POLITECNICO DI TORINO

Course in ICT for Smart Societies

Master Degree Thesis

# Cellular and DSRC approaches for vehicular traffic safety

**Supervisors**
Prof. Carla Fabiana Chiasserini
Prof. Claudio Ettore Casetti

**Candidate**
Mila Romana Cécile Tabacoff

**Department of Electronics and Telecommunications**

Academic Year 2017 – 2018

# Abstract

Nearly half of the total number of vehicle crashes occurs at road intersections, leading to severe health and economic consequences. Technological innovations in the transport field are playing a significant role in the inversion of this trend, with the purpose of ceasing this type of accidents in the near future. Connected vehicles represent the pillar of vehicular safety applications, and their connection to the surroundings is a key element in the implementation of awareness among the actors of Intelligent Transport Systems networks. In the present day, two main technologies exist for the establishment of the communication in vehicular applications: the cellular technology and the WiFi-based one, and the debate to determine the best suited one is still open. In this thesis, the Long Term Evolution and the Dedicated Short Range Communication are tested singularly and combined for the communication of safety-related messages in a Vehicular Ad-Hoc Network (VANET). Their performances are evaluated in terms of the number of avoided collisions and the reliability of the system is assessed through the study of the false positive detections.

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

The concept of Smart Cities is nowadays drawing attention to a set of new and existing technologies as a mean to solve problems and enhance the everyday life. This process is engaging several fields, from energy to home automation, health and transports. In particular, the transport field presents a large margin for technological improvements: traffic efficiency, electronic payments, infotainment on board and safety are only some of the new applications that transport systems are experiencing. All the innovation that this area is undergoing can be summarized by the concept of Intelligent Transport Systems (ITS). Safety applications in vehicular networks are one of the fields that is receiving most interest and focus, and their impact could play a striking role in today's society. In fact, road crashes cause each year the death of more than 1.3 millions of people, with an average of over 3000 deaths a day [1]. Moreover, almost half of the car accidents take place at intersections. The aftermath of these vehicle crashes not only has catastrophic impacts on health and human lives, but also presents non-negligible economical consequences. The United Nations Decade of Action for Road Safety [2] states that the car accidents economic repercussions represent from 1% to 3% of the countries GNP, amount of resources that could be destined for more productive uses. In fact, the costs range from property damage, to workplace productivity losses, traffic congestions and medical costs [3]. For these reasons, governments were called to establish research programs and initiatives with the purpose of drastically reducing the casualties caused by such accidents. Approaches like “Vision Zero” in the European Union (Sweden Traffic Safety) and “Road to Zero” by the United States National Security Council targeted the end of roadway fatalities by 2050. Furthermore, several automotive manufacturers decided to orient part of their research to connected vehicle technologies and Advanced Driver Assistance Systems (ADAS) through the adoption of sensing technologies and algorithms. This research led to the appearance of the first ITS services and autonomous vehicles testings debuted (e.g. Google Self-Driving Car, Uber, etc.).

Connected vehicle technologies represent one of the pillars for the pursuit of the aforesaid objectives, giving vehicles the possibility to communicate and exchange safety-related data inside Vehicular Ad-Hoc Networks (VANETs). In fact through the Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, VANETs allow to connect the vehicle to its surroundings. The latter becomes an entity comparable to smartphones, and can build cooperative systems capable of exchanging useful data such as position, speed and direction. These informations can be exploited by algorithms to detect anomalies in the mobility and prevent collisions. One of the major applications in the ITS environment is the Collision Detection, that exploits the data retrieved by cooperative systems in order to predict possible accidents among vehicles.

Two main types of technologies for the vehicular communication exist: the first approach is based on WiFi communication exploiting the DSRC standard, while the second one adopts the cellular network using the Long Term Evolution (LTE) standard. The performances of the two approaches are widely studied and analysed, and opened a debate on the best suited protocol for safety applications in ITS. In fact, VANETs present some communication challenges that the adopted system should satisfy. In particular, latencies represent a crucial aspect for safety applications [4], and as a matter of fact the V2X communication should not exceed 100 ms when dealing with collision warnings, while it can reach up to 1s for less critical situations. Furthermore, the communication needs to face the mobility issues of the nodes, that are characterized by a significant speed with respect to handheld-like devices of the classical cellular networks. The topology in VANETs is subject to constant changes, and both the speed and events as handover should be managed reliably and in an efficient way.

The purpose of this thesis work was to implement different scenarios in order to evaluate the performances of an Intersection Collision Avoidance (ICA) application with respect to the use of different communication protocols and architectures. The Cooperative Awareness Messages (CAMs) have been used in order to inform the environment of the state of each vehicle and have been exploited by the Collision Detection algorithm to predict collisions and hazards between pairs of vehicles. The location of this algorithm has been tested both in a centralized scenario, in which only the server has a computational capability, and in a distributed one, where all the vehicles were provided with such algorithm. Both the DSRC and the cellular approaches have been adopted singularly and combined for the transmission of CAM and Alert Messages using V2V and V2I communications. The aim was therefore to extract some statistics on the number of collisions that can be avoided using cellular and WiFi technologies, as well as obtaining measures of the system's reliability through the evaluation of the number of false positive hazards obtained. Performances are therefore obtained based on the parameters that characterize the safety application, giving a more direct understanding of the consequences of the protocols and architectures adopted on the efficiency of the system.

The outline that this thesis will follow is the following: the Chapter 2 covers the existing solutions for the V2X communication, stressing both the characteristics of cellular and WiFi-like approaches. Furthermore, a general review of related research in the literature is provided. The Chapter 3 focuses on the description of the personal contribution to the topic, presenting the characteristics of the developed applications as well as the adopted simulation tools. In the Chapter 4, the simulations settings are presented, and an assessment of the results that have been obtained is given. Finally, the Chapter 5 closes drawing conclusions on the work and delineating some future developments.

# Chapter 2

# State of the art

## 2.1 Standards

In the last few years, the huge development of Information and Communication Technologies (ICT) gave birth to a period of great innovation, introducing the possibility to enhance the everyday life on a wide range of areas. The introduction of the Smart Cities concept has drawn attention to the use of the ICT in fields that were not previously considered of strict interest, such as home automation, health and transport systems.

In particular, the innovation experienced by cellular networks with the appearance of new technologies such as the Long Term Evolution (LTE), the LTE-Advanced (4G) and more recently the 5G set a milestone for the development of vehicular communication and more in general vehicular safety. This evolution laid the ground for the notion of Intelligent Transport Systems (ITS), improving both safety and efficiency of the mobility. One of the key aspects of vehicular safety stands in the Vehicle-to-Vehicle (V2V) communication and in a more general way V2X communication. But these interactions can be extended beyond safety applications, allowing the exchange of data for information purposes, payments and entertainment during the journey. Vehicles can be therefore considered as smart objects capable of delivering services, connected to their environment as much as smartphones. Nowadays, two main approaches have been adopted for the communication in V2V scenarios: Dedicated Short-Range Communications and Cellular networks.

### 2.1.1 Long Term Evolution

LTE (Long Term Evolution), also called E-UTRAN (Evolved Universal Terrestrial Access Network), was developed and first released by the 3rd Generation Partnership Program (3GPP) in 2008. Its creation was conceived as an evolution of the GSM (Global System for Mobile Communications) and UMTS (Universal Mobile Telecommunication System) cellular technologies. The GSM technology based its architecture on the circuit switching method, where the calling party was connected to the called party through the establishment of a circuit across the whole telecommunications network. This type of connection was used to carry both real time (e.g. telephony) and data (e.g. SMS) services. When the GSM communication made way for the GPRS and UMTS technologies, the packet switching technique was introduced and added to the circuit switching, dividing the core network in two domains. In this case, telephony and messages are still transmitted through circuits, but data is now transferred in packets through the use of dedicated circuits. Developed as an evolution of the previous techniques, the LTE was the first system to only use packet switching for both data and voice services. The Evolved Packet Core (EPC) has been defined as the new core network used by the LTE system, solely based on the IP transport protocol. Therefore, no circuit-switched domain is necessary anymore. This evolution of the packet-switched architecture allowed to reach much higher data rates with respect to the previous systems, as well as a higher spectral efficiency.

### Architecture

The LTE access network is composed as a network of base stations called evolved NodeB (eNB). This architecture is characterized by the absence of a central controller, thus creating a flat structure as opposed to its predecessors. The adoption of a distributed solution is fundamental for what concerns the reduction of the setup time as well as the reduction of the time required for the handover, the process of transferring an ongoing call or data session from the control of an eNB to another. A further positive effect of the absence of a centralized controller is reflected on the MAC layer, responsible for the scheduling process. In fact, since the MAC is only present in the eNB and in the User Equipment (UE), scheduling can be done in a quicker manner leading to faster decisions thus faster communication. These enhancements can have crucial effects in vehicular networks, in particular for delay sensitive applications as safety and gaming.

### Interfaces

Several types of air communication interfaces are available in LTE, allowing different types of transmissions to take place. In fact, both Vehicle-to-Interface/Network (V2I/V2N) and Device-to-Device (D2D) communication are implemented, increasing the standard's flexibility and introducing the possibility to realize not only a V2V communication but also a Vehicle-to-Pedestrian (V2P) one by establishing a connection with a mobile device carried by an individual. To carry out such functionalities, the interfaces shown in Fig. 2.1 have been created:

- Uu: interface between the UE and the E-UTRAN (or eNB), used for conventional cellular traffic via the Uplink and Downlink;
- PC5: sidelink interface between two or more entities. In fact, it can be considered as a one-to-many communication interface, adopted for group communication;
- PC3: interface used for the communication between the UE and the ProSe Function, whose role is to provide informations for network-related actions, such as service authorization or Public Land Mobile Network (PLMN) informations acquisition. These exchanges of data adopt HTTP as transport protocol;
- S1: interface adopted between the eNB and the EPC, the core network of LTE, and can be considered as a many-to-many interface;
- PC4: interface between the EPC and the ProSe function.

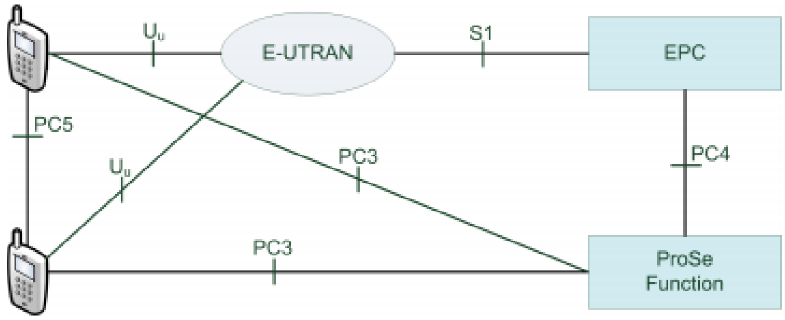


Fig. 2.1 LTE ProSe Architecture [5]

### Operation Scenarios for V2X

The described architecture and interfaces allow LTE to work in different operation scenarios for what concerns the V2X communication. In fact, depending on the interfaces that are being adopted, vehicles can decide to communicate directly or with the intervention of the eNB. Three main scenarios can be identified, and will described in their V2V version, even though the communicating nodes could also be pedestrians, infrastructure-like nodes or network devices.

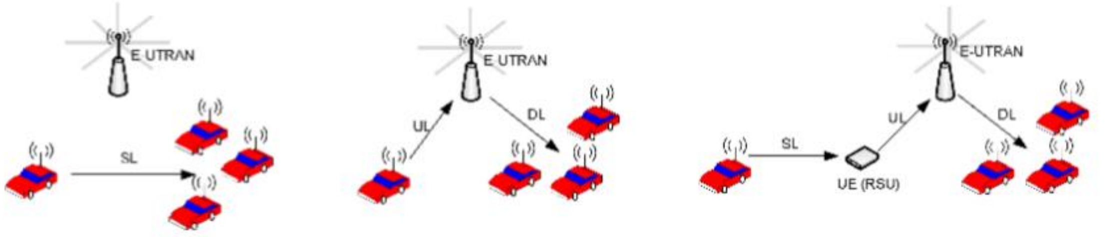


Fig. 2.2 LTE Operation Scenarios [6]

**Scenario 1** The first scenario that can be identified supports V2X operations based solely on the PC5 interface, thus the one connecting two or more different entities, called sidelink. In this case, a UE sends a V2X message to a group of UEs in a local area through the sidelink.

**Scenario 2** As opposed to the scenario 1, this scenario considers the communication to happen only on the Uu interface, thus the one connecting the UE to the eNB. In fact, for both V2V and V2P circumstances, the UE transmits in uplink a message to the eNB, that takes care of sending the message to the destination UEs in local area via downlink. This same situation can consider pedestrian UEs as transmitter or receiver nodes (V2P operation), as well as a eNB type RSU (V2I operation) or an application server (V2N operation). This particular scenario is the one that will be used in this work when adopting the LTE communication.

**Scenario 3** This last scenario is built as a combination of the previously described ones, in fact the communication is here based on both the Uu and PC5 interfaces. A vehicle here transmits a message in sidelink to UEs, one of which is a UE type RSU that will forward the message to the eNB in uplink. In due course, the eNB will use the downlink to transmit the message to one or multiple UEs, and may therefore choose to use broadcast. The actions just described can also be followed backwards, where a UE starts the communication by sending a message in uplink to the eNB, that transmits it to a UE type RSU that will then take care of broadcasting the message to the end destinations via sidelink.

These particular qualities lead to considering LTE as a possible solution for safety critical communications in the ITS field. The 3GPP LTE's evolutions lead to the current version denoted as release 14, that analyzes the "LTE enhancements and corresponding evaluations for LTE-based V2X services" [6].

