POLITECNICO DI TORINO

Master Degree in Communications and Computer Networks Engineering

Tesi di Laurea Magistrale

Performance of MEC Solutions in Automotive Applications

Relatore
Prof. Carla Fabiana Chiasserini

Correlatore:
Prof. Matteo Cesana

Candidato
Leonardo Lo Schiavo
ID number: s237082

Ottobre 2018

Sommario

L'elaborato di tesi concerne l'analisi di algoritmi sviluppati nell'ambito delle comunicazioni veicolari, mirati alla riduzione delle collisioni tra i veicoli al fine di garantire maggiore sicurezza stradale. Il focus del lavoro è sulla valutazione di parametri e performance di tali algoritmi attraverso una campagna simulativa basata sull'utilizzo di due diverse piattaforme di simulazione: il simulatore di tecnologie di rete OMNeT++ e il simulatore di traffico veicolare SUMO. L'approccio utilizzato consente ai due simulatori di comunicare reciprocamente in tempo reale affinché le condizioni di traffico possano influenzare il comportamento dei moduli implementati nel simulatore di rete e viceversa le azioni decise dall'attuazione degli algoritmi sviluppati in rete possano modificare le tracce di traffico dei veicoli coinvolti.
Il problema viene affrontato su incroci urbani considerando soluzioni centralizzate basate sulla tecnologia cellulare secondo il paradigma Multi-access Edge Computing (MEC), soluzioni di tipo distribuito sia con tecnologia cellulare sia con tecnologia wireless a corto raggio ed inoltre soluzioni di tipo ibrido in cui la tecnologia cellulare e la tecnologia wireless a corto raggio vengono integrate. Le soluzioni proposte vengono applicate sia ad automobili dotate di sistemi a frenata autonoma sia ad automobili con nessun livello di automazione al fine di determinare gli impatti di tali tecnologie sui risultati delle simulazioni.

Summary

The thesis work concerns the analysis of algorithms developed in the context of vehicular communications aimed at the reduction of the collisions among vehicles to guarantee higher levels of road safety. The focus of the work is on the evaluation of parameters and performances of such algorithms through a simulation campaign based on the utilization of two different simulative platforms: the simulator of network technologies OMNeT++ and the road traffic simulator SUMO. The adopted approach allows the two simulators to reciprocally communicate in real-time in such a way that traffic conditions may influence the behaviour of the modules implemented on the network simulator and the other way around the actions decided by the algorithms deployed in the network modify the traffic flows of the involved vehicles.

The problem is tackled on urban intersections by considering centralized solutions based on the cellular technology under the Multi-access Edge Computing (MEC) paradigm, distributed solutions leveraging both cellular technology and short-range wireless technology and also hybrid solutions in which cellular and short-range wireless technologies are integrated together. The proposed solutions will be applied either to cars equipped with autonomous braking systems and to cars without any level of automation to determine the impacts of such technologies on the results of the simulations.

Acknowledgements

I would like to thank the people I met in Telecom Italia, the company where I developed my thesis work from March 2018 to October 2018.

I want to start from my supervisors, Dr. Salvatore Scarpina and Dr. Marco Caretti. I wish everyone could work with them; they taught me how to think as an engineer and to adopt a critical attitude towards the problems to solve.

Thanks also to my colleagues at Politecnico di Torino: their collaboration and support helped me going through the hard studying days.

Finally, thanks to the people who have always been by my side: my family and my lifelong friends.

Contents

List of Tables VIII

List of Figures IX

List of Acronyms XI

1 Thesis Opening 1
1.1 Thesis outline . . . 1
1.2 Thesis structure . . . 3

2 Standard specifications for vehicular communications 5
2.1 Background . . . 5
2.2 The V2X communication . . . 5
2.3 The WLAN-based V2X . . . 6
2.3.1 DSRC Standard . . . 8
2.3.2 C-ITS Standard . . . 9
2.4 The Cellular based V2X . . . 11

3 V2X scenarios description 19
3.1 ETSI Intersection Collision Risk Warning . . . 19
3.2 Collision detection algorithm [23] . . . 20
3.2.1 Centralized collision detection algorithm [23] . . . 21
3.2.2 Distributed collision detection algorithm [23] . . . 22
3.3 General considerations for each scenario . . . 23
3.3.1 Reference road environment . . . 23
3.3.2 Vehicle and driver parameters . . . 24
3.3.3 Applicative messages parameters . . . 25
3.4 Cellular V2N scenario [23] . . . 25
3.4.1 Multi-access Edge Computing (MEC) [27] . . . 26
3.4.2 Server-eNodeB latency . . . 26
3.4.3 Application Messages . . . 27
3.5 Cellular V2V Scenario . . . 27
3.6 802.11p V2V Scenario . . . 28
3.7 Hybrid V2N Scenario . . . 28

4 Simulation setup and methodology 31
4.1 Simulation tools 31
4.2 Bidirectional coupling of OMNeT++ and SUMO 31
4.2.1 Traffic Control Interface 33
4.3 Simulation Methodology 34
4.3.1 Actions performed upon alert reception 35
4.3.2 Performances evaluation 36
4.3.3 Methodology Constraints 37
4.4 Scenarios implementation 38
4.4.1 Implementation of Cellular V2N scenario [23] 38
4.4.2 Implementation of Cellular V2V scenario 41
4.4.3 Implementation of 802.11p-based V2V scenario 43
4.4.4 Implementation of Hybrid V2N Scenario 44

5 Simulation Results 47
5.1 SUMO simulations results 47
5.2 Cellular V2N simulation results 48
5.2.1 C-V2N with Autonomous Braking 48
5.2.2 C-V2N with Human Driver 53
5.3 Cellular V2V simulation results 55
5.3.1 C-V2V with Autonomous Braking 55
5.3.2 C-V2V with Human Driver 58
5.4 802.11p-based V2V simulation results 60
5.4.1 802.11p-based V2V with Autonomous Braking 60
5.4.2 802.11p-based V2V with Human Driver 62
5.5 Hybrid V2N simulation results 64
5.5.1 Hybrid V2N with Autonomous Braking 65
5.5.2 Hybrid V2N with Human Driver 67
5.6 Application messages analysis 69

6 Conclusions 73
6.1 Remarks on simulation results 73
6.2 Future work proposals 73

Bibliography 75

List of Tables

3.1 Vehicle and driver parameters 25
3.2 ETSI CAM and DENM parameters 27
3.3 802.11p V2V Scenario specific parameters 28

5.1 SUMO simulations results 48
5.2 C-V2N with Autonomous Braking parameters 48
5.3 C-V2N with Autonomous Braking: Collision avoidance numerical results . . 51
5.4 C-V2N with Autonomous Braking: Detections quality numerical results . . 52
5.5 C-V2N with Human Driver parameters 53
5.6 C-V2N with Human Driver: Collision avoidance numerical results 54
5.7 C-V2N with Human Driver: Detections quality numerical results 54
5.8 C-V2V with Autonomous Braking: Collision avoidance numerical results . . 56
5.9 C-V2V with Autonomous Braking: Detections quality numerical results . . 57
5.10 C-V2V with Human Driver: Collision avoidance numerical results 58
5.11 C-V2V with Human Driver: Detections quality numerical results 59
5.12 802.11p-based V2V with Autonomous Braking: Collision avoidance numerical results 61
5.13 802.11p-based V2V with Autonomous Braking: Detections quality numerical results 61
5.14 802.11p-based V2V with Human Driver: Collision avoidance numerical results 63
5.15 802.11p-based V2V with Human Driver: Detections quality numerical results 64
5.16 Hybrid V2N with Autonomous Braking parameters 65
5.17 Hybrid V2N with Autonomous Braking: Collision avoidance numerical results 65
5.18 Hybrid V2N with Autonomous Braking: Detections quality numerical results 67
5.19 Hybrid V2N with Human Driver: Collision avoidance numerical results . . . 67
5.20 Hybrid V2N with Human Driver: Detections quality numerical results . . . 69
5.21 Numerical results for the reception rate of transmitted alerts 70
5.22 Numerical results for the reception rate of transmitted beacons 70

List of Figures

2.1 V2X communication types (taken from [2]) 6
2.2 V2V and V2I communications in VANET (taken from [7]) 7
2.3 DSRC protocol stack and related standards (taken from [3]) 9
2.4 C-ITS protocol stack and related standards (taken from [3]) 11
2.5 V2V communications through Uu interface (taken from [20]) 12
2.6 V2P and P2V communications through Uu interface (taken from [20]) . . . 13
2.7 V2I and I2V communications through Uu interface (taken from [20]) 13
2.8 V2N communications through Uu interface (taken from [20]) 14
2.9 V2V communications through PC5 interface (taken from [20]) 14
2.10 V2I and I2V communications through PC5 interface (taken from [20]) . . . 15
2.11 V2P and P2V communications through PC5 interface (taken from [20]) . . 15
2.12 Distributed scheduling through PC5 interface (taken from [21]) 16
2.13 ENodeB scheduling through Uu interface (taken from [21]) 17

3.1 End-to-end latency for ICRW application (taken from [17]) 20
3.2 Centralized collision detection algorithm (adapted from [23]) 22
3.3 Distributed collision detection algorithm 23
3.4 Reference road environment 24

4.1 State machines of OMNeT++ and SUMO (taken from [36]) 32
4.2 Syntax for changing the vehicle's state (taken from [38]) 33
4.3 Implementation of the command to set vehicle's deceleration 33
4.4 Syntax for getting vehicle's variables (taken from [39]) 34
4.5 Implementation of the command to get vehicle's magnitude of speed 34
4.6 Implementation of the function called upon alert reception 35
4.7 OMNeT++ implementation of C-V2N scenario 39
4.8 Connection of modules in the OMNeT++ Hybrid V2N scenario 44
4.9 OMNeT++ implementation of the RSU in the Hybrid V2N scenario 45

5.1 C-V2N with Autonomous Braking: Collision avoidance outcomes 49
5.2 C-V2N with Autonomous Braking: Detections quality 50
5.3 C-V2N with Autonomous Braking: Collision avoidance outcomes by varying eNodeB-server latency 51

5.4 C-V2N with Autonomous Braking: Detections quality by varying eNodeB-server latency . . . 52
5.5 C-V2N with Human Driver: Collision avoidance outcomes . . . 53
5.6 C-V2N with Human Driver: Detections quality . . . 55
5.7 C-V2V with Autonomous Braking: Collision avoidance outcomes . . . 56
5.8 C-V2V with Autonomous Braking: Detections quality . . . 57
5.9 C-V2V with Human Driver: Collision avoidance outcomes . . . 58
5.10 C-V2V with Human Driver: Detections quality . . . 59
5.11 802.11p-based V2V with Autonomous Braking: Collision avoidance outcomes 60
5.12 802.11p-based V2V with Autonomous Braking: Detections quality . . . 62
5.13 802.11p-based V2V with Human Driver: Collision avoidance outcomes . . . 63
5.14 802.11p-based V2V with Human Driver: Detections quality . . . 64
5.15 Hybrid V2N with Autonomous Braking: Collision avoidance outcomes . . . 66
5.16 Hybrid V2N with Autonomous Braking: Detections quality . . . 66
5.17 Hybrid V2N with Human Driver: Collision avoidance outcomes . . . 68
5.18 Hybrid V2N with Human Driver: Detections quality . . . 68
5.19 Reception rate of transmitted alerts in the different scenarios . . . 69
5.20 Reception rate of transmitted beacons in the different scenarios . . . 71

List of Acronyms

3GPP .... Third Generation Partnership Project

API .... Application Programming Interface

BSM .... Basic Safety Message

C-ITS .... Cooperative Intelligent Transport Systems

C-V2N .... Cellular V2N

C-V2V .... Cellular V2V

CAM .... Cooperative Awareness Message

CCA .... Cooperative Collision Avoidance

CCH .... Control Channel

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

D2D .... Device-to-Device

DENM .... Distributed Environmental Notification Message

DSRC .... Dedicated Short-Range Communications

E-UTRAN .... Evolved Universal Terrestrial Radio Access Network

EDCA .... Enhanced Distributed Channel Access

ENodeB .... Evolved Node B

ETSI .... European Telecommunications Standard Institute

EWM .... Emergency Warning Message

HMI .... Human-to-Machine Interface

ICA .... Intersection Collision Announcement

ICA .... Intersection Collision Avoidance

ICRW .... Intersection Collision Risk Warning

IEEE .... Institute of Electrical and Electronics Engineers

IP .... Internet Protocol

IT .... Information Technology

LOS .... Line-of-Sight

LTE .... Long Term Evolution

MAC .... Medium Access Control

MEC .... Multi-access Edge Computing

NED .... Network Description

NHTSA .... National Highway Traffic Safety Administration

NIC .... Network Interface Card

NLOS .... Non-Line-of-Sight

OBU .... On-Board-Unit

OFDM .... Orthogonal Frequency Division Multiplexing

PGW .... Packet Data Network Gateway

QoS .... Quality of Service

RAN .... Radio Access Network

RB .... Resource Block

RSU .... Road Side Unit

SAE .... Society of Automotive Engineers

SCH .... Service Channel

SINR .... Signal-to-Interference-plus-Noise Ratio

TCP .... Transmission Control Protocol

TraCI .... Traffic Control Interface

UDP .... User Datagram Protocol

UE .... User Equipment

V2I .... Vehicle-to-Infrastructure

V2N . . . . Vehicle-to-Network

V2P . . . . Vehicle-to-Pedestrian

V2V . . . . Vehicle-to-Vehicle

V2X . . . . Vehicle-to-everything

VANET . . . . Vehicular *Ad hoc* Network

WAVE . . . . Wireless Access in Vehicular Environment

WBSS . . . . WAVE Basic Service Set

WLAN . . . . Wireless Local Area Network

WSMP . . . . WAVE Short Message Protocol

Chapter 1

Thesis Opening

This chapter is written to give a general insight into the thesis work. Motivations, reasons and description of the work are presented, together with the general structure of the thesis.

1.1 Thesis outline

The recent expansion of metropolitan areas has determined the increase of socio-economic problems in many parts of the globe, in terms of vehicular accidents and urban congestion. To tackle such problems, nowadays information and communication technologies are at the core of systems aiming at improving safety and transportation efficiency in a sustainable manner. Since lethal vehicular accidents are the cause of death for millions of people in the world, the development of safety applications in the context of vehicular communications could bring practical benefits to the society.

For this reason, this thesis work concerns the analysis of the performances and parameters of a road safety application for urban intersections aimed at the reduction of the number of collisions among vehicles to guarantee higher levels of road safety. More specifically, the focus of the work is on the evaluation of such application for vehicular communications through a wide simulation campaign based on the utilization of two different simulative platforms: the simulator of network technologies OMNeT++ and the road traffic simulator SUMO. The adopted approach allows the two simulators to reciprocally communicate in real-time by exploiting a programming interface: in this way, traffic conditions influence the behaviour of the modules implemented in the network simulator and the other way around the actions decided by the algorithms deployed in the network modify the traffic flows of the involved vehicles.

The problem is tackled on urban intersection through centralized solutions based on LTE-cellular technology under the Multi-access Edge Computing (MEC) paradigm, in which an application server is responsible to collect the mobility information of vehicles subscribed to the collision avoidance application and eventually generate warning messages to vehicles potentially involved in a collision course. In this way, the drivers or eventual autonomous braking systems in the vehicles can take proper actions to avoid the potentially incoming collision. MEC paradigm results suitable for this kind of delay-sensitive safety applications since it can ensure low latency and high reliability by shifting computational resources and required functionalities closer to the mobile edge.

The thesis work also considers two distributed solutions for the safety application at urban intersection by leveraging two main technologies: on one side, the cellular technology for the vehicular environment providing support to direct vehicle-to-vehicle communications by exploiting the LTE assisted Device-to-Device standard of the 3GPP Release 12; on the other side, the Dedicated Short-Range Communications (DSRC) technology that natively supports *ad hoc* communications among vehicles and Road Side Units (RSUs) through the IEEE 802.11p standard. More specifically, a distributed version of the collision avoidance algorithm has been implemented to evaluate the performances and the main trade-offs of the aforementioned technologies in the proposed safety application. In general, distributed solutions are believed to be better for safety applications, since direct communications offer lower latencies than the communications passing through a network infrastructure. Nevertheless, the presence of Non-Line-of-Sight (NLOS) conditions between the transmitter and the receiver limits the reliability of the transmissions, which is something unacceptable in any safety application.

Furthermore, the thesis work foresees the possibility to integrate LTE-based cellular technology with 802.11p-based technology by creating a simulation scenario in which a MEC server exchanges packets with both the LTE base station and the 802.11p RSU. In this way, a collision avoidance application is deployed in an heterogenous environment in which vehicles might be equipped with On-Board-Units (OBUs) enabling LTE communications or with OBUs enabling 802.11p-based communications. Since in many parts of the world 802.11p is considered as the *de facto* standard for vehicular communications and 802.11p-based vehicles will be mandatory by 2020 in the United States, such scenario might be of crucial importance to avoid the divergence of just one technology in the context of vehicular communications and to enable new business models around the integration of the two technologies under the MEC paradigm.

Several simulations have been performed to test how far the centralized and distributed solutions support such vehicular safety application, and the main advantages and disadvantages of the two different approaches have been highlighted. At the end, considering simulations with only vehicles equipped with autonomous braking systems and simulations with only human drivers, it resulted that the centralized solution provides the best support for the collision avoidance application at urban intersection, thanks to the reliability and low latency ensured by the MEC paradigm. On the other hand, the distributed solutions are too much penalized by the unreliability of transmissions due to NLOS conditions between transmitters and receivers.

1.2 Thesis structure

The structure of the thesis work is summarized below.

Chapter 1: It is the current chapter, providing information about the reasons behind the choice of the work, the description of the work and the structure of the thesis.

Chapter 2: A theoretical introduction about the standard specifications for vehicular communications is given.

Chapter 3: This chapter is dedicated to the description of the vehicular communications scenarios considered to deploy the safety application.

Chapter 4: Simulation setup and methodology are presented in this chapter, by explaining how the software tools work and what has been implemented to simulate properly the scenarios of interest.

Chapter 5: The results of the simulation campaign are reported and commented in this chapter.

Chapter 6: This chapter deals with the remarks about the obtained simulation results and provides future work proposals.

Chapter 2

Standard specifications for vehicular communications

The chapter provides an overview about the main standard specifications adopted for vehicular communications to properly share information and support a wide range of applications. In such context, both cellular technology and short-range technology will be analysed, and potential advantages and eventual trade-offs of both approaches will be considered.

2.1 Background

The recent fast growth of metropolitan areas in many parts of the world, determining several socio-economic issues in terms of vehicular crashes and congestions within urban streets and highways, has fostered the development of systems which leverage information and communication technologies to improve road safety and transportation efficiency while keeping a sustainable impact on the environment. In this sense, since according to the World Health Organization lethal traffic accidents were the cause of death for about 1.4 million people in 2016 [1], the development of delay-critical safety applications in the context of vehicular communications might result to be significant for its potential social benefits.

2.2 The V2X communication

The Vehicle-to-everything (V2X) communication refers to a set of standards and technologies allowing the interaction between vehicles and other entities to exchange useful information which makes possible the development of a multitude of interesting services and applications, such as automated driving assistance, optimized road usage and collision avoidance, with the final goal of enhancing road safety and traffic efficiency and optimizing the energy savings. More specifically, considering both Wireless Local Area Network (WLAN) technology and cellular technology as underlying technologies used to support

V2X communication, it is possible to distinguish among four main different communication types according to the specific entity that interacts with the vehicle:

- Vehicle-to-Vehicle (V2V);
- Vehicle-to-Pedestrian (V2P);
- Vehicle-to-Infrastructure (V2I);
- Vehicle-to-Network (V2N).

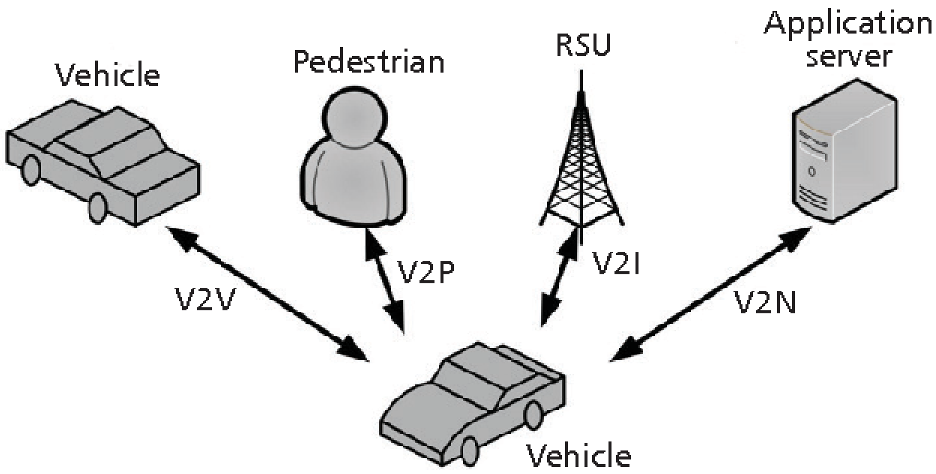

Figure 2.1. V2X communication types (taken from [2])

2.3 The WLAN-based V2X

The WLAN-based V2X communication technology has reached a mature stage thanks to the extensive standardization process of the recent past years, which nowadays makes such technology capable to meet the most challenging V2X applications' requirements. Therefore, in the next few years it is expected to deploy basic and improvable V2X systems [3]. More specifically, in the United States the standard to support WLAN-based V2X is the Dedicated Short-Range Communication (DSRC) wireless technology, which according to the U.S. National Highway Traffic Safety Administration (NHTSA) must be mandatory for new deployed vehicles by 2020. Whereas in Europe, the European Telecommunications Standard Institute (ETSI) has provided support to vehicular wireless short-range communications with the Cooperative Intelligent Transport Systems (C-ITS) Communications architecture [4] using at the access layer the so called ITS-G5 standard [5].
It is important to state that the two approaches have many technical similarities, since both rely on IEEE 802.11p [6], which provides wireless access in vehicular environments in the 5.9 GHz band by creating the so-called Vehicular *Ad Hoc* Networks (VANETs). Such

VANETs have as building blocks On-Board Units (OBUs), intended as the communication facilities on the vehicles, and Road Side Units (RSUs), which are the stationary infrastructure units deployed within the transportation environment, and they are characterized by *ad hoc* direct communications between OBUs or between OBUs and RSUs.
Therefore, such networks directly support two vehicular communications types: Vehicle to Vehicle (V2V) communications and Vehicle to Infrastructure (V2I) communications when vehicles communicate with the RSU, as it is possible to observe in the Figure 2.2. It must be noticed that in general V2V communications are low-cost communications because they do not require the high costs for deploying and maintaining roadside infrastructure, and that such communications might be suitable for applications with hard delay constraints (safety applications) because they provide short latency by avoiding the additional delay introduced by the multi-hop nature of the V2I communications. On the other side, V2I communications usually show a longer duration than V2V communications (indeed vehicles' mobility and speed cause frequent discontinuous and short-lived connections also in the V2I case) and therefore they could be suitable for applications requiring high data rates and moderate delay requirements.

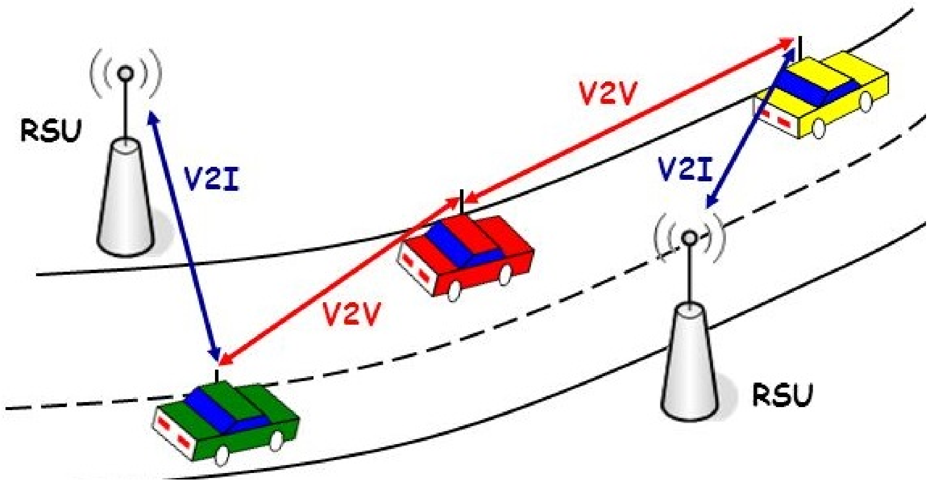

Figure 2.2. V2V and V2I communications in VANET (taken from [7])

In general, VANETs built upon 802.11p potentially support two different classes of application: safety applications, requiring timely and precise context awareness information, and non-safety infotainment applications, aiming to provide entertainment and comfort to passengers at the cost of higher data rates requirements than the safety applications [8]. On one hand, examples of safety applications are Cooperative Collision Avoidance (CCA), Emergency Warning Messages (EWM), Intersection Collision Avoidance (ICA), Platooning; on the other hand, examples of non-safety applications are Multimedia streaming, Internet Access through RSU, Advertisement, Parking Availability and many others [9].

