POLITECNICO DI TORINO

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATIONS

Master's Degree in Communications and Computer Networks Engineering

Master Thesis

C-V2X Applications for a 5G network testbed exploiting OpenAirInterface framework and Multi-Access Edge Computing

Academic supervisor:
Prof. Carla Fabiana Chiasserini

Company supervisor:
Ing. Giuliana Zennaro

Candidate:
Mirko Regaldi

OCTOBER 2018

Acknowledgements

I would like to thank my supervisors, professor Carla Chiasserini and Ing. Giuliana Zennaro, for teaching me, during these last six months, autonomy and precision, and for the huge opportunity to take part to 5G-T, one of the most formative experiences in my life.
I would like to express my gratitude to Ing. Paolo Bande, which has supported me during the technical part of the project and thought me that in Google you can find most of the solutions you were looking for.
A special thank to Giuseppe Avino, for its willingness and the time spent together at Politecnico, fixing problems and learning together from this experience.
A very thank to my friends, that give me the best support tools during these years: fun and laughs.
I thank my grandparents, my greatest fans, for sustaining me.
I would like to sends my most sincere gratitude to my mum, which has though me study and passion, to my dad, and my sister, that in difficult times bend over backwards to keep me peaceful and focused.
An important thank you to You, Francesca Rinero, for believing in me, even when I didn't, and for supporting me, always.

Abstract

5G Next Generation networks are envisioned as means to offer a wide range of services, with many different requirements, to final users. New cellular communication systems are configuring not just as a simple enhancement of the broadband mobile network service available nowadays, but as the key for future integration and continuity of technologies and industrial sectors traditionally disjointed. This revolution opens the door, for new players like the Vertical industries, to telecommunication market and apply new business models, as opportunity for new incomes. A network of services, managed by several Mobile Network Operators (MNO) and Verticals, is going to be built upon a common physical infrastructure, thanks to Virtualization of network resources and Softwarization of network functions (SDN/NFV). Two 5G pillars will be exploited to meet these tight requirements: network slicing and Multi-access Edge Computing (MEC). Network slicing helps 5G in the difficult task to reduce costs, managing all the resources efficiently. MEC enables very low latency services, nowadays impossible, but still mandatory for many future applications.
Automotive Vertical, one of the most invested sectors by this revolution, with connected cars and the ever-increasing level of automation is ready to enter definitely the mobile network environment. Several associations are born to support them in the deployment and management processes, in order to better meet their costumers' needs: car safety and infotainment.
5G-Transformer (5G-T) project is an initiative launched in 2017 by one of these associations, called 5G Private Public Partnership (5GPPP). 5G-T is aimed at supporting needs of main Vertical industries in the evolution of mobile communication networks towards a more flexible and scaling SDN and NFV based transport and communication infrastructure.
This Thesis project, developed in collaboration between Politecnico di Torino, Centro Ricerche FIAT and other European players, is part of 5G-T. The objective is demonstrating C-V2X communication, namely V2X communication based on proto-5G cellular network, between two applications, developed in C++ programming language. The exploited mobile network architecture is based on OpenAirInterface (OAI), an open-access framework that provides 3GPP standard-compliant implementation of LTE Release 10, on a standard Linux-based computing equipment. The former application, acting as a vehicle is named OAICodeCar (`oaicc`). `oaicc` is capable of interfacing either with a vehicle CAN bus and with GPS receiver or to a SUMO simulator file, extracting mobility-related values, used to construct ETSI CAM messages. The latter, called OAICodeCAlert (`oaica`), behaving as Road Side Unit collecting CAM messages, is an ETSI DENM generator. The two applications have been designed to communicate and cooperate. They communicate

transmitting and receiving messages. They cooperate saving log files of transmitted and received information and computing statistical analysis of the traffic flow over the network where they have been installed. The results of this Thesis is the core part of the Automotive demonstrator and shall represent a valuable starting point for the next phases of the project.