### 2.1.2 IEEE 802.11p

The IEEE 802.11p standard is the only Wi-Fi protocol that has been fully developed and tested for communication in vehicular networks. This version, published in 2010, supplies in fact all the enhancements to the IEEE 802.11 in order to support ITS applications. This specific standard is part of the Wireless Access Vehicular Environment (WAVE - IEEE 1609) communication protocol stack, also known as Dedicated Short Range Communication (DSRC). In fact, the 802.11p defines the physical transmission (PHY) and medium access protocol (MAC) layers of WAVE. A second standardization has been pursued in Europe in parallel to the one analyzed in this section, the ETSI ITS-G5. This version relies on a very similar standard and will be described in the next section.

| Features | IEEE 802.11p | LTE-V2V |
|---|---|---|
| Release | 2010 | 2016 |
| Improvements | No plans | Ongoing activity |
| Experimentations | Large scale testbed | Tests planned |
| V2I support | RSUs | eNodeB |
| Radio resources | CSMA/CA | SC-FDMA |
| Time synchronization | Not required | Required (GNSS) |

Table 2.1 IEEE 802.11p and LTE-V2V Main Features [7]

#### Physical Layer

The physical layer of IEEE 802.11p is based on the Wi-Fi version IEEE 802.11a, appropriately modified in order to meet the ITS applications requirements. In fact, both of the versions operate at a frequency of 5.9 GHz, in particular in the range between 5.850 and 5.925 GHz. The modulation adopted by the IEEE 802.11p is the Orthogonal Frequency-Division Multiplexing (OFDM), composed of orthogonal sub-carriers closely spaced in order to support parallel transmission streams. Even though the sidebands of each carrier overlap, it is possible on the receiver side to demodulate the signal without interferences as a consequence of the orthogonal property of such a modulation. This same modulation was used in the IEEE 802.11a, with the only difference that the considered version exploits the half-clocked mode with a bandwidth of 10 MHz per channel, half compared to the 11a. The symbol duration is therefore doubled with respect to the original version, allowing to deal with the Doppler effect created by the movement and speed of vehicles. This change allows to avoid

| | IEEE 802.11a | IEEE 802.11p |
|---|---|---|
| Data Rate (Mbps) | 6, 9, 12, 18, 24,<br>36, 48, 54 | 3, 4.5, 6, 9, 12,<br>18, 24, 27 |
| Modulation | BPSK , QPSK<br>16-QAM , 64-QAM | BPSK , QPSK<br>16-QAM , 64-QAM |
| ODFM Symbol Duration | 4.0 $\mu$s | 8.0 $\mu$s |
| Guard Period | 0.8 $\mu$s | 1.6 $\mu$s |
| Occupied Bandwidth | 20 MHz | 10 MHz |
| Frequency | 5 GHz ISM band | 5.850-5.925 GHz (dedicated) |

Table 2.2 Comparison IEEE 802.11a vs IEEE 802.11p

Inter Symbol Interference thanks to the presence of longer guard intervals (1.6 s) as well as dealing with Multipath effects. The signals encoded in each sub-carrier can use different modulations, as BPSK, QPSK, 16-QAM, 64-QAM, and with different bit rates (3, 4.5, 6, 9, 12, 18, 24, 27 Mbps). The choice among different modulation schemes gives the possibility to adapt the settings to the conditions of the channel using the Adaptive Modulation technique. Modulation, data rate and transmission power can be therefore selected in the most efficient way. The 75 MHz band used by this protocol is divided into 7 channels with 10 MHz width each: a central one known as Control Channel (CCH) and 6 more channels known as Service Channels (SCH) on the sides. The first one is used solely for security communications purposes, while the other ones can be used to transmit all informations with lower priority. Furthermore, Service Channels can be combined in order to increase the bit rate of the transmission. In fact, choosing a 64-QAM modulation characterized by a bit rate of 27 Mbps and using two SCH in parallel, the same performances of IEEE 802.11a can be reached with a rate of 54 Mbps.

### Lower MAC Layer

Following the same line as the Physical layer, also the MAC layer of IEEE 802.11p was inspired by previous versions, and adapted to meet vehicular safety requirements. The lower part of the MAC layer adopts the method of a contention-based channel access known as Enhanced Distributed Channel Access (EDCA). This protocol was derived from the IEEE 802.11e, and represents an enhanced version of the Distributed Coordination Function (DCF). In particular, EDCA defines the rules for what concerns the access to a channel and uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [8]. This mechanism aims at reducing packet collisions due to simultaneous transmissions over a given channel as a consequence of the half duplex property of nodes in WiFi networks. In order to do so, a node ready to transmit senses the medium and if the latter is free for an amount of time equal to an AIFS (Arbitration Inter-Frame Space), a random backoff time is extracted and after its expiration data can be sent. The MAC protocol of IEEE 802.11p also accounts for the priority of messages, ensuring that relevant and safety-related data can be exchanged promptly and reliably. Different messages and traffic categories are therefore classified into four Access Classes characterized by different priorities. The concept of Basic Service Set (BSS) defined in IEEE 802.11e that represents a group of nodes connected to an Access Point (AP) also had to be reassessed for the vehicular network scenario. In fact, in IEEE 802.11p a new type of BSS is created: the WAVE BSS (WBSS). In this scenario, one node will initiate the communication, broadcasting periodically a beacon, the Announcement Message, notifying the existence of a BSS. This message contains all the operational informations needed by other nodes to join the WBSS, e.g. which SCH can be used. Using the message data, other nodes may decide to join the WBSS, without any authentication required.

### 2.1.3 Dedicated Short Range Communication

DSRC is a short-range communication capability created by the US Department of Transportation (USDOT) with the purpose of supporting technologies enabling a reliable exchange of data among vehicles and road infrastructure. In fact, the communication implemented by DSRC is the only one that adopts a dedicated frequency band for safety-related applications in vehicular environments. Furthermore, the needs of such kinds of applications are satisfied by the property of DSRC to provide an immediate establishment of communication as well as low latencies in transmission. The reliability of the communication is assured despite the high speed at which vehicles operate and the extreme weather conditions that could interfere with the signal propagation. As previously explained, the IEEE 802.11p defines only the PHY and MAC layers of the DSRC protocol stack for vehicular communication, while the upper layers are defined by the IEEE 1609 family of standards, commonly referred to as WAVE.

#### Upper MAC Layer

The extension of the MAC layer covered by the IEEE 802.11p is defined by the IEEE 1609.4 standard, responsible of the multi-channel coordination. In fact, while the CCH only uses one channel, the SCH adopted for infotainment messages can rely on six different channels. Thus a single radio device can use different channels when alternating between the 50 ms dedicated to the CCH and the 50 ms assigned to the SCH transmission, with the possibility

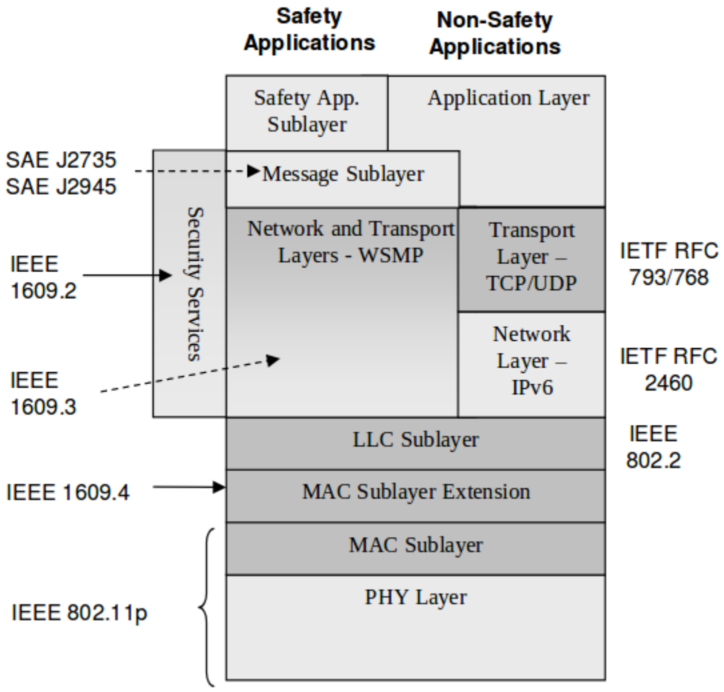


Fig. 2.3 DSRC Protocol Stack [9]

to choose the least congested one. The 100 ms time interval is known as Sync Interval. In order to be able to switch from one channel to another, nodes use the Universal Time Coordinated (UTC) GPS time. At the beginning of each UTC second, divided in 10 Synch Intervals, synchronization is performed thanks to GPS so to allow nodes to switch channel. A precaution used in this case to avoid bad synchronization consequences is the use of guard intervals with the purpose of absorbing possible delays. Furthermore, a transmitting node is required to include its time reference in the message, that if incorrect revokes its ability to transmit on the CCH. The multi-channel coordination technique adopted by the IEEE 1609.4 presents some drawbacks related to the Sync Intervals structure. In fact, possible latencies are introduced if an hazardous event takes place during the SCH interval. requiring to wait for the beginning of the next CCH interval. Moreover, in this same situation, collisions and losses of packets will be very probable due to the competition of all nodes to win the CCH interval for transmission. Such inconveniences classify a contention based system as not the best solution for vehicular safety purposes, but a possible solution implemented by ETSI will be described in the next section.

### WAVE Short Message Protocol

In the DSRC protocol stack, the network and transport layers are differentiated in order to meet both the requirements of standard communication and vehicular safety communication. In fact, the default protocols used in the majority of networks are part of the DSRC stack: the IP (Internet Protocol) for the network layer and the TCP (Transport Control Protocol) and UDP (User Datagram Protocol) for what concerns the transport layer. The adoption or these protocols is used to process all messages that are non-safety related. The communication in V2X scenarios is known to be critical and delay sensitive, for this reason protocols as IP, TCP or UDP are not a good fit in that they would introduce too much overhead. In order to solve such issue, a non-IP based communication has been developed: the WAVE Short Message Protocol (WSMP) provided by the IEEE 1609.3 standard. The WSMP supports a fast and reliable exchange of Wave Short Messages (WSM) and deals with the dynamicity of the vehicular scenario [10]. When a WSM data unit is received from upper layers, the WSMP includes in the unit a WSMP header, specifying the EtherType field value before passing the packet to lower layers. According to the value found in the above-mentioned field, the channel router can decide whether to assign the unit to the IP stack or to the WSMP one for higher priority data. The IEEE 1609.3 also provides an entity responsible for the management of networking services functions, the WAVE Management Entity (WME). In fact, this entity takes care of the processing of the service requests for higher layers as well as the monitoring of Service Announcements. The messages used by WAVE are defined in the IEEE 1609.2 standard, that describes the secure messages formats and their processing by WAVE-compliant devices. Furthermore, this standard defines methods used in order to secure WAVE management messages and application messages. Such a standard is fundamental in vehicular environments due to the safety-critical nature of the applications that require messages to be protected from all sorts of possible attacks, such as spoofing, eavesdropping and alteration.

### DSRC Messages

When considering the V2X communication, it is important to define the structure of possible messages that can be adopted when exchanging data. The Society of Automotive Engineers (SAE) drew up several technical standards for vehicular environments among which emerges the SAE J2735 on “DSRC Message Set Dictionary”. This standard describes the message structures and implementation instructions, as well as supporting interoperability among different DSRC applications. An exhaustive description of all the messages defined by the standard can be found in [11], but in the following lines only the more relevant structures for this thesis' work will be reported.

**Basic Safety Message (BSM)**

This message is in charge of providing to the network the data regarding the vehicle's state, needed in order to localize it and run safety applications. The BSM is broadcasted to the other entities in radio range on a periodic basis, with a frequency of 10 Hz. The BSM is mainly divided in two parts: a mandatory part (Part I) that contains the header and the body of the message, and an optional part (Part II) that can be used if the application requires it. The fields that compose the Part I of the message are:

- Message ID: only field of the BSM header, it represents the ID of the message;
- Message Count: field useful for the estimation of the Packet Error Rate;
- Temporary ID: ID that represents the transmitting vehicle;
- Time: retrieved from GPS receiver;
- Position Data: informations about the Latitude, Longitude, Elevation and Accuracy of the current position;
- Movement Data: informations about the vehicle's Speed. Heading, Transmission State, Steering Wheel Angle, Acceleration, Yaw Rate;
- Brake System Status: used for the informations about the brakes conditions;
- Vehicle's Dimensions: Length and Width.

The optional Part II of the BSM includes instead two different data frames: Vehicle Safety Extension: depending on the type of safety application that is being used, this frame can be exploited or not. It is mainly composed of:

- Event Flag: switched when an unexpected event occurs;
- Path History: stores the recent movements of the vehicle;
- Path Prediction: stores an estimation of the future positions of the vehicle;
- RTCM Package: field that can be used in order to carry some of the messages defined by the Radio Technical Commission for Maritime Services standards.
- Vehicle Status: frame used to carry informations for uncritical safety applications.

**Intersection Collision Avoidance (ICA)**
The ICA message has the objective of carrying vehicle's data in order to achieve a collision avoidance system at road's intersections. In addition to the standard informations required to identify the message and the vehicle, the ICA is equipped with fields related to the map structure (as the ID of the intersection and the Lane Number), as well as the Path History of the vehicle itself. These informations are used to build a proper safety application centered on road intersections.

### 2.1.4 Cooperative-ITS (C-ITS) Standard

The C-ITS is a protocol adopted by the European Union in 2016 with the aim of developing a European standard for V2X communication. This set of specifications has been developed by the European Telecommunications Standards Institute (ETSI), defining in the ETSI EN 302 665 the architecture used for the communication in ITS scenarios, and in the ETSI EN 302 663 the European version of the PHY and MAC layers. In this second case, ETSI used the specifications of the IEEE 802.11p to define the two layers as a support to safety applications. But safety applications are not the only ones taken into consideration by C-ITS, in fact also traffic efficiency and the infotainment applications are managed.