2.3.1 DSRC Standard

DSRC Vehicular communications are based on a comprehensive set of standards, including both traditional standards and standards developed by IEEE and Society of Automotive Engineers (SAE) to properly meet the requirements of many V2X use cases. More specifically, the IEEE 802.11 WLAN working group developed the IEEE 802.11p [6] standard for the physical and lower Medium Access Control (MAC) layers; the IEEE 1609 DSRC working group developed the family of IEEE 1609.x standards for the upper layers up to the transport layer; the SAE DSRC technical committee instead standardized the V2X message sets and the related performance requirements at the facilities layer. Usually, the aggregation of IEEE 802.11p standard with the family of IEEE 1609.x standards is known as Wireless Access in Vehicular Environment (WAVE) standard.

Starting from the physical layer, DSRC operates in the 5.9 GHz band ranging from 5.850 GHz to 5.925 GHz on seven 10 MHz-wide DSRC dedicated frequency channels. Among these channels, one is a Control Channel (CCH) dedicated to control and safety data, and the other six are Service Channels (SCHs) dedicated to non-safety data. Moreover, DSRC uses Orthogonal Frequency Division Multiplexing (OFDM) with doubled time parameters with respect to the traditional Wi-Fi to increase the robustness against fading and inter-carrier interference due to the Doppler effect caused by fast moving vehicles [10].

For what concerns the lower MAC layer, DSRC uses the IEEE 802.11p standard that relies on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) based Enhanced Distributed Channel Access (EDCA), introduced to provide Quality of Service (QoS) in 802.11e. Moreover, as 802.11p radio interfaces are still half-duplex and multiple channels might be accessed, a time division approach is needed to create respectively a CCH interval and a SCH interval (whose duration is for both 50ms) to operate with two different instances of EDCA protocol either in the CCH or in one of the SCHs. Therefore, a coordination of the different entities within the same coverage area is needed: for this purpose, an upper MAC protocol standardized as IEEE 1609.4 [11] manages such synchronization for vehicles belonging to the same WAVE Basic Service Set (WBSS) by means of a Global Positioning System (GPS) time reference, which allows to all the vehicles to operate at the same time in the CCH interval or in the SCH interval. One of the main drawbacks of this approach is that safety messages suffer of additional delay when they are issued in the SCH interval, as they need to wait for the next CCH interval to be properly transmitted over the unique CCH channel at which every vehicle is listening.

Considering the network and transport layer, the DSRC standard has two parallel sets of protocols: on one side, User Datagram Protocol (UDP), Transmission Control Protocol (TCP) and Internet Protocol (IP); on the other side the WAVE Short Message Protocol (WSMP) standardized by IEEE 1609.3 [12]. The usage of one specific set of protocols depends on the size of the upper layer messages, as the standard specifies that the maximum size of a WSMP packet must be of 1400 Bytes. Therefore, since usually safety messages are very small, they are encapsulated as WSMP packets, whose header of few bytes allows to introduce a very low overhead to better cope with the tight delay requirements of time-sensitive safety applications. Instead, usually non-safety messages have a very big size and thus they must go through the part of the stack with UDP/TCP and IP protocols.

Regarding the facilities layer, the syntax and semantics of V2X messages are specified by the SAE J2735 standard [13], among which significant importance is given to the Basic Safety Message (BSM): this message transports crucial state information about the sending vehicle in terms of position and dynamics. Moreover, the standard specifies that BSMs must be sent periodically with a maximum rate of 10Hz.

In the following figure (Figure 2.3), it is possible to observe the DSRC protocol stack and related standards:

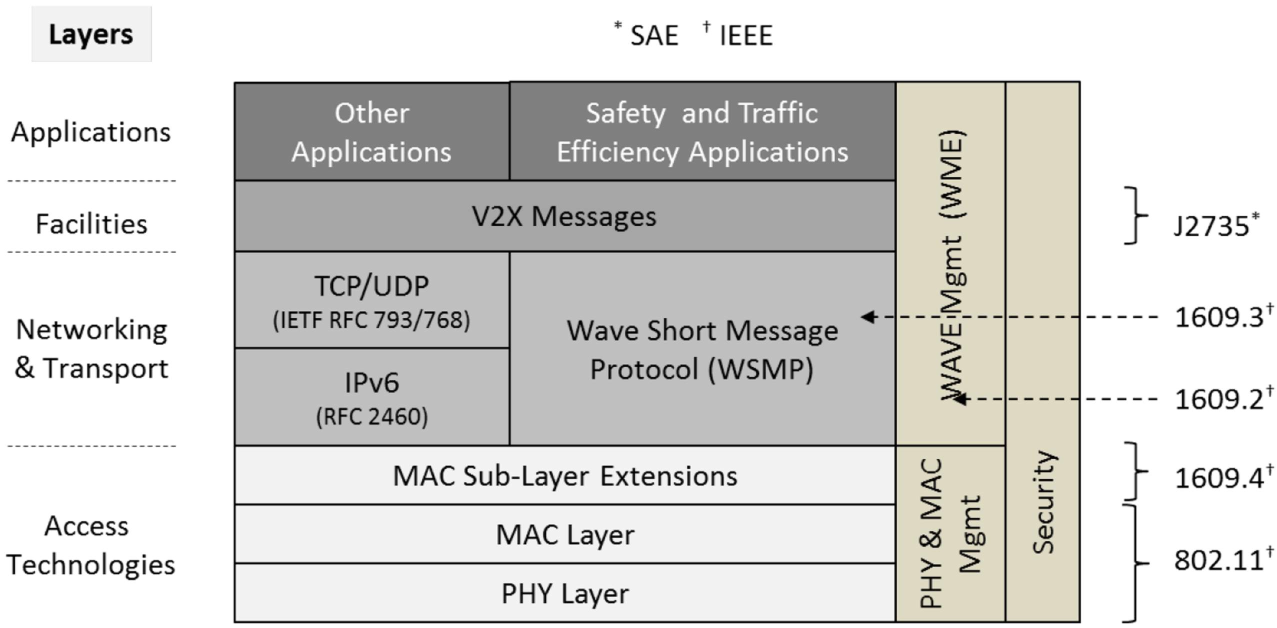

Figure 2.3. DSRC protocol stack and related standards (taken from [3])

2.3.2 C-ITS Standard

The European standard for V2X communications has been deployed in parallel with the equivalent standard in the United States, and for this reason the two standards present similarities in many aspects and the principal differences are especially at the upper layers, from the network layer to the facilities layer.

At the physical and lower MAC layer the C-ITS stack has an equivalent version of the IEEE 802.11p that is denominated ITS-G5: it operates in the 5.9 GHz frequency band, uses the same OFDM scheme of DSRC at the physical layer and adopts at the lower MAC layer an EDCA with CSMA/CA and with data traffic prioritization through access categories to guarantee Quality of Service. Nevertheless, the European spectrum is differently allocated, as it is sub-divided in four different parts: more specifically, the ITS-G5A is the 30 MHz wide primary frequency band employed for safety applications and the ITS-G5B has 20 MHz dedicated for non-safety applications [5].

As in the DSRC standard, TCP/UDP and IP protocols are used in the network and transport layers for non-safety applications. However, for network and transport layers the C-ITS standard also adopts in parallel the protocol called *GeoNetworking*, which is an *ad hoc* routing protocol for multi-hop communication specified in the ETSI EN 302 636 standard series [14]. Such protocol uses geographical coordinates for addressing and forwarding purposes: more specifically, all the vehicles located in a specific geographical area could become the destination of a packet with a delivery that is independent from the communication range of a single wireless hop, and locally each vehicle exploits the knowledge of its position and the neighbour positions to eventually forward packets. All these aspects enable an effective multi-hop routing which allows to create and maintain network routes in a very dynamic environment at the cost of low protocol overhead. Therefore, such protocol guarantees support for applications based on multi-hop communications, at the cost of an increased complexity and overhead with respect to the DSRC WSMP [3].

The main contribution of ETSI is concerned with standards at the facilities layer that define V2X messages with the goal of meeting application requirements. More specifically, ETSI EN 302 637-2 standard defines the Cooperative Awareness Message (CAM) [15], which it is possible to consider as the equivalent version of the BSM in the DSRC standard since such kind of message periodically carries vehicle state information in terms of position and movement that the other receiving vehicles may use to support safety applications. Moreover, the ETSI EN 302 637-3 standard defines the Distributed Environmental Notification Message (DENM) [16], whose transmission is usually triggered by safety applications to notify specific events occurred in the road environment.

For what concerns the application layer, the applications are not standardized precisely, but ETSI standardized minimum functional and performance requirements for three different kinds of applications, among which Intersection Collision Risk Warning (ICRW) [17] for potential vehicle collisions at intersections acquires crucial importance for the purposes of this thesis work.

In the following figure (Figure 2.4) it is possible to observe the C-ITS protocol stack and related standards:

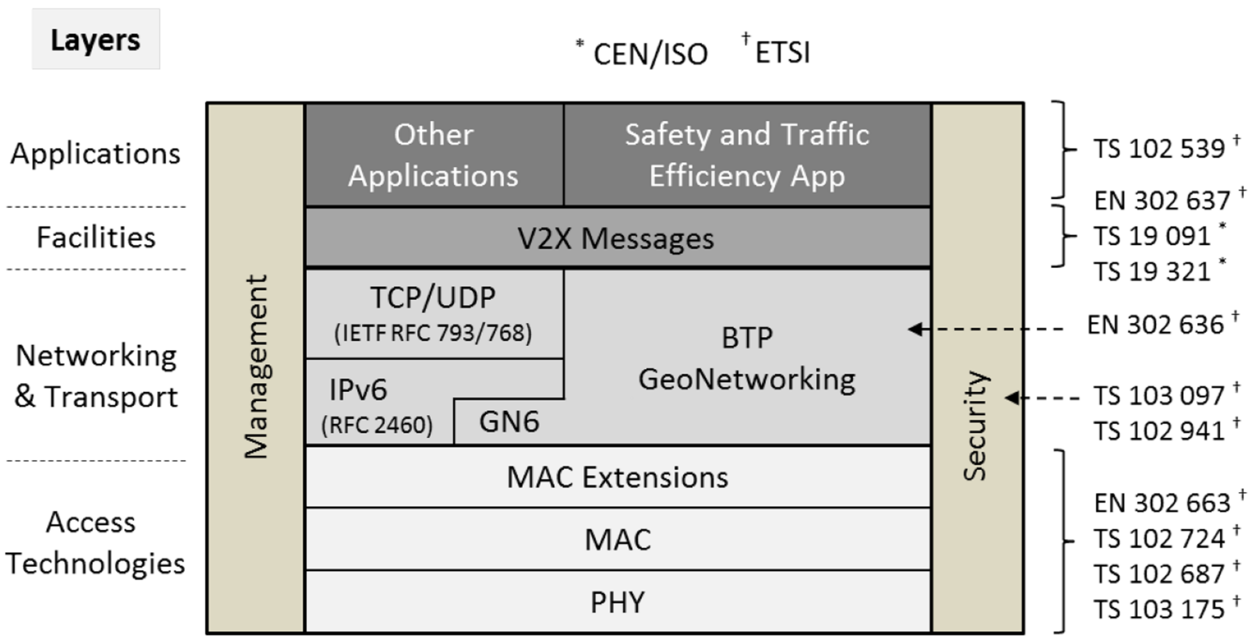

Figure 2.4. C-ITS protocol stack and related standards (taken from [3])

2.4 The Cellular based V2X

In the recent years many stakeholders have been investigating about the possibility of using the cellular technology to address the requirements of many V2X use-cases and try to overcome the main limitations of the vehicular networks relying on IEEE 802.11p in terms of inadequate deterministic Quality of Service guarantees and of intermittent connections due to limited radio range of the *ad hoc* communications and the lack of extensive roadside communication infrastructure [18].
Therefore, the 3rd Generation Partnership Project (3GPP) has been defining the technical specifications and standards to support V2X based on the cellular technology, which it is assumed to be advantageous since it can potentially provide better QoS support, higher data rates and better coverage also for fast-moving vehicles and it can exploit existing network infrastructures without the need of investing to deploy new ones, although one of its main challenges remains the capability to properly support latency-sensitive applications, especially in case of high mobility and congestion [19]. Moreover, the usage of cellular based V2X might result to be crucially important as it allows to provide services also to vulnerable users like pedestrians and cyclists communicating with vehicles, since it is relatively easy to import the technological advancements of the cellular technology made for V2X into the hardware of the devices used by such road users.
3GPP first standardized the Cellular V2X (C-V2X) with the specifications of the Release 14 [20], in which LTE is used as underlying technology. In this case, all the V2X communication types are supported thanks to the fact that the network entities can exploit two different radio interfaces: the one-to-many PC5 interface and the Uu interface between

the User Equipment (UE) and the Evolved Node B (eNodeB).

More specifically, the Uu interface permits to have the traditional cellular network connections, which are mainly concerned with the support of non-safety and infotainment applications, usually requiring a server. Moreover, the usage of only such interface makes possible the following V2X communication types: [20]

- V2V, when a vehicle sends a message in uplink to the LTE Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and the E-UTRAN performs downlink transmission of that message to multiple vehicles, potentially by using a broadcast mechanism (Figure 2.5);

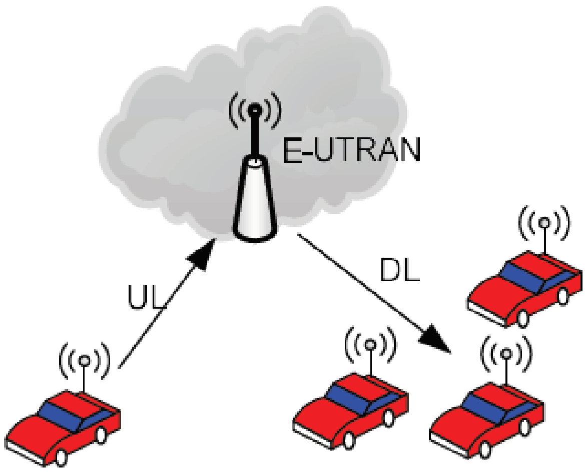

Figure 2.5. V2V communications through Uu interface (taken from [20])

- V2P and P2V, respectively when a vehicle transmits a message in uplink to the E-UTRAN and it transmits that message in downlink to multiple pedestrians and when a pedestrian sends an uplink message that the E-UTRAN sends in downlink to multiple vehicles, potentially by using a broadcast mechanism in both cases (Figure 2.6);

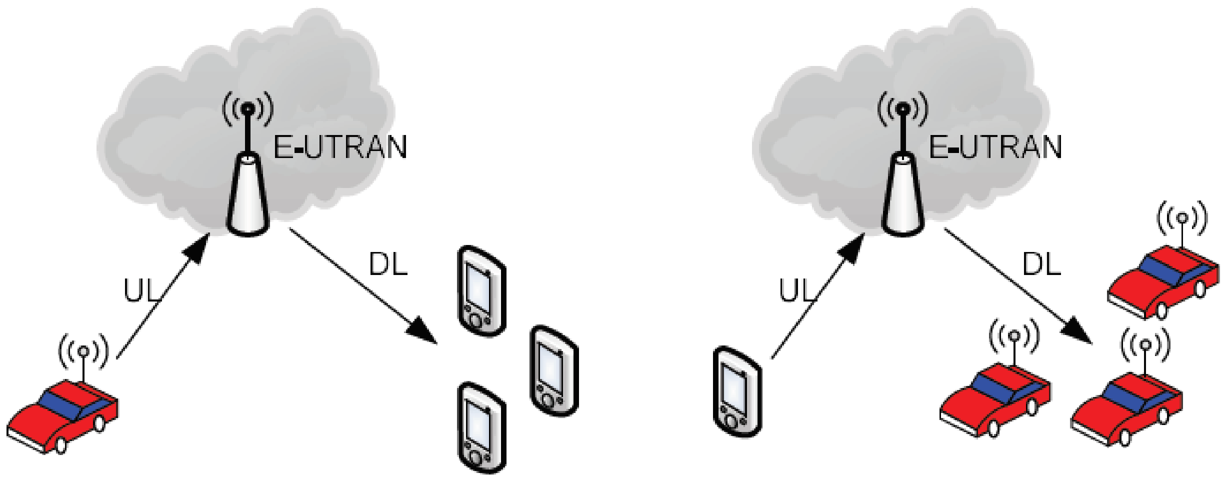

Figure 2.6. V2P and P2V communications through Uu interface (taken from [20])

- V2I and I2V, respectively when a vehicle transmits a V2I message in uplink to an eNB type RSU and when a eNB type RSU transmits a I2V message to multiple vehicles in downlink (Figure 2.7);

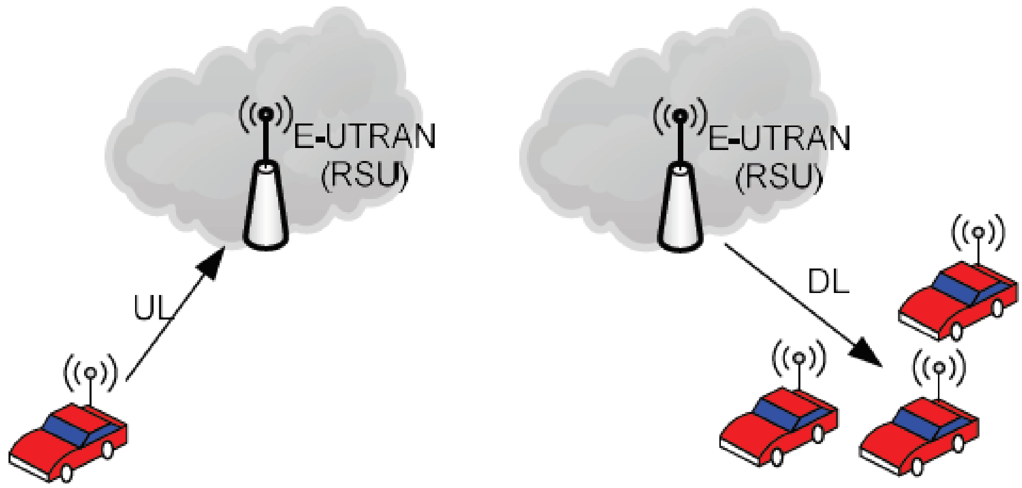

Figure 2.7. V2I and I2V communications through Uu interface (taken from [20])

- V2N, when a vehicle communicates with an application server (Figure 2.8);

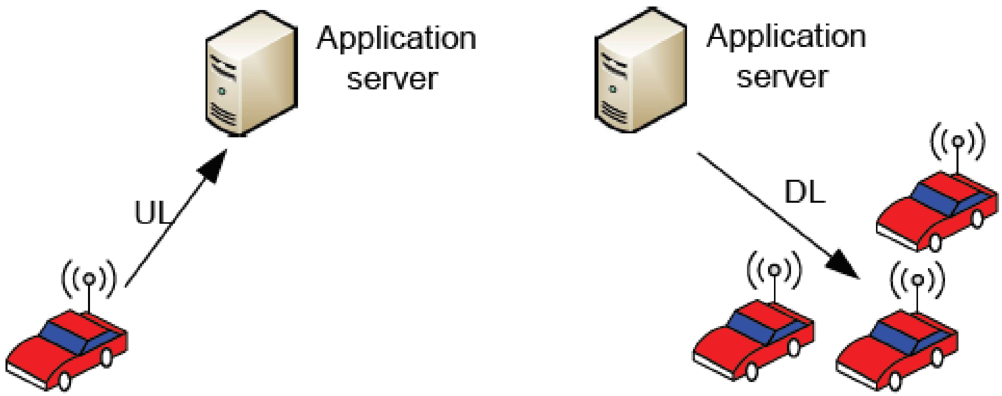

Figure 2.8. V2N communications through Uu interface (taken from [20])

Instead, PC5 interface allows to have direct communication among network entities without the need of using the traditional cellular network communication, thus potentially reducing the related latency and providing support to safety-critical applications. Furthermore, the usage of only such interface makes possible the following V2X communication types: [20]

- V2V, when a vehicle transmits a message to multiple vehicles (Figure 2.9);

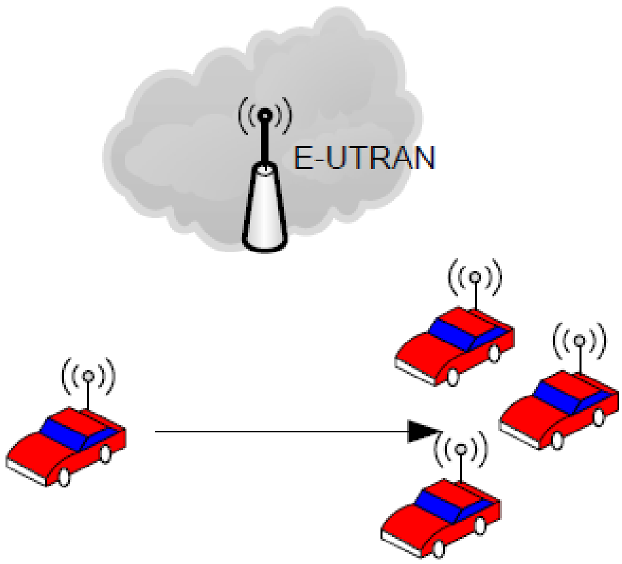

Figure 2.9. V2V communications through PC5 interface (taken from [20])

- V2I and I2V, respectively when a vehicle transmits a message to multiple UE-type RSU and when a UE-RSU transmits a message to multiple vehicles (Figure 2.10);

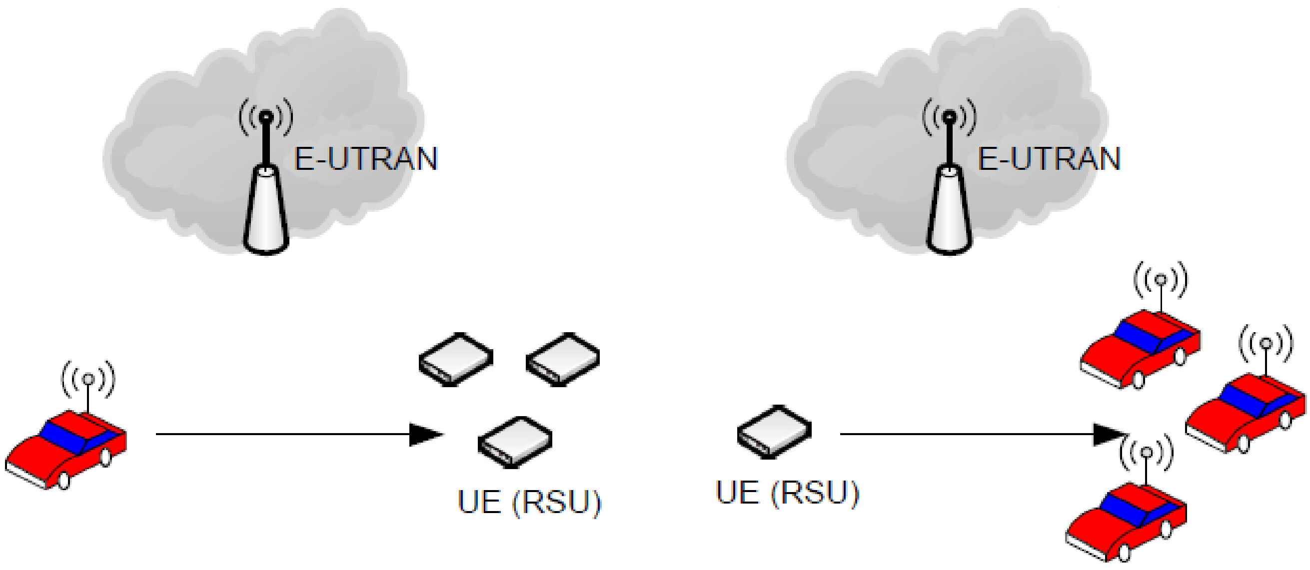

Figure 2.10. V2I and I2V communications through PC5 interface (taken from [20])

- V2P and P2V, respectively when a vehicle transmits a message to multiple pedestrian UE and when a pedestrian UE transmits a message to multiple vehicles (Figure 2.11).

