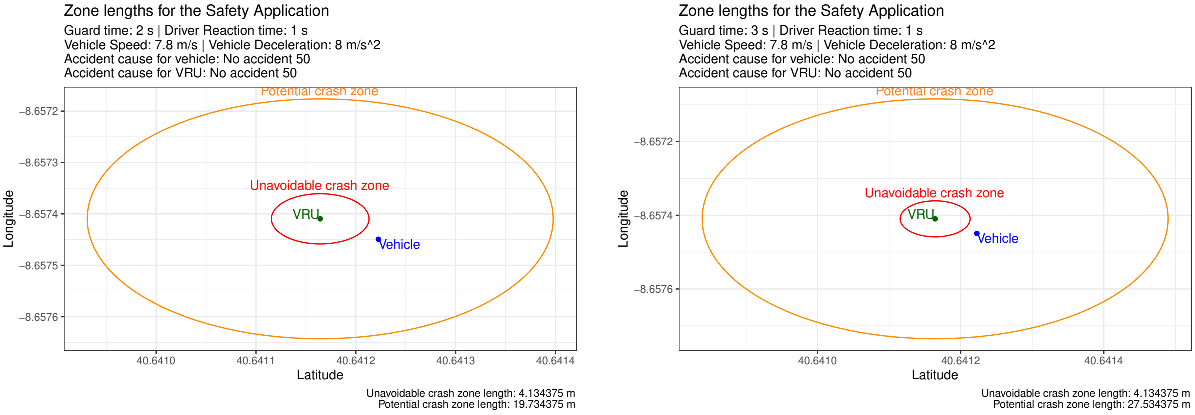
Even considering the dynamics of the nodes, the system might not be susceptible enough for a city manager's requirements. For example, a particular city area might be too dangerous, with the system needing to be more cautious/conservative. By the opposition, a very calm city area where the algorithm can be less conservative/more relaxed.

To tune such behavior, the t_{guard} parameter (defined and described in Section 5.4.2) can be changed to make the algorithm less or more conservative.

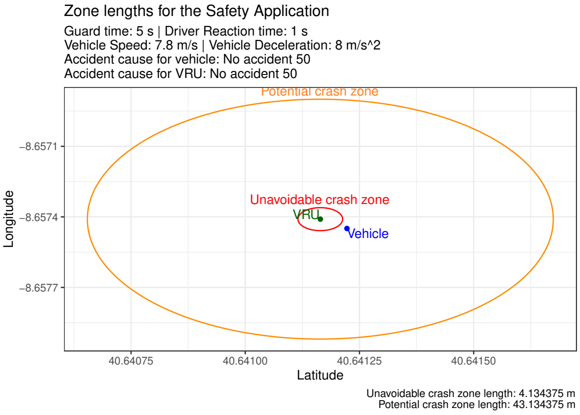
Figure 6.24 presents an initial situation and the different situations obtained for different values of t_{guard}.



(a) With original $t_{guard} = 1.0s$ **(b)** $t_{guard} = 0.5s$



(c) $t_{guard} = 2s$ **(d)** $t_{guard} = 3s$



(e) $t_{guard} = 5s$

Figure 6.24: Analysis of the effect the t_{guard} has in the overall situation

The results show that the t_{guard} does not influence the unavoidable crash zone length, which is static for a determined situation. However, and as expected, it controls the length of the potential crash zone, with decreasing the values of t_{guard} meaning the reduction of the length of the potential crash zone - as it can be seen in Figure 6.24b - and increasing values of t_{guard} also meaning the expansion of the said zone - as it can be seen in Figure 6.24c, Figure 6.24d and Figure 6.24e.

Since the potential crash zone implies a more detailed analysis of a potential accident (*i.e.* uses more information about VRU and vehicle), this effectively means that increasing the t_{guard} renders the algorithm more conservative by analyzing more potential accidents. By opposition, decreasing the t_{guard} renders the algorithm less conservative.

6.5 SUMMARY

This chapter presented and discussed the principal results concerning the timing analysis, scalability analysis and accuracy.

The results show that the system is capable of fulfilling the defined use cases while leaving room for future improvements with the usage of 5G in the VRU ITS-S application, and the full exploration of edge computing in the case of the vehicle information. Timing constraints made essential to consider the edge approach in vehicles and 5G in smartphones. The 5G network is becoming more widespread and requires no changes beyond the need for new compatible equipment.

In the case of edge computing, it is not possible to conclude that an edge deployment is without issues. It also means a more complex system and will introduce the need for orchestration between multiple system deployments split into the different edge points.

Results from scalability tests have shown that the system is capable of scaling within reasonable cases, while also showing the shortcomings of the current deployment, especially in terms of the Safety Application times and Notification times for the VRU ITS-S application.

On the algorithm side, a set of tests allowed to understand the need for fusion of Radar and OBU data for locating the vehicle. A set of situations was also analyzed, proving that the potential collision algorithm is dynamic and its decision can be considerably changed upon the tuning of a set of hyper-parameters. City managers or other entities can change such parameters to tune the algorithm behavior to their needs.

The next chapter will provide conclusions of the overall work, and some remarks about future work that can be considered.