Contents

List of Figures

Chapter 1

Introduction

Fifth generation of mobile networks, commercially known as 5G, is very close to its commercialization, foreseen in 2020. Discussion about it has not been just limited to telecommunication companies and standardization communities, but has trasversally penetrated either the scientific and the industry world. That is because this new technology is configuring as more disruptive with respect to its predecessors, much more than an enhancement of already existing broadband mobile cellular networks. Strong of the maturity reached by Virtualization and Softwarization techniques (SDN/NFV), this network of services is going to incorporate several traditional sectors, impacting over many different aspects of our everyday life. This revolution opens the door, for new players like the Vertical industries, to telecommunication market and applies new business models, as opportunity for new incomes. Automotive industry will be strongly invested by the revolution in mobility with important advantages. To give an idea, by 2030, the automotive revenue pool is estimated to increase considerably, becoming a 1.5 trillion market, according to a 2016 analysis conducted by McKinsey [1].
This fast growing is justified by the fact that, when yesterday selling a car was limited to the product itself, today the range of services that we can find within a vehicle is increased. This trend is going to explode and translates into higher incomes and possibility of high revenues for car manufacturers and stakeholders.
Thanks to the huge bandwidth enhancement, the possibility to connect new devices to this ever-growing mobile network, merged with the capability to develop sophisticated and smart algorithms running everywhere, facilitated by high computational capabilities offered by Cloud Computing, has paved the way to a completely new idea to intend the car and the driving, that are going to move more and more from a product to a service. These favourable conditions has also made that concepts like *autonomous driving* and *connected cars* are ready to becoming real applications. Powered by a new network, cars can communicate each other and with network infrastructure, to serve connected mobility or to make the journey more entertaining. Car makers have started to pioneer these new ways to conceive vehicles and

mobility, that goes under the name of Vehicle to Everything (V2X) communication and several associations are born to support them in the deployment and management processes, in order to better meet their costumers needs: car safety and infotainment.
Cars are evolving rapidly. Originally they were means to move fast and freely from a point to another, to reach our workplace, in a world where speed and timing is strictly relevant, or have a journey with family or friends. Now they are ready to enter definitely the mobile network environment, equipped with on board Internet access and behaving like smartphones or IoT device in a cellular network. In the early years, electronics before and software after, have become core of car value. They can reach very high speeds and the risks connected to driving multiplies. The number of external stimuli we are subjected to is considerable and our attention during driving is reduced; new multimedia on board possibilities, new connected devices as wearable technology we are surrounded by and a life which requires our presence everywhen and everywhere make us more distracted.
According to PwC [2], by 2030, 40% of the mileage driven in Europe could be covered by autonomous vehicles, with China leader in the market (50%), followed by Europe (40%) and USA (36%). As we can state, Europe plays a relevant role in this revolution; European Telecommunications Standardization Institute (ETSI), non-profit organization with head quarter in Sophia Antipolis (France), is a aimed at defining the common guidelines that car manufacturers, from all over the world, must follow to guarantee technology interoperability and car safety. In this effort ETSI is not lonely. In fact, from one side the 3rd Generation Partnership Project (3GPP) is working on Releases 15 and 16 for 5G, to support also vehicular communication, on the other side car manufacturers join their forces associating within communities, as the 5G Automotive Association (5GAA) or the 5G infrastructure Public Private Partnership (5GPPP), the organization which launched the project this Thesis work is inserted in. Communities like the above mentioned 5GAA and 5GPPP are devoted at investigating the most challenging use cases, namely the most relevant situations, that 5G deployment, thus Automotive, needs to face with in the next years.
Scientific community in Europe has made huge steps, and before 5G, other technologies have been indicated by many to serve V2X, as ITS-G5, based on IEEE 802.11p, from the family of the commercial standard Wi-Fi.
Points of failure of already existing Commercial Off-The-Shelf (COTS) vehicular communication systems, and the reason why they could be very well supported by mobile network architectures, are many, since some of them, like radar and cameras, are line-of-sight technologies, which it means they are weak against situations in which an obstacles cover the street (intersections, fog) and they provide additional processing delays in their detection procedures. Some other technologies communicate using very crowded spectrum frequencies, making communication difficult and even dangerous within an environment in which a message could make difference