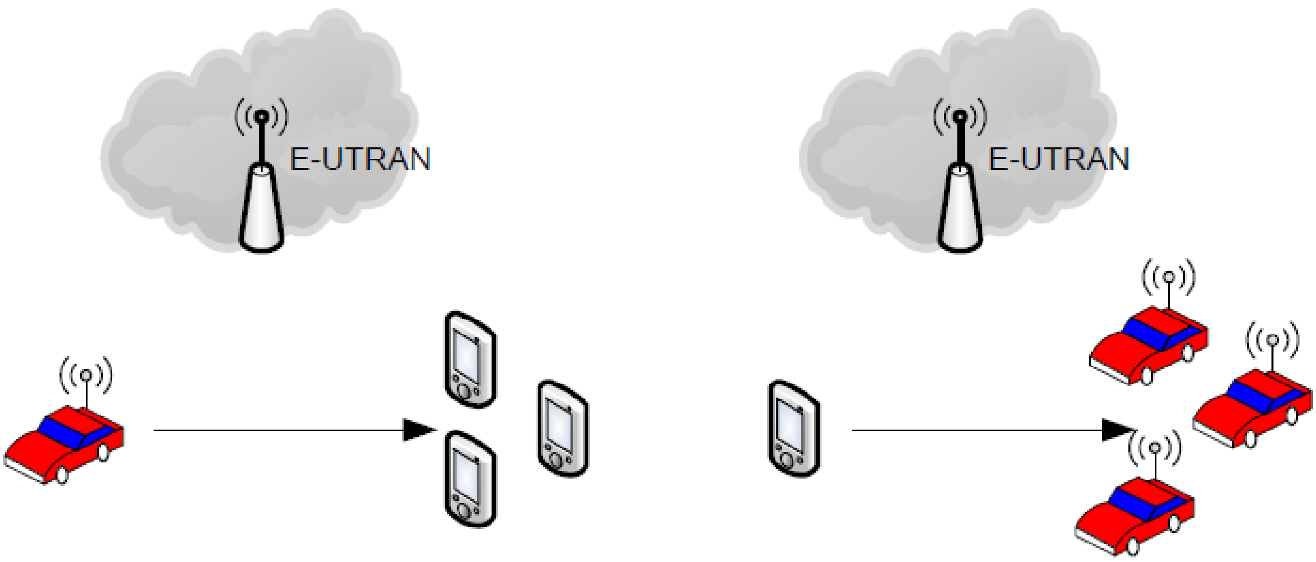

Figure 2.11. V2P and P2V communications through PC5 interface (taken from [20])

More in detail, the direct communication through the PC5 interface specified in the 3GPP Release 14 is an extension of the 3GPP Release 12 for Device-to-Device (D2D) functionality, which lacked support for many safety V2X scenarios. For this reason, Release 14 specifications have been made to meet the several V2X requirements by introducing new low-latency transmission modes, improving the support for high speed vehicles and high node density and proposing a new distributed algorithm for scheduling the resources in the V2V direct communication based on sensing with semi-persistent transmission to optimize the use of the channel [21].

In general, both Release 12 and Release 14 state that devices can directly communicate by using LTE uplink transmission and spectrum resources with two different operational modes: the first one can work also without the coverage of a base station and is based on a distributed algorithm implemented between the vehicles to schedule the communication resources by using only PC5 interface (Figure 2.12); instead, the second one must work with the continuous coverage of a base station which guarantees a centralized synchronization and resources scheduling through control signalling over the Uu interface [22] (Figure 2.13).

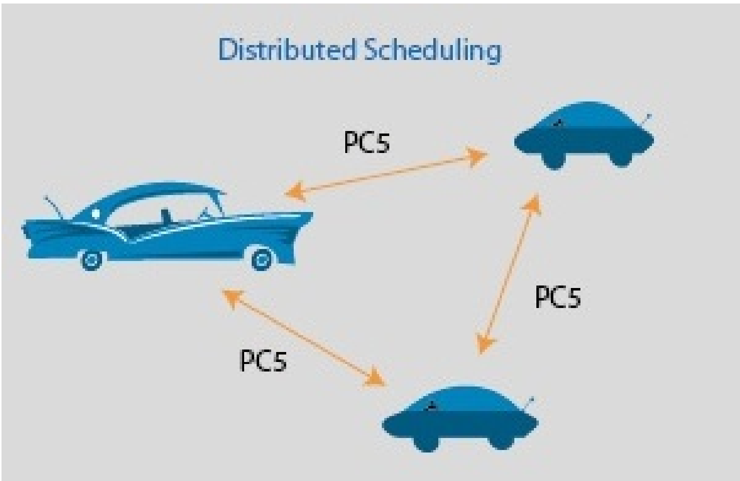

Figure 2.12. Distributed scheduling through PC5 interface (taken from [21])

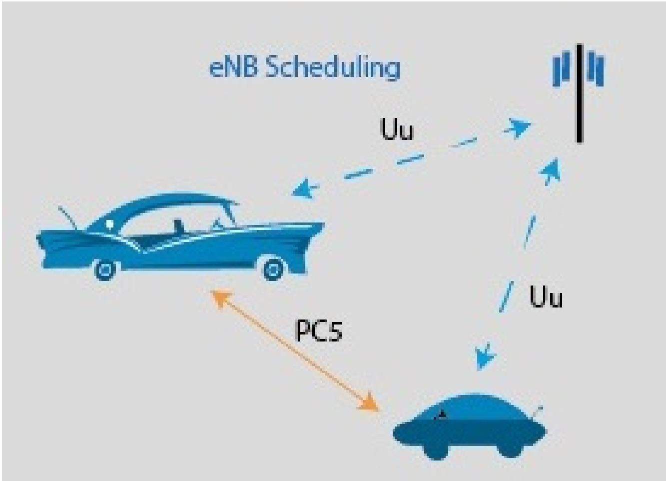

Figure 2.13. ENodeB scheduling through Uu interface (taken from [21])

Chapter 3

V2X scenarios description

This chapter describes the V2X scenarios which have been analysed and simulated in this thesis work. Given the crucial importance of safety applications in the context of vehicular networks, all the proposed scenarios are concerned with a collision detection application developed to avoid accidents at intersections in urban environment.

3.1 ETSI Intersection Collision Risk Warning

Considering the V2X safety application for avoiding accidents at intersections, ETSI provides in [17] technical specifications about the requirements of the so-called Intersection Collision Risk Warning application (ICRW) by specifying the parameters and conditions needed to properly operate with such kind of application. More specifically, in case of potential collisions at intersections, such application provides warnings to drivers in such a way that they can immediately react to avoid potential accidents or, if automatic braking system is available at vehicles, it will directly react in response of the warning, thus guaranteeing in the latter case a faster reaction towards the potential collision.
Such application must leverage the processing of two different kinds of messages:

- **Beacons**, periodic messages sent by vehicles to notify their current mobility status in terms of position, speed and direction;
- **Alerts**, event-based messages generated to notify the detection of collision risk.

ETSI standardizes both kind of messages respectively with the Cooperative Awareness Message (CAM) [15] and Decentralized Environmental Notification Message (DENM) [16]. Moreover, ETSI highlights the fact that this kind of application requires a short end-to-end latency time to work properly and avoid the collisions: as it is possible to observe in the Figure 3.1, such end-to-end latency is defined as the interval between the time instant (T0) at which the beacon data (eventually triggering a collision risk) is available at the vehicle electronic system and the time instant (T6) at which the warning message is showed on the vehicle Human-to-Machine Interface (HMI) or is delivered to the vehicle electronic system for automatic braking, if available. In particular, ETSI specifies that such latency must not be more than 300ms [17].

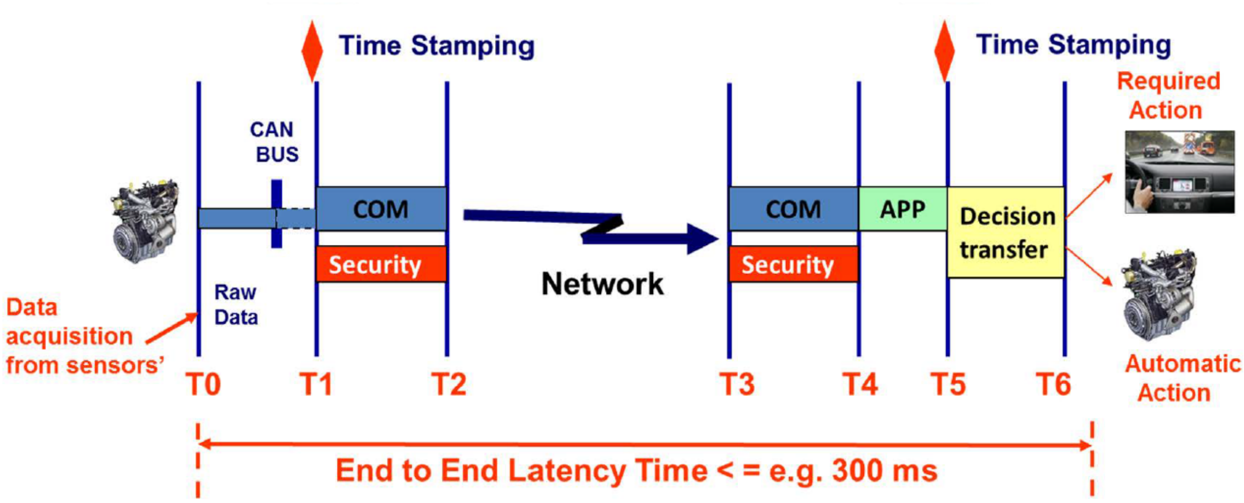

Figure 3.1. End-to-end latency for ICRW application (taken from [17])

Another key aspect of ETSI specifications for ICRW consists in the definition of the functional modes which the vehicles might be compliant with to support this kind of application:

- **Vehicle originating mode**: it refers to the capability of the vehicle to generate CAMs according to the specifications of [15] and to trigger DENMs transmission upon the detection of the collision risk;
- **Minimum Vehicle receiving mode**: it is related to the capability of the vehicle to process the received DENM messages and eventually provide a warning to the driver or to the automatic braking system;
- **Full Vehicle receiving mode**: it refers to the capability of the vehicle to process the received DENM message as previously described in the previous statement and also to process the received CAMs to be able to analyse the potential collision risks with other vehicles.

The definition of such functional modes results to be of significant importance, since vehicles implementing both the originating mode and the full receiving mode could potentially support distributed scenarios which do not require the presence of a road side network infrastructure element that processes the CAMs received to evaluate potential collisions at intersections.

3.2 Collision detection algorithm [23]

This paragraph is dedicated to the description of the collision detection algorithm used in the scenarios that will be analysed in this chapter. Being a generic trajectory-based algorithm, it can be potentially exploited to detect collisions between couples of different road entities, such like two vehicles or between one vehicle and a pedestrian. An important assumption that must be done is that the considered algorithm evaluates collisions of entities following straight paths. Moreover, the detector is triggered as soon as a beacon (CAM or BSM) is received with the final aim of determining whether the sender of the beacon is at collision risk with another entity. For this reason, any entity subscribed to the service must send periodical information about its mobility status by means of beacons.

3.2.1 Centralized collision detection algorithm [23]

In the case of a centralized solution (V2I or V2N), the detector is implemented in a road infrastructure entity or in a remote server. Furthermore, another key aspect that is worth to observe is that the potential collision risks considered are the ones involving the entity whose beacon triggered the detection algorithm with all the entities (or an optimized subset of them) which have previously sent beacons.
In this case, the inputs required by the algorithm are the following:

- Position, speed and acceleration of the entity triggering the algorithm;
- The latest beacons previously sent by other entities and stored in a data structure B.

Firstly, for every entity for which recent versions of received beacons are stored and for one entity at the time, the algorithm takes the old position, speed and acceleration stored in the beacon of the considered entity and updates its position according to a uniformly accelerated rectilinear motion by considering as time horizon the time difference between the time instant at which the triggering beacon has been received and the time instant at which the beacon of the considered entity has been stored.
Consequently, by exploiting the obtained position, speed and acceleration of the triggering entity and of the considered entity, the algorithm computes how the square distance between such entities varies with time starting from the current time instant and then finds the time instants in which that function is minimum. Among such time instants, the algorithm selects the ones which are positive and under a given time threshold.
Then, the algorithm computes the values of the minimum distance at the selected time instants and verifies whether the minimum distance value is below a given spatial threshold, in which case a potential collision between the triggering entity and the considered entity is detected, and an alert is issued for both.
In the following Figure (Figure 3.2) it is possible to observe the pseudo-code for the presented algorithm:

Algorithm Collision detection (with acceleration)

Require: $\vec{p}_0, \vec{v}, \vec{a}, \mathcal{B}$

$C \leftarrow \emptyset$

for all $b \in \mathcal{B}$ **do**

  read $\vec{\tilde{p}}^0, \vec{\tilde{v}}, \vec{\tilde{a}}$ from b

  $\tilde{p}_x(t) \leftarrow \tilde{p}_x^0 + \tilde{v}_x t + \frac{1}{2}\tilde{a}_x t^2$

  $\tilde{p}_y(t) \leftarrow \tilde{p}_y^0 + \tilde{v}_y t + \frac{1}{2}\tilde{a}_y t^2$

  $D(t) \leftarrow (p_x - \tilde{p}_x)^2 + (p_y - \tilde{p}_y)^2 = \left[p_x^0 - \tilde{p}_x^0 + (v_x - \tilde{v}_x)\, t + \frac{1}{2}\,(a_x - \tilde{a}_x)\, t^2\right]^2 + \left[p_y^0 - \tilde{p}_y^0 + (v_y - \tilde{v}_y)\, t + \frac{1}{2}\,(a_y - \tilde{a}_y)\, t^2\right]^2$

  $\mathcal{T} \leftarrow t\colon \frac{\mathrm{d}}{\mathrm{d}t} D(t) = 0$

  for all $t^\star \in \mathcal{T}$ **do**

    if $t^\star > 0$ **or** $t^\star \leq t_{\min}$ **then**

      $d^\star \leftarrow \sqrt{D(t^\star)}$

      if $d^\star \leq d_{\min}$ **then**

        $C \leftarrow C \cup \{b\}$

        break

return

Figure 3.2. Centralized collision detection algorithm (adapted from [23])

3.2.2 Distributed collision detection algorithm [23]

In the distributed scenarios (V2V and V2P), the detector is implemented locally within the entities subscribed to the collision detection service, like vehicles and pedestrians. Moreover, in this case each entity triggers the detector to evaluate only potential collision risks between itself and the entities from which it has received a triggering beacon, thus external collisions are not considered. Therefore, this behaviour might not optimize the overall road safety, but it reduces the energy consumption due to additional computational power that would be spent to evaluate collisions involving other entities. This choice might result to be crucial in high-density conditions, especially to save battery of pedestrian devices used to run the collision detection algorithm.
In this case, the input of the algorithm is just the information about an entity in terms of position, speed and acceleration contained in the beacon whose reception triggers the detector. Indeed, since in this case the entity running the algorithm evaluates only potential collision courses with itself, there is no need to store previously received beacons to eventually detect collisions among the triggering entity and other entities. Therefore, whenever the detector is triggered, it just uses the mobility information included in the beacon received from the triggering entity and the mobility information obtained directly from the sensors of the entity which is running the algorithm. With such information available, as in the centralized case, the algorithm still computes how the square distance between the two entities varies with time to verify whether the time instants of minimum distance and minimum distance values will be under a temporal and spatial threshold

respectively. In the affirmative case, a collision is detected, and an alert is issued for both entities.
The pseudo-code for such algorithm is presented in the following Figure (Figure 3.3):

Algorithm Collision detection (with acceleration)

Require: $\vec{p}_0, \vec{v}, \vec{a}$

$D(t) \leftarrow (p_x - \tilde{p}_x)^2 + (p_y - \tilde{p}_y)^2 = \left[p_x^0 - \tilde{p}_x^0 + (v_x - \tilde{v}_x)\,t + \frac{1}{2}(a_x - \tilde{a}_x)\,t^2\right]^2 + \left[p_y^0 - \tilde{p}_y^0 + (v_y - \tilde{v}_y)\,t + \frac{1}{2}(a_y - \tilde{a}_y)\,t^2\right]^2$

$\mathcal{T} \leftarrow t\colon \frac{\mathrm{d}}{\mathrm{d}t} D(t) = 0$

for all $t^\star \in \mathcal{T}$ **do**

  if $t^\star > 0$ **or** $t^\star \leq t_{\min}$ **then**

    $d^\star \leftarrow \sqrt{D(t^\star)}$

    if $d^\star \leq d_{\min}$ **then**

      Collision

      break

return

Figure 3.3. Distributed collision detection algorithm

3.3 General considerations for each scenario

The aim of this paragraph is to present the general considerations and parameters holding for every scenario that will be analysed in the next paragraphs. Indeed, despite the intrinsic differences among the different scenarios, there are some parameters which are generally valid as they are determined by technical specifications and environmental variables which are kept general to permit reasonable comparisons among the different scenarios.

3.3.1 Reference road environment

To analyse the performances of the collision detection algorithm developed to avoid vehicular accidents at intersections, an urban area made up of two 6m-wide perpendicular roads and one unregulated road crossing have been considered.
More specifically, on the four quadrants created by the intersection of the two roads, four buildings with a height of 20 meters were placed in order to properly account for Non-Line-Of-Sight (NLOS) conditions.
In such environment, no pedestrian lanes were included, thus vehicles are the only entities appearing in the scenarios.

In the following figure (Figure 3.4), the reference road environment is presented:

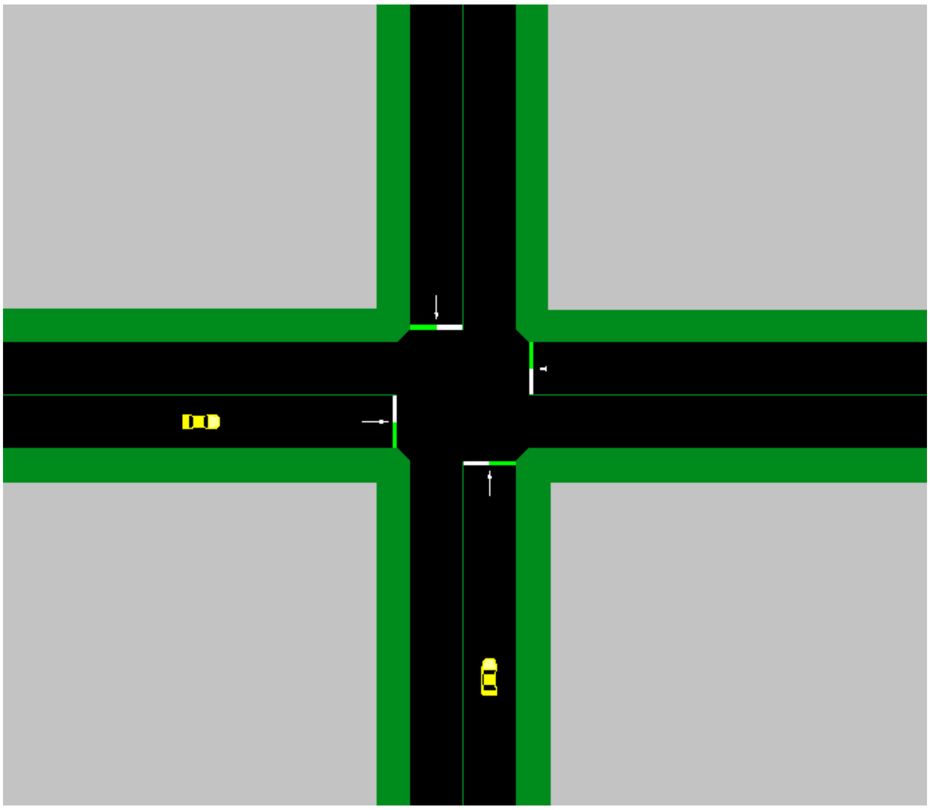

Figure 3.4. Reference road environment

3.3.2 Vehicle and driver parameters

In the considered scenarios, it has been assumed that the vehicles are travelling at a maximum speed of 13.89 m/s (50 km/h), with a maximum acceleration of 2.9 m/s^2 and a maximum deceleration of 7.5 m/s^2 [24].
Moreover, the vehicles are generated randomly from the extremity of one of the four different roads of the road environment and they follow only a straight path, thus no turning is foreseen at intersections.
Considering the messages presented in the paragraph 3.1 to support the safety application of collision avoidance at intersections, it has been assumed a reasonable elaboration time of 400 ms needed by the on-board units (OBUs) of the vehicles to process such messages.
For what concerns the driver, the only parameter considered was the reaction time needed to take an action after an alert message is presented to the Human-to-Machine interface (HMI) of the vehicle, and it was set to one second [25].
The parameters presented in this paragraph are summarized in the following table (Table 3.1):

Maximum vehicle speed	13.89 m/s
Maximum vehicle acceleration	2.9 m/s^2
Maximum vehicle deceleration	7.5 m/s^2
OBU elaboration time	400 ms
Human reaction time	1 s

Table 3.1. Vehicle and driver parameters

3.3.3 Applicative messages parameters

As specified in the paragraph 3.1, two kind of messages are needed to support the collision detection application at road intersections: the beacon, a periodic message sent by vehicles to notify their current mobility status, and the alert, an event-based message generated to notify the risk of collision.
To properly support safety applications, it is required to have as much as possible updated information about the current mobility status of vehicles, therefore beacon messages should be sent with a high frequency: according to ETSI such frequency should range from 1 to 10 Hz for CAMs [15], whereas in SAE standard there is no specific indication to regulate such frequency, although most safety applications require 10 Hz [26].
For the considered scenarios, such frequency has been set to the value of 10 Hz, so that the detector can potentially work with updated information that permits to reduce the probability of errors and increase the reliability of the detection.

3.4 Cellular V2N scenario [23]

In this first scenario, vehicles are connected to a cellular infrastructure through OBUs supporting cellular V2N (C-V2N) communications and they are subscribed to the collision avoidance application as clients: therefore, they are responsible only to send periodic beacons and eventually receive and process alerts. On the other side, a collision avoidance application server is responsible to process the received beacons, run the collision detector and send eventual alerts to the vehicles involved in the potential collision course. More specifically, LTE technology was considered for the access layer, IP for the network layer and UDP for the transport layer and all the vehicles are under coverage of an LTE eNode-B, that was located in the upper-right quadrant of the urban road environment.
The great advantage of this centralized scenario consists in the fact that the transmissions are reliable and synchronized by a network infrastructure, as vehicles do not communicate directly and thus are not significantly penalized by NLOS conditions.
Nevertheless, a significant disadvantage of any centralized approach is the latency due to the absence of *ad hoc* communications and the necessity of exploiting a network infrastructure to communicate. For this reason, the position of the server within the network infrastructure is an important aspect, since the latency needed to reach the server from the LTE base station may result crucial for the performances of the analysed safety application, as already discussed in paragraph 3.1. Under this perspective, a new paradigm

for the location of computational resources has been investigated, as it will be explained in the next section.

3.4.1 Multi-access Edge Computing (MEC) [27]

The Multi-access Edge Computing paradigm provides cloud-computing capabilities and an Information Technology (IT) service environment within the edge of the network at the proximity of mobile users, guaranteeing ultra-low latency, high bandwidth and real-time access to radio network information. In this way, by exploiting servers at the Radio Access Network (RAN) of network operators, delay-sensitive and context-aware applications can be properly and flexibly deployed, thus enabling new vertical business segments and services for private consumers and enterprise customers [27].
More in detail, the main characteristics of the MEC paradigm are the following: [28]

- **Low Latency**: Latency is reduced since services run at the edge of the network, very close to the end users' devices. This aspect is fundamental to improve the user experience and to react faster in a wide range of applications;
- **Independence**: The edge can use local resources to offer different services to subscribers while keeping a complete independence from the rest of the network. This aspect is important to provide high resilience to safety systems;
- **Proximity**: The fact that MEC servers are physically very close to the end users facilitates the possibility to capture from them key information, which can be useful for analytics and big data;
- **Location awareness**: The location of the devices connected to the network can be easily determined by a local service leveraging low-level signalling information. This aspect allows the creation of many location-aware services;
- **Network context awareness**: The possibility of using real-time network information is crucial to offer context-aware applications which differentiate the mobile users' experience and create new business models.

3.4.2 Server-eNodeB latency

In this scenario, the server has been deployed in two different locations of the network infrastructure, determining two different delay values needed to reach the server from the base station (or vice versa the base station from the server).
The two considered locations are the following:

- Server at the Metro Node in a Multi-access Edge Computing (MEC) fashion, determining a one-way latency from the base station of $5ms$;
- Server in the Cloud, determining a one-way latency from the base station of $20ms$;

These values are provided by Telecom Italia as conservative (Metro-eNB) and typical (Cloud-eNB) references. It is important to underline the fact that in this vehicular safety scenario, the usage of the MEC paradigm might be significant as it allows to obtain lower latencies by keeping the advantages of any centralized approach.

3.4.3 Application Messages

As far as application messages are concerned, in this scenario both the beacon and the alert follow the ETSI specifications respectively for CAM [15] and DENM [16], which standardize many parameters of such messages, especially in terms of maximum transmission frequency, maximum message age and size. In the following table, the values of such parameters used in this scenario are reported (Table 3.2):

CAM frequency	10 Hz
Maximum DENM frequency	1 Hz
Maximum CAM Age	0.8 s
CAM Size	240 $Bytes$
DENM Size	263 $Bytes$

Table 3.2. ETSI CAM and DENM parameters

3.5 Cellular V2V Scenario

In this scenario, vehicles are equipped with OBUs supporting cellular V2V (C-V2V) communications by being compliant to the 3GPP Release 14 standard, which was introduced to support V2V use-cases by using LTE [20]. More specifically, this scenario will consider as Transmission Mode the Mode 3 of LTE-V2X, according to which both resource scheduling and interference of V2V traffic is managed by an assisting LTE eNodeB by means of signalling control information with UEs through the Uu interface and the direct data exchange among vehicles is ensured through the PC5 interface. Therefore, the version of LTE-V2X considered here works only under the coverage of a base station.
Moreover, in this case vehicles are subscribed to the collision avoidance application as both clients and servers: as clients, they send periodic beacons and eventually process received alerts; as servers, they receive other vehicles' beacons to evaluate possible collision courses and in case send alert messages, as already described in the paragraph 3.2.
This scenario has been taken into consideration since V2V communications are seen suitable for safety applications due to the instant communications between entities which allows to reduce significantly the latency, which may result decisive to transmit in time warnings in case of dangerous situations.
However, the scenario has been proposed to also evaluate the eventual shortcomings and how such shortcomings may affect the performances in the specific application of collision avoidance at intersections. In particular, the main drawback considered in the scenario is

the presence of NLOS conditions for the direct communication between vehicles, as those conditions may significantly increase the signal attenuation, thus reducing the probability of correctly receive packets from other entities and making the overall system less reliable.