The ETSI ITS-G5 is an evolution of the C-ITS standard that supports autonomous driving and includes wireless short range communications, dedicated to ITS scenarios as well as Road Transport and Traffic Telematics (RTTT). As announced during the description of the IEEE 802.11p protocol stack, the US and European standards for ITS are very similar, and their differences will be reported as follows.

#### Frequency Spectrum

The PHY layer of ITS-G5 exploits, to this day, five channels with a 10 MHz width each in the 5.9 GHz spectrum, as opposed to the IEEE version that uses seven channels. The channels are known in the European standard by their types instead of their numbering. In particular, three channel types can be identified:

1. ITS G5A band: corresponds to the IEEE channels 176, 178 and 180, and spans a 30 MHz width. It contains the the G5-CCH, G5-SCH1 and G5-SCH2 used for ITS safety applications;

2. ITS G5B band: corresponds to the IEEE channels 172 and 174 and is composed of the G5-SCH3 and G5-SCH4 channels. This band is exploited for non-safety applications;

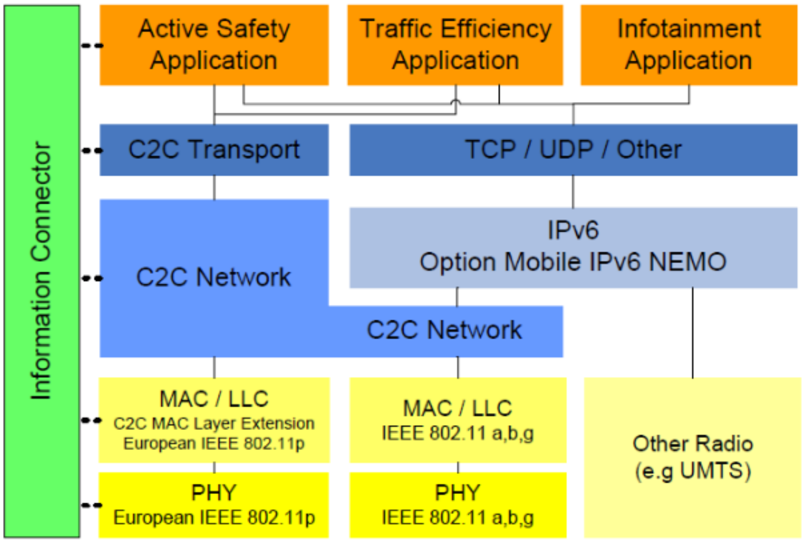


Fig. 2.4 ITS ETSI protocol stack [9]

3. ITS G5D band: corresponds to the IEEE channels 182 and 184, and its use is reserved for future ITS applications.

One important requirement set for the ITS-G5 is for its spectrum to limit interferences with the adjacent 5.8 GHz Electronic Fee Collection (EFC) system [12]. The mitigation techniques used for this purpose are defined in the ETSI TS 102 792 standard. For what concerns the rest of the PHY layer specifications, ITS-G5 follows the line of IEEE 802.11p, adopting the OFDM modulation with the same set of parameters as well as the half-clocked mode.

### MAC Layer

As for the DSRC protocol stack, the ITS-G5 relies for the non-safety applications on the IP-based network and transport layers. However, the protocols adopted for the routing differ. In fact, in this case an ad-hoc routing protocol is employed for multi-hop communication: the GeoNetworking, specified in the ETSI EN 302 636 standard. This protocol is based on the use of geographical coordinates for what concerns the addressing and forwarding operations. A packet can therefore be sent to all vehicles located in a specific geographical area, avoiding problems related to non-line-of-sight and communication range conditions, augmenting the performances of classical broadcasting. Its use in ITS safety applications is crucial in that it allows to have a low protocol overhead and deal with the frequent topology changes of vehicular networks. The GeoNetworking routing also allows to transmit IPv6 packets on the standardized adaptation sub-layer GN6, the IPv6 over GeoNetworking. A further difference with the IEEE 802.11p protocol stands at the access layer, where instead of a contention based system ETSI proposes the Decentralized Congestion Control (DCC) mechanism, defined in the ETSI TS 102 687 standard [13]. This choice comes as a consequence of ITS applications' need to count on the reliability and low latencies of data transmission. In fact, the limited bandwidth of ITS-G5 as well as the nature of the MAC protocol of IEEE 802.11 may cause an excess of data load on some wireless channels. DCC has been adopted in order to avoid instabilities in the system's behaviour, by dynamically changing channel in order to avoid local congestions.

### Messages

The C-ITS protocol stack describes in the standard for the facilities layer the types of messages that can be used by applications along with the definition of their structure. The most relevant messages in this case are the V2X ones, whose structure is very similar to the messages described in the DSRC protocol stack. The dissemination of the messages relies on a point-to-multipoint communication. In particular, the CAMs shall be transmitted from the origin ITS-Station (ITS-S) to the destination ITS-Ss in a single hop. The latter shall thus be located in the direct communication range of the transmitting ITS-S.

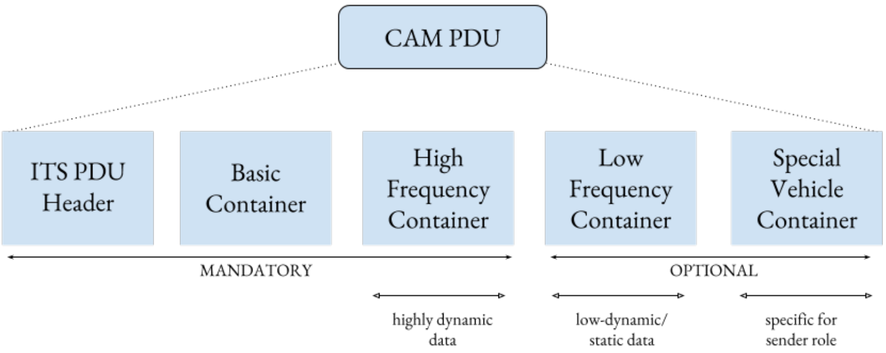


Fig. 2.5 CAM Structure

The Cooperative Awareness Message (CAM), defined in the ETSI EN 302 637-2, can be considered as the heartbeat messages of the ITS scenarios. In fact, their objective is to convey periodically the critical state informations of the vehicles, as the BSM in IEEE protocols, supporting both safety and traffic efficiency purposes. The reception of a CAM allows a third-party vehicle to track the position and movements of other vehicles, allowing to predict the occurrence of hazardous situations and enabling cooperative services. CAM messages are broadcasted to the neighbour nodes periodically with a frequency that can vary from 1 Hz to 10 Hz. The frequency of their transmission can also be adjusted dynamically according to the variations of the vehicle's state, adopting the Dynamic CAM functionality, described as follows. The entity responsible for the creation and management of a CAM is the Cooperative Awareness basic service, located in the facilities layer of the ITS-G5 architecture. The structure of a CAM id divided in a fixed mandatory part and an optional part, and generally divided into containers, as pictured in the image below.

The Distributed Environmental Notification Message (DENM), unlike the CAM, is an event-based message triggered by a specific application. Its main objective is to alert other vehicles of an event that has been detected and that may have an impact on road safety. However, this type of message can also be adopted for traffic efficiency purposes.

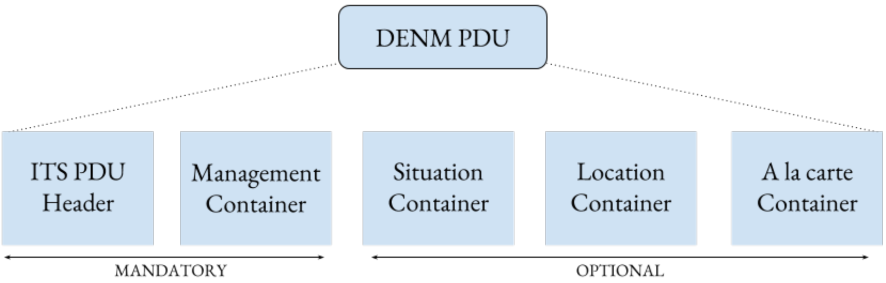


Fig. 2.6 DENM Structure

### CAM generation frequency management

In order to provide a constant awareness system in the road network, CAMs are usually generated in a periodic way by the CA basic service in a given ITS-S. But the frequency at which the messages are generated can also be determined taking into consideration the mobility parameters such as the change of position and speed so to calibrate the rate accordingly. This definition implies that the generation interval between CAMs can be dynamic, evolving along with the status of the traffic. As a general rule, an upper and lower bound can be identified for the generation of messages:

- Lower limit *T_GenCamMin* = 100 ms, corresponding to the generation frequency of 10 Hz;
- Upper limit *T_GenCamMax* = 1000 ms, corresponding to the generation frequency of 1 Hz.

The CAM is triggered based on the originating ITS-S status, that shall be checked every *T_CheckCamGen*, value that usually corresponds to the lower limit of the generation time interval. The currently valid upper limit *T_GenCam* is initially set to its default value *T_GenCamMax*. In a more specific way, the trigger conditions of the CAM generation are the following:

**Condition 1**

The time elapsed since the previous CAM generation is equal to *T_GenCam_Dcc* and one of the following mobility conditions is verified:

- the difference of heading between the last CAM and the current condition is larger than 4°;
- the difference of position between the last CAM and the current condition is larger than 4 m;
- the difference of speed between the last CAM and the current condition is larger than 0,5 m/s.

In this case, the upper limit *T_GenCam* shall be set to the time elapsed from the previous CAM generation.

**Condition 2**

The time elapsed since the previous CAM generation is equal to (or greater than) *T_GenCam* and also of *T_GenCam_Dcc*. In this case, if the CAM is triggered N_GenCam consecutive times due to this condition, the parameter *T_GenCam* shall be set to its default value *T_GenCamMax*.

## 2.2 Related Research

When evaluating different technologies, both pros and cons emerge from studies and performances shall be evaluated from different perspectives. In particular, one of the main topics that have been studied is whether to opt for a given technology or to build an hybrid solution that combines two different technologies. This comparison has been analysed in [14], where the theoretical performances of LTE and DSRC approaches are individually evaluated gathering the results of several other studies, and then individually compared based on field simulations. For what concerns the LTE, studies as [15] state that the LTE is not a lead candidate for the support of vehicular safety scenarios due to its tendency to overload the network easily and therefore generate collisions and packet losses during the transmission of safety-related beacons. This concept is also examined in [16], where the the advantages of the LTE performances are praised in sparse networks where the requirements on reliability, mobility support and scalability are met, but challenges appear when considering the same performances in networks with a higher cellular traffic load presence. In fact in this case the applications requirements on latencies are not satisfied. This type of behaviour is a consequence of working with entities characterized by a high mobility because of the Doppler effect that signals experiment as well as the cellular handover process. In those cases, LTE has been characterized with a Round Trip Time (RTT) larger than 100 ms as opposed to the the one underwent by DSRC that is lower than 10 ms.

The DSRC approach is also investigated in [16], where its performances are classified as acceptable for what concerns topologies characterized by a low concentration of vehicles as well as a limited mobility. The WiFi based communication is in fact not suited for the communication among entities characterized by a significant speed.

The two technologies have also been been studied and compared in [17], where an ICA system has been implemented and tested both with cellular-V2V (C-V2V) and IEEE 802.11p. The two types of communication have been assessed with respect to the number of collisions avoided with different Penetration Rates (PR) percentages as well as different channel conditions. The study proved that the performances obtained are very similar for both technologies, with a slightly better performance of C-V2V that reaches the 92% of collisions detected in time compared to the 90% obtained with IEEE 802.11p. These values are obtained with the 100% of PR, thus considering that all the considered vehicles are provided with the technologies and algorithm used. For lower values of PR, the system cannot be considered as reliable.

In the present literature, the comparisons between DSRC and cellular network communication for vehicular safety applications are in most of the cases based on typical parameters as reliability and latencies of the system, but rarely on parameters more strictly related to the collision detection application. Supporting a study related to collision avoidance systems using statistics on the number of collisions avoided in different scenarios would be interesting to have a higher insight on the consequences of the chosen protocol directly on the system performances. Furthermore, the great majority of studies that analysed this topic proposed qualitative comparisons between communication systems, but quantitative considerations are still hard to find. In the next chapter, the architecture of the system and the developed applications adopted in order to provide quantitative observations will be described.

# Chapter 3

# System Architecture

The applications that have been developed in this thesis are based on an heterogeneous network that allows the use of both WiFi-based and cellular technologies. While the V2V communication can be implemented through the IEEE 802.11p communication standard, the LTE standard can only be adopted for V2I communication. In fact, the simulators that have been used do not support yet the LTE sidelink transmissions on the PC5 interface, enabling only the connection of the UE to the eNB on the Uu interface. All the vehicles are therefore provided with an On-Board Unit (OBU) that enables the computation and communication capabilities, both using the DSRC and the LTE networks. The road infrastructure is instead represented by an eNB for the reception and transmission of data, connected to a server located on a METRO node that holds the computational capabilities.

## 3.1 Applications Description

The aim of this thesis is to compare the DSRC technology with the cellular LTE communication in terms of performances for safety applications. In particular, the objective was to implement a Collision Avoidance system able to detect and prevent collisions among vehicles at road intersections exploiting the functionalities of the previously stated technologies. In order to compare the efficiency of the latter, four scenarios have been staged, and the two technologies have been used both alone and combined for the delivery of safety-related informations. Furthermore, all the scenarios implemented have been tested both considering a human driver and an autonomous vehicle, making it possible to compare quantitatively the results obtained for the two methods.

All the scenarios implement an exchange of CAM among vehicles and with the server in order to provide the basic informations about the state of the nodes. When the analysis of those messages results in the prediction of a collision, alert messages are transmitted.