CHAPTER 7

Conclusion

I have not failed.
I've just found 10,000 ways that won't work.
– Thomas A. Edison

Chapter 6 presented and discussed the evaluation of the VRU safety system through results obtained from a set of laboratory and real-life tests. This chapter provides further conclusions on the overall developed system and future improvements that can be considered.

7.1 Conclusions

In this dissertation, a solution to predict potentially dangerous situations - such as collisions - between vehicles and VRUs was developed and evaluated. The infrastructure from the ATCLL project was leveraged to allow gathering in real time data from vehicles and VRUs through different sources, and notify both vehicles and VRUs on these dangerous situations. The system architecture followed a hybrid approach by considering an edge and cloud approach to obtain the data about localization and other dynamics of vehicles and VRUs, in order to compute the length of risk zones and predict a point of collision. In order to prove the interoperability, the system was integrated with vehicles from Bosch.

Keeping in mind the current number of fatal crashes involving VRUs, this solution is crucial. The amount of previous work exploring potential solutions to mitigate these exact problems corroborate that.

The developed solution distinguishes itself from others by the fact it used real infrastructure and aggregated and processed real data from the smart city of Aveiro (within the context of ATCLL project), creating a customizable system that can be used to predict about collisions and notify vehicles and VRUs.

The developed system was developed having in mind both reliability - by ensuring appropriated notification times through a temporal evaluation - and interoperability - since it uses standardized formats, it can easily be deployed and reused by other vendors, as the integration with Bosch CCUs proves.

With the advent of autonomous vehicles that can react to obstacles, this system could be considered redundant. Since autonomous vehicles need to act on the road while avoiding accidentally injuring or killing someone, they are prepared to avoid collisions with VRUs. However, these systems are expensive and only equipped in newer vehicles, leaving behind most vehicles currently on the road. In addition, these solutions, being vehicle-centric, do not provide VRUs with any feedback on the situation, and do not fully explore the potential of smart city infrastructure. Therefore, the developed system can be seen as a complement to autonomous vehicles, to be used in conjunction with their technology, or instead of it when vehicles are older and do not have that equipment. Even for a potential fully autonomous vehicle capable of reacting to an imminent collision with a VRU, this system can be valuable. By warning the driver and VRU before an accident happens, both do not get surprised - and potentially scared - about an emergency braking performed by the autonomous vehicle.

In addition, the developed system can be customized and monitored by smart city managers, providing potentially essential statistics on the number and types of potentially dangerous situations and distribution of them over time. This set of information can support city managers for example in optimizing the traffic management of the city.

Results show the system potential, with the system being capable of full-filling the defined use cases by detecting real situations that did not correspond to an accident but that, if the situation continued, would become a collision between VRU and vehicle. The definition of potential accident is challenging since a simple change in the vehicle, or VRU dynamics might change the overcome - *e.g.* the vehicle might turn, or the VRU might run away. However, real-life tests within the smart city of Aveiro showed the system to be able to detect a potential accident when VRU and vehicle were in collision route. In addition, the system proved to react in an expected way to changes in hyper-parameters (such as driver reaction delay) and dynamics information (*e.g.* vehicle speed).

Finally, results from timing analysis showcased the need for an edge approach in vehicles and usage of 5G or 6G in smartphones. At the same time, scalability tests proved that the system is capable of scaling within reasonable cases.

7.2 Future Work

While the developed system fulfills the defined use cases, more improvements can be considered now or in the future, with the development and enhancement of ITS related technologies:

- **Improvement of the accuracy of the detection of potentially dangerous situations**: The integration of information from other sensors, such as Lidar, would further improve the quality of the potential collision detection. The improvement of the underlying algorithm, possibly with Machine Learning techniques, is now a possibility with the current aggregation of data.

- **Improvement of the system scalability and temporal performance:** The usage of 5G and 6G by VRUs and future improvements in WAVE, C-V2X and alternatives, will mean a further decrease in transport and notification times. In an edge computing approach, deploying the system in several points throughout a smart city can further improve the processing times at the cost of necessary orchestration. Optimizations in the logical layer of the system, for example, by clustering several VRUs within a single message, can also be an impactful optimization in highly dense urban environments.
- **Extension of the system to consider characteristics of other types of VRUs:** Non animal VRUs can't carry a ITS-S but they can be considered as well by the system. Cyclists' and motorcyclists' differences in terms of dynamics - different levels of speed, acceleration, or more ease in abrupt changes of trajectory - shall also be considered. Current projects like Bosch *2Wheelers* will attempt to address those issues.
- **Additional real-life stress-testing:** Having in mind the critically of the system, and even if the system was tested during its development, more stress testing with real-life scenarios is highly advisable to ensure it will always perform as designed.
- **Road safety is more than collision avoidance and detection:** The possibility of defining potentially dangerous situations beyond the ones defined in the algorithm can be critical for a city manager. A dashboard to give city managers control over how to notify VRUs and vehicles and which conditions can trigger potentially dangerous situations (*e.g.*, in a school zone, a certain speed threshold can be considered problematic, even if in other parts of the city is not), can be helpful.
- **Actuation on vehicles:** The developed system's output can be used as an input to a vehicle decision pipeline to support the decision of the path the vehicle will take to a particular destination. Such capability would allow, for example, a vehicle to avoid areas with many VRU - areas where potential accidents are more usual - or, in the limit, support emergency braking scenarios.
- **Orchestration of multiple dangerous situations:** The developed system considered several VRUs and vehicles, and is capable of computing if a potential dangerous situation is imminent, and notify both vehicles and VRUs. In the situation of several VRUs, this means 1 vehicle will receive multiple different warnings that might mean different incompatible actions to mitigate the accidents (*e.g.* in one situation braking, in the other reducing the speed without braking). To fully support a self-driving scenario, the aggregation and orchestration of multiple warnings of dangerous situations must be done, in conjunction with the vehicle's own sensors, in order to produce the best possible action.