between life and death.
5G promises, by its side, low latency, ultra-high availability, resilience, high reliability and high data rate performance, KPIs that assign it as the leader technology to fulfil strict needs required by automotive industry, that, together with cooperation introduced by ETSI in Intelligent Transportation Systems, is capable to sustain an environment, as the street, in which car accidents are mainly matter of human errors.
The Thesis project, developed in collaboration between Politecnico di Torino, Centro Rcerche FIAT in Orbassano (TO) and other European players, is aimed at demonstrating communication between two C-V2X applications, exchanging basic safety messages, standardized by the ETSI, called Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Messages (DENM) over a proto-5G network, implemented using Open Air Interface (OAI) framework and equipped with a Multi-Access Edge Computing Platform. The work is the core step of a project called 5G Transformer (5G-T), launched by 5GPPP in June 2017, which involves several companies from academic and industry world, supporting Vertical industries services implementation in next generation networks.
The structure of the Thesis is deployed as follows: Chapter II will provide an overview of the two main groups of communication technology supporting V2X: ETSI ITS-G5, relaying on IEEE 802.11p WAVE and the evolution 802.11px, and cellular based C-V2X, nowadays served by LTE, but destined to evolve with 5G. After that, Chapter III will focus on the newest concepts and architectural elements of 5G , describing the New Radio, the Non-Stand-Alone and Stand-Alone architecture and then dwelling more on two enabling 5G technology, Network Slicing and ETSI Multi-access Edge Computing, whose theoretical architecture will be reviewed in details. In Chapter IV is described the subject and most concrete part of the Thesis. Two applications, named *OAICodeCar* (`oaicc`) and *OAICodeCAlert* (`oaica`) have been developed. `oaicc`, acting as a vehicle, is capable of interfacing either with a vehicle CAN bus and with GPS receiver or to a SUMO simulator file, extracting mobility-related values, used to construct ETSI CAM messages. `oaica`, behaving as Road Side Unit (RSU) and collecting CAM messages, is an ETSI DENM generator. The two applications have been designed to communicate and cooperate. Chapter V will focus on the results obtained by this work, completing the development part, to measure applications and MEC performance, also in relation with future steps of 5G-T. Conclusions are given in Chapter VI.

Chapter 2

Communication technologies: road towards C-V2X

In Vehicle to Everything (V2X) communication, the exchange of information shall take place between two edges: a vehicle and entity, which could be another vehicle (V2V), a road infrastructure (V2I), a pedestrian (V2P) or an application server (V2N).
Originally, it was conceived with the objective to create a safer road environment.

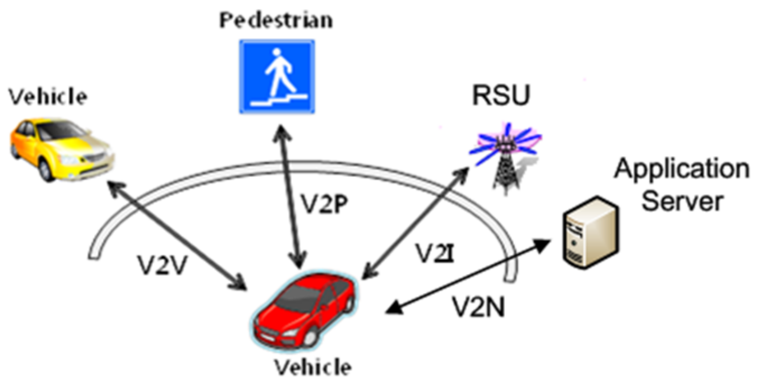


Figure 2.1: V2X communication. Taken from [24]

Car accident situation in Europe is really worrying. Opening the *Accident Report 2017* of the European Commission [3] reveals that, in 2015, the number of accidents registered in Europe were about 1 million, with Italy at second place just after Germany. Even if these data are decreasing by 260 000 with respect to 2006, symptom of a deeper sensibility from governs and car manufacturers concerning

the danger, it still remains one of the main cause of death in Europe, especially between young people. Car accidents are due mainly to recognition errors, decision errors and performance errors, all conductible to human mistakes [4], that can be eventually be avoided with the intermediation of an automated intelligence in the vehicle or in the infrastructure.
In second analysis, a comfortable driving experience can be obtained by means of V2X. The possibility for the driver to consult Local Dynamic Maps, containing layered traffic information, or to use Automated Parking System applications, in order to ease the process of finding a parking slot during a congested event, joint with the convenience for the passengers to enjoy a movie streamed from the Internet, to consult the weather forecast and watch a video on Youtube, are all examples of how car experience can become more complete, entertaining and satisfying.
An Intelligent Transportation System (ITS), in a directive of the European Commission in 2014, available here [5], is a system in which Information and Communication Technologies are applied to support road environment in terms of optimal use of road, traffic and travel data, road safety and security, connection between vehicles and infrastructure and management of the data traffic. Such a system is composed by three functional entities: vehicles and applications, road infrastructure and VANET. The paradigm of Vehicular Ad Hoc NETworks (VANET), indeed, has been introduced in 2001 and consists in wireless ad hoc networks, namely networks with partial or absent infrastructure support, specifically thought for V2X. Such systems are characterized by:

- Mobility: nodes of the network are in movement. Mobility feature can be divided into macro mobility, namely models determined by driving rules and road layout, and micro mobility, namely models determined by interconnection between vehicles.

- Network topology: the network topology has a very fast dynamics, high speed and low connectivity, because cars could exchange information for e brief period before moving away, directed to the next network.