The alert messages are built following the model of the DSRC ICA messages, and their generation is a consequence of the *CollisionDetection* algorithm results. In the different scenarios, the generation process of alert messages changes according to the location of the *CollisionDetection*. In fact, two application versions are implemented:

- centralized version: both the computational capabilities and the *CollisionDetection* algorithm are restricted to the server. In this case, the latter is the only entity capable of predicting possible collisions and therefore send alert messages to the concerned vehicles;
- distributed version: the algorithm is established inside all the vehicles present in the system. Each vehicle is able to run the *CollisionDetection* and detect hazardous situations confronting its mobility data with the ones received from other vehicles via CAMs. Different tests can be assembled based on the possibility for the vehicle to run the algorithm for itself, generating a local alert message, or to broadcast instead the alert messages to the surrounding vehicles.

The four scenarios simulate different systems in which the the application can be centralized, distributed or located both in the server and in the vehicles. The alerts can therefore be transmitted by different entities according to the scenario.

### 3.1.1 Scenario I

In the first scenario, vehicles were provided with a double communication interface, one for the DSRC communication and one for the LTE one. While the DSRC protocol was used to broadcast CAM among vehicles in a V2V mode, providing constant updates on the state of vehicles, the LTE was only used for the communication with the road infrastructure and vice versa. In this case, CAMs were transmitted via LTE to the eNB server, the only entity equipped with a Collision Detection algorithm able to predict possible accidents. In the case of the detection of an hazardous situation, the server was responsible to transmit in unicast mode an alert message to the concerned vehicles. This type of LTE operation scenario is described in the 3GPP Technical Report 36.885 (section 4.2) [6]. While the cooperative data is transmitted using both technologies, the safety messages are only spread using the cellular network. Furthermore, the computation is here centralized and confined to the road infrastructure.

### 3.1.2 Scenario II

The second scenario provides for vehicles to interact once again using DSRC for the V2V communication and LTE for the V2I one. But as opposed to the previous case, the Collision Detection algorithm is here located not only in the eNB server, but also inside vehicles. This way, the infrastructure can send alert messages to warn vehicles of possible hazards, but the vehicles themselves can also perform the calculations based on the CAM received from the surrounding entities and their own mobility parameters. Two types of alerts can thus be received: a local alert generated by a vehicle's computer, and subsequently processed in order to show a visual warning in the vehicle's display, and an alerts sent in V2I mode by the server. The computation in this case is no more centralized, but distributed in the vehicular network.

### 3.1.3 Scenario III

As an evolution of the second scenario, the third one adopts the architecture previously described, but includes the broadcast of alert messages by vehicles in V2V mode. Three different types of alerts can thus be received:

- local alert self-generated by the vehicle;
- V2I alert transmitted by the server;
- V2V alert broadcasted by an external vehicle.

Both the Scenarios II and III can be defined as mixed scenarios, where two technologies are used in order to deliver safety-related informations.

### 3.1.4 Scenario IV

In the fourth and last scenario, the architecture used in the Scenario III is reused, avoiding this time any communication with the server. Only the V2V communication via DSRC is left for both CAM and alert messages, but still discriminating between self-generated warnings and alerts coming from the outside.

Generally, if two vehicles are not in Line of Sight (LOS) they do not receive each others CAMs, and they can therefore not run collision detection on their data. For this reason, if they a collision is conceivable, they will not be warned unless some other entity in LOS with both of them can run the algorithm. This issue can be solved both in scenarios in which a centralized approach is adopted, both in scenarios that implement the alert transmission in V2V.

## 3.2 System Characteristics

The main actors that have been used in order to build the scenarios previously described are described as follows.

### 3.2.1 Messages

The system taken into consideration for this work is based on a constant exchange of messages, both in a V2V mode and in V2I mode. Two different types of messages are adopted:

**Cooperative Awareness Message:** periodic message sent by vehicles with a frequency of 10 Hz that contains all the data characterizing the status of a vehicle at the moment of transmission. In particular, the CAM contains:

- CarID: id that uniquely identifies a vehicle, retrieved from the SUMO configuration;
- Source Address: address of the sender vehicle;
- Destination Address: address of the receiver vehicle;
- Sending Time: time at which the CAM was sent;
- Arrival Time: time at which the CAM was received;
- Network Type: protocol used for the communication (802.11p or LTE);
- Current Position: position of the vehicle at sending time;
- Current Speed: speed of the vehicle at sending time;
- Current Acceleration: acceleration of the vehicle at sending time;
- Angle: direction of the vehicle.

The CAM used is defined as an *HeterogeneousMessage*, built as an extension of the Wave Short Message (WSM) and used by the heterogeneous network. Its structure allows to set the type of network protocol to be used to send the packet, according to which the *DecisionMaker* module determines the correct output port. PARLARE QUI DELLE PORTE.

The messages can therefore be sent to the surrounding vehicles using 802.11p and to the server (eNodeB) using LTE. In order to avoid flooding the network with CAM even when the the vehicle's mobility parameters do not change substantially or when the vehicle is not moving, the dynamic CAM approach defined by ETSI EN 302 637-2 has been adopted. For simplicity purposes, only the conditions ITS dynamics-related described in the standard have been implemented. In particular, three conditions have been identified as crucial to determine whether a vehicle's dynamics should be notified to the surrounding environment or not: the variations of position, speed and direction. Each 100 ms, before sending the CAM, the parameters are analysed with respect to the values taken at the time of the previous CAM, and the conditions checked. More in detail, a CAM should be sent if:

$$\Delta position > 4m \text{ and } \Delta speed > 0.5m/s \text{ and } \Delta angle > 4$$

A further condition is established on the maximum time period that can elapse between two CAM transmissions. In fact, if even though the conditions above stated are not satisfied but 1 second went by since the previous transmission, a new CAM should be generated and sent.

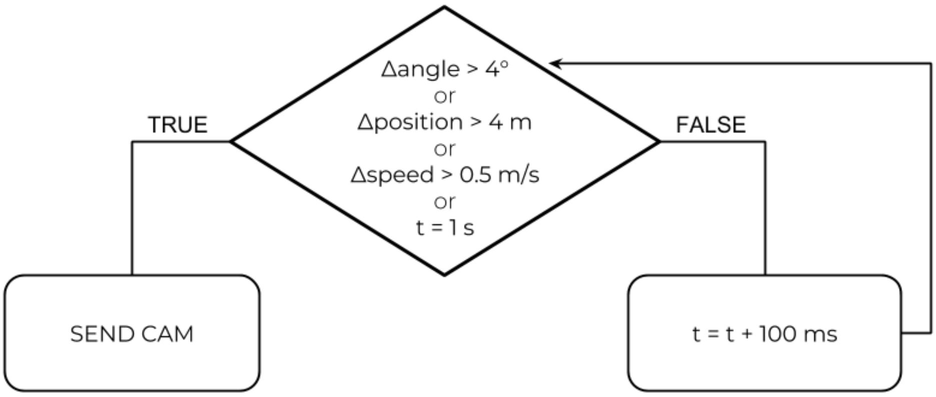


Fig. 3.1 Dynamic CAM flow chart

**Alert Message:** event-based message triggered by the occurrence of a possible collision detected by the *CollisionDetection* algorithm. When this sort of message is received by a vehicle, a visual signal is activated on the dashboard in order to inform the driver of the possible accident.

The Alert is composed of:

- GenerationTime: time at which alert is generated;
- Car1: ID of the first car involved in the possible collision;
- Car2: ID of the second car involved in the possible collision;
- Other_type: type of entity to whom we may collide, in this case a vehicle. This parameter is set for scenarios in which other entities are introduced, such as pedestrians and bicycles;
- collision_type: collision due to a rear-end or to an intersection;
- T2C: Time to Collision, so in how long vehicles could collide;
- S2C: Space to Collision, so in how many meters vehicles could collide;

### 3.2.2 Data Structures

The creation of some data structures has been fundamental for the proper functioning of the system. The most relevant ones are here described.

#### VehicleData

This structure is composed of all the relevant parameters that characterize a vehicle status at a given time. The purpose of this structure is to be able to pass the mobility parameters to functions in a compact and legible way. VehicleData is composed of:

- VehicleId: id of the considered vehicle;
- Timestamp: time at which the vehicle is inserted in the structure;
- Position: coordinates of the vehicle;
- Speed: speed of the vehicle on the cartesian axes;
- Acceleration: acceleration of the vehicle on the cartesian axes;
- Angle: heading of the vehicle.

#### Storage

The main function of the Storage is to record all the informations extracted from CAMs received and is adopted by both vehicles and server. The Storage is built as a map that associates to the id of a vehicle the VehicleData of the latter. Each time a CAM is received, the data contained in the message is gathered in a VehicleData structure, whose values are saved inside the Storage. It is crucial for the system to always have the most recent vehicle parameters, for this reason if the CAM transmission took more than 0.8 seconds to reach the destination, the values are discarded. This structure is the one used to pass data to the Collision Detection algorithm, thus its integrity and reliability are crucial for the system.

#### Alert Data Structure

The Alert Data Structure is a map that associates a pair of vehicle IDs to the time at which the alert was sent to them. This structure's aim is to avoid an excessive flooding of alerts in the network. In fact, this type of message is controlled in such a way to avoid transmissions at less than 1 second of delay from the previous one. This structure comes with methods created in order to ease its management, such as an *Update* function and *Check_and_Send_Alert* used to check the time elapsed from the previous transmission.

### 3.2.3 System Workflow

The whole system behaviour can be globally divided in two parts: the actions taken at the vehicle level and the ones taken at the server level. The description of the operations followed by the system is reported below.

#### Vehicle

CAM messages are generated by each vehicle with a frequency of 10 Hz, thus each 0.1 seconds, and if the conditions of the *DynamicCAM* function are verified the CAM is sent both to the surrounding vehicles in radio visibility with 802.11p and to the server with LTE. The vehicles fill the messages with their most recent informations retrieved by the TraCI Mobility module of Omnet++. This behaviour is fundamental in the moment in which other vehicles receive a CAM through the *ReceiveCAMPacket* function. The data contained the message is indeed first extracted in order to be inserted in the Storage of the vehicle, and then passed to the *CollisionDetection* function so to be compared with the receiving vehicle's data. In fact, the *CollisionDetection* is in charge of detecting whether a collision among two vehicles is likely to happen. According to the scenario that has to be simulated, the *CollisionDetection* algorithm located inside the vehicle can be called to compare the vehicle itself to the sender of the CAM, or more in general iterating on all the vehicles that have been saved inside the Storage structure. In the case in which a collision is foreseen by the algorithm, an Alert message shall be sent to the concerned vehicles. The *AlertDataStructure* has been adopted with the purpose of storing the last warning message sent regarding a given collision. In order to avoid sending a too large number of Alerts, the Check_Send_Alert function controls when the last alert was sent, and limits the transmission to one message per second. When an Alert is received by a vehicle, the *ReceiveAlert* function is called with the purpose of printing the message in a file used for post-processing statistics.

### Server

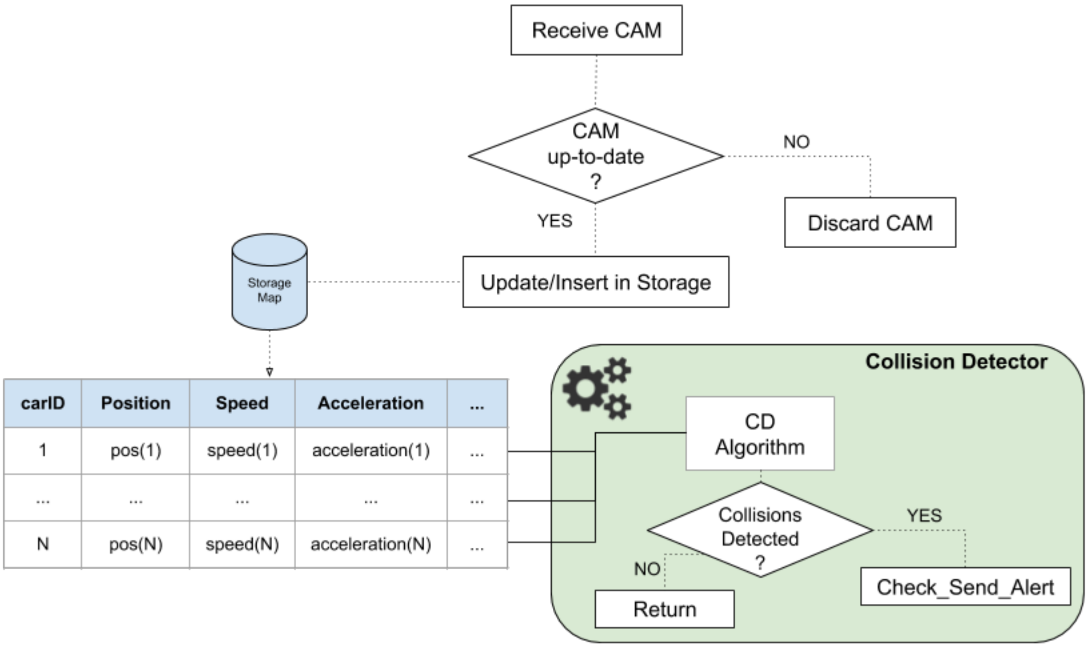
As previously stated, the CAM is not only sent to vehicles but also to the server. In this case, the steps taken are very similar to the ones followed by vehicles. Data is once again extracted from the message and used by the *UpdateStorage* function, and consequently passed to the *CollisionDetection* algorithm. The latter iterates on all the vehicles saved in the storage in order to predict possible collisions. In this case, a further precaution is taken in order to guarantee that the data used to predict crashes is up-to-date. In fact, while iterating the storage, the positions of the vehicles are updated using the interval of time elapsed from their insertion in the structure through the following formula:

$$x = x + v\Delta t + \frac{1}{2}a\Delta t^2$$

This way, possible inaccuracies due to stale entries in the storage can be further mitigated. If an hazardous situation is detected among two vehicles, the SendAlert function is called and in the case in which Check_Send_Alert function has a positive outcome the messages are transmitted to the concerned vehicles. For all the scenarios analyzed, the server has been considered to run on a Metro node, that represents one of the topology solutions adopted nowadays by mobile operators as an alternative to the use of the Cloud. The Metro node is a metropolitan node, located not far from the eNB. A message received by the eNB should therefore be forwarded to the server, and an alert generated by the latter should follow the same route backwards. In order to model this topology, a Round-Trip Time (RTT) of 10 ms is considered for the connection between eNB and Metro node. In the code implementation, this behaviour has been simulated introducing in the server a delay of 5 ms at the reception of a message, and 5 ms before sending an alert if all the conditions required are verified.