References

[1] W. H. O. Social Determinants of Health Team, “Global Status Report on Road Safety 2018: Summary,” World Health Organization, Tech. Rep. 1, 2018, pp. 1–403. [Online]. Available: `https://www.who.int/publications/i/item/9789241565684` (visited on 10/22/2021).

[2] European Commission, “2019 road safety statistics: what is behind the figures? Trends in fatality numbers 2010-2019,” no. June, pp. 1–6, 2020. [Online]. Available: `https://ec.europa.eu/commission/presscorner/detail/en/qanda_20_1004` (visited on 10/21/2021).

[3] Z. Lv, L. Qiao, Q. Wang, and A. K. Singh, “AI-empowered IoT Security for Smart Cities,” *ACM Transactions on Internet Technology*, vol. 21, no. 4, p. 99, 2021. DOI: `10.1145/3406115`. [Online]. Available: `https://doi.org/10.1145/3406115` (visited on 10/22/2021).

[4] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, “Smart cities of the future,” *Eur. Phys. J. Special Topics*, vol. 214, pp. 481–518, 2012. DOI: `10.1140/epjst/e2012-01703-3`.

[5] A. Caragliu, C. Del Bo, and P. Nijkamp, “Smart Cities in Europe,” *Journal of Urban Technology*, vol. 18, no. 2, pp. 65–82, 2011, ISSN: 1466-1853. DOI: `10.1080/10630732.2011.601117`. [Online]. Available: `https://www.tandfonline.com/action/journalInformation?journalCode=cjut20` (visited on 10/22/2021).

[6] K. Darlington, *Fully Autonomous Cars. Will They Become a Reality?* [Online]. Available: `https://www.bbvaopenmind.com/en/technology/innovation/fully-autonomous-cars-how-and-when-will-become-reality/` (visited on 10/23/2021).

[7] J. Bartlett, *Daimler, Waymo, and GM Make Big Gains in Level 4 Self-Driving.* [Online]. Available: `https://mindmatters.ai/2020/11/daimler-waymo-and-gm-make-big-gains-in-level-4-self-driving/` (visited on 10/23/2021).

[8] S. Liu, L. Li, J. Tang, S. Wu, and J. L. Gaudiot, “Creating Autonomous Vehicle Systems,” *Synthesis Lectures on Computer Science*, vol. 6, no. 1, pp. 1–198, 2018. DOI: `10.2200/S00787ED1V01Y201707CSL009`.

[9] ETSI (European Telecommunications Standards Institute), *TS 102 637-1 V1.1.1 (2010-09) - Basic Set of Applications - Part 1 : Functional Requirements*, 2010.

[10] ——, *TR 103 300-1 V2.1.1 (2019-09) - Vulnerable Road Users (VRU) awareness - Part 1: Use Cases Definition*, 2019.

[11] *AveiroOpenLab.* [Online]. Available: `https://aveiro-open-lab.pt/` (visited on 10/21/2021).

[12] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, “A comprehensive survey on vehicular Ad Hoc network,” *Journal of Network and Computer Applications*, vol. 37, no. 1, pp. 380–392, Jan. 2014, ISSN: 10958592. DOI: `10.1016/j.jnca.2013.02.036`. [Online]. Available: `http://dx.doi.org/10.1016/j.jnca.2013.02.036`.

[13] N. Dasanayaka, K. F. Hasan, C. Wang, and Y. 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.

[14] S. Zeadally, J. Guerrero, and J. Contreras, “A tutorial survey on vehicle-to-vehicle communications,” *Telecommunication Systems*, vol. 73, no. 3, pp. 469–489, 2020, ISSN: 15729451. DOI: `10.1007/s11235-019-00639-8`. [Online]. Available: `https://doi.org/10.1007/s11235-019-00639-8`.

[15] D. Flore, "The automotive use-case for 5G," 2017. [Online]. Available: `https://www.3gpp.org/ftp/information/presentations/Presentations_2017/A4Conf010_Dino%20Flore_5GAA_v1.pdf`.

[16] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister, and I. Thibault, "3GPP C-V2X and IEEE 802.11p for Vehicle-to-Vehicle communications in highway platooning scenarios," *Ad Hoc Networks*, vol. 74, pp. 17–29, May 2018, ISSN: 15708705. DOI: `10.1016/j.adhoc.2018.03.004`.