As in the case of the Cellular V2N scenario, the applicative messages follow the specifications of the standard ETSI for CAMs and DENMs (paragraph 3.4.3).

3.6 802.11p V2V Scenario

In this scenario, vehicles are equipped with an OBU enabling V2V communication through 802.11p and DSRC standard. Differently from the cellular V2V scenario, the entities access the channel by means of a CSMA/CA protocol as already seen in the previous chapter, thus there is no need for a central network infrastructure element (e.g. RSU) responsible of synchronizing the nodes in accessing the available resources.
The 802.11p technology has been analysed to have a direct comparison with the cellular V2V technology in terms of performances in supporting the proposed collision avoidance application. Also in this case, vehicles are subscribed to the application as both clients and servers and NLOS conditions are considered to see how much this technology is capable to overcome its main limitation.
Nevertheless, the parameters used in this scenario differ from the ones used in the C-V2V case, because they are defined accordingly to the DSRC standard for 802.11p communications already described in the first chapter.

More specifically, the applicative messages are specified by the SAE J2735 standard [13] which indicates an equivalent version of the ETSI CAM termed Basic Safety Message (BSM) and another kind of message called Intersection Collision Announcement (ICA) [29] offering an equivalent service of the ETSI DENM alert.
The specific parameters of the scenario in terms of data rate [30], BSM frequency [26], BSM and ICA sizes [31] are reported in the following table (Table 3.3):

Data rate	6 $Mbps$
BSM frequency	10 Hz
BSM Size	190 $Bytes$
ICA Size	200 $Bytes$

Table 3.3. 802.11p V2V Scenario specific parameters

3.7 Hybrid V2N Scenario

In this scenario, two different kinds of vehicles are considered: vehicles equipped with an OBU enabling C-V2N communications in an LTE network and vehicles equipped with an OBU enabling V2V and V2I through 802.11p standard.

To ensure the interoperability between the two different technologies and make possible the communication between LTE-based vehicles and 802.11p-based vehicles, an 802.11p-based RSU has been placed at the centre of the urban intersection and has been connected to a UDP/IP MEC server. In this way, on one hand 802.11p-based vehicles send their beacons to and receive eventual alerts from the RSU, which is responsible to forward beacons to the server and alerts from the server to the involved 802.11p-based vehicles. On the other hand, the LTE-based vehicles still communicate with an LTE eNodeB connected to the server, with the same fashion described in the paragraph 3.4.
It is worth to underline that in this case all the vehicles are subscribed as clients of the collision avoidance application and the parameters used in the scenario are the same of the paragraph 3.4 for the LTE-based vehicles and of the paragraph 3.6 for the 802.11p-based vehicles and RSU.

This scenario has been created to achieve these three goals:

- Overcoming the NLOS problem of *ad hoc* communications by putting an RSU at the centre of the road topology;
- Creating the basis for an heterogenous scenario in which the main advantages of the single technologies could be selectively and dynamically exploited;
- Integrating two different technologies, in such a way to mesh and make valuable in the market vehicles supporting any of the two available technologies for vehicular communications. This gives the possibility of creating a business model around the MEC server paradigm.

Chapter 4

Simulation setup and methodology

This chapter provides a description of the approach adopted for simulating the scenarios presented in the previous chapter in terms of setup of simulation tools, methodology and modelling of real-world properties of vehicular communications.

4.1 Simulation tools

To perform the simulations for the different V2X scenarios of interest for this thesis work, two different simulative environments have been considered respectively to model the communication network and the vehicular mobility within the road infrastructure: the open-source and discrete-event simulator OMNeT++ [32] at the version 5.1.1 and the SUMO mobility simulator [33] at the version 0.30.0.
More specifically, within the OMNeT++ environment it has been necessary to use the SimuLTE project [34] at the version 0.9.1 and Veins project [35] at the version 4.6 for the following reasons:

- Couple OMNeT++ simulator with SUMO mobility simulator through the Traffic Control Interface (*TraCI*) provided by the Veins framework;
- Support 802.11p-based communications by using Veins framework;
- Support LTE-based communications by using the SimuLTE framework.

4.2 Bidirectional coupling of OMNeT++ and SUMO

The bidirectional coupling of network simulator OMNeT++ and road traffic simulator SUMO is ensured by the fact that both simulators include dedicated communications modules which exchange in real-time mobility traces and road environment properties throughout the simulation time by means of commands and related answers [36].
To guarantee a tight combination, considering that SUMO progresses its simulation time at regular discrete timesteps, OMNeT++ advances the SUMO simulation only at fixed time intervals according to the following steps [36]:

- OMNeT++ advances its simulation within a given interval and in the meanwhile buffers the generated commands;
- At the end of the interval, OMNeT++ sends the buffered commands to SUMO, which in turn buffers such received commands;
- OMNeT++ triggers the corresponding time interval on SUMO and waits for the modified mobility trace;
- SUMO executes the received commands, advances its simulation within the considered interval and at the end of it sends back to OMNeT++ the resulting mobility trace;
- OMNeT++ processes the received mobility trace to move accordingly the existing nodes and eventually inserts new ones or deletes those already arrived at destination;
- OMNeT++ repeats the procedure considering the next time interval.

The state machines of both simulators implementing the previously described behaviour can be observed in the following figure (Figure 4.1):

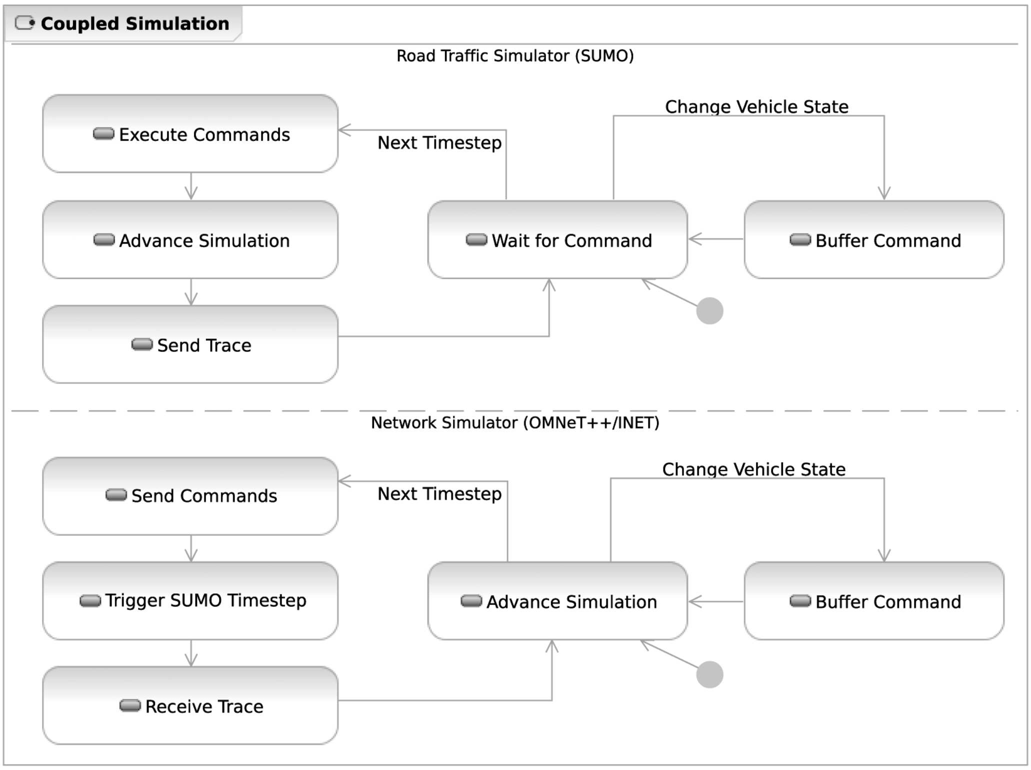

Figure 4.1. State machines of OMNeT++ and SUMO (taken from [36])

4.2.1 Traffic Control Interface

The presented coupled simulation framework is ensured by the so called Traffic Control Interface (*TraCI*), that uses a TCP based architecture in which on one hand SUMO acts as the server responsible for processing the received commands and providing the related responses, and on the other hand OMNeT++ acts as the client that sends specific commands to SUMO and then waits for the related answers [37].
In particular, the Veins project of the OMNeT++ environment provides a middleware for coupling the two simulators by making available a C++ client library for the *TraCI* Application Programming Interface (API) [37]. Nevertheless, in such library not all the *TraCI* commands have been implemented and this might create problems in simulating the considered scenarios, as it will be clarified in the next sections. For this reason, general guidelines for implementing such commands in OMNeT++ are now presented:

- Determine the type of command to be implemented among the types specified in [37];
- Consider the syntax of the specific command of the previously determined type, as for example in [38] or in [39];
- In the *TraCICommandInterface* class located in the *veins/modules/mobility/traci* folder of the Veins project, create a new method with the name of the command in which the command is filled according to its syntax and then buffered.

According to the previously described guidelines, two different missing commands have been implemented for the purposes of this thesis work. In the following list, such commands are presented in terms of their *TraCI* syntax and C++ implementation in OMNeT++:

- Command to set vehicle's deceleration:

ubyte	string	ubyte	<value_type>
Variable	**Vehicle ID**	**Type of the value**	**New Value**

Figure 4.2. Syntax for changing the vehicle's state (taken from [38])

```
void TraCICommandInterface::Vehicle::setDecel(double speed) {
    uint8_t variableId = VAR_DECEL;
    uint8_t variableType = TYPE_DOUBLE;
    TraCIBuffer buf = traci->connection.query(CMD_SET_VEHICLE_VARIABLE,
       TraCIBuffer() << variableId << nodeId << variableType << speed);
    ASSERT(buf.eof());
}
```

Figure 4.3. Implementation of the command to set vehicle's deceleration

- Command to get vehicle's magnitude of speed:

ubyte	string	ubyte	<return_type>
Variable	**Vehicle ID**	**Return type of the variable**	**<VARIABLE_VALUE>**

Figure 4.4. Syntax for getting vehicle's variables (taken from [39])

```
double TraCICommandInterface::Vehicle::getSpeed() {
    return traci->genericGetDouble(CMD_GET_VEHICLE_VARIABLE,
            nodeId,VAR_SPEED, RESPONSE_GET_VEHICLE_VARIABLE);
}
```

Figure 4.5. Implementation of the command to get vehicle's magnitude of speed

4.3 Simulation Methodology

To evaluate the performances of the collision avoidance application, the simulative approach adopted in this thesis work foresees the bidirectional coupling of the network simulation and the road traffic simulation. Such coupling is obtained by means of real-time interactions between SUMO and OMNeT++: in this way, on one hand the road traffic conditions of SUMO might trigger algorithms implemented in the network components of OMNeT++; on the other hand, the results of such algorithms might translate into commands destined to SUMO to properly change the road traffic conditions and road entities' status.
More specifically, considering as example the application proposed in this work, the server of the application might generate and send alerts in OMNeT++ as the result of a dangerous traffic condition in SUMO due to vehicles involved in a potential collision course. As soon as a vehicle receives an alert it must properly change its current mobility status to avoid the collision (by braking or reducing its current speed).
This approach provides the benefits of obtaining significant insights about the impact of network protocols and applications on the road traffic and making more realistic the results obtained from the vehicular communications simulations [36].

Moreover, this approach implies two different methodological requirements:

- Implementation of specific actions to be performed by vehicles and drivers in response to alert receptions;

- Execution of two kinds of simulations of the same scenario to properly evaluate the performances of the proposed collision avoidance application.

These two aspects, as well as the principal constraints of this methodology, will be better clarified in the next subparagraphs.

4.3.1 Actions performed upon alert reception

The actions performed upon alert reception by vehicles must be done to properly avoid the potential collision signalled by the alert itself. For such reason, the function *stopcar* has been implemented in the client part of the collision avoidance application. More specifically, such function will stop the vehicle just before the stop line through the command *stopAt* of *TraCI*, provided that SUMO finds out that such situation is possible according to the current speed and the maximum deceleration of the vehicle. This usually happens when the alert is received sufficiently in advance. Otherwise, the function will cause an abrupt braking by forcing the vehicle to stop according to its maximum deceleration: in this case, very likely the alert has been received late and the vehicle is too close to the junction or worse has already taken the junction.
The implementation of the described function is showed in the following figure (Figure 4.6):

```
void myClientCD::stopcar() {

    EV << mobility->getId() << "is braking...\n";
    traciVehicle->setColor(Veins::TraCIColor(255, 0, 0, 255));


    if (junc==false){
        try {
            double x,y;
            x = stopcond.find(nextjun)->second;
            y = stoptimes.find(nextjun)->second;
            traciVehicle->stopAt(currentroad, x, 0, 0, y);
            } catch (const std::exception& e) {
                traciVehicle->setSpeed(0);
            }
        }

    else
        traciVehicle->setSpeed(0);
}
```

Figure 4.6. Implementation of the function called upon alert reception

This function provides a first possible solution to address the problem of performing actions upon the reception of alerts. Indeed, the implementation of more realistic and adaptive actions is limited by the number and behaviour of the commands that can be transmitted to SUMO. Nevertheless, this function allows to modify the vehicle's current speed according to a specific deceleration value. Since the collision detection algorithm described in the previous chapter accounts also for acceleration and deceleration, this might lead to a more realistic behaviour of the server of the application: indeed, the deceleration value notified by the vehicles together with their position and speed in the beacons might lead the server to understand that a previously alerted situation is not at collision risk anymore thanks to the imposed braking. In this way, the server does not have to continuously alert a risky situation just because vehicles do not react to the received alerts. Therefore, this approach might result to be crucial in the case that other traffic is already consuming the server's computational resources.

4.3.2 Performances evaluation

The performances of the collision avoidance application will be mainly evaluated in terms of collisions avoided in time, collisions detected too late (not avoided in time) and collisions not detected (thus also not avoided). Furthermore, since the application is based on the detection of collisions, it is important also to evaluate the performances in terms of number of false positive alerts triggered by the detector.
To properly evaluate such performances, in the proposed simulation methodology two different simulative instances of the same scenario (i.e. same road reference environment, same vehicular parameters and same routes for the vehicles) must be executed: one simulation run with only the traffic simulator SUMO, and another simulation in which SUMO and OMNeT++ are coupled as described in the previous sections. The output considered in the first case is the file produced by SUMO including the collisions occurred during the simulation; whereas in the second case it is important to collect a file including the alerts received by all the vehicles, a file including the alerts transmitted by all the vehicles, the number of detections counted once per couple of vehicles (positives) and again the file produced by SUMO including the collisions occurred in the simulation.
In this way, it is possible to define the following performance indexes:

- **True positives percentage**: it is given as the ratio between the number of true positives and the total number of positives. The number of true positives is the number of collisions occurred in the simulation with only SUMO that are matched with detections of the second simulation with SUMO and OMNeT++ coupled;

- **False positives percentage**: it is the complementary of the true positive percentage;

- **Percentage of not detected collisions**: it is given as the ratio between the number of not detected collisions in the simulation with SUMO and OMNeT++ and the total number of collisions occurred in the simulation with only SUMO. The number of not detected collisions is the number of collisions occurred in the simulation with SUMO and OMNeT++ for which no detections were triggered.

- **Percentage of too-late detected collisions**: it is given as the ratio between the number of too-late detected collisions in the simulation with SUMO and OMNeT++ and the total number of collisions occurred in the simulation with only SUMO. The number of too-late detected collisions is the number of collisions occurred in the simulation with SUMO and OMNeT++ for which at least one detection was triggered.

- **Percentage of successfully avoided collisions**: it is given as the ratio between the number of successfully avoided collisions in the simulation with SUMO and OMNeT++ and the total number of collisions occurred in the simulation with only SUMO. The number of successfully avoided collisions is the difference between the total number of collisions occurred in the simulation with SUMO and the number of collisions occurred in the simulation with SUMO and OMNeT++.

4.3.3 Methodology Constraints

This subparagraph clarifies the main constraints of the proposed methodology, which are mainly due to the limitations of the traffic simulator SUMO and the kind of performance evaluation adopted.
The first aspect that must be underlined is that to allow SUMO to record the collisions at intersections, the analysed intersections must be created of type *unregulated* in NETEDIT, the road network editor of SUMO. This implies that there are no rules to regulate how the vehicles approaching an intersection should traverse the intersection itself and, in this way, the probability of having collisions at the intersection potentially increases. Nevertheless, SUMO is natively a collision-free simulator and its default behaviour aims at reducing the collision probability: indeed, provided that vehicles have a positive value for the maximum deceleration, by default SUMO tries to avoid the collisions at intersections by forcing the vehicles not yet in the area of a given intersection to brake whenever there is already at least another vehicle in the area of the same intersection. Therefore, such behaviour may significantly affect the evaluation of the performances of the proposed application, as some collisions may be avoided because of SUMO default behaviour and not because of alerts properly and timely received by the vehicles.
Therefore, to be able to properly evaluate the performances and keep at the same time a positive value for the maximum deceleration (useful for the reasons explained in the subparagraph 4.3.1), in the considered simulated scenarios all the vehicles start with a null deceleration and have their maximum deceleration value changed only as soon as they receive an alert. For this reason, it was necessary to implement the command to set a vehicle's maximum deceleration described in the subparagraph 4.2.1.

The second important constraint of this methodology is related to the evaluation of the performances of the collision avoidance algorithm. As described in the previous subparagraph, a significant performance index is given by the percentage of false positives. Indeed, despite false positives are not as critical as false negatives, anyway they have a crucial impact on the quality of experience perceived by the final user. For this reason, it is fundamental that the driver is rarely alerted in case of situations of low or no danger, thus the probability of false positives should be kept as low as possible to make the application efficient and valuable.
According to the proposed methodology, the false positive evaluation needs the execution

of two different simulations: the one with only SUMO and the one with SUMO and OMNeT++. As already discussed, the simulation with SUMO is run to observe the collisions occurred without any vehicular communication and the simulation with SUMO and OMNeT++ is run to check whether those collisions are avoided thanks to the deployment of the collision avoidance algorithm in the network. Since this methodology foresees that the results of OMNeT++ influence the traffic conditions of SUMO (i.e. when the vehicles brake after receiving an alert), such situation may create “secondary” collisions between new vehicles approaching the intersection and vehicles which have completed their braking and are now ready to traverse the intersection. The problem with these "secondary" collisions is that they do not exist in the simulation with only SUMO as they are consequence of the modifications of the SUMO default traffic pattern caused by the commands from OMNeT++. Therefore, any eventual detection related to such "secondary" collisions can not be matched with the collisions occurred in the simulation with only SUMO, thus it would be not possible to rigorously evaluate whether the detections of "secondary" collisions are false positives or true positives.
To overcome this limitation, couples of vehicles are properly shifted in time in the SUMO traffic pattern in such a way to ensure independence among consecutive potential collisions and avoid the "secondary" collisions.

4.4 Scenarios implementation

This paragraph deals with the implementation in OMNeT++ of the V2X scenarios proposed in this thesis work, by focusing mainly on the modelling of channel models for the different communication technologies and the high-level description of the developed pieces of software.

4.4.1 Implementation of Cellular V2N scenario [23]

To implement the cellular V2N scenario in the SimuLTE project of OMNeT++, it is needed to create a Network Description (NED) file containing as submodules an LTE eNodeB, a Packet Data Network Gateway (PGW), an IP router and a server from the class *StandardHost*. Moreover, a manager from the class *VeinsInetManager* must be added to insert SUMO vehicles in OMNeT++ and manage their mobility through the interface between SUMO and OMNeT++. The graphic representation of the presented NED file in OMNeT++ is showed in the following figure (Figure 4.7):

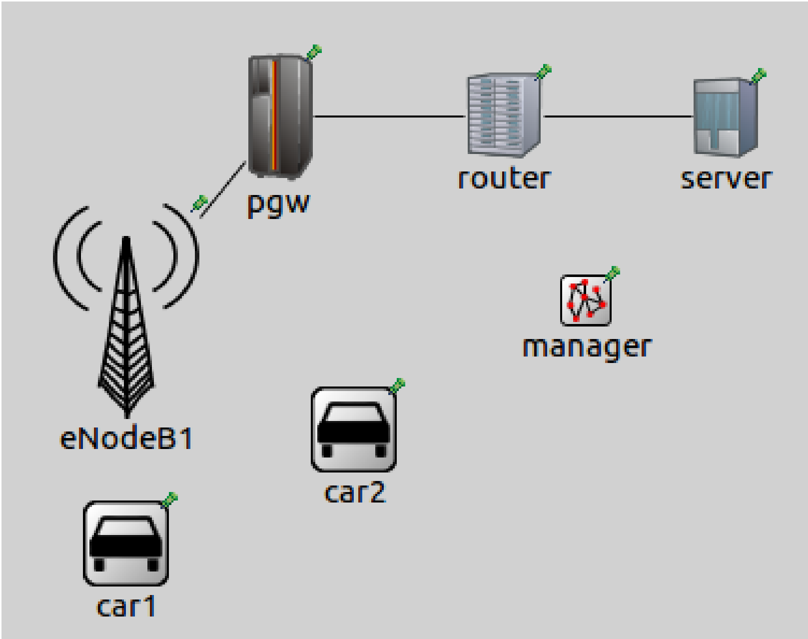

Figure 4.7. OMNeT++ implementation of C-V2N scenario

Furthermore, in the configuration file (*.ini*) several parameters must be specified: the kind of vehicular module to be launched by the mobility manager throughout the simulation; the kind of application to be run by the server and the vehicles; the application parameters; LTE parameters; channel parameters; simulation specific parameters.
In this scenario, the network server is the server of the collision avoidance application, thus it is responsible of running the detector and eventually sending alerts to the involved vehicles. The vehicles, instead, are the clients of the collision avoidance application, thus they have to send beacons and process received alerts by exploiting LTE at the access layer [23]. Moreover, in this case the client part of the application contains indications about the vehicles' behaviour in case of alert reception, as already described in the subparagraphs 4.3.1 and 4.3.3.

In this scenario, the channel model considered to simulate the LTE communications between the vehicles and the eNodeB is the one presented in [40], according to which the Signal-to-Interference-plus-Noise Ratio (SINR) at every signal reception is given by the following formula:

$$SINR = \frac{P_{RX}[mW]}{\sum_i P_{RX_i}[mW] + N[mW]} \tag{4.1}$$

where P_{RX} is the power of the received signal, P_{RX_i} the interfering power received from interfering eNodeBs, and N is the Gaussian noise. For the sake of simplicity, in the simulations only one eNodeB was considered and therefore the component of interfering power from other base stations can be considered negligible. The value of SINR is fundamental as it affects the probability of losing data packets at the physical layer: the higher the SINR value the higher is the probability of forwarding to the upper layers the received packets [40].

The power of the received signal P_{RX} is computed as follows:

$$P_{RX}[dBm] = P_{TX}[dBm] - P_{loss}[dB] - S[dB] \tag{4.2}$$

where P_{TX} is the power of the transmitted signal, P_{loss} is the path loss attenuation and S is the Shadowing attenuation. In this scenario, the P_{loss} and S attenuations are modelled according to [41] and considering the Urban Macro scenario. More specifically, in [41] a probabilistic model determines whether the transmitter and the receiver (in this case the vehicle and the eNodeB) are in Line-of-Sight (LOS) or in Non-Line-of-Sight (NLOS) according to a probability which depends on the distance among the two involved entities. The idea is that the closer the entities the higher the probability of being in LOS conditions. Being in LOS or NLOS conditions change the way the simulation model computes the P_{loss} attenuation and the Shadowing attenuation S.

The LOS probability P_{LOS} is determined according to the following formula [41] :

$$P_{LOS} = \min(18/d,1) * (1 - e^{-d/63}) + e^{-d/63} \tag{4.3}$$

where d is the distance between the transmitter and the receiver.