### CollisionDetection

As it can be noticed from the descriptions of the server and vehicle's workflows, the *CollisionDetection* algorithm is one of the pillars of the system. This function in fact receives as input parameters all the informations that characterize two vehicles at a given time, and analyzes them in order to predict if a collision is likely to occur. This prediction is based on the analysis of the mobility parameters, that allows to compute the Time-to-Collision (T2C) and the Space-to-Collision (S2C) values. Those two parameters represent the remaining time for two vehicles before reaching their mutual minimum distance and the distance that cars will reach when at T2C. The T2C and S2C values are checked against two thresholds, the $T2C_t$ and $S2C_t$ equal to 2.5 and 3.7 respectively, under which a collision is detected.
The first step taken in order to predict a collision is to identify that the type of collision that could happen is a collision due to an intersection, and therefore discard all the cases in which the collision type represents a rear end crash. The *CollisionDetection* function, itself composed of two more methods that compute the T2C and S2C, is then called. Both the T2C and S2C values are computed, and the values they could take are either equal to "NO_COLLISION" in the case in which a crash is not foreseen in the near future or if the value is larger than the given threshold, or equal to a positive value if an hazard is possible. The first value to be evaluated is the T2C through the *TimeToCollision* method, that if returns a negative outcome automatically sets also the S2C to a "NO_COLLISION" value and the pair is returned. If the T2C is instead a positive number, the *SpaceToCollision* method is called. In the case in which the returned S2C is larger than the corresponding threshold, both the values are set to "NO_COLLISION" and returned. On the opposite, if the S2C is lower than the $S2C_t$, the pair that now contains the two parameters that identify the future collision is returned. The thresholds adopted have been chosen in such a way to minimize the number of false positives detected and set to zero the number of false negatives considering the chosen vehicles speed. In fact, embracing larger threshold values would have led to a higher number of collisions detected, but also a higher number of false positives.



| carID | Position | Speed | Acceleration | ... |
|---|---|---|---|---|
| 1 | pos(1) | speed(1) | acceleration(1) | ... |
| ... | ... | ... | ... | ... |
| N | pos(N) | speed(N) | acceleration(N) | ... |

Fig. 3.2 Collision Detection Mechanism

**Algorithm 1** Collision detection pseudocode

```
Require: $\vec{p}_0, \vec{v}, \vec{a}, \mathscr{B}$
$C \leftarrow \emptyset$
$\vec{p}(t) \leftarrow \vec{p}_0 + \vec{v}t + \frac{1}{2}\vec{a}t^2$
$p_x(t) \leftarrow p_x^0 + v_x t + \frac{1}{2}a_x t^2$
$p_y(t) \leftarrow p_y^0 + v_y t + \frac{1}{2}a_y t^2$
for all $b \in \mathscr{B}$ do
    read $\vec{\tilde{p}}^0, \vec{\tilde{v}}, \vec{\tilde{a}}$ from $b$
    $\tilde{p}_x(t) \leftarrow \tilde{p}_x^0 + \tilde{v}_x t + \frac{1}{2}\tilde{a}_x t^2$
    $\tilde{p}_y(t) \leftarrow \tilde{p}_y^0 + \tilde{v}_y t + \frac{1}{2}\tilde{a}_y t^2$
    $D(t) \leftarrow (p_x - \tilde{p}_x)^2 + (p_y - \tilde{p}_y)^2 =$
    $= \left[p_x^0 - \tilde{p}_x^0 + (v_x - \tilde{v}_x)t + \frac{1}{2}(a_x - \tilde{a}_x)t^2\right]^2 +$
    $+ \left[p_y^0 - \tilde{p}_y^0 + (v_y - \tilde{v}_y)t + \frac{1}{2}(a_y - \tilde{a}_y)t^2\right]^2$
    for all $t^\star \in T$ do
        if $t^\star < 0$ or $t^\star > t2c_t$ then
            continue
        $d^\star \leftarrow \sqrt{D(t^\star)}$
        if $d^\star \leq s2c_t$ then
            $C \leftarrow C \cup \{b\}$
            break
return $\mathscr{C}$
```

# Chapter 4

# Simulations and Results

## 4.1 Simulation Tools

A Mobile Ad-Hoc Network (MANET) can be defined as a network of nodes that are not under the control of a top-down network administrator. These free nodes composing MANETs are usually mobile nodes (smartphones, mobile devices, PCs, etc.), but vehicles can employ the same kind of principle generating a Vehicular Ad-Hoc Network (VANET). The requirements in the case of VANETs are different from the ones of MANETs due to the characteristics of the nodes. In fact, in this specific case nodes have high transmission power as well as high computational capabilities, but at the same time the network is highly variable and unstable due to the movement if the nodes. The main characteristics of a VANET can be summarized as follows[18]:

- High Mobility: nodes in a VANET are usually moving at high speed, increasing the complications in the transmission of data especially in safety critical situations. This issue is in part solved by the fact that vehicles are restricted in their movements by the street's topology, making it easier to predict their position;
- Network Topology: due to the high mobility of nodes, the network topology tends to be variable and the connection among nodes unstable;
- Energy Resources: in a VANET, nodes do not have the same energy constraints as mobile devices. In fact, vehicles have a large rechargeable battery that provides power to the electrical systems. The computational capabilities can therefore be larger with respect to the ones in MANETs;
- Geolocalization: the vehicle's data (position, speed, direction, . . . ) can be augmented by satellite navigation systems in order to have more precise informations;

- Time Critical: VANETs require extremely low latencies communication in order to provide the useful data necessary for taking decisions in safety applications.

The development of VANETs in the past few years created the necessity to adopt specific simulation tools. In fact, their network performances can be considered more complex to analyze with respect to the ones of classic Ad-Hoc networks due to the peculiarities stated above. For these reasons, the workbench both requires network and mobility simulators that shall interact through a VANET simulator. In particular:

- Mobility Simulator: in charge of creating realistic traffic models and simulating the parameters that characterize the traffic flow;

- Network Simulator: in charge of the simulation of the network traffic parameters for V2X transmissions by exploiting the communication stack and routing protocols.

For the purposes of this thesis, both V2V and V2I communications need to be simulated. In order to do so, OMNeT++ has been chosen as Network Simulator, SUMO as Mobility Simulator and VeinsLTE as VANET Simulator.

### 4.1.1 OMNeT++

Objective Modular Network Testbed in C++ (OMNeT++) is a simulation library implemented in C++ used in order to build network simulators [19]. The platform has a component-based architecture that allows to program single modules that can be assembled into larger components, the compound modules, using the NED language. Such a structure lays the ground for a hierarchical simulation model, where components can be reused in order to enable large-scale simulations. Modules are characterized by some parameters used to deliver data used for configuration purposes, usually set in the omnetpp.ini configuration file. OMNeT++ is a discrete event simulator, thus actions can be taken at discrete and defined instances of time instead of being continuous, allowing simulations to run faster.

### 4.1.2 SUMO

Simulation of Urban MObility (SUMO) is an open source simulation tool implemented in C++ and developed by the Institute of Transportation Systems at the German Aerospace Center. It is a microscopic simulator, meaning that the single vehicles (or pedestrians, public transports, etc.) are modelled explicitly with their own route, speed, acceleration and characteristics.

### 4.1.3 VeinsLTE

Vehicular In Network Simulation (Veins) is an open source software platform that connects the previously described simulators: Omnet++ to model the network communication among vehicles and SUMO for what concerns the mobility simulation. The simulators are connected through a TCP socket using the Traffic Control Interface (TraCI) protocol, allowing them to run in parallel and exchange bidirectionally data so to provide realistic simulations. Veins can be considered as one of the most popular VANET simulators, and its frameworks is reported in figure x. VeinsLTE is the extension of Veins that provides the LTE support to models. This software version provides both the simulation of IEEE 802.11p and the simulation of the LTE complete stack using the SimuLTE module that allows the implementation of heterogeneous vehicular networks. On top of those stacks, applications can be developed by accessing and modifying parameters from the codes of both protocols[20]. Applications can be built in order to send messages only through IEEE 802.11p or LTE, but the decision could also be left to the lower layers that identify smartly the best option at a given time using a Decision Maker module. In order to provide an accurate and realistic VANET simulator, Veins supplies noise and interference models like Simple Path Loss, Shadowing and Fading.

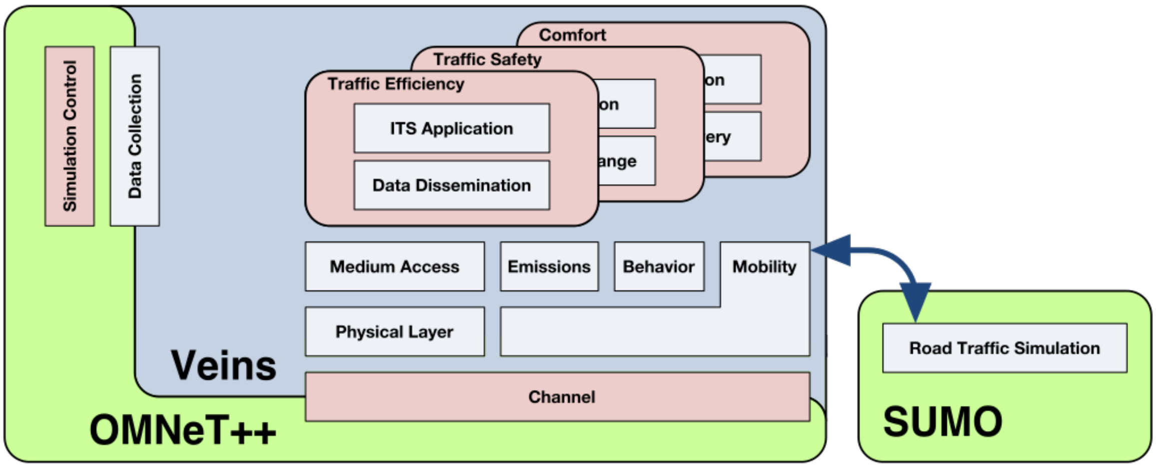


Fig. 4.1 Veins Architecture

## 4.2 Reference Scenario

### 4.2.1 Mobility Settings

In order to analyze the behaviour of the system described previously with the use of different technologies as well as different architectures, an urban scenario composed of two intersections between three main roads has been considered. Both of the crossings have been left unregulated, setting the traffic lights to green for the whole simulation time and avoiding the use of traffic signs, so to increase the collision probability among vehicles. The scenario is reported in Fig. 4.2. The vehicles enter the system with a constant speed of 13.89 m/s, thus 50 km/h, from six possible accesses. A null deceleration has been set in order to force the vehicle's journey even if hazardous situations are foreseen. Furthermore, the vehicles are restricted to straight rides because of the willingness to simplify the rules dominating the junctions, and therefore observe and analyze the basic scenario.

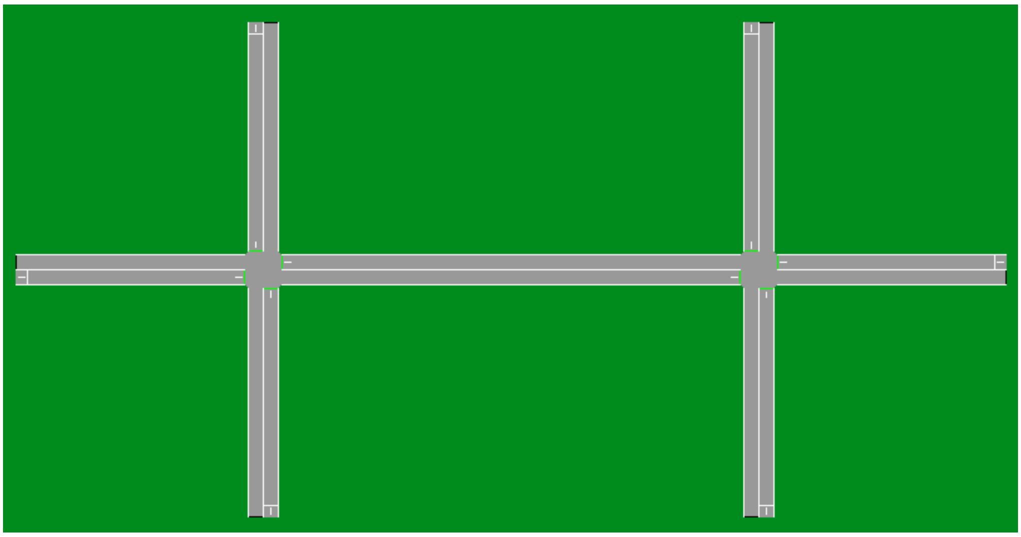


Fig. 4.2 Reference Scenario in SUMO

The generation rate of the entities that enter the map in SUMO follows a Poisson distribution with input rate $\lambda$. A Poisson distribution can be defined as a discrete distribution that predicts the occurrence of events in a given time period, knowing the average instances in that same period of time. Such a distribution behaves following a negative exponential function, and can be defined by the following formula:

$$P_{\lambda}(n) = \frac{\lambda^n}{n!} e^{-\lambda}$$

For this reason, the instants of time at which vehicles are generated are distributed according to a negative exponential. For the sake of the realistic behaviour of the system and to avoid the saturation of one particular street, the vehicles have been injected iterating in a circular way on the six entrances. In fact, the insertion of too many vehicles in a row in a single road can lead to severe slowdowns in the system's mobility, and the engendered collisions would not be trustworthy due to the type of traffic and the slow-pace of vehicles. The generation rate adopted for the simulations has been identified in the light of the study on the maximum generation rate, in which several lambdas have been tested for the above mentioned scenario. Evaluating the average number of vehicles in the system with respect to the variation of the generation rate $\lambda$ in the range [0.1;2.0], the following graph has been obtained.