[17] K. Samdanis and T. Taleb, "The Road beyond 5G: A Vision and Insight of the Key Technologies," *IEEE Network*, vol. 34, no. 2, pp. 135–141, Mar. 2020. DOI: `10.1109/MNET.001.1900228`.

[18] C. J. e. a. Bernardos, "European Vision for the 6G Network Ecosystem," *5G IA*, 2021. [Online]. Available: `https://5g-ppp.eu/wp-content/uploads/2021/06/WhitePaper-6G-Europe.pdf` (visited on 11/27/2021).

[19] ITU FG-Net-2030, "Network 2030," *ITU Web Site*, 2020. [Online]. Available: `https://www.itu.int/en/ITU-T/focusgroups/net2030/Documents/White_Paper.pdf` (visited on 10/21/2021).

[20] S. Zeadally, R. Hunt, Y. S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): Status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012, ISSN: 10184864. DOI: `10.1007/s11235-010-9400-5`.

[21] J. Santa, F. Pereñíguez, A. Moragón, and A. F. Skarmeta, "Experimental evaluation of CAM and DENM messaging services in vehicular communications," *Transportation Research Part C: Emerging Technologies*, vol. 46, no. February 2018, pp. 98–120, 2014, ISSN: 0968090X. DOI: `10.1016/j.trc.2014.05.006`.

[22] ETSI (European Telecommunications Standards Institute), *EN 302 637-2 V1.4.1 (2019-04) - Basic Set of Applications - Part 2: Specification of Cooperative Awareness Basic Service*, 2019.

[23] ——, *EN 302 637-3 V1.3.1 (2019-04) - Basic Set of Applications - Part 3: Specifications of Decentralized Environmental Notification Basic Service*, 2019.

[24] A. G. de Figueiredo, "Mobility sensing and V2X communication for emergency services," *MSc Dissertation*, 2021.

[25] ETSI (European Telecommunications Standards Institute), *TS 103 300-3 V2.1.1 (2020-11) - Vulnerable Road Users (VRU) awareness - Part 3: Specification of VRU awareness basic service*, 2020.

[26] 5. A. Association, "Vulnerable Road User Protection," 2020. [Online]. Available: `www.5gaa.org` (visited on 09/28/2021).

[27] J. Tong, N. Nassir, P. Lavieri, M. Sarvi, S. Ryan, S. Harris, and I. Australia, "Putting the Connectivity in C-ITS-Investigating pathways to accelerate the uptake of road safety and efficiency technologies," 2020. [Online]. Available: `https://imoveaustralia.com/news-articles/intelligent-transport-systems/case-for-connected-vehicles/` (visited on 12/05/2021).

[28] ETSI (European Telecommunications Standards Institute), *TS 103 300-2 V2.1.1 (2020-05) - Vulnerable Road Users (VRU) awareness - Part 2: Functional Architecture and Requirements definition*, 2020.

[29] ——, *TR 103 562 V2.1.1 (2019-12) - Basic Set of Applications: Analysis of the Collective Perception Service (CPS)*, 2019.

[30] U. S. D. of Transportation, *J2735E: V2X Communications Message Set Dictionary - SAE International.* [Online]. Available: `https://www.sae.org/standards/content/j2735_202007/` (visited on 10/19/2021).

[31] *SAE J2735 DSRC Message List - SNIP Support.* [Online]. Available: `https://www.use-snip.com/kb/knowledge-base/sae-j2735-dsrc-message-list/` (visited on 10/19/2021).

[32] J. Kocic, N. Jovicic, and V. Drndarevic, "Sensors and Sensor Fusion in Autonomous Vehicles," *2018 26th Telecommunications Forum, TELFOR 2018 - Proceedings*, pp. 2018–2021, 2018. DOI: `10.1109/TELFOR.2018.8612054`.

[33] B. S. Jahromi, T. Tulabandhula, and S. Cetin, "Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles," *Sensors (Switzerland)*, vol. 19, no. 20, pp. 1–23, 2019, ISSN: 14248220. DOI: `10.3390/s19204357`.

[34] J. Casademont, A. Calveras, D. Quinones, M. Navarro, J. Arribas, and M. Catalan-Cid, "Cooperative-intelligent transport systems for vulnerable road users safety," *Proceedings - 2019 International Conference on Future Internet of Things and Cloud, FiCloud 2019*, pp. 141–146, 2019. DOI: `10.1109/FiCloud.2019.00027`.

[35] D. J. Yeong, G. Velasco-hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, pp. 1–37, 2021, ISSN: 14248220. DOI: `10.3390/s21062140`.

[36] W. Liu, S. Muramatsu, and Y. Okubo, "Cooperation of V2I/P2I communication and roadside radar perception for the safety of vulnerable road users," *Proceedings of 2018 16th International Conference on Intelligent Transport System Telecommunications, ITST 2018*, Dec. 2018. DOI: `10.1109/ITST.2018.8566704`.

[37] J. J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi, and J. E. Naranjo, "Vehicle to pedestrian communications for protection of vulnerable road users," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 1037–1042, 2014. DOI: `10.1109/IVS.2014.6856553`.