The path loss attenuation in LOS conditions P_{loss}^{LOS} is given by the formula in [41] :

$$P_{loss}^{LOS} = \begin{cases} 22log_{10}(d) + 28 + 20log_{10}(f_c) & 10m < d < d'_{BP} \\ 40log_{10}(d) + 7.8 - 18log_{10}(h'_{BS}) - 18log_{10}(h'_{UT}) + 2log_{10}(f_c) & d'_{BP} < d < 5000m \end{cases} \tag{4.4}$$

where d is the distance between the transmitter and the receiver, f_c is the carrier frequency, h'_{BS} is the effective antenna height of the base station (given by the actual height minus 1 meter), h'_{UT} is the effective antenna height of the user terminal (given by the actual height minus 1 meter). Moreover, the d'_{BP} is the effective break point distance, which is computed as follows [41] :

$$d'_{BP} = \frac{4h'_{BS}h'_{UT}f_c}{c} \tag{4.5}$$

where c is the value of the speed of light.

The path loss attenuation in NLOS conditions $P_{loss}^{NLOS}$is given by the formula in [41] :

$$P_{loss}^{NLOS} = 161.04 - 7.1log_{10}(W) + 7.5log_{10}(h) - (24.37 - 3.7(h/h_{BS})^2)log_{10}(h_{BS})$$
$$+(43.42 - 3.1log_{10}(h_{BS}))log_{10}(d-3) + 20log_{10}(f_c) - (3.2(log_{10}(11.75h_{UT}))^2 - 4.97) \quad (4.6)$$

where W is the street width, h is the average building height, h_{BS} is the actual antenna height of the base station and h_{UT} is the actual antenna height of the user terminal.

For what concerns the Shadowing attenuation S, in [41] it is modelled with a log-normal distribution with zero mean and a standard deviation that depends on LOS or NLOS conditions: in case of LOS conditions the standard deviation is of $4dB$, otherwise it is of $6dB$.

4.4.2 Implementation of Cellular V2V scenario

To implement the cellular V2V scenario in OMNeT++, it is needed to extend the default NED file termed *SingleCell_D2D.ned* located in the folder of the available networks for the SimuLTE project by including the manager from the *VeinsInetManager* class.
Furthermore, in the configuration file (*.ini*) several parameters must be specified: the kind of vehicular module to be launched by the mobility manager throughout the simulation; LTE parameters; channel parameters; simulation specific parameters; support of LTE D2D capability for vehicles and eNodeB; the kind of application to be run by the vehicles; application parameters. Since it is a distributed scenario, the vehicles act as both clients and servers of the collision avoidance application: therefore, they have to send their beacons and alerts and process beacons and alerts from other vehicles by exploiting the LTE-based D2D communication.
In this scenario, the LTE communications between the vehicles and the eNodeB (useful for resources scheduling) follow the same approach of the previous subparagraph 4.4.1. Instead, the modelling of the SINR associated to the transmission (and reception) from vehicle i to vehicle j is given by the following formula: [42]

$$SINR(i,j) = \frac{P_{RX}^{ij}[mW]}{\sum_k P_{RX}^{kj}[mW] + N[mW]} \quad (4.7)$$

where P_{RX}^{ij} is the power received by vehicle j from vehicle i, N is the Gaussian noise and the summation indicates the interference from other vehicles, as the term P_{RX}^{kj} is the power received by vehicle i from an interfering vehicle k. In this case, the interference is caused whenever a generic vehicle k has used the same Resource Block (RB) of vehicle i at the time of the transmission [42]. Again, the value of the SINR affects the probability of losing data packets at the physical layer.

The power of the received signal P_{RX} (which can be a useful term or an interference term depending on the particular transmission considered) is computed as in the equation

(4.2). Nevertheless, the path loss P_{loss} and Shadowing S attenuations are different in this case and modelled according to [43], which defines the channel models for D2D in Outdoor to Outdoor communications.
Starting from the LOS probability P_{LOS}, it is determined by the following formula [44]:

$$P_{LOS} = \min(18/d,1) * (1 - e^{-d/36}) + e^{-d/36} \tag{4.8}$$

where d is the distance between the transmitter and the receiver (both are vehicles).

Before presenting the path loss attenuation P_{loss} according to [43], it is necessary to introduce two elements: the free space path loss $P_{loss-freespace}$ and a default path loss $P_{loss-default}$. More specifically, the $P_{loss-freespace}$ is given by the following formula [44] :

$$P_{loss-freespace} = 20log_{10}(d) + 46.4 + 20log_{10}(f_c/5) \tag{4.9}$$

where d is the distance between vehicles and f_c is the carrier frequency.

The default path loss $P_{loss-default}$ is specified in [45] according to LOS or NLOS conditions. More specifically, the formula for the LOS conditions $P^{LOS}_{loss-default}$ is the following:

$$P^{LOS}_{loss-default} = \begin{cases} 22.7log_{10}(d) + 27 + 20log_{10}(f_c) & 10m < d < d'_{BP} \\ 40log_{10}(d) + 7.56 - 17.3log_{10}(h'_{UT}) - 17.3log_{10}(h'_{UT}) + 2.7log_{10}(f_c) & d'_{BP} < d < 5000m \end{cases} \tag{4.10}$$

where the h'_{UT} is obtained as in the previous subparagraph, but in this case the effective breakpoint distance is computed in a different way:

$$d'_{BP} = \frac{4h'_{UT}h'_{UT}f_c}{c} \tag{4.11}$$

Here the effective antenna height of the base station is not considered anymore as such effective breakpoint distance is related to D2D communications.

The formula for the default path loss in case of NLOS conditions $P^{NLOS}_{loss-default}$ is instead the following:

$$P^{NLOS}_{loss-default} = (44.9 - 6.55log_{10}(h_{UT}))log_{10}(d) + 5.83log_{10}(h_{UT}) + 16.33 + 26.16log_{10}(f_c) \tag{4.12}$$

At the end, the final path loss attenuation P_{loss} is given by the following formula [43] :

$$P_{loss} = \max(P_{loss-default}, P_{loss-freespace}) \tag{4.13}$$

Finally, for what concerns the shadowing attenuation S, [43] specifies that it is modelled with a log-normal distribution with zero mean and a fixed standard deviation of $7dB$.

4.4.3 Implementation of 802.11p-based V2V scenario

To implement the 802.11p-based V2V scenario in the Veins project of OMNeT++, it is necessary to change the default road scenario with the considered road scenario and make vehicles able to support a customized application for collision avoidance.
Such application has been adapted from the one of the C-V2V scenario and has been deployed on top of the base WAVE application layer of Veins. In this case, vehicles act as both clients and servers of the application.
Furthermore, in the configuration file (*.ini*) several parameters must be specified: 802.11p Network Interface Card (NIC) parameters; application layer parameters; simulation specific parameters; coordinates of the obstacles (buildings) within the reference road scenario; shadowing model associated with the presence of obstacles. Considering the last two parameters, their presence in the configuration file is justified in [46], according to which it is possible to match real-world measurements of 802.11p transmissions carried out in urban environments with the results of an empirical model that computes the power received after a transmission through the following formula:

$$P_r[dBm] = P_t[dBm] + 10log_{10}(\frac{G_t G_r \lambda^2}{16\pi^2 d^\alpha}) - \beta n - \gamma d_m \tag{4.14}$$

where P_r is the power received, P_t is the power transmitted, G_t and G_r are respectively the transmitter and receiver antenna gains, λ is the wavelength, d is the distance between the sender and the receiver, α is a path loss exponent, β is the attenuation per wall in dB, n is the number of times that the line of sight between the vehicles intersects a building wall, γ is the attenuation per meter in dB/m and d_m is the total length of the buildings' intersection by the line of sight.
Therefore, such model allows to deterministically account for the geometry of the buildings placed in the simulation scenario: whenever two vehicles are in NLOS conditions because of a building between them, then their transmission power will be subject to a shadowing effect which causes an attenuation due to the number of building walls between them and another attenuation due to the length of the penetration.
It is important to underline that this deterministic model is very specific and has been considered only because of the related validation in [46].
On the other hand, in the C-V2V scenario a more general and probabilistic model has been considered because no similar empirical models, exploiting the buildings geometry and matching real-world C-V2V measurements, have been created.

4.4.4 Implementation of Hybrid V2N Scenario

To implement the Hybrid V2N scenario in OMNeT++, it has been necessary to extend the C-V2N scenario by considering also vehicles equipped with 802.11p technology and by inserting a 802.11p RSU at the middle of the intersection. More specifically, such RSU ensures the integration between the 802.11p world and the LTE world through an IP-based interface connected to a router which in turn is connected to the network server, as it is possible to observe from the following figure (Figure 4.8):

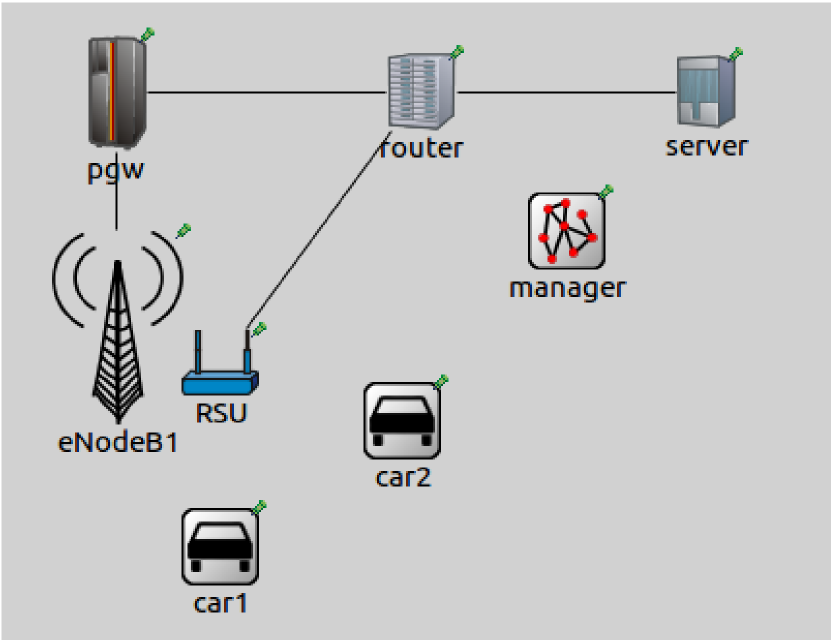

Figure 4.8. Connection of modules in the OMNeT++ Hybrid V2N scenario

Since this scenario includes vehicles equipped with OBUs enabling C-V2N LTE communications and vehicles equipped with OBUs enabling V2V and V2I through 802.11p NICs, the channel models described for the implementation of the C-V2N scenario and the 802.11p-based V2V scenario can be kept unchanged respectively for LTE communications and 802.11p communications of this implementation.

In this case, all vehicles are only clients of the collision avoidance application, thus the LTE vehicles send their beacons to the server as already seen in the C-V2N scenario, whereas the 802.11p vehicles send their beacons in broadcast and only the RSU processes them after reception.

To ensure the integrability between the two technologies, it is required that the RSU processes the beacons from 802.11p vehicles, adapts them for the IP world and finally sends them to the application server. Furthermore, the network server has to process the received beacons and generate alerts in case of potential collision detected among couples of vehicles. In the latter case, the server tries to resolve the IP addresses of the intended

destinations and whenever it is not able to resolve at least one address per couple, then it means that there is at least one 802.11p vehicle involved in the collision: at this point, the server has to send the alert to the IP interface of the RSU. Then, the RSU will receive and process the alert, adapt it for the WAVE world and finally send it in broadcast through its 802.11p NIC. In this way, the intended destination(s) can properly receive the alert.
To achieve such behaviour, it has been necessary to implement an RSU which adapts the beacons of the 802.11p world into beacons of the IP world and the other way around the alerts from the IP world into alerts of the 802.11p world. This adaptation is performed by the module *Adapter* implemented in the RSU, which is on one side directly connected to the application responsible of processing the 802.11p beacons and sending the 802.11p alerts and on the other side directly connected to the application responsible of processing the IP alerts and sending the IP beacons.
In the following figure (Figure 4.9), it is possible to observe the internal implementation of the RSU in OMNeT++:

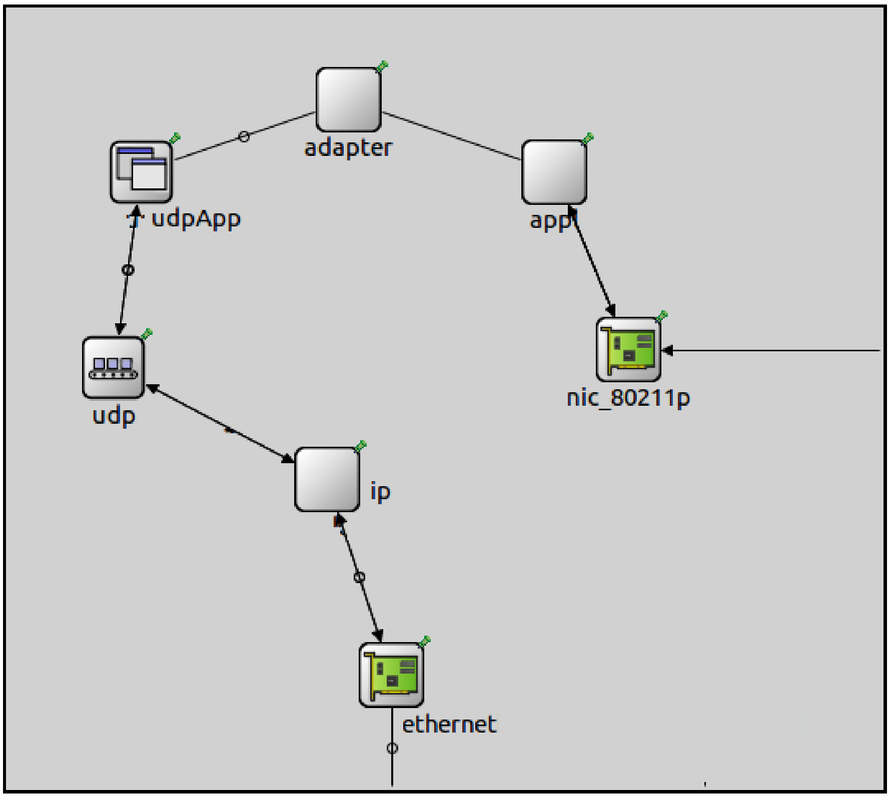

Figure 4.9. OMNeT++ implementation of the RSU in the Hybrid V2N scenario

Chapter 5

Simulation Results

This chapter deals with the results coming from the simulation scenarios described in the previous chapters, according to the simulation methodology presented in the paragraph 4.3. More specifically, the reported results concern with the evaluation of the proposed collision avoidance application's performances in terms of successfully avoided or not avoided collisions and in terms of true or false positives triggered by the detection algorithm of the application, as already described in the paragraph 4.3.2.
Furthermore, for every scenario considered, two different kinds of simulations with SUMO and OMNeT++ have been launched: one with only vehicles equipped with autonomous braking systems and the other with only vehicles without any level of automation (human drivers).

5.1 SUMO simulations results

According to the simulation methodology presented in the paragraph 4.3, it is necessary to run simulations with only SUMO before running the simulations with SUMO and OMNeT++ coupled. SUMO simulations take as inputs a file containing the road environment and a file containing the properties of the vehicles to launch in terms of maximum speed, maximum deceleration, maximum acceleration, route and generation time. The only output to consider is represented by the occurred collisions, which should be avoided by introducing the support of vehicular communication technologies.
Considering the same road environment input file and different input files containing the properties of the vehicles to launch, a total of 10 SUMO simulations lasting 1500 seconds have been launched and their results in terms of number of occurred collisions are reported in the following table (Table 5.1):

Run number	Number of collisions
#1	16
#2	12
#3	22
#4	18
#5	9
#6	16
#7	12
#8	19
#9	18
#10	18

Table 5.1. SUMO simulations results

It is important to underline that in the simulations with SUMO and OMNeT++ coupled, the SUMO part will take the same input files described in this paragraph, but the output produced will be different thanks to the functionalities added by OMNeT++ to avoid the collisions. Therefore, also the simulations with SUMO and OMNeT++ coupled will be 10 for each considered scenario, and the related results will be averaged over such 10 runs.

5.2 Cellular V2N simulation results

In this paragraph, the results of the LTE cellular V2N scenario are presented considering simulations with only vehicles equipped with autonomous braking systems and simulations with only vehicles driven by human drivers.

5.2.1 C-V2N with Autonomous Braking

In this specific scenario, all the vehicles are equipped with autonomous braking systems that allow to have a fast reaction towards the received alerts and the thresholds of the collision detection algorithm (paragraph 3.2.1) have been chosen to maximize the number of successfully avoided collisions by keeping at the same time a reasonably low number of false positives. Moreover, the Metro Node and the Cloud have been considered as positions of the network server, determining different values for the latency to reach the base station (paragraph 3.4.2). In the following table, the parameters of this scenario are reported:

Collision detector time threshold	1.5 s
Collision detector space threshold	3.7 m
Latency eNodeB-Metro Node Server	5 ms
Latency eNodeB-Cloud Server	20 ms

Table 5.2. C-V2N with Autonomous Braking parameters

In the following figure (Figure 5.1), the performances of the application in terms of collision avoidance are shown, considering the network server placed at the Metro Node in a MEC fashion and in the Cloud:

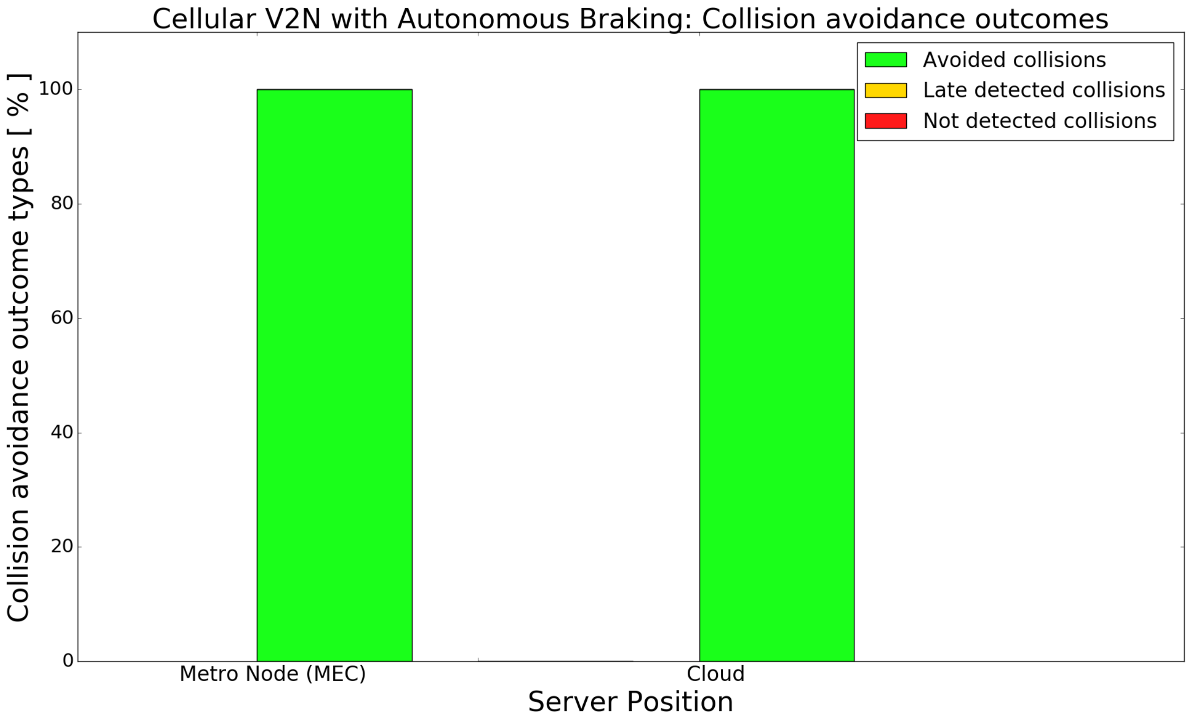

Figure 5.1. C-V2N with Autonomous Braking: Collision avoidance outcomes

The figure above show that the application allows in both cases to completely avoid the 100% of the collisions that there would have been in the simulations with only SUMO. This happens thanks to the timing with which beacons and alerts are transmitted and received by the involved entities. Nevertheless, to properly evaluate the performances of the application it is needed also to quantify the number of false positives triggered by the detector. Indeed, a detector which is able to avoid the 100% of the collisions by triggering alerts at any situation would reduce the quality of the application, since also not really dangerous situations would be alerted.

In the following figure (Figure 5.2), the performances of the application in terms of detections quality are represented, considering again the two positions for the network server:

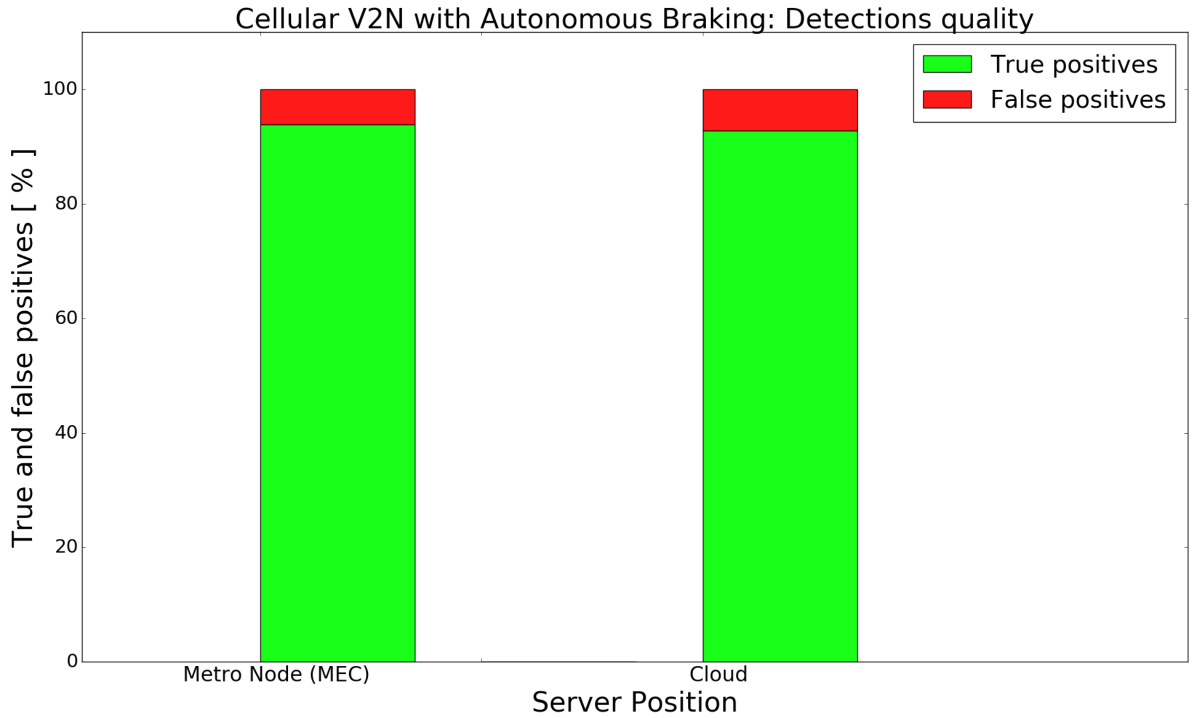

Figure 5.2. C-V2N with Autonomous Braking: Detections quality

According to the figure above, in both cases the detector generates a very low percentage of false positives and the detections are mainly due to alert vehicles that are going to actually collide. In this case, the high detection accuracy is ensured by the availability of timely updated information about the mobility status of the vehicles. Moreover, the case with the server at the Metro Node ensures a slightly better detection quality than the case with the server in the Cloud because of the lower value of latency to reach the base station.