- Resources: resources, in terms of battery life, memory and processing capabilities. This point doesn't represent a real issue for VANET, since plenty of those resources are nowadays available inside a vehicle.

- Localization: localization is a key concept in VANET, since the synchronization, obtained by means of Global Positioning System (GPS), is fundamental feature to guarantee updated information to car passengers and avoid frame collisions in a wireless environment.

VANETs are composed by two elementary devices: the Road Side Unit (RSU) and the On Board Unit (OBU). The RSU is a stationary device, capable to support V2X applications and which can exchange messages together with other entities that support its same applications. The RSU is instead a communication facility device, located inside the vehicle, including processing, connectivity, sensors interfaces, power management and antenna. It can be either stationary or in movement. In principle, any radio access technology could be used to provide connectivity to the vehicle. From 2001 there have been identified two types of VANETs, that present convenient and advantageous features for V2X:

- Wireless LAN based, which exploits Dedicated Short Range Communication (DSRC). In Europe standardization effort is in charge of IEEE 802.11p amendment. Connectivity is realized by means of a Wi-Fi modem.
- Cellular based, which exploits LTE mobile cellular networks and that, in the near future, is going to be supported also by 5G. Globally, standardization effort is managed by 3GPP. Connectivity is realized by the LTE User Equipment (UE) modem.

This chapter has the objective to revise the main concurrent technologies for V2X deployment. We provide an overview of 802.11p WAVE (US) and ITS-G5 (EU), standardized respectively by IEEE 802.11p working group and ETSI; then we inspect C-V2X, introduced in Release 14 of 3GPP, in order to have an idea of the analogies and differences among them and clarify how nowadays V2X communication via 5G is so important.

2.1 DSRC vehicular systems between United States and Europe

In 1999, the USA's Federal Communications Commission (FCC) allocated the spectrum of 5.9 GHz, namely 5.85 GHz to 5.925 GHz, to DSRC, in order to support either *safety applications*, for instance to support the lane change, to cope with road intersection, and *non-safety applications*, like parking assistance applications. At the beginning, the working group in charge to standardize the technology was ASTM 2313, but, in 2004 [6] the standardization task moved to IEEE 802.11 standard group. DSRC turns into *802.11p WAVE* amendment, which stands for Wireless Access for Vehicular Environment.
In Europe, DSRC is known instead as ETSI ITS-G5. In 2008, 30 MHz of spectrum were allocated, [7], from 5.875 to 5.905 GHz for ITS communication (from 5.905 to 5.925 for future ITS applications).

2.1.1 IEEE 802.11

IEEE 802.11p is an amendment of IEEE 802.11, which defines lower layers, namely the Medium Access Control (MAC) and physical (PHY) layers, for DSRC-based

operations, to provide vehicular ad hoc communication. On the top of the above mentioned stack layers, depending on the continent where they have been deployed, various protocols have been added to provide additional services to 802.11p.
It borrows features from 802.11a and 802.11e, enhancing them with specific characteristics, to support tight traffic requirements for this very dynamic environment. In order to understand which innovations has brought 802.11p and what is planned for 802.11px, we will give a survey of IEEE 802.11x components and operations. The following description relies on this book [8], and on this article [9], which contain a very clear and detailed vision of the topics.
IEEE 802.11x is a family of standards for wireless LAN. In order to generalize the main concepts, we will abstract the versions x from the family and we will refer to it as IEEE 801.11. The architecture of IEEE 802.11 is characterized by:

- Basic Service Set (BSS): it consists in a group of at least two stations (STA), the starting and ending point of a communication. It can be either *direct* (ad hoc or independent BSS), without pre-planning of the communication resources by a central coordination point, or by means of a *network infrastructure* (infrastructure BSS or simply BSS). In this second case, the wireless LAN network is established by an Access Point (AP), that centralizes the control of the network.
- Basic Service Area (BSA) or cell, the geographical area in which two STAs can transmits their frames.