[38] *Tesla AI Day 2021 Review — Part 1: The Promise of Fully Self-Driving Cars | by Alberto Romero | Sep, 2021 | Towards Data Science.* [Online]. Available: `https://towardsdatascience.com/tesla-ai-day-2021-review-part-1-the-promise-of-fully-self-driving-cars-8e469265509b` (visited on 11/01/2021).

[39] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*, Springer, 2020, pp. 405–421.

[40] F. Guayante, A. Diaz-Ramirez, and P. Mejia-Alvarez, "Detection of vulnerable road users in smart cities," *Proceedings - 2014 8th International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST 2014*, pp. 307–312, Dec. 2014. DOI: `10.1109/NGMAST.2014.60`.

[41] B. Fardi, U. Neubert, N. Giesecke, H. Lietz, and G. Wanielik, "A fusion concept of video and communication data for VRU recognition," in *2008 11th International Conference on Information Fusion*, 2008, pp. 1–8.

[42] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013, ISSN: 15662535. DOI: `10.1016/j.inffus.2011.08.001`.

[43] W. Elmenreich, "A review on system architectures for sensor fusion applications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4761 LNCS, pp. 547–559, 2007, ISSN: 16113349. DOI: `10.1007/978-3-540-75664-4_57`.

[44] P. Lytrivis, G. Thomaidis, and A. Amditis, "Sensor Data Fusion in Automotive Applications," *Sensor and Data Fusion*, no. February, 2009. DOI: `10.5772/6574`.

[45] B. P. L. Lau, S. H. Marakkalage, Y. Zhou, N. U. Hassan, C. Yuen, M. Zhang, and U. X. Tan, "A survey of data fusion in smart city applications," *Information Fusion*, vol. 52, no. May, pp. 357–374, 2019, ISSN: 15662535. DOI: `10.1016/j.inffus.2019.05.004`. arXiv: `1905.11933`.

[46] J. Gomez-Gil, R. Ruiz-Gonzalez, S. Alonso-Garcia, and F. J. Gomez-Gil, "A Kalman filter implementation for precision improvement in Low-Cost GPS positioning of tractors," *Sensors (Switzerland)*, vol. 13, no. 11, pp. 15 307–15 323, 2013, ISSN: 14248220. DOI: `10.3390/s131115307`.

[47] D. Kang and D. Kum, *Camera and Radar Sensor Fusion for Robust Vehicle Localization via Vehicle Part Localization*, 2020. DOI: `10.1109/ACCESS.2020.2985075`.

[48] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database," *Proceedings - 2011 6th International Conference on Pervasive Computing and Applications, ICPCA 2011*, pp. 363–366, 2011. DOI: `10.1109/ICPCA.2011.6106531`.

[49] F. Fredlund and L. Smailji, “Performance comparison of different NoSQL structure orientations,” *MSc Dissertation*, pp. 1–39, 2020.

[50] M. Wang, C. Perera, P. P. Jayaraman, M. Zhang, P. Strazdins, R. K. Shyamsundar, and R. Ranjan, “City data fusion: Sensor data fusion in the internet of things,” *The Internet of Things: Breakthroughs in Research and Practice*, pp. 398–422, 2017. DOI: `10.4018/978-1-5225-1832-7.ch019`. arXiv: `1506.09118`.

[51] F. J. Martinez, C. K. Toh, J. C. Cano, C. T. Calafate, and P. Manzoni, “Emergency services in future intelligent transportation systems based on vehicular communication networks,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 2, pp. 6–20, 2010, ISSN: 1939-1390. DOI: `10.1109/MITS.2010.938166`.

[52] C. Voigtmann, S. L. Lau, and K. David, “Evaluation of a collaborative-based filter technique to proactively detect pedestrians at risk,” *IEEE Vehicular Technology Conference*, 2012. DOI: `10.1109/VTCFALL.2012.6399117`.

[53] O. M. Carsten, “Can road transport informatics help vulnerable road users?” *Proceedings - Society of Automotive Engineers*, no. P-253 pt 2, pp. 1031–1036, 1991. DOI: `10.1109/VNIS.1991.205847`.

[54] A. Ferrara, “Automatic pre-crash collision avoidance in cars,” *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 133–138, 2004. DOI: `10.1109/IVS.2004.1336369`.

[55] D. M. Gavrila, “Cooperative Systems for Vulnerable Road Users: The Concept of the WATCH-OVER Project,” 2006. [Online]. Available: `https://www.researchgate.net/publication/229004941`.

[56] D. Lill, A. Gutjahr, M. Schappacher, and A. Sikora, “Cooperative sensor networks for VRU eSafety,” *Proceedings - 5th International Conference on Systems and Networks Communications, ICSNC 2010*, pp. 52–57, 2010. DOI: `10.1109/ICSNC.2010.14`.

[57] M. Luís, C. Gomes, S. Sargento, J. Ortiz, J. Santa, P. J. Fernández, M. G. Pérez, G. M. Pérez, S. Barmpounakis, N. Alonistioti, J. Cieślak, and H. Gierszal, “Exploring Cloud Virtualization over Vehicular Networks with Mobility Support,” *Connected and Autonomous Vehicles in Smart Cities*, pp. 223–258, Dec. 2020. DOI: `10.1201/9780429329401-8`.