In reality, the assumed values for the eNodeB-server latency might be different because of many real-world phenomena and circumstances. Therefore, analysing the performances when the value of the considered latency is assumed to be bigger might be useful to evaluate the sensitivity of the application with the chosen parameters.
In the following figures and tables, the results of such analysis are presented, starting from the collision avoidance outcomes:

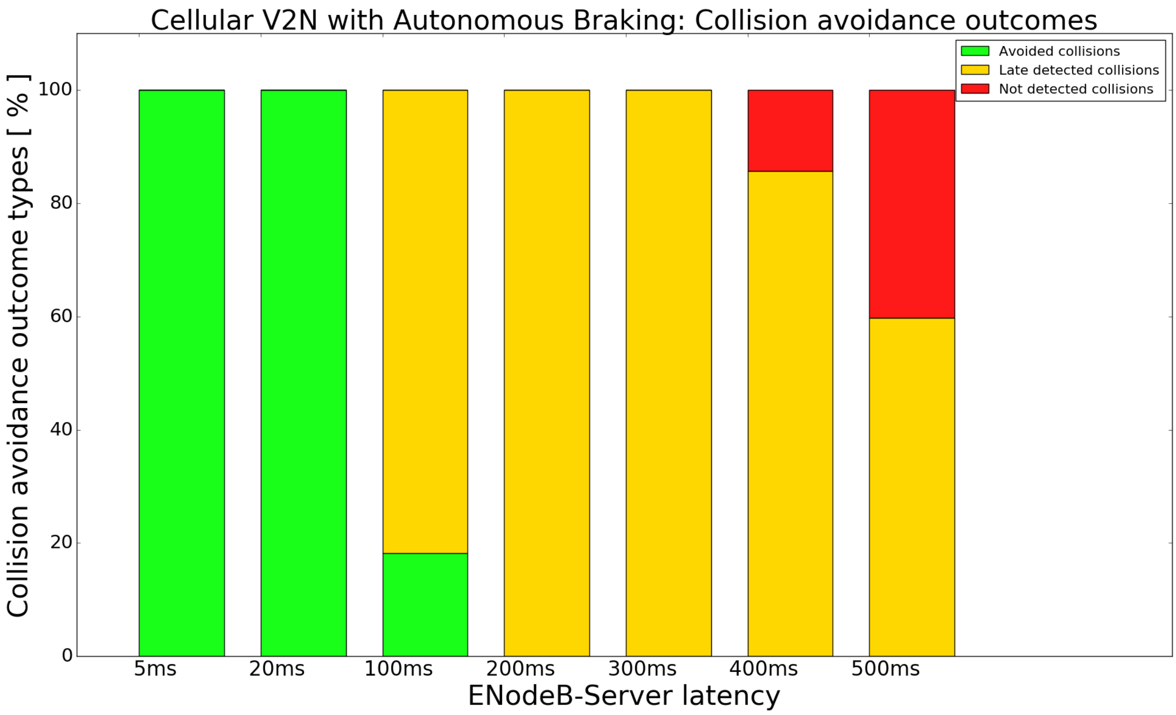

Figure 5.3. C-V2N with Autonomous Braking: Collision avoidance outcomes by varying eNodeB-server latency

	Avoided collisions	**Late detected collisions**	**Not detected collisions**
5ms	100%	0%	0%
20ms	100%	0%	0%
100ms	18.2%	81.8%	0%
200ms	0%	100%	0%
300ms	0%	100%	0%
400ms	0%	85.7%	14.3%
500ms	0%	59.7%	40.3%

Table 5.3. C-V2N with Autonomous Braking: Collision avoidance numerical results

Results show that the application has worse performances with increasing values of the considered latency: indeed, also with 100ms of latency about the 82% of collisions are not avoided since the alerts are not triggered and received in time. Furthermore, starting from 400ms of latency some occurred collisions are not even detected since the detector does not have updated information about the current situation in the road environment because of too late received beacons.
Potentially, to reduce the number of not avoided collisions when high values of the eNodeB-server latency are available it could be possible to impose higher thresholds in the detection

algorithm. Nevertheless, using higher thresholds means to increase the probability of generating false positives, thus the performances of the application would gain in terms of number of avoided collisions but lose in terms of detection quality.
Consider now the performances in terms of detection quality:

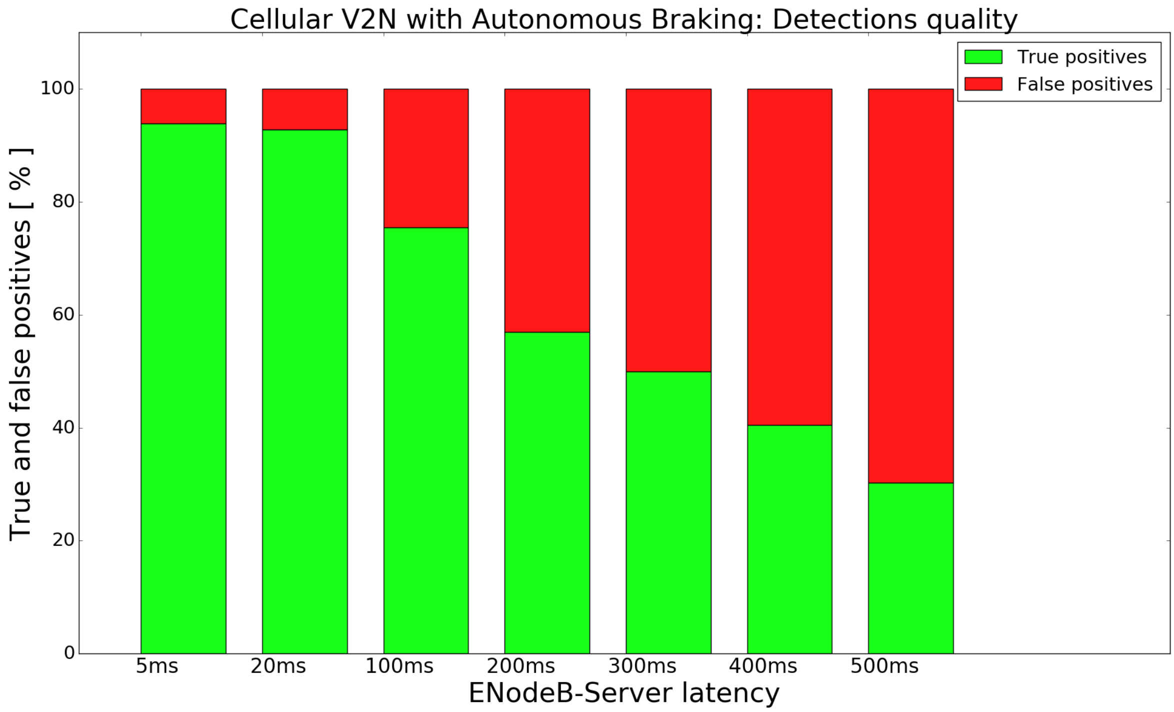

Figure 5.4. C-V2N with Autonomous Braking: Detections quality by varying eNodeB-server latency

	True positives	**False positives**
5ms	93.9%	6.1%
20ms	92.8%	7.2%
100ms	75.5%	24.5%
200ms	57%	43%
300ms	50.4%	49.6%
400ms	40.7%	59.3%
500ms	30.3%	69.7%

Table 5.4. C-V2N with Autonomous Braking: Detections quality numerical results

For what concerns the false positives evaluation, with higher values of latency the detector is less accurate and thus more unnecessary alerts are generated.

5.2.2 C-V2N with Human Driver

In this scenario, all the vehicles do not present any autonomous braking system and thus human reaction is required upon alert reception. Also in this case, the thresholds of the collision detection algorithm (paragraph 3.2.1) have been chosen to maximize the number of successfully avoided collisions by keeping at the same time a reasonably low percentage of false positives. More specifically, the time threshold has been increased with respect to the C-V2N scenario with autonomous braking in order to account for the human reaction time, already described in paragraph 3.3.2. In Table 5.5, the specific parameters of this scenario are reported:

Human reaction time	1 s
Collision detector time threshold	2.5 s
Collision detector space threshold	3.7 m

Table 5.5. C-V2N with Human Driver parameters

In the following figure and table, the performances of the application in terms of collision avoidance are shown, considering the eNodeB-server latency values described in paragraph 3.4.2 and also bigger values for such latency:

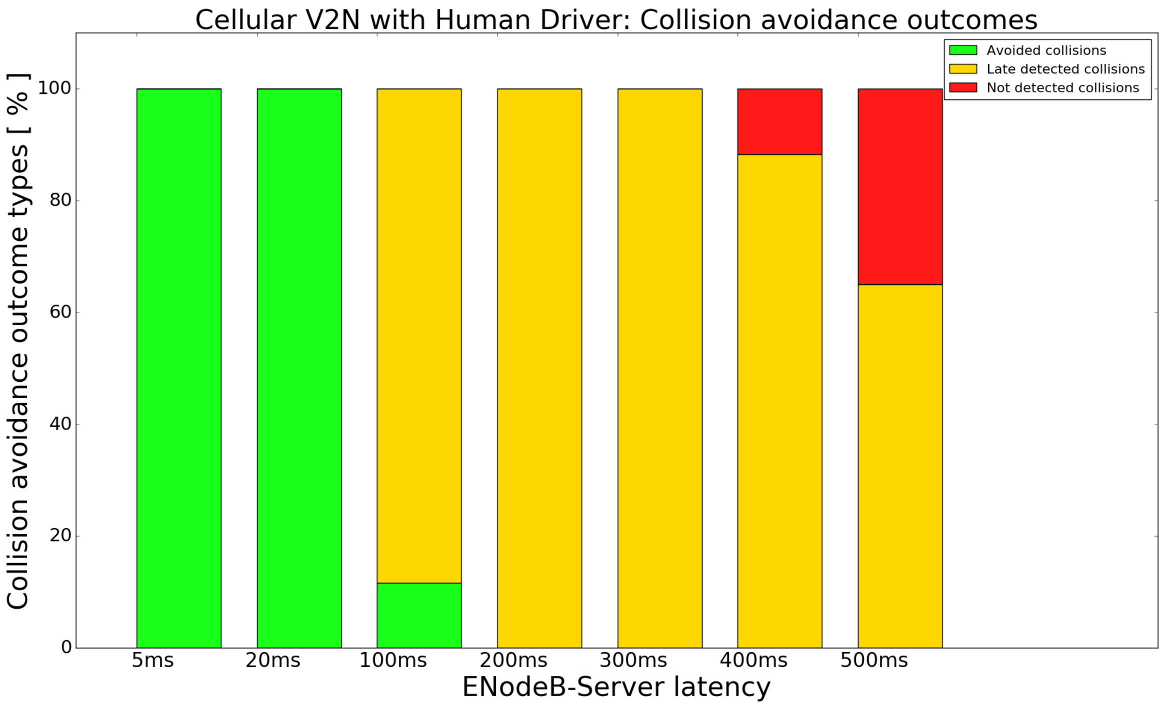

Figure 5.5. C-V2N with Human Driver: Collision avoidance outcomes

	Avoided collisions	**Late detected collisions**	**Not detected collisions**
5ms	100%	0%	0%
20ms	100%	0%	0%
100ms	11.7%	88.3%	0%
200ms	0%	100%	0%
300ms	0%	100%	0%
400ms	0%	88.3%	11.7%
500ms	0%	65%	35%

Table 5.6. C-V2N with Human Driver: Collision avoidance numerical results

The reported results state that the application shows worse performances with increasing values of the eNodeB-server latency, as late detections and not detected collisions begin to occur respectively at 100ms and 400ms. A significant consideration that comes out from the obtained results is that the percentage of not detected collisions is smaller than the one of the case with autonomous braking, considering latency values of 400ms and 500ms: this result is expected, since increasing the time threshold of the detection algorithm implies to reduce the false negative probability.
It is important to note also that the 100% of the potential collisions are avoided with values of latency of 5ms and 20ms, as in the previous case with autonomous braking: this is possible only because the time threshold has been increased to account for the human reaction time, otherwise the time threshold of the previous case (1.5s) would have caused several late detected collisions. Therefore, for these values of latency it is possible to achieve the same performances of the case with autonomous braking in terms of successfully avoided collisions, but the higher threshold adopted comes at the cost of a higher false positive probability, as it is possible to observe in the following table and figure:

	True positives	**False positives**
5ms	90.6%	9.4%
20ms	89.5%	10.5%
100ms	72.3%	27.7%
200ms	55.6%	44.4%
300ms	48.7%	51.3%
400ms	39.7%	60.3%
500ms	28.4%	71.6%

Table 5.7. C-V2N with Human Driver: Detections quality numerical results

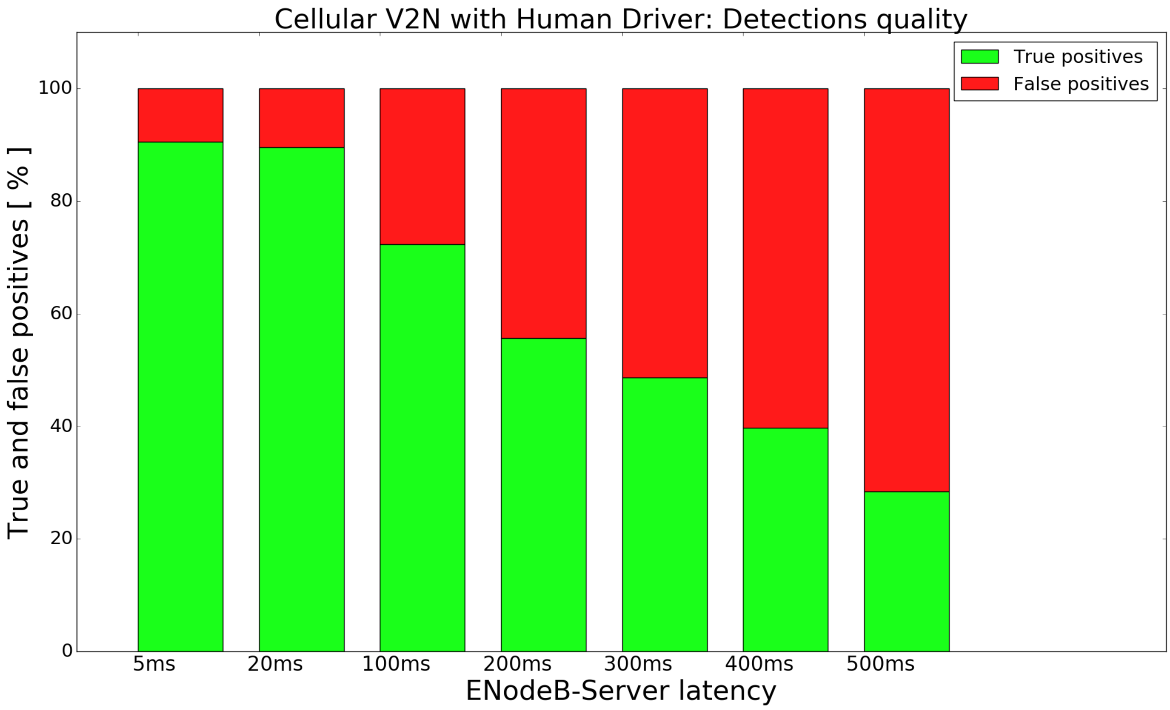

Figure 5.6. C-V2N with Human Driver: Detections quality

From the reported table and figure, it is possible to observe the usual trend of increasing false positives percentage with higher values of the eNodeB-server latency due to the less accuracy of the detector. Moreover, by fixing a specific value of the latency, the performances in terms of false positives are worse in this case with respect to the autonomous braking scenario because of the higher time threshold used in the detection algorithm.

5.3 Cellular V2V simulation results

In this paragraph, the results of the distributed LTE cellular V2V scenario are presented considering simulations with only vehicles equipped with autonomous braking systems and simulations with only vehicles driven by human drivers.

5.3.1 C-V2V with Autonomous Braking

In this scenario, all the vehicles are equipped with autonomous braking systems and with OBUs implementing the LTE D2D communication standard described in the paragraph 3.5. Furthermore, different time thresholds for the collision detection algorithm have been analysed in order to see how the performances of the application vary.

In the following figure and table, the collision avoidance results for this scenario are presented:

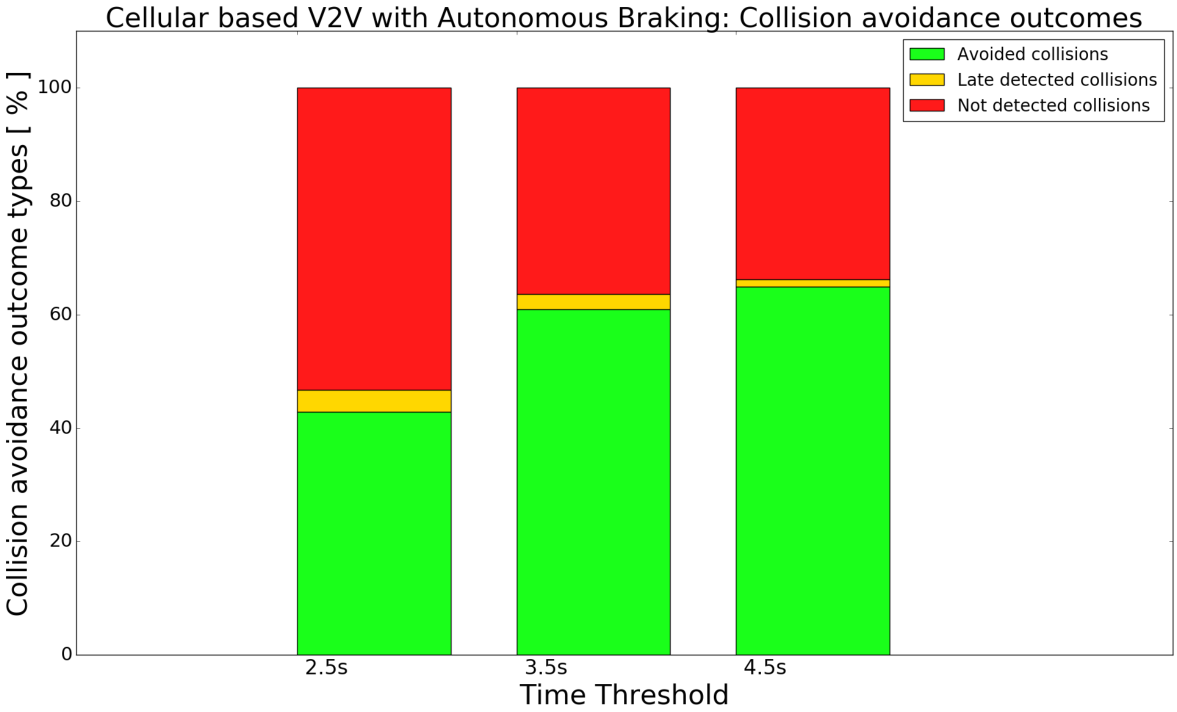

Figure 5.7. C-V2V with Autonomous Braking: Collision avoidance outcomes

Time threshold	**Avoided collisions**	**Late detected collisions**	**Not detected collisions**
2.5s	42.9%	3.9%	53.2%
3.5s	61%	2.6%	36.4%
4.5s	64.9%	1.3%	33.8%

Table 5.8. C-V2V with Autonomous Braking: Collision avoidance numerical results

Results show that a significant percentage of collisions is not even detected by the application: this behaviour is caused by the fact that NLOS conditions between transmitter and receiver increase the probability of not receiving beacons and thus the detection algorithm implemented at the vehicles cannot use updated information about the mobility status of the other vehicles to notice potential collision courses. Moreover, by increasing the value of the time threshold, higher percentages of avoided collision are obtained, although this implies worse performances in terms of false positives.

In the following figure and table, results in terms of detection quality are reported:

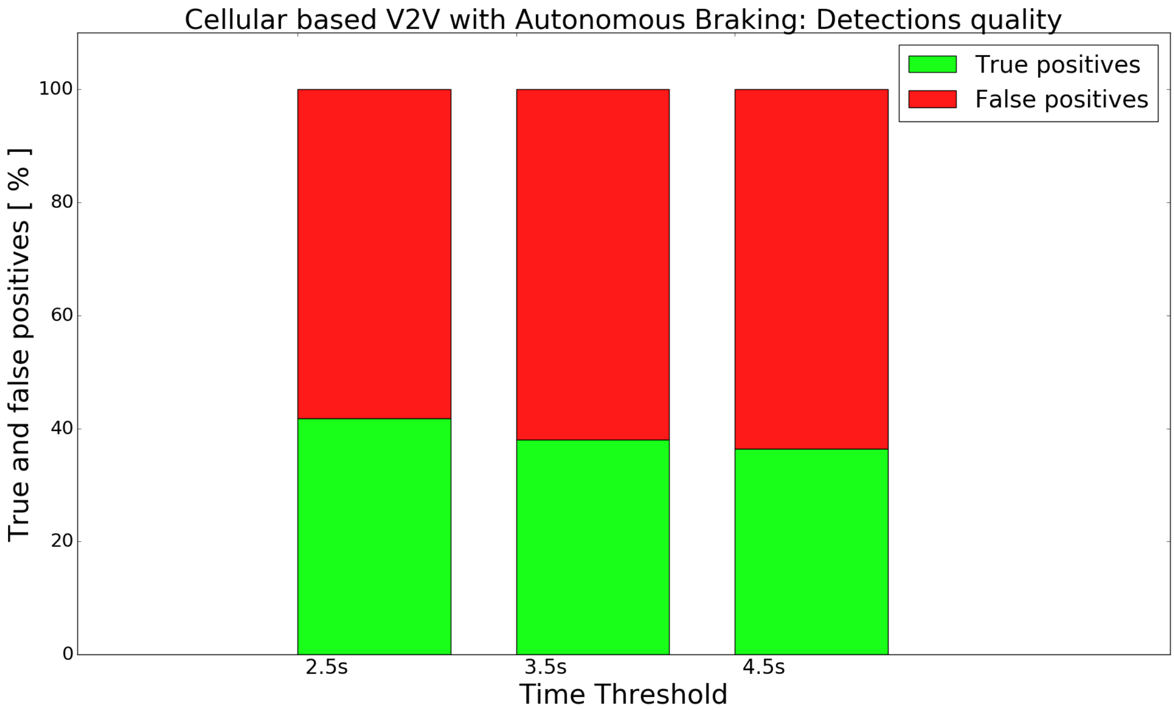

Figure 5.8. C-V2V with Autonomous Braking: Detections quality

Time threshold	**True positives**	**False positives**
2.5s	41.7%	58.3%
3.5s	38.1%	61.9%
4.5s	36.4%	63.6%

Table 5.9. C-V2V with Autonomous Braking: Detections quality numerical results

According to the reported results, as expected, higher percentages of false positives are obtained with increasing value of the time threshold. Moreover, by observing the numerical values it can be stated that in any case the application does not perform well because of the very high number of false positives. It is interesting to notice that the percentage of false positives of this C-V2V scenario is very similar to the equivalent percentage obtained in the C-V2N scenario with an eNodeB-server latency of 400ms: indeed, in this scenario, vehicles act also as servers of the application by processing beacons and the elaboration time needed by the OBUs to process such beacons has been assumed to be 400ms. Therefore, it is expected to have a similar degradation of the detection accuracy in the two cases.

5.3.2 C-V2V with Human Driver

In this scenario, all the vehicles are equipped with OBUs implementing the LTE D2D communication standard described in the paragraph 3.5 and human drivers are supposed to react in case of received alert. Moreover, also in this case different values for the time threshold of the detection algorithm have been analysed: in particular, they have been shifted of one second with respect to the autonomous braking scenario in order to account for the human reaction time.
In the following figure and table, the collision avoidance results for this scenario are shown:

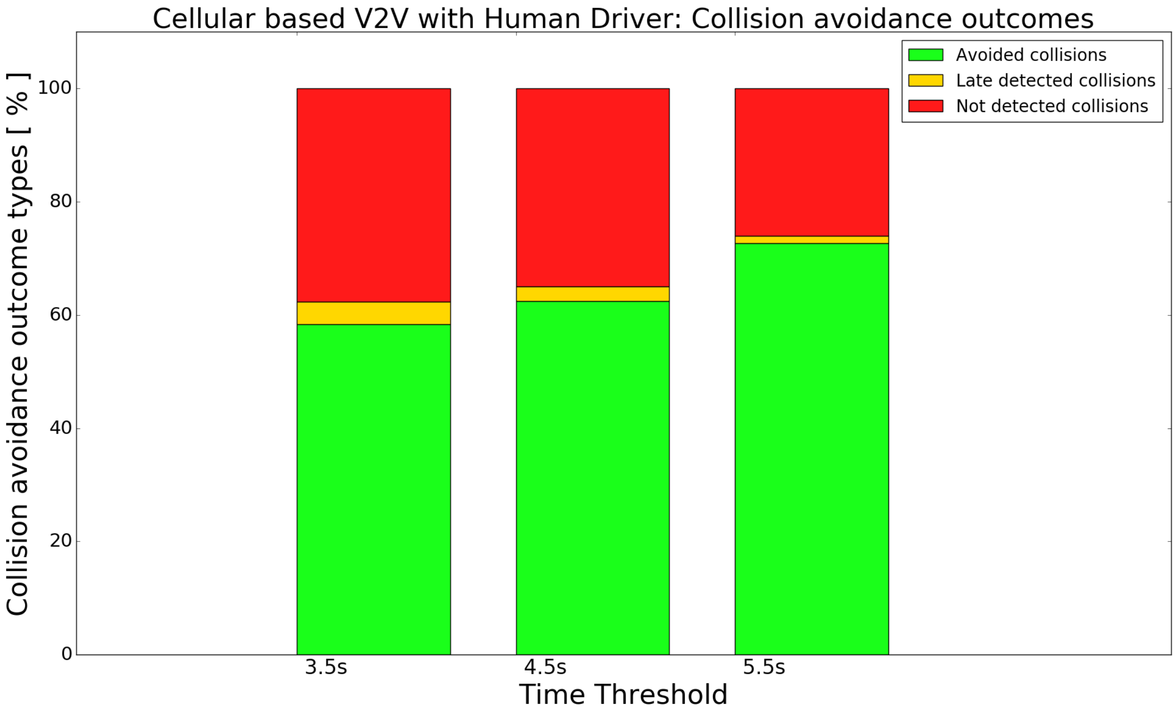

Figure 5.9. C-V2V with Human Driver: Collision avoidance outcomes

Time threshold	**Avoided collisions**	**Late detected collisions**	**Not detected collisions**
3.5s	58.4%	4%	37.6%
4.5s	62.4%	2.5%	35.1%
5.5s	72.7%	1.3%	26%

Table 5.10. C-V2V with Human Driver: Collision avoidance numerical results

The obtained results show the usual trend for the collision avoidance application: higher threshold values allow to avoid more collisions and reduce the false negatives. Comparing these results with the ones of the previous scenario by considering the same time threshold (for example, 3.5s or 4.5s), it is possible to observe that the false negative percentage does not change too much, as it is mainly driven by the values of the thresholds and does not depends on the time needed to react towards a received alert. Moreover, under this perspective slightly better performances are obtained in the scenario with autonomous braking, in which some of the late detected collisions of this scenario turns into avoided collisions thanks to the faster reaction in case of alert reception.