Whereas an ad hoc BSS has the advantage for STAs to wake up and communicate whenever they need to, infrastructure BSS allows two LANs to extend the service area, since they can be logically connected, through a Distribution Service (DS) and form an Extended BSS (EBSS).
BSS are known to users through SSID, an up to 32 bytes identifier of the AP. Instead, the Basic Service Set Identification (BSSID) is a number of 48 bytes, coinciding with the MAC address of the AP, and shared by each component in a BSS. The goal of such BSSID is to allow the receiver to distinguish frames belonging to the its BSS. For each IBSS an independent MAC address is used.
In a BSS, frames can be control frames, data frames and management frames. When a user wants to connect to a BSS, it must follow three steps. The first one is scanning phase, in which a Beacon Frame, broadcasted by the AP and containing synchronization information, is caught by the STA (passive mode), or in which the STA itself sends a Probe Frame, looking for an AP (active mode). Authentication is the phase, which can be leaded by the AP, that allows or not the STA to authenticate on the basis of a MAC address list, or a shared key system authentication. The last phase is called association, in which the exchange of information, concerning the wireless interface (e.g. 802.11b, 802.11p etc), mobility support and basic and the minimum rate of service is signalled.
At PHY layer, the set of frequencies used by IEEE 802.11 is 2.x GHz, for the early

versions and 5.x GHz for the newest, like 801.11p and beyond. Channel width varies from 20 MHz to 40 MHz channels and data rates goes from 1.2 to 600 Mbps. Techniques used at physical layer have been Frequency Hop Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), Orthogonal Frequency Division Multiplexing (OFDM) and Multiple Input Multiple Output (MIMO). Apart from OFDM, revised in the next sections, we won't discuss them in the details, because this would drives us out from the main speech. The time granularity in IEEE 802.11 is the timeslot, whose duration relies on the specific PHY layer implementation.
The MAC layer is purposed to define the MAC frame (MPDU), to segment data and perform error checksum on the header. According to MAC architecture, time is divided into *superframes*. Each superframe may be formed at least by a Distributed Coordination Function (DCF), to which a Point Coordination Function (PCF) can be added. DCF is a Contention Period (CP), in which nodes access the channel according to Carrier Sense Multiple Access Collision Avoidance (CDMA/CA) scheme and is common to refer to as *WiFi MAC*. The latter mode, implementable in lonely infrastructure BSS, corresponds to a Contention Free Period, where no CDMA is used and STAs and access to the channel is based on AP instructions. This last function is not mandatory in 802.11 and it is not our interest discussing that.
Waiting time that STAs have to wait before, after and during a dialogue, is defined

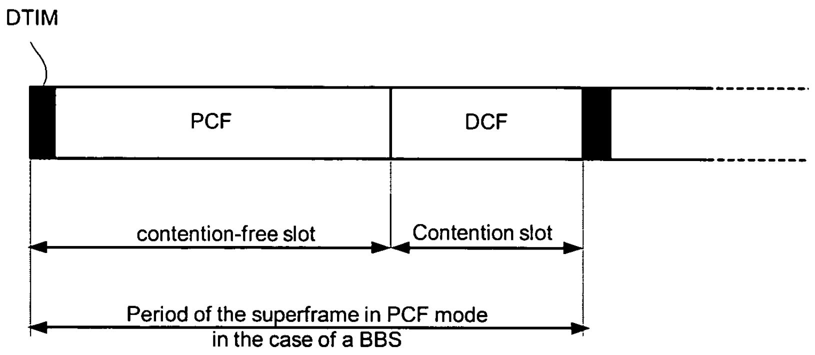


Figure 2.2: 802.11 superframe

as Inter Frame Space (IFS), with duration proportional to the priority of the user frame. We will present 802.11b IFS, from the shortest to the logest one:

1. SIFS (Short IFS): period waited during the same dialogue, in order that it cannot be interrupted (for instance between a frame and its relative ACK);

2. PIFS (Point coordination IFS): equal to (SIFS + 1 timeslot duration). It is waited only by the AP in the PCF period, before carrying very important information, like the Beacon Frame.;

3. DIFS (Distributed IFS): it is equal to (PIFS + 1 timeslot duration). It is generally waited when a node wakes up and waits for a free channel before starting transmission, and is the most common used IFS during operational routines;
4. EIFS (Extended IFS): it can last up to 100 ms and is used when the PHY layer notifies to the MAC an error on a transmission and it cannot understand what is happening.

Recalling that wireless medium is by nature half duplex, therefore each STA cannot sense, transmit and receive at the same time, Carrier Sense mechanism can be performed in two modes:

- Physical carrier sensing: if the power level is higher than a certain threshold, then the channel is sensed busy, otherwise, it is sensed as free and the Clear Channel Assessment (CCA) status is sent to MAC layer from PHY;
- Virtual carrier sensing: the user senses the channel and demodulates the header of frames sent over BSS by other STAs.