[58] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, “Edge-Based Collision Avoidance for Vehicles and Vulnerable Users: An Architecture Based on MEC,” *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 27–35, Mar. 2020. DOI: `10.1109/MVT.2019.2953770`.

[59] A. Napolitano, G. Cecchetti, F. Giannone, A. L. Ruscelli, F. Civerchia, K. Kondepu, L. Valcarenghi, and P. Castoldi, “Implementation of a MEC-based vulnerable road user warning system,” *2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2019*, Jul. 2019. DOI: `10.23919/EETA.2019.8804497`.

[60] R. Miller and Q. Huang, “An adaptive peer-to-peer collision warning system,” *IEEE Vehicular Technology Conference*, vol. 1, no. April, pp. 317–321, 2002, ISSN: 07400551. DOI: `10.1109/VTC.2002.1002718`.

[61] J. Aghav, P. Hirwe, and M. Nene, “Deep Learning for Real Time Collision Detection and Avoidance,” in *Proceedings of International Conference on Communication, Computing and Networking*, 2017.

[62] F. Basma and H. H. Refai, “Collision avoidance system at intersections,” University of Oklahoma, Tech. Rep. 2216, 2009.

[63] S. B. McLaughlin, J. M. Hankey, T. a. Dingus, and S. G. Klauer, “Development of an FCW algorithm evaluation methodology with evaluation of three alert algorithms,” *National Highway Traffic Safety Administration, DOT HS*, vol. 811, no. June, p. 145, 2009.

[64] A. Tahmasbi-Sarvestani, H. Nourkhiz Mahjoub, Y. P. Fallah, E. Moradi-Pari, and O. Abuchaar, “Implementation and Evaluation of a Cooperative Vehicle-to-Pedestrian Safety Application,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 62–75, 2017, ISSN: 19411197. DOI: `10.1109/MITS.2017.2743201`.

[65] A. Breg, Y. L. Murphey, and T. Yu, “Neural Network Predicting Remote Vehicle Movement with Vehicle-to-Vehicle Data,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 555–560. DOI: `10.1109/SSCI.2018.8628886`.

[66] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook, "Simple and complex activity recognition through smart phones," *Proceedings - 8th International Conference on Intelligent Environments, IE 2012*, pp. 214–221, 2012. DOI: `10.1109/IE.2012.39`.

[67] T. Wang, G. Cardone, A. Corradi, L. Torresani, and A. T. Campbell, "WalkSafe: A Pedestrian Safety App for Mobile Phone Users Who Walk and Talk While Crossing Roads," *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications - HotMobile '12*, DOI: `10.1145/2162081`.

[68] *Activity Recognition API | Google Developers.* [Online]. Available: `https://developers.google.com/location-context/activity-recognition` (visited on 10/22/2021).

[69] S. M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network," *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, pp. 131–134, Mar. 2017. DOI: `10.1109/BIGCOMP.2017.7881728`.

[70] M. Morold, Q. H. Nguyen, M. Bachmann, K. David, and F. Dressler, "Requirements on Delay of VRU Context Detection for Cooperative Collision Avoidance," *IEEE Vehicular Technology Conference*, vol. 2020-Novem, 2020, ISSN: 15502252. DOI: `10.1109/VTC2020-Fall49728.2020.9348627`.

[71] S. Yadav and S. Payandeh, "Understanding Tracking Methodology of Kernelized Correlation Filter," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018*, Institute of Electrical and Electronics Engineers Inc., Jan. 2019, pp. 1330–1336, ISBN: 9781538672662. DOI: `10.1109/IEMCON.2018.8614990`.

[72] A. Milosavljević, D. Rančić, A. Dimitrijević, B. Predić, and V. Mihajlović, "A method for estimating surveillance video georeferences," *ISPRS International Journal of Geo-Information*, vol. 6, no. 7, 2017, ISSN: 22209964. DOI: `10.3390/ijgi6070211`.

[73] smartMicro, *UMRR-11 Type 44 Traffic Management Sensor Product Information.* [Online]. Available: `https://www.smartmicro.com/fileadmin/media/Downloads/Traffic_Radar/Sensor_Data_Sheets__24_GHz_/UMRR_Traffic_Sensor_Type_44_Data_Sheet.pdf` (visited on 09/27/2021).

[74] S. Antipolis, *ETSI - ETSI experts complete specifications for Vulnerable Road Users*, 2020. [Online]. Available: `https://www.etsi.org/newsroom/news/1852-2020-11-etsi-experts-complete-specifications-for-vulnerable-road-users` (visited on 12/17/2020).

[75] P. Choi, J. Gao, N. Ramanathan, M. Mao, S. Xu, C.-C. Boon, S. A. Fahmy, and L.-S. Peh, "A case for leveraging 802.11p for direct phone-to-phone communications," in *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2014, pp. 207–212. DOI: `10.1145/2627369.2627644`.

[76] X. Li, Y. Feng, F. Wang, and Q. Qian, "When Smart Phone Meets Vehicle: A New On-Board Unit Scheme for VANETs," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 1095–1100. DOI: `10.1109/CIT/IUCC/DASC/PICOM.2015.165`.