In the following figure and table, results in terms of detection quality are reported:

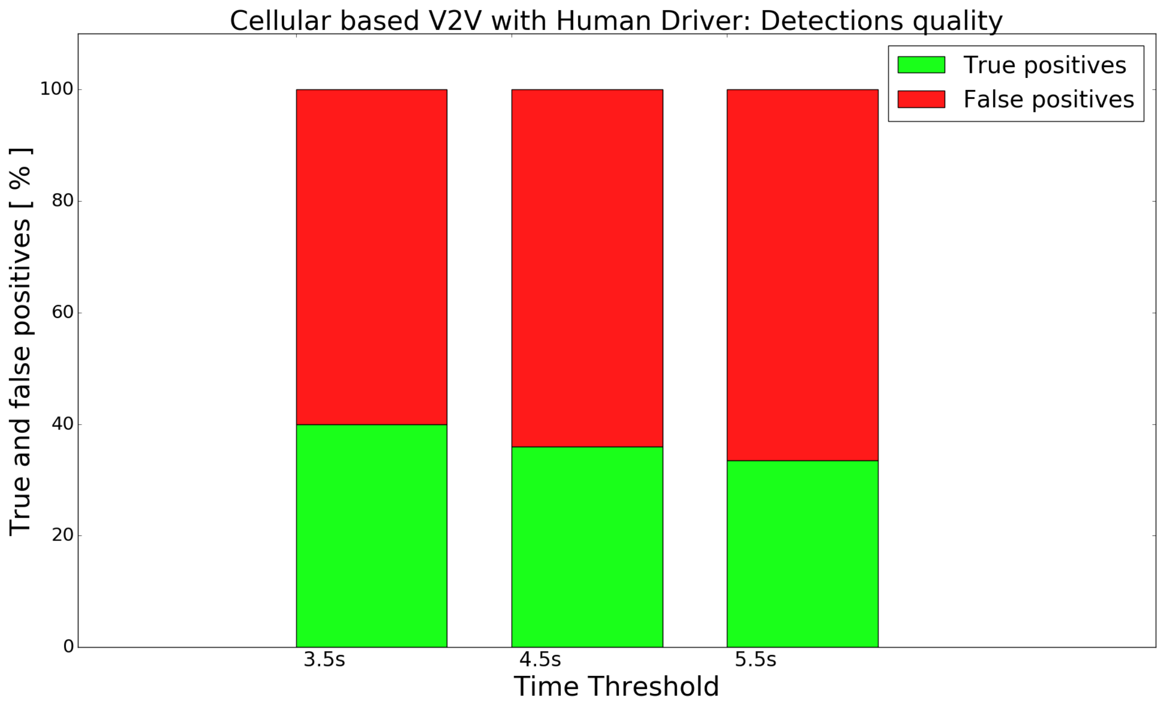

Figure 5.10. C-V2V with Human Driver: Detections quality

Time threshold	**True positives**	**False positives**
3.5s	40%	60%
4.5s	36%	64%
5.5s	33.5%	66.5%

Table 5.11. C-V2V with Human Driver: Detections quality numerical results

Similar considerations to the ones of the autonomous braking scenario can be done: the higher the threshold the higher the percentage of false positives, which is very high in every case. Furthermore, comparing again this scenario with the autonomous braking scenario by considering the same time threshold (for instance 3.5s or 4.5s), it can be stated that the results are practically the same, since the detection accuracy does not depend on how fast the vehicles or drivers react to the received alerts.

5.4 802.11p-based V2V simulation results

In this paragraph, the results of the distributed 802.11p-based V2V scenario are presented considering simulations with only vehicles equipped with autonomous braking systems and simulations with only vehicles driven by human drivers.

5.4.1 802.11p-based V2V with Autonomous Braking

In this scenario, all the vehicles are equipped with autonomous braking systems and with OBUs implementing the DSRC standard described in the paragraph 3.6. Furthermore, different time thresholds for the collision detection algorithm have been analysed in order to see whether the performances of the application vary.

In the following figure and table, the collision avoidance results for this scenario are presented:

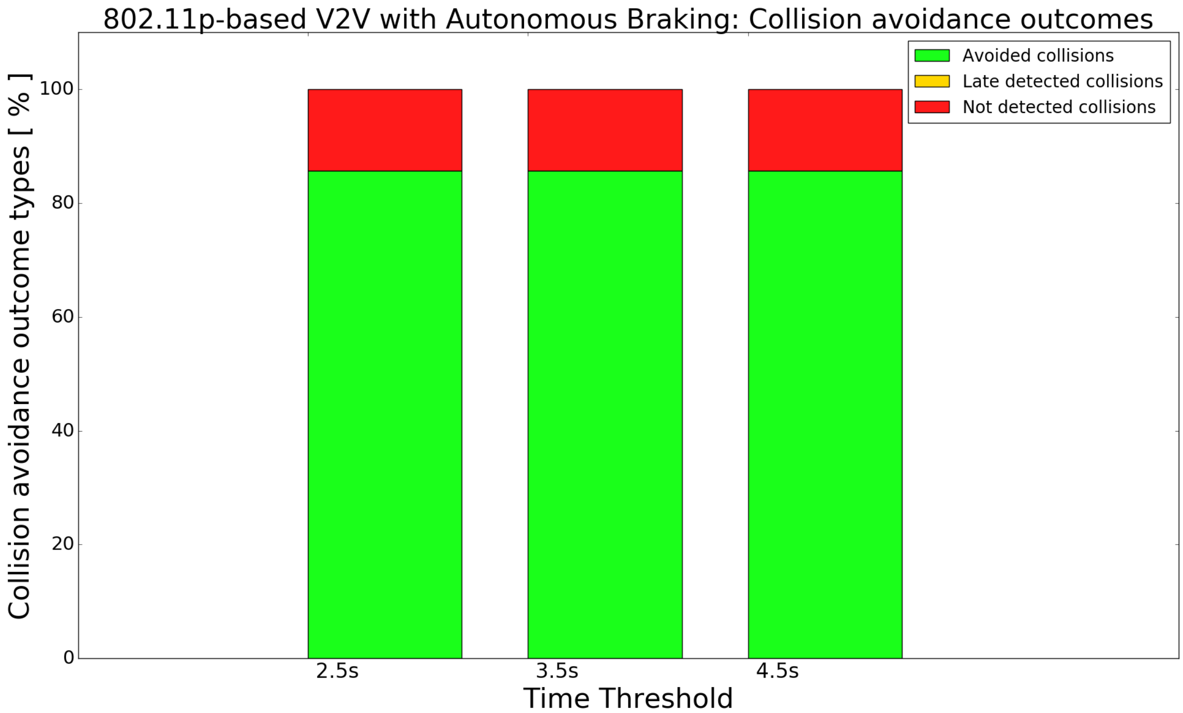

Figure 5.11. 802.11p-based V2V with Autonomous Braking: Collision avoidance outcomes

Time threshold	**Avoided collisions**	**Late detected collisions**	**Not detected collisions**
2.5s	85.7%	0%	14.3%
3.5s	85.7%	0%	14.3%
4.5s	85.7%	0%	14.3%

Table 5.12. 802.11p-based V2V with Autonomous Braking: Collision avoidance numerical results

The obtained results show that by varying the time threshold there are not significant changes: this behaviour might be due to the fact that a deterministic channel model exploiting the coordinates of the buildings to consider NLOS conditions is used. Therefore, in this scenario vehicles are surely able to successfully exchange beacons only when they are very close, since in that case the signal attenuation due to buildings and distance is not enough to penalize the reception. For this reason, increasing the time threshold to try to facilitate and anticipate the generation of alerts is not so effective.
Moreover, also in this case the detection quality is mainly affected by the 400ms needed to process the beacons at vehicles. Thus, the obtained results in terms of not detected collisions are very similar to the ones of the C-V2N scenario with autonomous braking and 400ms of eNodeB-server latency. Nevertheless, with respect to that C-V2N scenario, in this case the rest of collisions are successfully avoided: this happens because in this distributed scenario the alerts are directly exchanged between the vehicles, thus they do not have to reach a base station from a network server and do not suffer from the related latency.
Furthermore, a possible comparison with the C-V2V case is limited by the fact that the simulative environment uses two different channel models for the two different technologies. More specifically, according the probabilistic channel model implemented for the C-V2V, there is not a deterministic distance below which vehicles are surely able to exchange packets. For this reason, in the C-V2V scenario, increasing the time threshold might improve on average the avoided collisions percentage, but at the same time beacons and alerts might be lost also with distances between vehicles that are acceptable for the 802.11p channel model, determining overall worse performances in terms of avoided collisions with respect to this 802.11p-based V2V scenario.

In the following table and figure, results in terms of detection quality are reported:

Time threshold	**True positives**	**False positives**
2.5s	41%	59%
3.5s	39.2%	60.8%
4.5s	38.1%	61.9%

Table 5.13. 802.11p-based V2V with Autonomous Braking: Detections quality numerical results

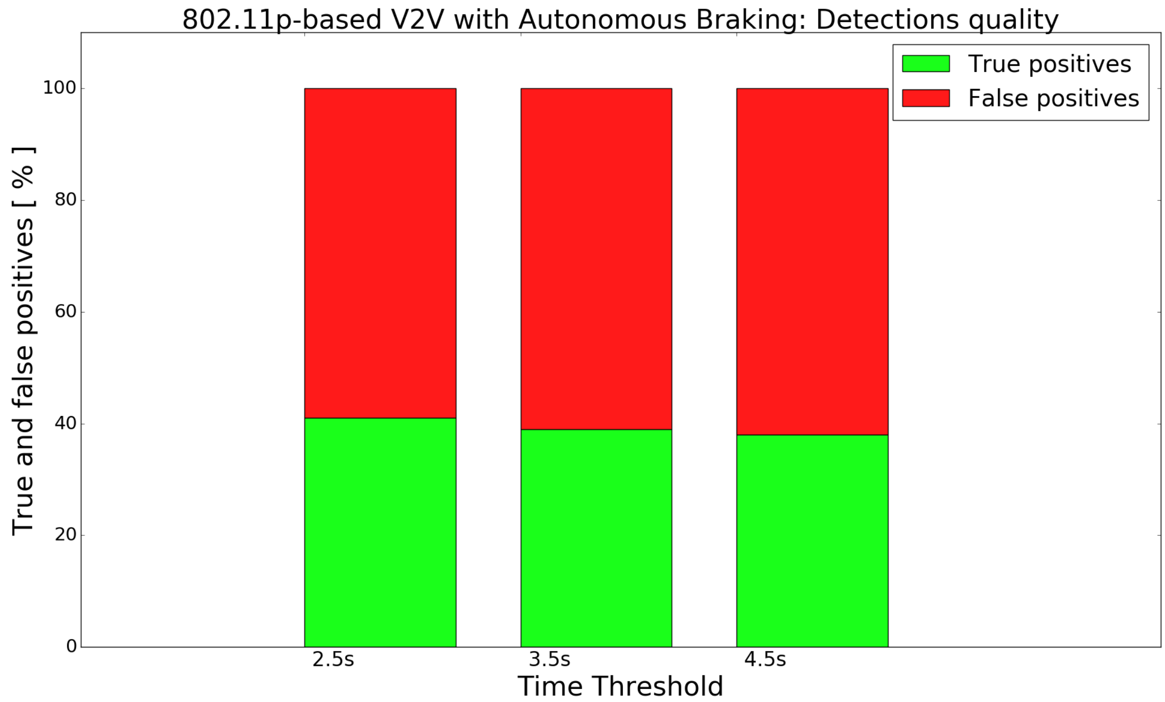

Figure 5.12. 802.11p-based V2V with Autonomous Braking: Detections quality

According to the reported results, it is possible to state that the application does not perform well because of the very high number of false positives. Moreover, also in this case the detection accuracy is mainly driven by the 400ms needed to process the beacons at vehicles and the values of the time threshold do not have a significant impact. Therefore, similar results for false positives are obtained in the C-V2V scenario and in the C-V2N scenario with an eNodeB-server latency of 400ms.

5.4.2 802.11p-based V2V with Human Driver

In this scenario, human drivers and vehicles with OBUs implementing the DSRC standard described in the paragraph 3.6 are considered. Moreover, also in this case different values for the time threshold of the detection algorithm have been analysed and shifted of one second to account for the human reaction time.

In the following figure and table, the collision avoidance results for this scenario are presented:

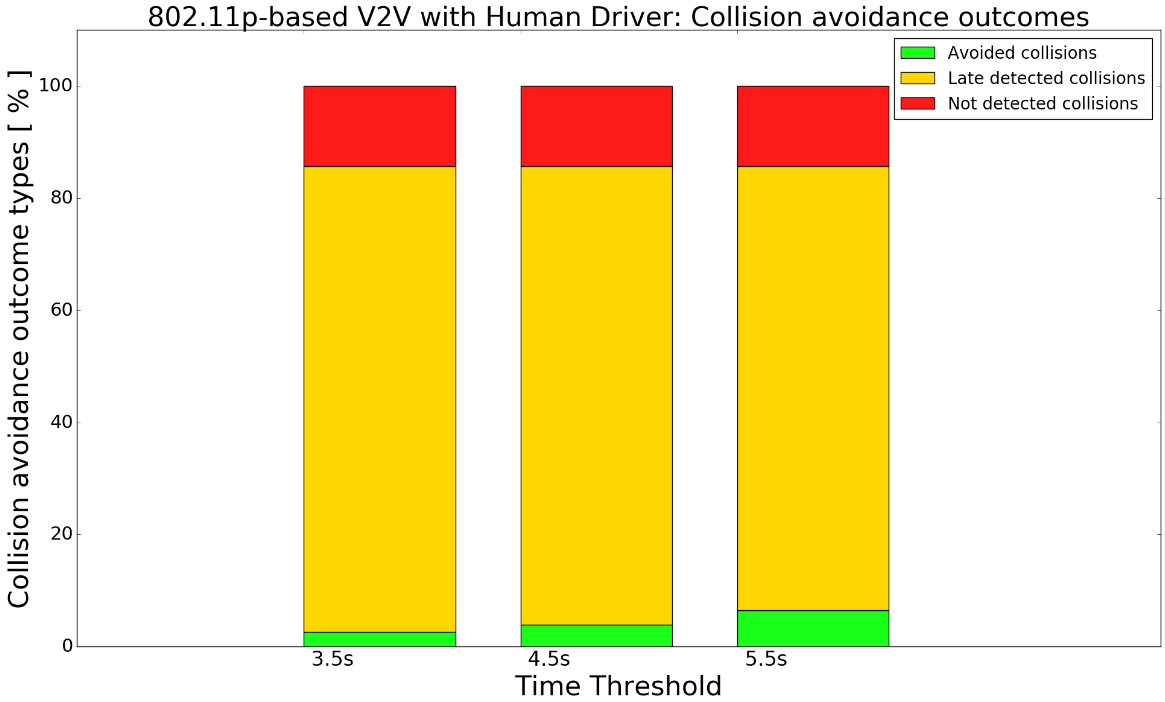

Figure 5.13. 802.11p-based V2V with Human Driver: Collision avoidance outcomes

Time threshold	**Avoided collisions**	**Late detected collisions**	**Not detected collisions**
3.5s	2.6%	83.1%	14.3%
4.5s	3.9%	81.8%	14.3%
5.5s	6.5%	79.2%	14.3%

Table 5.14. 802.11p-based V2V with Human Driver: Collision avoidance numerical results

Results show that the amount of not detected collisions does not change with respect to the case of autonomous braking, since increasing of one second the values of the time threshold does not have a great impact on the false negative probability for this specific scenario. Therefore, collision avoidance performances are expected to be worse than the case of autonomous braking because of the human reaction time: indeed, in this scenario the great majority of collisions is detected too late and a very small percentage of collisions is detected in time.

In the following table and figure, results in terms of detection quality are reported:

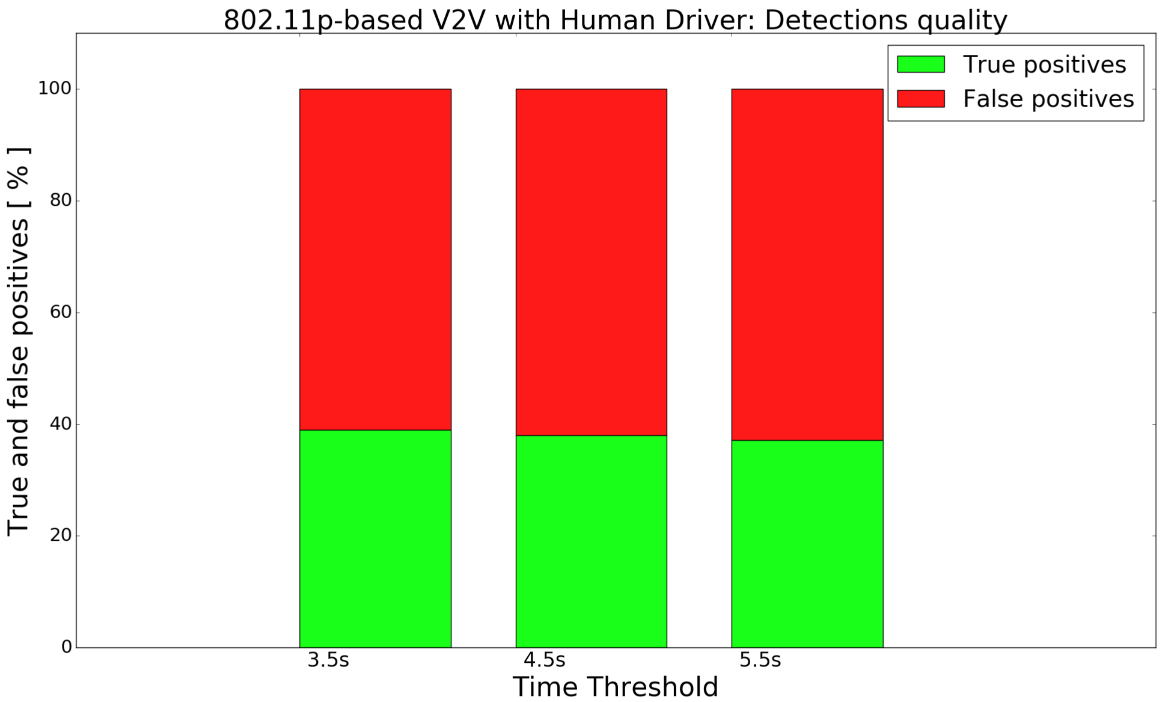

Figure 5.14. 802.11p-based V2V with Human Driver: Detections quality

Time threshold	**True positives**	**False positives**
3.5s	39.2%	60.8%
4.5s	38.1%	61.9%
5.5s	37.2%	62.8%

Table 5.15. 802.11p-based V2V with Human Driver: Detections quality numerical results

According to these results, the application does not perform well because of the high number of false positives, which very slightly increases considering higher values for the time threshold. Anyway, also in this case the detection accuracy is mainly driven by the 400ms to process the beacons at vehicles.

5.5 Hybrid V2N simulation results

This paragraph deals with the simulation results of the Hybrid V2N scenario described in the paragraph 4.4.4, considering simulations with only vehicles equipped with autonomous braking systems and simulations with only vehicles driven by human drivers.
It is important to underline that this scenario is considered to evaluate the integrability of the cellular technology with 802.11p in the context of this specific safety application for

vehicular communications. Indeed, as shown in the previous sections, the applicative performances of 802.11p without infrastructure are too much penalized by NLOS conditions, but anyway 802.11p performs better in many other vehicular applications, as for example platooning. Therefore, having the possibility to dynamically and selectively exploit the advantages of the two proposed approaches would be very advantageous: one on hand, the presence of an infrastructure (LTE eNodeB or 802.11p RSU) allows to better cope with NLOS conditions and will guarantee good performances in the considered intersection collision avoidance application also with the presence 802.11p-equipped vehicles. On the other hand, at the same time the 802.11p OBUs might be exploited to optimize the performances of those applications suitable for direct communications.

5.5.1 Hybrid V2N with Autonomous Braking

In this scenario, all the vehicles are equipped with autonomous braking systems and they are generated in such a way that their OBUs have the same probability to use either LTE or 802.11p at the access layer. Furthermore, in this case the results have been obtained considering the server only at the Metro Node in a MEC fashion, determining an eNodeB-server and also an RSU-server latency of 5ms (paragraph 3.4.2). The thresholds of the collision detection algorithm (paragraph 3.2.1) used are the same of the scenario described in paragraph 5.2.1. The specific parameters of this scenario are reported in the following table (Table 5.16):

Collision detector time threshold	1.5 s
Collision detector space threshold	3.7 m
Latency eNodeB-Metro Node Server	5 ms
Latency 802.11p RSU-Metro Node Server	5 ms
Probability of launching LTE OBUs	0.5
Probability of launching 802.11p OBUs	0.5

Table 5.16. Hybrid V2N with Autonomous Braking parameters

In the following figures and tables, the performances of the application in terms of collision avoidance and detection quality in this hybrid scenario are reported and compared with the C-V2N scenario with the server placed at the Metro Node:

	Avoided collisions	**Late detected collisions**	**Not detected collisions**
Hybrid V2N	100%	0%	0%
Cellular V2N	100%	0%	0%

Table 5.17. Hybrid V2N with Autonomous Braking: Collision avoidance numerical results

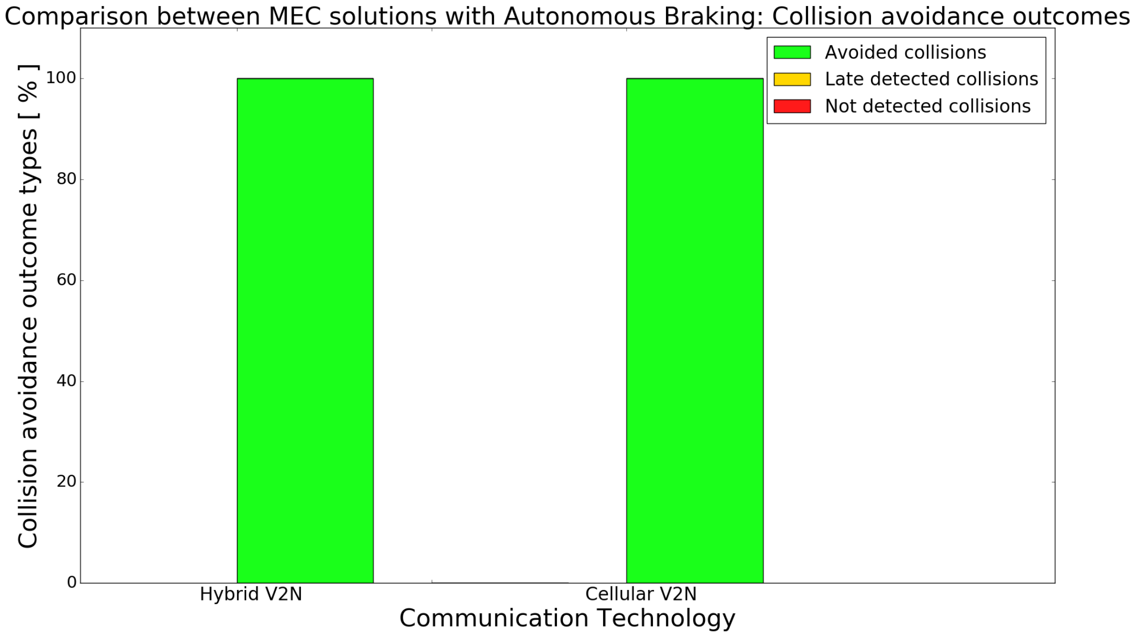

Figure 5.15. Hybrid V2N with Autonomous Braking: Collision avoidance outcomes

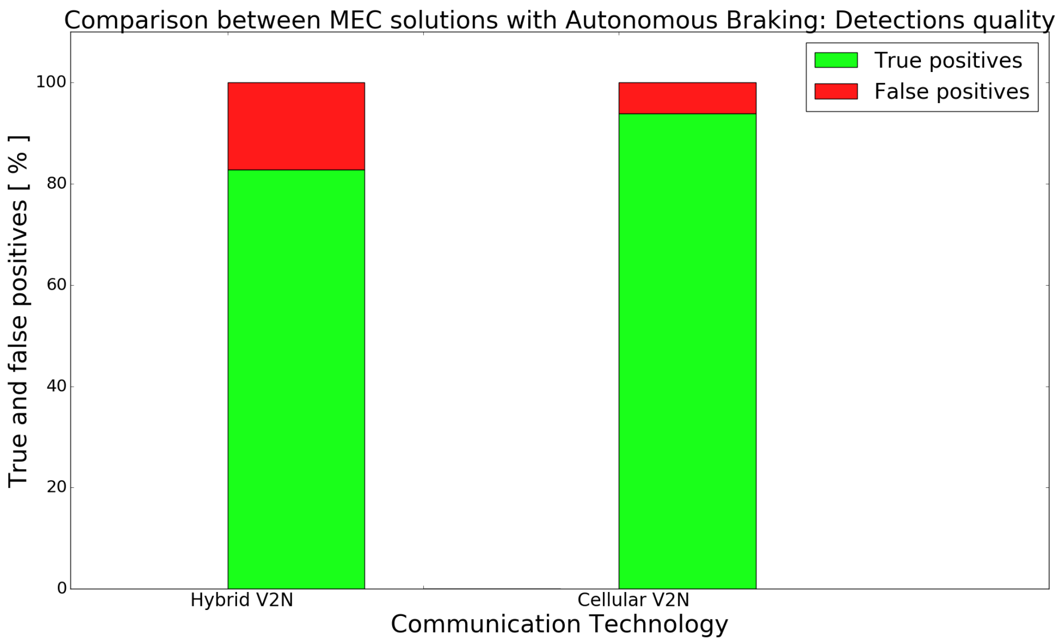

Figure 5.16. Hybrid V2N with Autonomous Braking: Detections quality

	True positives	**False positives**
Hybrid V2N	82.8%	17.2%
Cellular V2N	93.9%	6.1%

Table 5.18. Hybrid V2N with Autonomous Braking: Detections quality numerical results

The obtained results show that by using the hybrid solution the 100% of collisions are successfully avoided, thus confirming the significance of the integration performed in this scenario. Moreover, the performances in terms of avoided collisions are obtained by keeping a low value of false positives percentage (17.2% only). The difference in terms of false positives with respect to the C-V2N scenario might be justified by the different technology used at the access layer by about the 50% of vehicles launched in the simulations.