If the channel is sensed as busy, either at PHY or at MAC layer, than a procedure based on Exponential Backoff algorithm, is performed, to allow that two STAs detecting channel busy at the same time avoid to start transmitting contemporary. In CSMA/CA, the sender STA, before transmitting, has to wait for a period equal to DIFS, then sends in uplink, to the AP, a Request To Send (RTS) control packet. After a SIFS period, the AP answers with a Clear To Send (CTS) packet. This protocol avoid collision in the well-known hidden terminal scenario problem, in which a STA is communicating with the AP, and there is another terminal, hidden to the STA, that wakes up and wants to transmit. In fact, whereas it cannot sense frames sent in uplink by STA, it can demodulate CTS messages and can sense the upcoming collision with the STA. IEEE 802.11 has been deployed in many versions and we are going to discuss the one dedicated to Wireless Access in Vehicular Environment: 802.11p.

2.1.2 IEEE 802.11p WAVE PHY and lower MAC

In 2010, IEEE published [10], drafted to support Wireless Access in Vehicular Environment (WAVE), thus describing modifications of the pre-existent PHY and MAC layer architecture for ITS VANET environment.
The main idea behind 802.11p is that, because passengers safety is a central problem in vehicular networks, a vehicle needs to communicate with another vehicle without setting up a BSS, namely excluding authentication and association phase described in IEEE 802.11. 802.11p has introduced the so-called WAVE or OCB mode (Outside the Context of a BSS). Through this, an STA can immediately communicate with

other STAs, using the same BSSID and channel, which belong to what is configured as WAVE BSS.
Wi-Fi operations in 80.11p take place in 5.9 GHz band; each subcarrier is then modulated with traditional techniques as 64-QAM or BPSK, and data rate varies from 3 to 27 Mbps consequently to the applied modulation scheme.
PHY layer in 802.11p is very similar to 802.11a. The 75 GHz of the spectrum are divided into seven 10 MHz channels, to cope with increased delay spread in vehicular environments [11]. These channels are logically divided into 6 Service Channels (SCH) and 1 Control Channel (CCH) and OFDM modulation is used. SCH are used to carry data traffic, for instance traffic of car safety applications, while CCH is exploited to carry control traffic.
MAC layer is divided into two parts, lower and upper MAC. The first one defines how to handle a single channel, whereas the upper MAC defines how to handle a group of channels (CCH or SCH). IEEE 802.11p specifies just lower MAC. At lower MAC layer, channel access mode is called Enhanced Distributed Channel Access or EDCA, firstly introduced in IEEE 802.11e, which extends basic operations in the Distributed Coordination Function (DCF), discussed before, with advanced QoS functionalities. In EDCA, STAs experience differentiated channel access to the wireless medium; eight class of services, the User Priorities (UP), are mapped into four MAC queues called Access Categories (AC), each defined by an index (ACI) and individual set of parameter duration. These are, presented with increasing priority:

- Background (AC_BK)
- Best effort (AC_BE)
- Video (AC_VI)
- Voice (AC_VO)

In EDCA the additional IFS called Arbitration IFS (AIFS) is waited, by each STA, before starting a dialogue, and is the higher, the lower it is the priority of the UP.
Superframe is composed by two intervals, one dedicated to CCH and the other one related to SCH, each of them lasting 50 ms. That means, half of the bandwidth is used to carry control information and, after 50 ms one of the SCH is selected.
Critical issues of this technology are:

- Alternating channel access, which gets uncontested benefits in terms of costs but, due to the fact that, when a single CCH is used, seven SCH are left uninitialized, frequency resources are used inefficiently. Inefficient use of frequency resources is flanked by poor safety, since safety application traffic, served by a SCH cannot be transferred during CCH period.

- Frame collision is still a problem because the EDCA is a contemption-based mechanism. RTS/CTS mechanism cannot be exploited because collision happens more frequently in broadcast environment.

ETSI has posed some solutions that partially solve these issues, as we will see in the next subsection.

2.1.3 ETSI ITS-G5: CAM and DENM messages

In [12], Cooperative Intelligent Transport Systems (C-ITS) are defined as an subset of ITS systems in which BSS nodes cooperate, exchanging and sharing information collected, in order to improve safety, convenience and efficiency with respect to stand-alone systems.
ETSI, as we will deepen in next chapters, has started working on C-ITS to standardize, in Europe, Vehicular Ad Hoc Communication based on IEEE 802.11p WAVE. In particular, the access layer technology specified in [13] is known as ITS-G5, since it operates at 5 GHz, and specifies operations and characteristics of PHY and MAC layer.
In the process of describing differences and similarities between ITS-G5 and IEEE WAVE, has been particoularly useful Eckhoff et al. study [14]. In Europe the DSRC spectrum has been divided into three sub-bands, each one, devoted to a specific application type:

- ITS-G5A: devoted to safety and traffic efficiency applications,
- ITS-G5B: devoted for non-safety applications,
- ITS-G5C: devoted to Broadband Radio Access Network (BRAN) WLAN and RLAN,
- ITS-G5D: devoted to future services.