[77] S.-l. Developers, *sklearn.metrics.pairwise.haversine_distances — scikit-learn 1.0 documentation*, 2021. [Online]. Available: `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.haversine_distances.html` (visited on 10/18/2021).

[78] H. Summala, "Brake Reaction Times and Driver Behavior Analysis," *Transportation Human Factors*, vol. 2, pp. 217–226, 3 Sep. 2000. DOI: `10.1207/STHF0203_2`.

[79] Nationwide Vehicle Contracts, *Understanding Car Size and Dimensions*, 2017. [Online]. Available: `https://www.nationwidevehiclecontracts.co.uk/guides/ask-nvc/understanding-car-size-and-dimensions` (visited on 10/18/2021).

[80] The Council of the European Union, "Council Directive 96/53/EC laying down for certain road vehicles circulating within the Community the maximum authorized dimensions in national and international traffic and the maximum authorized weights in international traffic," *Official Journal of the European Communities*, no. L 235/59, pp. 59–75, 1996. [Online]. Available: `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:31996L0053`.

[81] N. Kudarauskas, "Analysis of emergency braking of a vehicle," *Transport*, vol. 22, no. 3, pp. 154–159, Jan. 2007, ISSN: 16483480. DOI: `10.1080/16484142.2007.9638118`.

[82] P. S. Bokare and A. K. Maurya, “Acceleration-Deceleration Behaviour of Various Vehicle Types,” *Transportation Research Procedia*, vol. 25, pp. 4733–4749, 2017, ISSN: 23521465. DOI: `10.1016/j.trpro.2017.05.486`.

[83] SIMCom, *SIM7600 Series PCIE Hardware Design*. [Online]. Available: `http://mt-system.ru/sites/default/files/documents/sim7600_series_pcie_hardware_design_v1.02.pdf` (visited on 09/27/2021).

Appendix

In this appendix, examples of the information contained vehicular messages CAMs, of the VRU VAMs and of camera and radar information are presented. The information is presented in a dictionary in JSON format.

1. Example of CAM in JSON format

```
{
    "header": {
        "protocolVersion": 2,
        "messageID": 2,
        "stationID": 3
    },
    "cam": {
        "generationDeltaTime": 26673,
        "camParameters": {
            "basicContainer": {
                "stationType": 8,
                "referencePosition": {
                    "latitude": 41.5340871,
                    "longitude": -8.4371093,
                    "positionConfidenceEllipse": {
                        "semiMajorConfidence": 139,
                        "semiMinorConfidence": 139,
                        "semiMajorOrientation": 13.9
                    },
                    "altitude": {
                        "altitudeValue": 159.6,
                        "altitudeConfidence": 13
                    }
                }
            },
            "highFrequencyContainer": {
                "basicVehicleContainerHighFrequency": {
                    "heading": {
                        "headingValue": 0.0,
                        "headingConfidence": 23
                    },
                    "speed": {
                        "speedValue": 0.01,
```

```
                        "speedConfidence": 20
                    },
                    "driveDirection": "UNAVAILABLE",
                    "vehicleLength": {
                        "vehicleLengthValue": 50,
                        "vehicleLengthConfidenceIndication": "noTrailerPresent"
                    },
                    "vehicleWidth": 25,
                    "longitudinalAcceleration": {
                        "longitudinalAccelerationValue": 161,
                        "longitudinalAccelerationConfidence": 102
                    },
                    "curvature": {
                        "curvatureValue": 1023,
                        "curvatureConfidence": "unavailable"
                    },
                    "curvatureCalculationMode": "yawRateNotUsed",
                    "yawRate": {
                        "yawRateValue": 32767,
                        "yawRateConfidence": "unavailable"
                    },
                    "accelerationControl": null,
                    "lanePosition": null,
                    "steeringWheelAngle": null,
                    "lateralAcceleration": null,
                    "verticalAcceleration": null,
                    "performanceClass": 0,
                    "cenDsrcTollingZone": null
                },
                "rsuContainerHighFrequency": null
            },
            "lowFrequencyContainer": {
                "basicVehicleContainerLowFrequency": {
                    "vehicleRole": "default_",
                    "exteriorLights": {
                        "lowBeamHeadlightsOn": false,
                        "highBeamHeadlightsOn": false,
                        "leftTurnSignalOn": false,
                        "rightTurnSignalOn": false,
                        "daytimeRunningLightsOn": false,
                        "reverseLightOn": false,
                        "fogLightOn": false,
                        "parkingLightsOn": false
                    },
                    "pathHistory": []
                }
            },
            "specialVehicleContainer": null
        }
    }
```

```
}
```