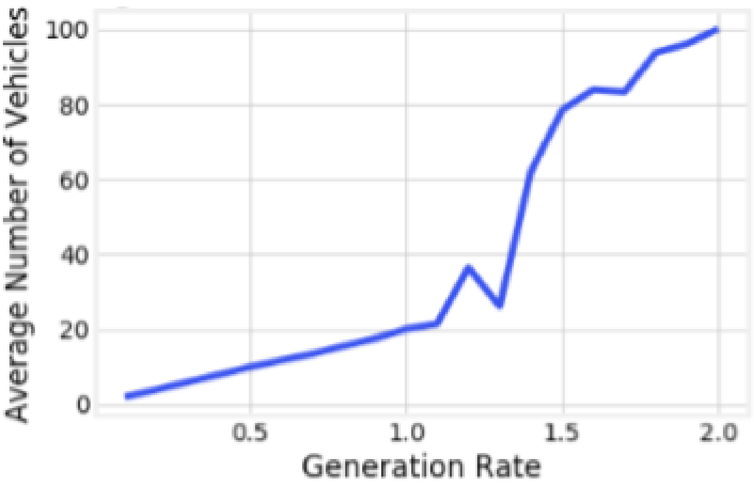


Fig. 4.3 Maximum Rate Study

In order to obtain simulations where the behaviour of the vehicles is meaningful enough to obtain some statistics, the $\lambda$ should neither be too small, since the number of collisions would be very small or null, nor too high for saturation purposes. In the figure x, the stability interval of the generation rate can be identified between 0.1 and 1.1, where the growth is linear. For this reason and for the motivations stated above, a generation rate $\lambda = 0.7$ has been chosen for the simulations. Twenty different mobility traces have been created with SUMO taking into consideration all the parameters described. For each scenario that has been tested, 20 simulations each one with a 300 s duration have been runned.

### 4.2.2 Evaluation of the effectiveness

For the sake of the system's realistic behaviour, some delays have been introduced in order to simulate real-life timings for the communication among vehicles and with the infrastructure. Several kinds of latencies are considered at different levels, covering both network-related delays, hardware delays and human-responsiveness delays.

#### Transmission and Propagation time

The transmission time represents the time needed to send the packets' bits on the link, and therefore represents the bit rate of the communication link. The propagation time instead accounts for the time required to reach the destination from the transmitter, and can be computed using the distance between the two nodes and the speed of light:

$$t_P = \frac{d}{c}$$

Both of these delays are automatically managed by the OMNeT++ simulator, that introduces latencies when transmitting the packets.

#### Processing time

For what concerns the reception of alert messages inside vehicles, a processing time is needed in order to transform the message into an audio or visual signal that can warn the driver of a possible collision. For this reason, a delay of 400 ms has been adopted in order to take into account the time needed by the vehicle to issue an alert via Human-to-Machine Interface (HMI).

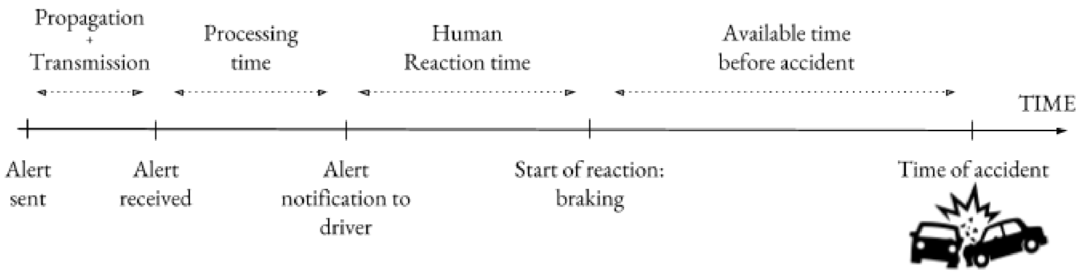

Propagation + Transmission
Processing time
Human Reaction time
Available time before accident
TIME
Alert sent
Alert received
Alert notification to driver
Start of reaction: braking
Time of accident

Fig. 4.4 Timeline in Human Driver scenario

### Reaction time

From the moment in which an alert signal is given by the vehicle to the driver, a short quantity of time is required for the the driver to start reacting, thus braking. This time is generally modeled with a 1 second delay. Of course, when modeling scenarios that adopt the autonomous vehicle technology instead of a human driver, this specific latency is not considered due to the zero reaction time needed to answer to an input.

The time left from the beginning of the reaction to the instant in which the collision is foreseen is the available time $T_A$. The time required to stop the vehicle, derived from the speed at which the latter is running, can be compared with $T_A$ in order to understand if the collision can be avoided. In fact, if $T_{BRAKING} > T_A$, the crash cannot be avoided. In the scenario in which an autonomous vehicle is adopted (Fig. 4.5), the $T_A$ is longer due to the absence of human reaction time, thus the number of avoided collisions should be higher.

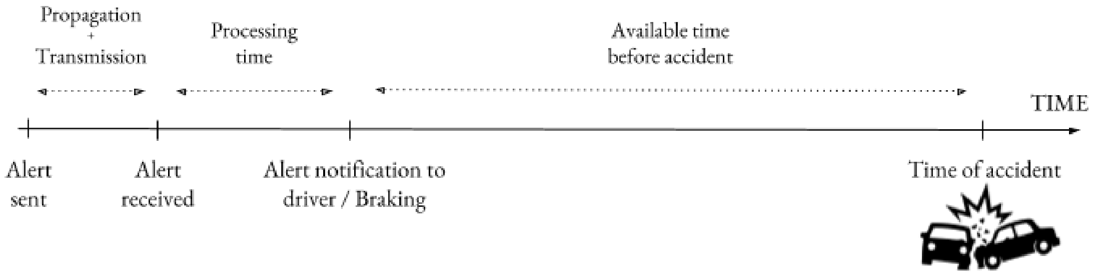


Fig. 4.5 Timeline in Autonomous Vehicle scenario

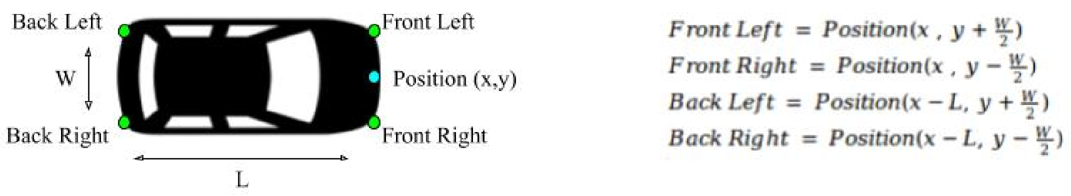
Those parameters are used in order to evaluate the effectiveness of the developed application, in that they provide some indicators, as the percentage of collisions detected in time. too late or not detected, that can help in the assessment of the different scenarios.

## 4.3 Simulation Results

As described in the previous chapter, four scenarios have been tested in order to obtain some statistics on crucial parameters for vehicular safety applications. For the three first scenarios that involve the use of a server, the latter has been considered to be located on a Metro node, characterized by a latency of 5ms both in the reception and transmission of messages. Furthermore, for each scenario statistical results on central parameters have been retrieved, both considering the human driver case and the autonomous vehicle one. The consequences of this distinction are considered in the post-processing of the simulation outputs, in which the timings corresponding to those two situations are evaluated.

### 4.3.1 Collision Evaluation

The methodology adopted in post-processing to determine if two entities really collided is based on building polygons around the position of the vehicles that received an alert from the system. In fact, the position that can be retrieved by the TraCI Scenario Manager in OMNeT++ corresponds to the coordinates of the centre of the vehicle's front bumper. Considering the width (1.8 m) and length (4.3 m) of the cars, a polygon is built around the given coordinates in order to model the vehicle's shape in the following figure:



$$
\begin{aligned}
Front\,Left &= Position(x\,,\ y + \tfrac{W}{2})\\
Front\,Right &= Position(x\,,\ y - \tfrac{W}{2})\\
Back\,Left &= Position(x - L,\ y + \tfrac{W}{2})\\
Back\,Right &= Position(x - L,\ y - \tfrac{W}{2})
\end{aligned}
$$

Fig. 4.6 Polygon building method for Collision Evaluation

This way, for each pair of entities that received a warning message, possible intersections among the corresponding polygons are evaluated, and real collision situations are differentiated with respect to false positives.

### 4.3.2 Scenario I

The scenario I is characterized by vehicles exchanging CAM both among them and with the server, but with the *CollisionDetection* algorithm only located in the server. For this reason, this first case provides for the generation of alert messages only from the server, thus using the LTE communication. Before analysing the behaviour of the system locating the server on the Metro node, an ideal case has been tested in which no latencies are considered in the communication with the server. This way, a comparison between the two scenarios, both for the human driver and the autonomous vehicle, can be done. Generally, we could expect a trend in which the percentage of collisions detected in time is higher in the case of an ideal server, since the alert is sent more promptly. Furthermore, a larger number of collisions detected in time should be noticed in the autonomous vehicle results.

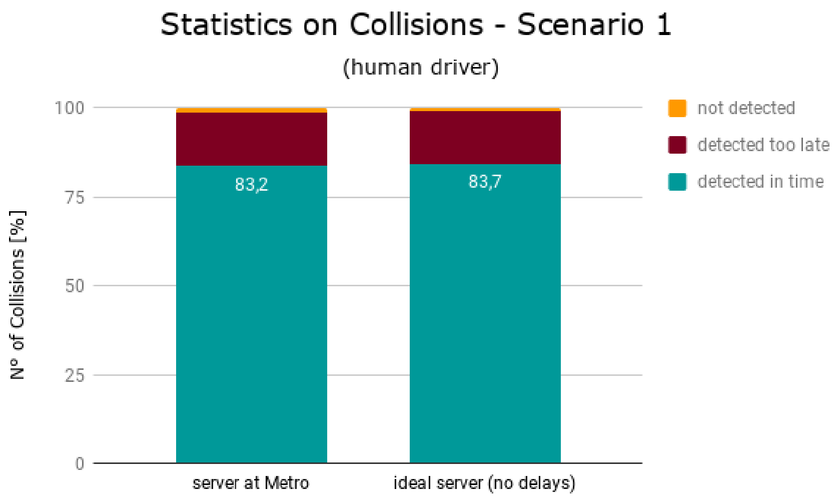


Fig. 4.7 Statistics Scenario I - Human Driver

These expectations are satisfied by the results obtained, in which both for the human driver scenario and for the autonomous vehicle one, a slightly higher number of collisions are detected in time when adopting an ideal server with respect to the Metro one. Considering the human driver scenario, a decrease of detections in time of 1.55% can be noticed. With the autonomous vehicle scenario instead the contraction is less decisive because of the impact played by the absence of the human reaction time. In fact, a larger time is available for the vehicle to stop, and therefore avoid the collision. For this reason, the decrease of detected collisions in time using the server located at the Metro node is only 0.5%.

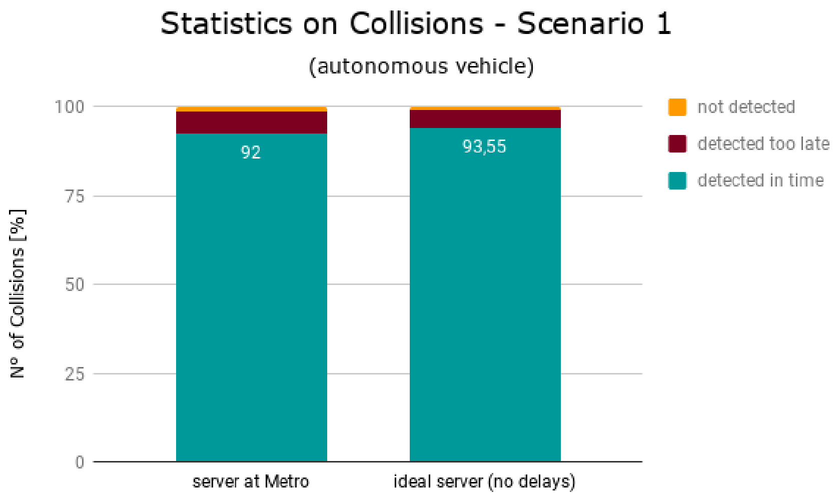


Fig. 4.8 Statistics Scenario I - Autonomous Vehicle

### 4.3.3 Scenario II

In the second scenario, the *CollisionDetection* algorithm is located in the vehicles as well as in the server. Two types of alert messages can thus be received by vehicles: the ones issued by the server and the ones issued locally by vehicles themselves, to auto-notify a warning. An interesting parameter to study in this case is the type of the first alert received by a vehicle, which is the one that potentially allows to avoid a collision. Generally, we could expect a much larger number of alert messages originated from the vehicle to show up first, since the latencies coming into play are lower compared to the ones of an alert issued by the server, in particular when the latter is located on the Metro node.