In ITS-G5, spectrum is divided, as in WAVE, into seven channels, of which six SCH and one CCH. Differently from what we have seen, in the previous section, with IEEE 802.11p WAVE, in ITS-G5 there not exists the alternating access scheme. This, by one side, makes the OBU cost increasing, but allows, by the other side, to deliver safety messages in time and avoid packet losses due to switching between channels.
The PHY layer exploits Orthogonal Frequency Division Multiplexing (OFDM), with the same parameter set seen or 802.11p. OFDM is a channel access technique in which transmission time is divided into slots, called OFDM symbols, since each transmitting device can pick one of the orthogonal partially overlapping carriers (the peak of a carrier coincide with the zero of another one), belonging to a set of frequencies in that time slot, and multiply its modulating symbol with that carrier. The orthogonality property of the OFDM carriers make possible, for a set

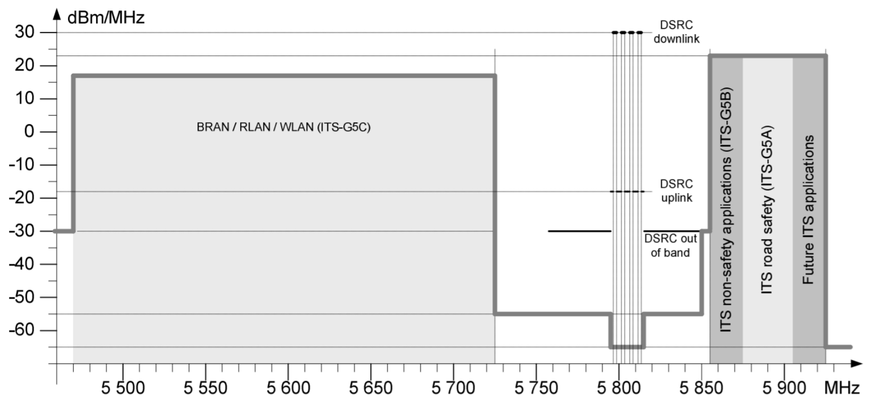


Figure 2.3: ITS-G5 frequency allocation

of transmitters, to access the channel at the same time, using a different carrier, and, for a receiver, to retrieve the transmitted symbols by applying a scalar product with the received signal.

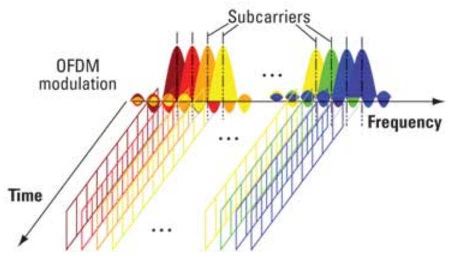


Figure 2.4: OFDM channel access

At lower MAC layer, EDCA is used, with CSMA/CA and QoS service based on Access Categories.
In addition to EDCA, in order to cope with high loads, Decentralized Congestion Control mechanism (DCC) is one of the main access layer techniques for ITS-G5. It is a cross-layer function, namely a function not limited just to one layer of the ITS protocol stack. It is used by each node, which assigns a state to the channel load, deciding, eventually, what to do when the channel is overwhelmed, in order to avoid congestion problems, to save resources for transmission of higher priority traffic, to be more adaptive with respect to the fast varying channel, and providing equal access to channel resources. It is enabled mandatory just in ITS-G5A and ITS-G5B. DCC consists mainly of four components:

- DCC access

- DCC net
- DCC app
- SCC mgmt

Each of them is located in the correspondent layer. We will inspect in the next lines the DCC access, since we are interested more in the access layers. It consists in transmit power control (TPC), DCC sensitivity control (DSC), transmit rate control (TRC), transmit datarate control (TDC) and transmit access control (TAC). Each of these mechanisms is based on a threshold system, based on which, if a certain MSDU is received or must be transmitted with a certain parameter, the STA has firstly to check if it meets the threshold value.
A very important feature of DCC is the DCC access control loop, that allows an STA to assign a state to the channel, that can be classified as RELAXED if it is mainly free, RESTRICTIVE if it is overloaded or ACTIVE, if it is working normally.
The IEEE 802.11p physical layer has great benefits in resistance to the Doppler spread effect and it has very short frames, which makes it very suitable for fast vehicular communication. But it is sensitive to collision,s when the load of the channel start increasing, and in these cases, it loses the high reliability KPI which is fundamental for V2X communication.