Listing 1: Example of the information in a CAM

2. Example of VAM in JSON format

```
{
   "header": {
      "protocolVersion": 1,
      "messageID": 14,
      "stationID": 639779574
   },
   "vam": {
      "generationDeltaTime": 26084,
      "timestamp": 561823901156,
      "sequenceNumber": 53,
      "vamParameters": {
         "stationType": 1,
         "referencePosition": {
            "latitude": 4063469390,
            "longitude": -865977380,
            "positionConfidenceEllipse": {
               "semiMajorConfidence": 0,
               "semiMinorConfidence": 0,
               "semiMajorOrientation": 0
            },
            "altitude": {
               "altitudeValue": 6095176625,
               "altitudeConfidence": 0
            }
         },
         "heading": {
            "headingValue": 12278309888,
            "headingConfidence": 0
         },
         "sizeClass": 4,
         "weightClass": 1,
         "speed": {
            "speedValue": 16568309,
            "speedConfidence": 1
         },
         "direction": 2,
         "orientation": {
            "roll": -292233888,
            "pitch": -8975349,
            "yaw": 142002
         }
      }
   }
}
```

Listing 2: Example of the information in a VAM

3. EXAMPLE OF CAMERA INFORMATION IN JSON FORMAT

```
{
    "id": "urn:ngsi-ld:Total:aveiro_camera:2",
    "type": "Total",
    "altitude": {
        "type": "Number",
        "value": 0,
        "metadata": {}
    },
    "dateObserved": {
        "type": "DateTime",
        "value": "2021-05-26T14:49:37.966Z",
        "metadata": {}
    },
    "detectedPerson": {
        "type": "Boolean",
        "value": True,
        "metadata": {}
    },
    "heading": {
        "type": "Number",
        "value": 0,
        "metadata": {}
    },
    "listOfObjects": {
        "type": "Array",
        "value": [
            {
                "id": {
                    "type": "Integer",
                    "value": 0
                },
                "label": {
                    "type": "Text",
                    "value": "Person"
                },
                "confidence": {
                    "type": "Number",
                    "value": 0.646969
                },
                "bbbox": {
                    "type": "Array",
                    "value": {
                        "topLeft_x": {
                            "type": "Number",
                            "value": 1327
```

```
                        },
                        "topLeft_y": {
                            "type": "Number",
                            "value": 426
                        },
                        "width": {
                            "type": "Number",
                            "value": 22
                        },
                        "height": {
                            "type": "Number",
                            "value": 41
                        }
                    }
                },
                "location": {
                    "type": "geo:json",
                    "value": {
                        "type": "Point",
                        "coordinates": [
                            -8.660141,
                            40.634869
                        ],
                        "crs": {
                            "type": "name",
                            "properties": {
                                "name": "EPSG:4326"
                            }
                        }
                    }
                }
            },
            {
                "id": {
                    "type": "Integer",
                    "value": 0
                },
                "label": {
                    "type": "Text",
                    "value": "Person"
                },
                "confidence": {
                    "type": "Number",
                    "value": 0.629238
                },
                "bbbox": {
                    "type": "Array",
                    "value": {
                        "topLeft_x": {
                            "type": "Number",
```

```
                            "value": 1300
                        },
                        "topLeft_y": {
                            "type": "Number",
                            "value": 425
                        },
                        "width": {
                            "type": "Number",
                            "value": 22
                        },
                        "height": {
                            "type": "Number",
                            "value": 38
                        }
                    }
                },
                "location": {
                    "type": "geo:json",
                    "value": {
                        "type": "Point",
                        "coordinates": [
                            -8.659949,
                            40.635002
                        ],
                        "crs": {
                            "type": "name",
                            "properties": {
                                "name": "EPSG:4326"
                            }
                        }
                    }
                }
            }
        ],
        "metadata": {}
    },
    "location": {
        "type": "geo:json",
        "value": {
            "type": "Point",
            "coordinates": [
                -8.65992939,
                40.634982
            ],
            "crs": {
                "type": "name",
                "properties": {
                    "name": "EPSG:4326"
                }
            }
```

```
        },
        "metadata": {}
    },
    "zoom_level": {
        "type": "Number",
        "value": 0,
        "metadata": {}
    }
}
```

Listing 3: Example of the information in a camera message

4. Example of Radar Information in JSON format

```
{
    "id": "urn:ngsi-ld:Values:aveiro_radar:p3",
    "type": "Values",
    "acceleration": {
        "type": "Number",
        "value": 0,
        "metadata": {}
    },
    "car_id": {
        "type": "Integer",
        "value": 62,
        "metadata": {}
    },
    "class": {
        "type": "Integer",
        "value": 1,
        "metadata": {}
    },
    "dateObserved": {
        "type": "DateTime",
        "value": "2021-05-28T17:36:43.180Z",
        "metadata": {}
    },
    "heading": {
        "type": "Number",
        "value": 52.569,
        "metadata": {}
    },
    "location": {
        "type": "geo:json",
        "value": {
            "type": "Point",
            "coordinates": [
                -8.657148311,
                40.640851872
            ],
            "crs": {
```

```
                "type": "name",
                "properties": {
                    "name": "EPSG:4326"
                }
            }
        },
        "metadata": {}
    },
    "road": {
        "type": "Number",
        "value": 275,
        "metadata": {}
    },
    "road_osmid": {
        "type": "Array",
        "value": [
            41403155,
            41403156
        ],
        "metadata": {}
    },
    "speed": {
        "type": "Number",
        "value": 9.7,
        "metadata": {}
    }
}
```

Listing 4: Example of the information in a radar message