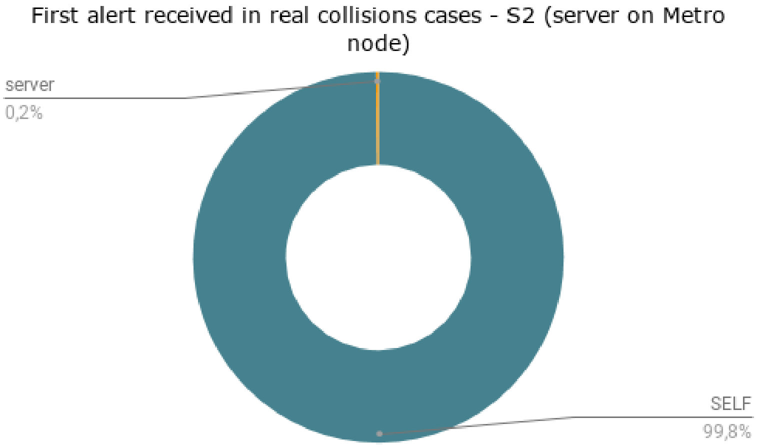


Fig. 4.9 First Alert Analysis with Real Collisions - S2

For what concerns the real collision case, the percentage of alerts transmitted locally by the vehicle represents the 99.8% of the total number, leaving only the 0.2% coming from the server. On the other hand, the number of first alert issued by the server increases significantly when considering the false positives instead of the real collisions. In fact, the percentage obtained is states that the 27,4% of the total alerts is transmitted by the server. This specific behaviour could be explained by the fact that while the vehicle operates the *CollisionDetection* on current mobility parameters for both itself and the vehicle that sent the CAM, the server works with only the current parameters of one vehicle, while the second set of parameters is retrieved from the storage. Even if those parameters are appropriately updated, the precision of the server could suffer from a small inaccuracy with respect to the vehicle's system.

On the side of the statistics related to the number of collisions detected, for both the human driver scenario and the autonomous vehicle's one, some interesting evaluations can be extracted. Due to the fact that two different systems are working on the prediction of possible

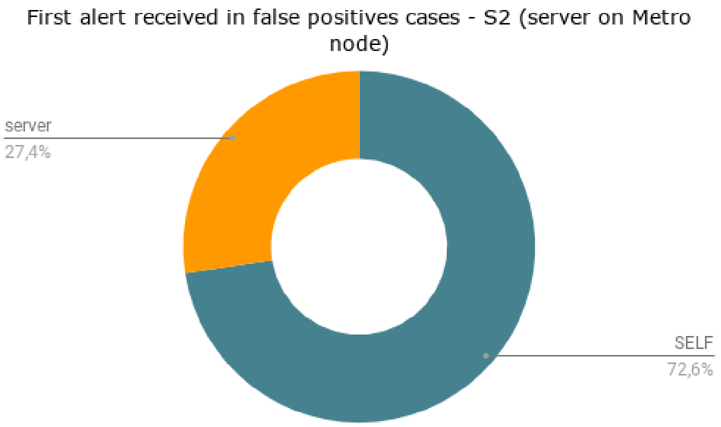


Fig. 4.10 First Alert Analysis with False Positives - S2

collisions, a higher number of collisions are detected with respect to the previous scenario. In fact, an increase of 8.8% can be noticed in the collisions detected in time when considering a human driver case, and an increase of 0.25% when focusing on the autonomous vehicle case. Furthermore, it can be observed that the number of collisions not detected is now equal to zero for both the autonomous vehicle and the human driver scenarios. This result marks an improvement of the 1.15% compared to the previous scenario.

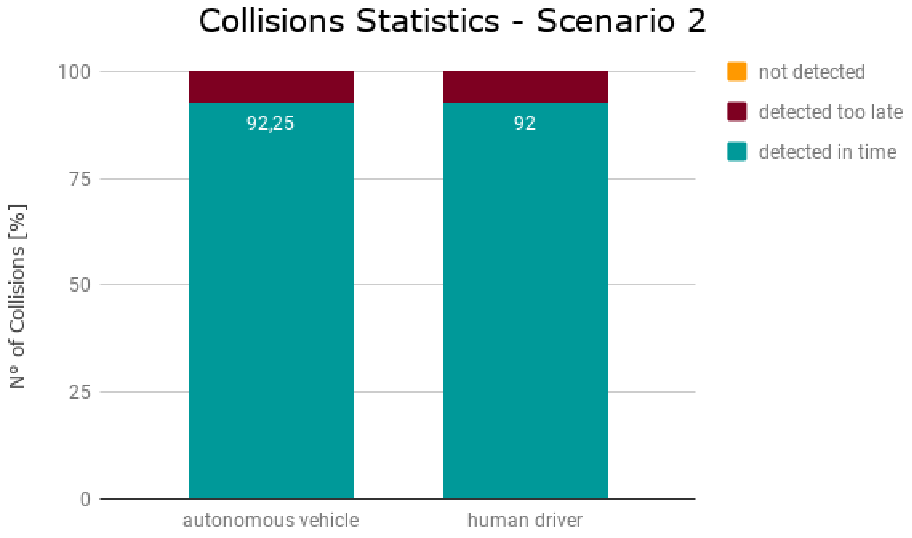


Fig. 4.11 Collision Statistics - S2

### 4.3.4 Scenario III

The third scenario that has been implemented is very similar to the one just described, with the only difference that the alerts generated by vehicles are now also broadcasted to the other vehicles in range as well as being sent locally by a vehicle to itself. Three types of alerts can therefore be identified, allowing to make some comments on the type of the first alert received by vehicles.

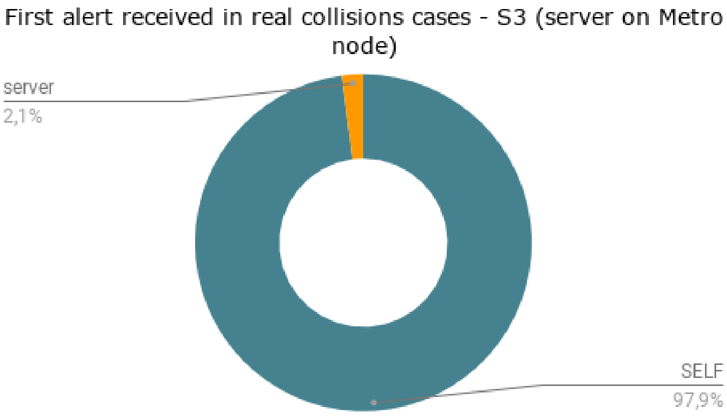


Fig. 4.12 First Alert Analysis with Real Collisions - S3

The first trend worth highlighting concerns the fact that neither in real collisions cases nor in false positive ones the first alert received is issued by an external vehicle. In fact, as for the Scenario II, the alerts are transmitted only by the server and by the vehicle itself. This behaviour could be justified by the fact that latencies are smaller for the I2V communication, achieved using unicast mode, and of course for the self-notice case compared to the ones that characterize the V2V communication. In fact, the alerts released from external vehicles are transmitted using the broadcast mode, implemented in OMNeT++ through the iteration on all the active nodes in the system. This process could be at the origin of a lower reactivity when compared to the two other methods. Generally, while the self-generated alerts are immediately received by the vehicle, the alerts coming from the outside experience some latencies. In particular, those coming from the server take 0.405 s to reach the destination vehicle, while the ones transmitted by other vehicles take in average 0.400 s. The lack of their presence among the first alerts received is therefore not a consequence of the transmission and propagation times, but it may be a consequence of the process at the origin of the alert generation. In fact, the alert coming from an outside vehicle is generated milliseconds up to seconds later with respect to the one generated by the server. A possible reason could be the nature of the vehicles to run the Collision Detection first on their own data to control if they can be involved in a collision and then proceed with the iterations on the data located in their storage.

Another explanation could be based on the larger traffic generated in the WiFi network as a consequence of the alert messages sent in broadcast. This approach could in fact generate some collisions among packets, even though the system has been built in such a way to avoid an excessive flooding of alerts in the network by limiting the transmission to one per second by each vehicle. It is still possible to notice, as for the previous scenario, a much larger number of alerts received by the vehicle's local notification system with respect to the ones sent by the server.

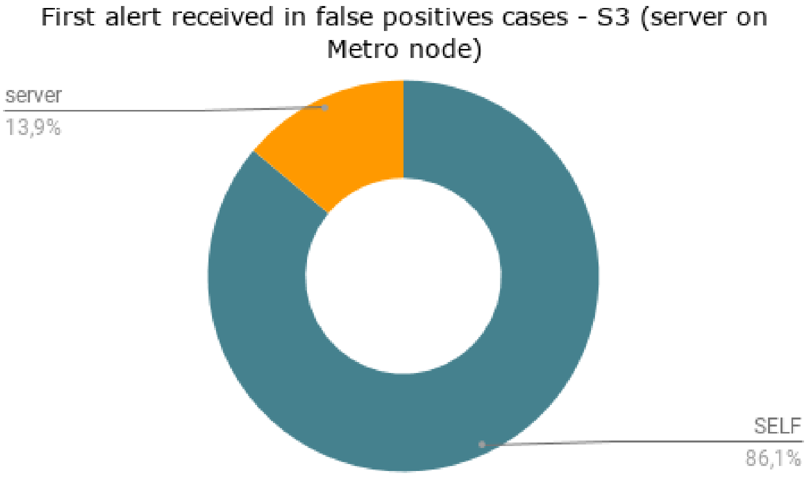


Fig. 4.13 First Alert Analysis with False Positives - S3

For what concerns the number of collisions obtained with the Scenario III, it is interesting to notice that as for the Scenario II, the number of the collisions that have not been detected is equal to zero. In fact, both scenarios implemented a solution in which the responsibility related to the detection of collisions is taken by two different types of entities, the server and the vehicles. This approach increases the possibility to detect in time possible hazardous situations. The number of collisions detected in time is this time equal to the 92.25% for the human driver scenario and 93.35% for the autonomous vehicle as shown in Fig. 4.14, thus very similar to the Scenario II results.

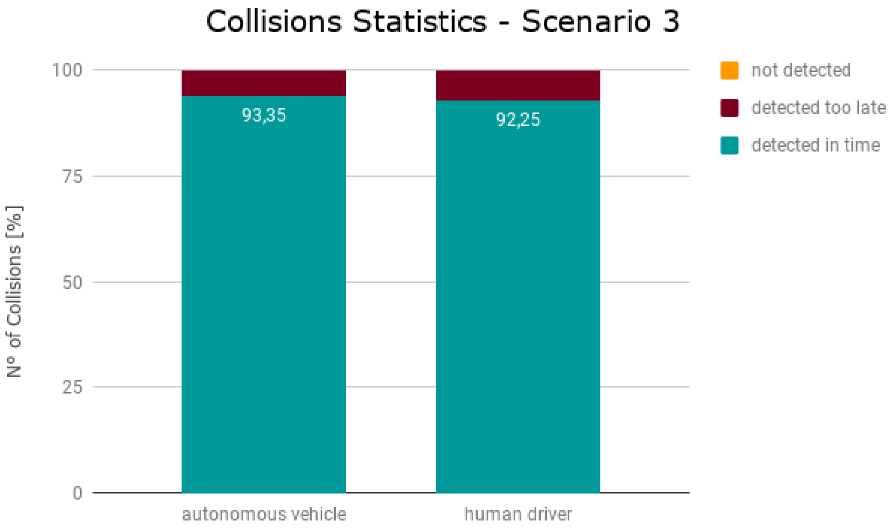


Fig. 4.14 Collision Statistics - S3

### 4.3.5 Scenario IV

The last scenario that has been studied is derived from the Scenario III, phasing the server out of the system. The *CollisionDetection* algorithm is therefore used only by the vehicles, that generate alert messages for themselves and for the surrounding entities.
The performances of this scenario in terms of the number of collisions detected in time do not change significantly compared to the ones of the Scenarios II and III. This type of behaviour was predictable due to the predominance of self-generated alerts as first type of warning received. In fact, since this type of message is still present in this scenario, it is reasonable to obtain very similar results when compared to the previous ones. In particular, a percentage of 92.65% is obtained for autonomous vehicles, while the 92.25% is achieved for human drivers. Once again, all the collisions are detected, either in time or too late, leaving the not detected value equal to 0.
For this scenario as well a consideration on the first alert received can be done, even though only the IEEE 802.11p standard is used thus no comparison among technologies can be done. The first alert received is in all cases the locally generated one as it is easy to imagine. But the external alerts generated by other vehicles in the system become essential to warn vehicles that may not be in mutual visibility and for which it is impossible to have knowledge on each other's status.

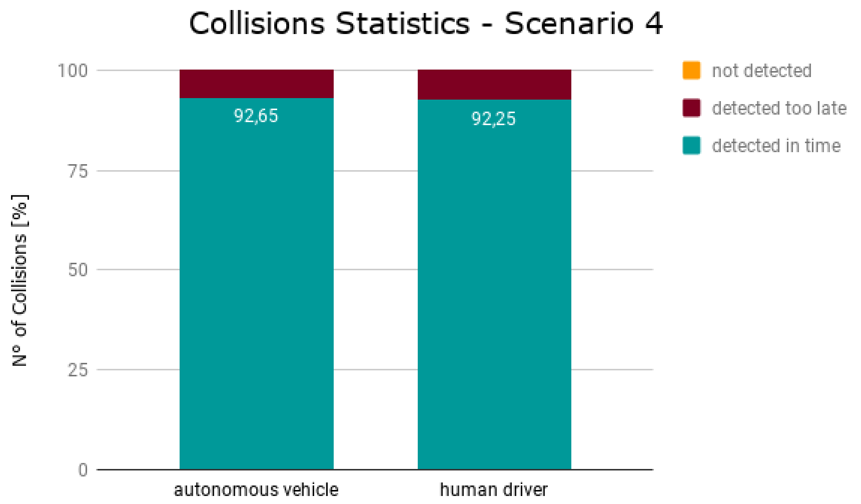


Fig. 4.15 Collision Statistics - S4

### 4.3.6 False Positives Analysis

Among all the alerts that have been generated both by vehicles and by the server, a given percentage was destined to vehicles that did not ended up colliding. Those situations can be defined as false positives, thus events that were identified as collisions but that during the post-processing analysis have been denied. In fact, in such cases the verifications performed on the polygons representing the vehicles did not show any superposition among the vehicles virtual shapes. In the four scenarios that have been analysed, the values of false positives fluctuated around the 30% of the total number of hazards predicted, and the more detailed values are shown in Fig 4.16.