5.5.2 Hybrid V2N with Human Driver

In this scenario, only human drivers are considered, and the parameters are all kept the same with respect to the previous scenario with the autonomous braking, except the time threshold of the collision detection algorithm, which has been increased of one second in order to cope with the human reaction time.

In the following figures and tables, the performances of the application in terms of collision avoidance and detection quality are reported and compared with the C-V2N scenario with the server placed at the Metro Node:

	Avoided collisions	**Late detected collisions**	**Not detected collisions**
Hybrid V2N	100%	0%	0%
Cellular V2N	100%	0%	0%

Table 5.19. Hybrid V2N with Human Driver: Collision avoidance numerical results

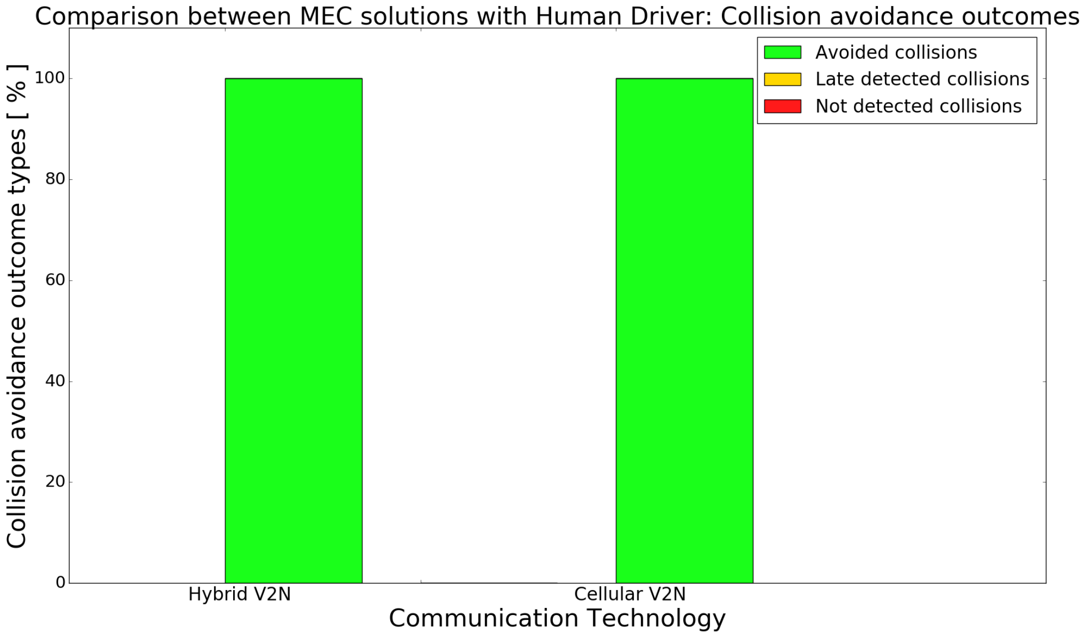

Figure 5.17. Hybrid V2N with Human Driver: Collision avoidance outcomes

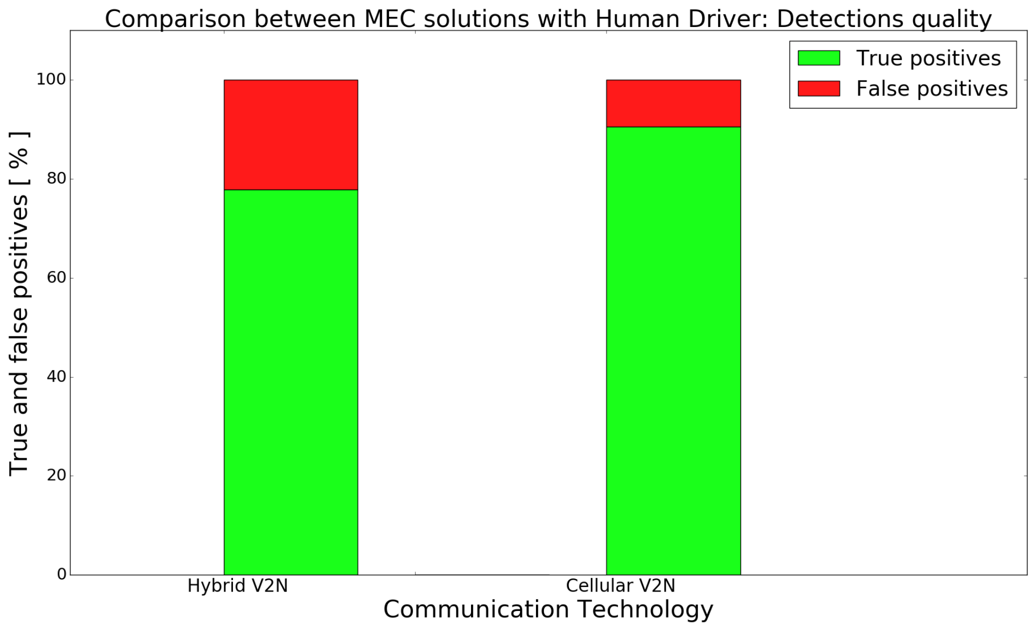

Figure 5.18. Hybrid V2N with Human Driver: Detections quality

	True positives	False positives
Hybrid V2N	77.8%	22.2%
Cellular V2N	90.6%	9.4%

Table 5.20. Hybrid V2N with Human Driver: Detections quality numerical results

The obtained results show that by using the hybrid solution the 100% of collisions are successfully avoided also in this scenario. Moreover, since a higher time threshold for the collision avoidance algorithm has been used, a higher percentage of false positives is expected with respect to the case with autonomous braking. Indeed, in this case the percentage of false positives is about 22%, which is still a reasonably low value.

5.6 Application messages analysis

This paragraph reports statistics about the applicative messages used to support the proposed collision avoidance application. More specifically, for each of the four considered scenarios, the percentages of successfully received and not received beacons and alerts are analysed.

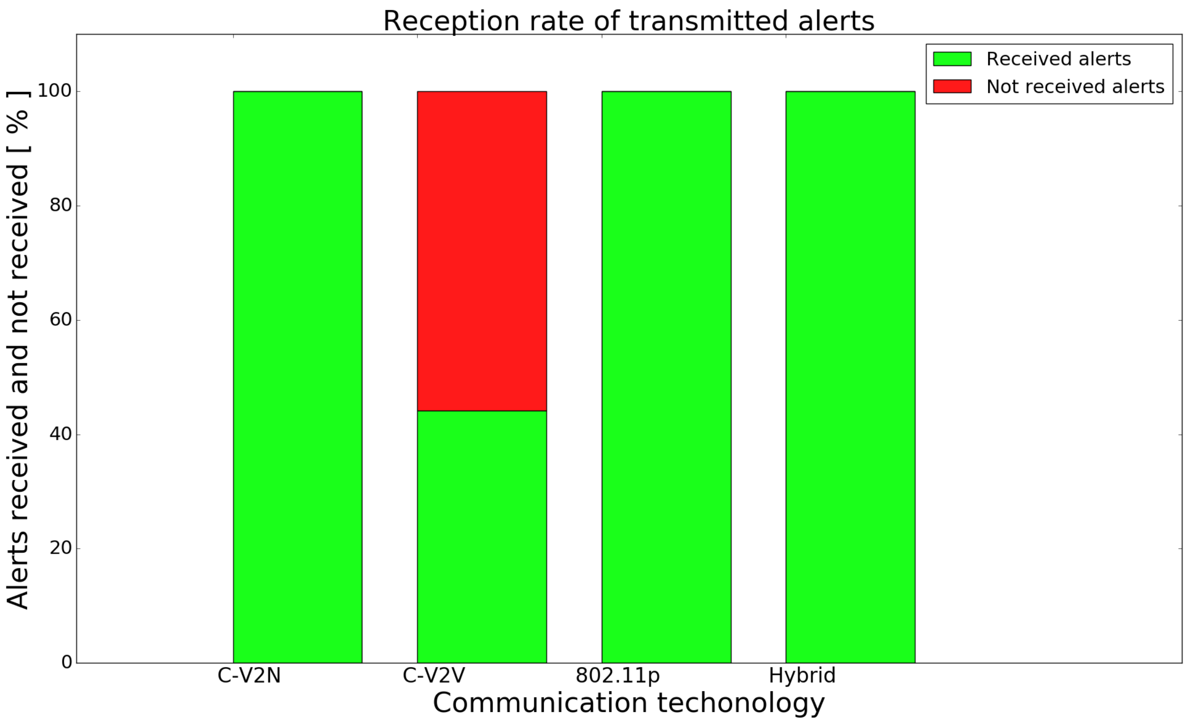

Figure 5.19. Reception rate of transmitted alerts in the different scenarios

	Received alerts	**Not received alerts**
Cellular V2N	100%	0%
Cellular V2V	44.2%	55.8%
802.11p V2V	100%	0%
Hybrid V2N	100%	0%

Table 5.21. Numerical results for the reception rate of transmitted alerts

From the results above, it is possible to state that the worse reception rate of transmitted alerts is in the C-V2V scenario, as the NLOS conditions considered in the channel model reduce the probability of correctly receiving packets. It is interesting to note that in the 802.11p scenario, the reception rate of transmitted alerts is 100%, as these packets are usually sent by vehicles that are very close each other, thus when the signal attenuation imposed by the deterministic channel model is not enough to limit the reception probability. Instead, the two centralized scenarios have an intrinsic reliability which allows to successfully receive all the alerts coming from the network server. Furthermore, results show that not receiving an alert in the C-V2V scenario is not so critical, because whenever a vehicle triggers a detection, at the same time it sends an alert to the other vehicle but also schedules an alert for itself, which is never lost and in most of the dangerous situations allows to actually avoid the incoming collision. This finds its confirmation on the fact that the percentage of late detected collisions is very small in the C-V2V scenario when compared to the percentage of avoided collisions, since whenever a potential collision is detected, in most of the cases the same collision is also avoided in time.

In the following table and figure, the statistics about the beacons are presented:

	Received beacons	**Not received beacons**
Cellular V2N	100%	0%
Cellular V2V	26.8%	73.2
802.11p V2V	27.1%	72.9%
Hybrid V2N	97.2%	2.8%

Table 5.22. Numerical results for the reception rate of transmitted beacons

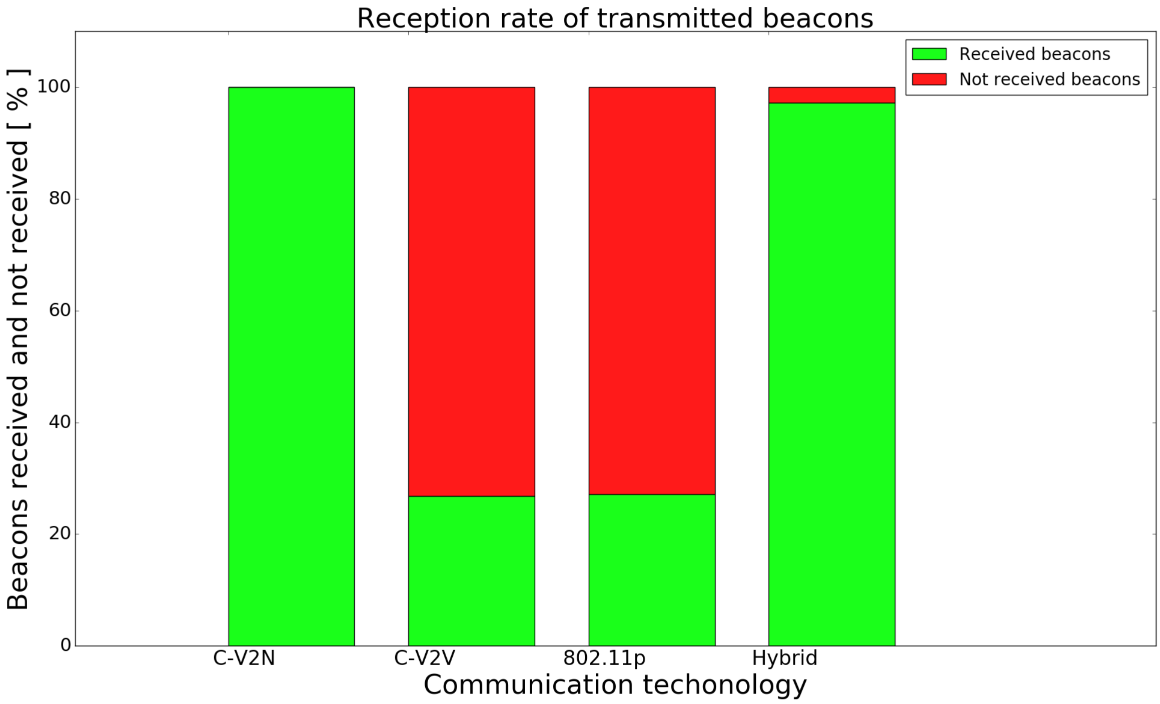

Figure 5.20. Reception rate of transmitted beacons in the different scenarios

Observing the obtained results, the V2N scenarios show the best performances in terms of beacons reception rate, although in the Hybrid V2N scenario some beacons are not received because vehicles equipped with 802.11p NIC just inserted in the simulation are not yet in communication range with the RSU, which is located at the centre of the road topology described in the paragraph 3.3.1. This does not impact on the performances of the Hybrid V2N scenario, since the RSU does not receive those beacons when the vehicles are still far from the central intersection. For what concerns the distributed scenarios, it is possible to state that a great number of beacons is not received for sure when vehicles are very far from each other. Anyway, due to NLOS conditions, some beacons might not be received also when vehicles are relatively close each other, causing the degradation of the performances in terms of a higher percentage of false negatives.

Chapter 6

Conclusions

The last chapter of this thesis work concerns the conclusive remarks about the obtained results and provides future work proposals.

6.1 Remarks on simulation results

The simulation results have shown that to obtain good performances in the analysed collision avoidance application in terms of percentage of successfully avoided collisions and detection quality, it is required that the entities running the detection algorithm receive in time as much as possible beacons in order to have up-to-date information about the current traffic conditions. This is achieved in the cellular V2N scenario when low values of the eNodeB-server latency are considered: indeed, on one hand such latency allows the server to receive in time the beacons transmitted by the vehicles; on the other hand, the intrinsic reliability of the centralized cellular approach ensures the reception of a high percentage of the transmitted beacons, which in the presented simulations is 100% since low traffic levels and vehicles very close to the base station are considered. In the distributed scenarios, instead, worse performances are obtained for this specific use case, since some beacons are not received because of NLOS conditions between vehicles and the time required to process the received beacons at vehicles' OBUs makes out-of-date such beacons.

6.2 Future work proposals

Future work proposals are summarized in the following list:

- Enhancing the collision detection algorithm to better deal with real-world vehicle dynamics;
- Optimizing the thresholds of the collision detection algorithm considering different values of the maximum speed and deceleration of vehicles;

- Improving the comparison between C-V2V and 802.11p-based V2V by using similar channel models for the two technologies;
- Increasing the number of vehicles in the scenario and/or the distance among vehicles and base station;
- Introducing pedestrians in the scenario.

Bibliography

[1] http://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, last check in September 2018.

[2] S. Sun, J. Hu, Y. Peng, X. Pan, L. Zhao and J. Fang, *Support for vehicle-to-everything services based on LTE*, in IEEE Wireless Communications, Vol. 23, pp. 4-8, 2016.

[3] A. Festag, *Standards for vehicular communication - from IEEE 802.11p to 5G*, in Elektrotechnik & Informationstechnik, Springer, Vol. 132, pp. 409-416, 2015.

[4] ETSI, *Intelligent Transport Systems (ITS); Communications Architecture*, EN 302 665, 2010.

[5] ETSI, *Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHZ frequency band*, EN 302 663, 2012.

[6] IEEE, *IEEE standard for information technology telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements, part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, 2010.

[7] A.M. Vegni, C. Campolo, A. Molinaro and T. Little, *Modeling of Intermittent Connectivity in Opportunistic Networks: The Case of Vehicular Ad hoc Netowrks*, in Routing in Opportunistic Networks, Chap. 7, Springer, ISBN 978-1-4614-3514-4, 2013.

[8] C. Campolo and A. Molinaro, *On Vehicle-to-Roadside Communications in 802.11p/WAVE VANETs*, in IEEE WCNC 2011, pp. 1010-1015, 2011.

[9] V. Kumar, S. Mishra and N. Chand, *Applications of VANETs: Present Future*, in Communications and Network, Vol. 5, pp. 12-15, 2013.

[10] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi and H. Hartenstein, *Design methodology and evaluation of rate adaptation based congestion control for Vehicle Safety Communications*, in Proceedings of IEEE Vehicular Networking Conference (VCN), Amsterdam, Netherlands, pp. 116-123, 2011.

[11] IEEE, *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)– Multi-Channel Operation*, 2016.

[12] IEEE, *IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Services*, 2016.

[13] SAE, *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, J2735, 2016.

[14] ETSI, *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking*, EN 302 636, Parts 1-6, 2014.

[15] ETSI, *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of*

Applications; Part 2: Specification of Cooperative Awareness Basic Service, EN 302 637-2, 2014.

[16] ETSI, *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*, EN 302 637-3, 2014.

[17] ETSI, *Intelligent Transport Systems (ITS); V2X Applications; Part 2: Intersection Collision Risk Warning (ICRW) application requirements specifications*, TS 101 539-2, 2018.

[18] G. Araniti, C. Campolo, M. Condoluci, A. Iera and A. Antonella, *LTE for Vehicular Networking: A survey*, in IEEE Communications Magazine, Vol. 51, pp. 148-157, 2013.

[19] A. Filippi, K. Moerman, G. Daalderop, P.D. Alexander, F. Schober and W. Pfliegl, *Ready to roll: Why 802.11p beats LTE and 5G for V2x*, Paper by NXP Semiconductors, Cohda Wireless and Siemens, 2016.

[20] 3GPP, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on LTE-based V2X Services; (Release 14)*, TR 36.885, 2016.

[21] 3GPP, *Initial Cellular V2X standard completed*, http://www.3gpp.org/news-events/3gpp-news/1798-v2x_r14 , last check in September 2018.

[22] A. Filippi, K. Moerman, V. Martinez, A. Turley, O. Haran and R. Toledano, *IEEE 802.11p Ahead of LTE-V2V for Safety Applications*, NXP, Autotalks, 2017.

[23] M. Malinverno, G. Avino, F. Malandrino, C. Casetti, C.F. Chiasserini and S. Scarpina, *Performance Analysis of C-V2I-based Automotive Collision Avoidance* in IEEE WoW-MoM 2018 , 2018.

[24] Wikipedia, *SUMO - Vehicle Type Parameter Defaults*, http://sumo.dlr.de/wiki/Vehicle_Type_Parameter_Defaults, last check in October 2018.

[25] Cop Radar, *Braking Factors - Driver Reaction Times*, https://copradar.com/redlight/factors/, last check in September 2018.

[26] NHTSA, *Identify Intelligent Vehicle Safety Applications Enabled by DSRC*, Vehicle Safety Communications Project Task 3 Final Report, 2005.

[27] ETSI, *Mobile-Edge Computing Introductory Technical White Paper*, https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing, last check in September 2018.

[28] M. Patel, Y. Hu, P. Hédé, J. Joubert, C. Thornton, B. Naughton, J.R. Ramos, C. Chan, V. Young, S.J. Tan, D. Lynch, N. Sprecher, T. Musiol, C. Manzanares, U. Rauschenbach, S. Abeta, L. Chen, K. Shimizu, A. Neal, P. Cosimini, A. Pollard and G. Klas, *Mobile Edge Computing - Introductory Technical White Paper*, in ETSI Portal, 2014.

[29] SNIP, *SAE J2735 DSRC Message List*, https://www.use-snip.com/kb/knowledge-base/sae-j2735-dsrc-message-list/, last check in September 2018.

[30] M. Sepulcre, J. Mittag, P. Santi, H. Hartenstein and J. Gozalvez, *Congestion and Awareness Control in Cooperative Vehicular Systems*, in IEEE Proceedings, Vol. 99, No. 7, pp. 1260–1279, 2011.

[31] K. Ansari, C. Wang and Y. Feng, *Exploring Dependencies of 5.9 GHz DSRC Throughput and Reliability on Safety Applications*, in Proceedings of the 10th IEEE VTS Asia

Pacific Wireless Communications Symposium, pp. 448-453, 2013.

[32] OMNeT++, *OMNeT++ Discrete Event Simulator*, https://www.omnetpp.org/, last check in September 2018.

[33] SUMO, *SUMO - Simulation of Urban MObility*, http://sumo.dlr.de/index.html, last check in September 2018.

[34] SimuLTE, *SimuLTE - LTE User Plane Simulation Model for INET and OMNeT++*, http://simulte.com/, last check in September 2018.

[35] Veins, *Veins - Vehicles in Network Simulation*, http://veins.car2x.org/, last check in September 2018.

[36] C. Sommer, R. German and F. Dressler, *Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis*, in IEEE Transactions on Mobile Computing, Vol. 10, no. 1, pp. 3-15, 2011.

[37] Wikipedia, *TraCI*, http://www.sumo.dlr.de/wiki/TraCI, last check in September 2018.

[38] Wikipedia, *TraCI/Change Vehicle State*, http://www.sumo.dlr.de/wiki/TraCI/Change _Vehicle_State, last check in September 2018.

[39] Wikipedia, *TraCI/Vehicle Value Retrieval*, http://sumo.dlr.de/wiki/TraCI/Vehicle_ Value_Retrieval, last check in October 2018.

[40] A. Virdis, G. Stea and G. Nardini, *Simulating LTE/LTE-Advanced Networks with SimuLTE*, in Simulation and Modeling Methodologies, Technologies and Applications, pp. 83-105, Springer, ISBN 978-3-319-26469-1, 2016.

[41] 3GPP, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects (Release 9)*, TR 36.814, 2010.

[42] A. Virdis, G. Nardini and G. Stea, *Modeling unicast device-to-device communications with SimuLTE*, in 1st International Workshop on Link-and System Level Simulation (IWSLS), Vienna, Austria, 2016.

[43] 3GPP, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on LTE Device to Device Proximity Services; Radio Aspects (Release 12)*, TR 36.843, 2014.

[44] P. Kyösti, J. Meinilä, L. Hentila, X. Zhao, T. Jämsä, C. Schneider, M. Narandzi'c, M. Milojevi'c, A. Hong, J. Ylitalo, V. Holappa, M. Alatossava, R.J.C Bultitude, Y. Jong and T. Rautiainen, *IST-4-027756 WINNER II D1.1.2 v1.2 WINNER II channel models*, 2007. Available online at: https://www.cept.org/files/8339/winner2%20-%20final%20report.pdf , last check in September 2018.

[45] P. Heino, J. Meinilä, P. Kyösti, L. Hentila, T. Jämsä, E. Suikkanen, E. Kunnari and M. Narandzic, *CP5-026 WINNER+ D5.3 v1.0 WINNER+ Final Channel Models*, 2010. Available online at: https://www.researchgate.net/publication/261467821_CP5-026_WINNER_D53_v10_WINNER_Final_Channel_Models, last check in September 2018.

[46] C. Sommer, D. Eckhoff, R. German and F. Dressler, *A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments*, in Eighth International Conference on Wireless On-Demand Network Systems and Services, 2011.