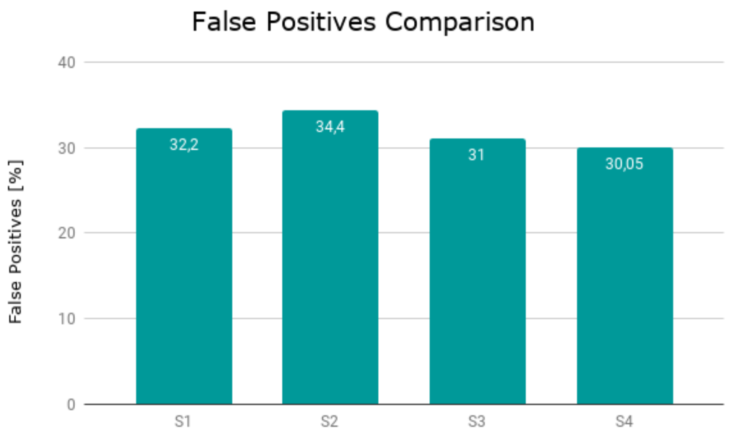


Fig. 4.16 False Positives Percentages

Generally, when the amount of false positives is too high, a system cannot be defined as reliable and well calibrated. But when considering vehicular safety applications, even false positives can be useful in the chance in which these predictions represent hazardous events, thus situations where the distance between two vehicles was small enough to be classified as dangerous. An indicator that determines whether the false positives represent hazardous situations or not is the Cumulative Distribution Function (CDF) computed from the distribution of the false positives' minimum distances. In fact, some crucial statistics can be retrieved, such as the percentile of a given distance. Analysing the CDF obtained for the results of the Scenario 3 in Fig. 4.17, it can be observed how the 80% of the false positive cases reach a minimum distance of 1 meter and that the 90% fall under a distance of 3 meters. The great majority of hazards that did not result in collisions can be therefore classified as useful for the safety purposes of the system.

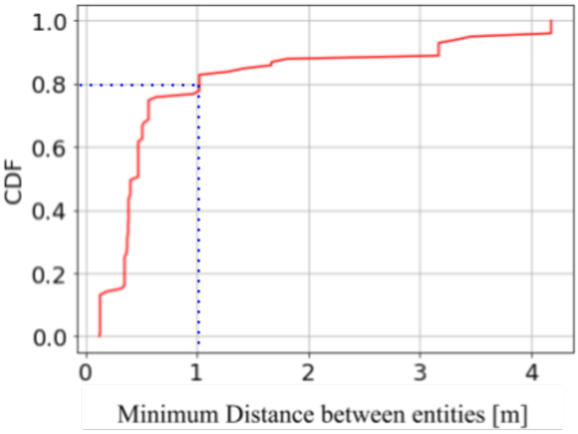


Fig. 4.17 Minimum Distance CDF in False Positive cases

### 4.3.7 General Comparison

Ultimately, some general considerations can be discussed on the behaviour of the four scenarios that have been studied. Looking at the statistics obtained for the scenarios when considering the autonomous vehicle case in Fig. 4.18, it can be noticed how only the Scenario 1 presents a small yet existing percentage of not detected collisions. As previously explained, this result is a consequence of the fact that only the server is in this case responsible for the generation of alerts. For all the other scenarios instead, at least two types of alerts are available, increasing the possibility to detect a collision in time.

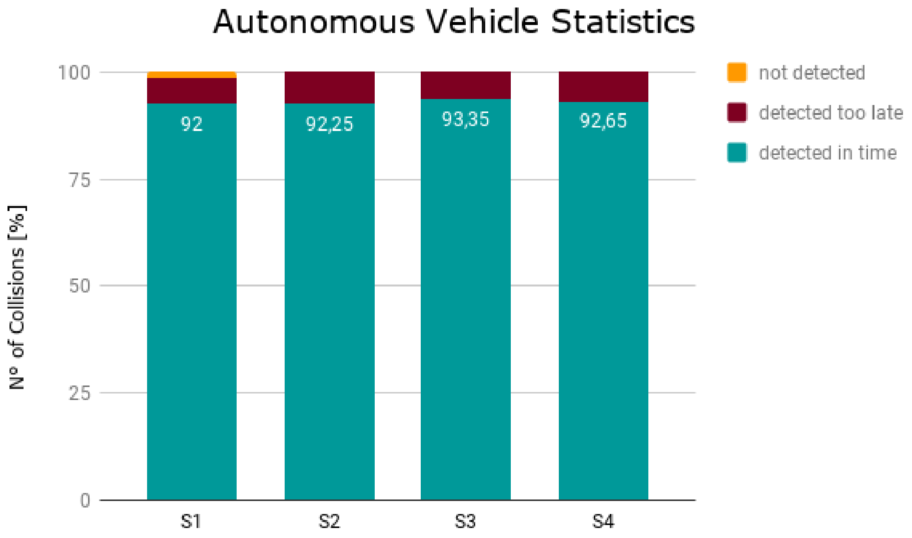


Fig. 4.18 Comparison on Collisions Statistics - Autonomous Vehicle

A similar trend is visible for the values obtained considering the human driver cases (Fig. 4.19), where slightly lower percentages of collisions detected in time are verified.

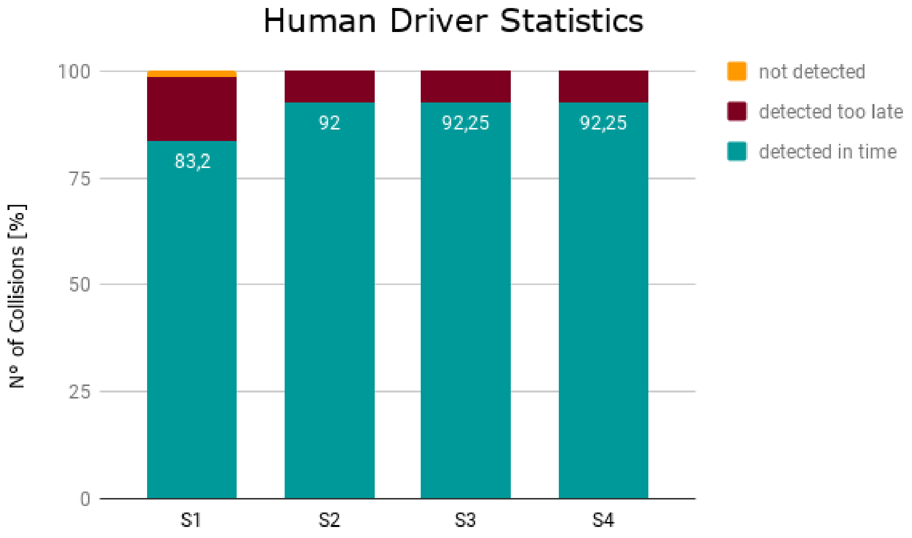


Fig. 4.19 Comparison on Collisions Statistics - Human Driver

Another important evaluation to consider is the quantification of the improvements given by the autonomous vehicle scenarios with respect to the human driver cases. For each scenario, an increase in the detection percentage of the autonomous vehicle statistics is verified. More in particular, the higher improvement is obtained for the Scenario I for which the collisions detected in time increase by 8.8%. The second best improvement is scored by the Scenario 3 where a growth of 1.1% is verified. For what concerns the scenario 2 and 4, the improvements are less significant, and reach the 0.25% and 0.4% respectively. The larger improvements could be experimented by the scenarios in which the the alerts take a longer time to be received, leaving the driver less available time to brake and stop the vehicle. In fact, in those situations the effects of an autonomous vehicle can have more influence with respect to the ones in which more time is available. The Scenario I, that only used the server in order to run the *CollisionDetection* algorithm and therefore send alerts is in fact the scenario that experienced the larger latencies, proved by the low percentage of collisions detected in time in the human driver case.

# Chapter 5

# Conclusion

In this thesis, the performances of an Intersection Collision Avoidance system have been tested in four scenarios enabled by the V2V communication using IEEE 802.11p, and V2I communication via the LTE cellular network. The development of the ICA application followed four different models:

- Scenario I in which the Collision Detection is only located on the server, generating an alert messages transmission in V2I;
- Scenario II in which the Collision Detection is located on the server but also distributed inside vehicles, generating two types of alert messages: a V2I transmission from the server and a locally generated alert by the vehicle itself;
- Scenario III that completes the second one introducing a third type of alert, broadcasted in V2V by the vehicles;
- Scenario IV in which the server is no longer part of the system, leaving only the self-generated alert and the V2V ones.

The results obtained from the four scenarios highlight a general trend for which the higher percentage of avoided collisions is obtained with mixed scenarios, as the II and III, for which at least two types of alert messages are received by vehicles using both the cellular and WiFi network. The Scenario IV also presented very good statistics with respect to the Scenario I, behaviour that could lead to characterize the V2I transmission of alerts as less effective than the V2V and local ones. In fact, the even though the lower performances of the Scenario I can be explained by the fact that the server is the only entity running the Collision Detection, the Scenarios II and III in which the both the server and the vehicles work on the prediction do not present substantial improvements with respect to the Scenario IV in which the server is not present. If the purpose is to increase as much as possible the amount of avoided collisions, the Scenario III is the best suited. But if the objective is to simplify the vehicle architecture, avoiding the insertion of a LTE receiver, the Scenario IV has very similar performances even considering the absence of the infrastructure.

Furthermore, all the scenarios have been assessed both considering the human driver and the autonomous vehicle cases. The number of collision detected in time is generally larger when considering an autonomous vehicle, since the time available to take an action in view of a possible collision is larger.

For what concerns the reliability of the system, the study of the minimum distance between false positives proved that the 80% of false positives reaches a minimum distance lower than 1 m, proving that the false positives still represent hazardous situations.

In the future, the same simulations could be augmented by adding informations retrieved by GPS on-board receivers in order to increase the precision of the timing and position data. The simulations could also be extended in order to include the Vehicle-to-Pedestrian (V2P) communication, considering smartphones as pedestrians UEs. In this case, the LTE communication could be adopted for the V2P transmissions, generating higher traffic on the cellular network. In this case, it would be interesting to assess the performances of the system when both V2I and V2P communications are based on the cellular technology. The growth of the LTE traffic could generate packet losses, thus reduce the effectiveness of the ICA system. Finally, a Penetration Rate (PR) study can be performed in order to understand the consequences of the extension of the technologies adopted on the performances of the system.

# References

[1] Association for safe international road travel. http://asirt.org/Initiatives/Informing-Road-Users/Road-Safety-Facts/Road-Crash-Statistics, 2015.

[2] UN General Assembly. Global plan for the decade of action for road safety 2011-2020. page 25, 03 2010.

[3] Together for safer roads. http://www.togetherforsaferroads.org/4-ways-road-crashes-impact-the-economy/, 2016.

[4] Y. L. Tseng. Lte-advanced enhancement for vehicular communication. *IEEE Wireless Communications*, 22(6):4–7, Dec 2015.

[5] J. Schlienz and A. Roessler. Device to device communication in lte. Technical Report 1MA264_0e, ROHDE & SCHWARZ, 2016.

[6] 3rd Generation Partnership Project. 3rd generation partnership project technical specification group radio access network study on lte-based v2x services (release 14). Technical Report TR 36.885 V14.0.0, 3GPP, 2016.

[7] Alessandro Bazzi, Barbara M. Masini, Alberto Zanella, and Ilaria Thibault. On the performance of ieee 802.11p and lte-v2v for the cooperative awareness of connected vehicles $l_1$. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 66, 2017.

[8] Lusheng Miao, Karim Djouani, Barend Van Wyk, and Yskandar Hamam. A survey of ieee 802.11p mac protocol. 09 2011.

[9] L. Ward and Dr. M. Simon. Intelligent transportation systems using ieee 802.11p - application note. Technical Report 6.2015-1MA153_4e, ROHDE & SCHWARZ, 2015.

[10] Shereen A. M. Ahmed, Sharifah H. S. Ariffin, and Norsheila Fisal. Overview of wireless access in vehicular environment (wave) protocols and standards. *Indian Journal of Science and Technology*, 2013.

[11] *Dedicated Short Range Communications (DSRC) Message Set Dictionary™ Set*, mar 2016.

[12] Andreas Festag. Standards for vehicular communication - from ieee 802.11p to 5g. *Elektrotechnik & Informationstechnik*, 132, 2015.

[13] ETSI. Draft etsi en 302 663 v1.2.0 (2012-11) - intelligent transport systems (its); access layer specification for intelligent transport systems operating in the 5 ghz frequency band. Technical Report N 302 663 V1.2.0, ETSI, 2012.

[14] Zhigang Xu, Xiaochi Li, Xiangmo Zhao, Michael H. Zhang, and Zhongren Wang. Dsrc versus 4g-lte for connected vehicle applications: A study on field experiments of vehicular communication performance. *Journal of Advanced Transportation*, 2017.

[15] A. Vinel. 3gpp lte versus ieee 802.11p/wave: which technology is able to support cooperative vehicular safety applications? *IEEE Wireless Communications Letters*, 1, 2012.

[16] Z. H. Mir and F. Filali. Lte and ieee 802.11p for vehicular networking: a performance evaluation. *EURASIP Journal on Wireless Communications and Networking*, 89, 2014.

[17] G. Avino, M. Malinverno, C. Casetti, C. F. Chiasserini, F. Malandrino, M. Rapelli, and G. Zennaro.

[18] Manjyot Saini and Harjit Singh. Vanet, its characteristics, attacks and routing techniques: A survey. *International Journal of Science and Research (IJSR)*, 2015.

[19] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. page 60, 01 2008.

[20] F. Hagenauer, F. Dressler, and C. Sommer. Poster: A simulator for heterogeneous vehicular networks. In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 185–186, Dec 2014.