Messages

The two access layer technologies presented so far, IEEE WAVE and ETSI ITS-G5, are the basement for the full-stack architecture showed in figure 2.14. Above them, these two standards define network and transport layer, that vary consistently each other, and they will be not revised, since, as we will see later on, we have decided to exploit simple UDP transport to carry C-V2X data the applications developed for this Thesis.
On the top of the stack pile, a Facility/Application layer is defined, and foresees broadcasting of periodically transmitted messages. In WAVE, these messages are called Basic Safety Messages, for ETSI these are called Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Messages (DENM). In this Thesis project, ETSI standard has more relevance with respect to WAVE, that we have explained in order to introduce ITS-G5; thus, CAM and DENM are described in the following sections.

ETSI CAM

CAM [16] messages are periodically exchanged by ITS-Station (ITS-S), namely each node belonging to an ITS system, and they contain information about the vehicle current status like positioning, velocity, acceleration and external light status.

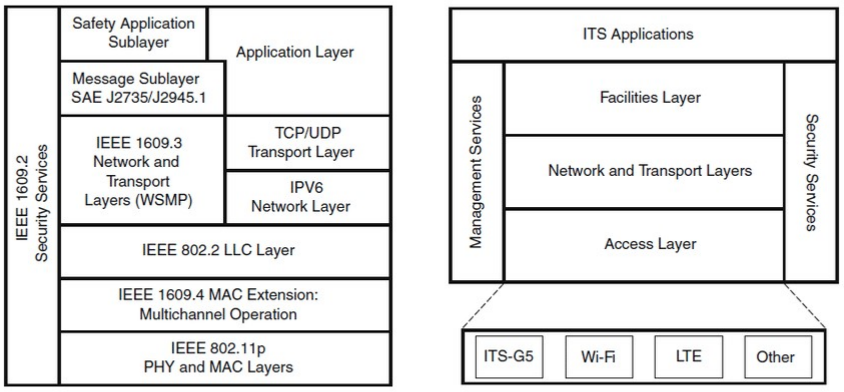

IEEE 1609.2 Security Services
Safety Application Sublayer
Application Layer
Message Sublayer SAE J2735/J2945.1
IEEE 1609.3 Network and Transport Layers (WSMP)
TCP/UDP Transport Layer
IPV6 Network Layer
IEEE 802.2 LLC Layer
IEEE 1609.4 MAC Extension: Multichannel Operation
IEEE 802.11p PHY and MAC Layers
ITS Applications
Management Services
Facilities Layer
Network and Transport Layers
Access Layer
Security Services
ITS-G5
Wi-Fi
LTE
Other

Figure 2.5: IEEE WAVE (on the left) and ITS-G5 (on the right)

ITS-S, transmitting a CAM message, is said originating ITS-S. These messages are very relevant for vehicular safety applications, since represent a direct interface to the vehicle itself for all the occupants of the road environment. Receiving a CAM can make the difference, for a vehicle, between collision and avoidance of the same collision, realizing, for instance, that another hidden vehicle is crossing an intersection at high speed, ignoring the red color of a traffic light.
CAM format is defined in ETSI EN 302 637-2, together with Cooperative Awareness (CA) basic service, and is represented with a stratified structure.
The CA basic service is a functional block, part of the Facility Layer described in ETSI EN 302 665, consisting of four main sub functions: encode CAM, decode CAM, CAM transmission and CAM reception management. The first two blocks perform CAM serialization and de-serialization, according to Unaligned Packet Encoding Rule (UPER). The transmission CAM management determines the frequency at which CAMs must be generated, turns on and off the generation and transmission of CAMs. The reception CAM management, instead, starts the decoding CAM procedure, provides the ITS applications or the Local Dynamic Map of the received CAMs.
For a vehicle ITS-S, CAM generation period shall not be greater than 1 sec and not inferior to 100 ms, respectively frequency of 1 and 10 Hz. CAM generation is regulated by two main factors:

- Channel condition, controlled by DCC parameter `T-GenCamDcc`, that provides the minimum time interval between two consecutive CAMs, in order to reduce the load on the channel, as required by DCC. In 2018 version of the standard, LTE-V2X support is added, underlying that no DCC support is foreseen, therefore CAM generation must satisfy just the following point.

- Vehicle dynamics, depending on how rapidly some parameters in vehicle status are changing.

CAM has to be immediately generated when:

- heading absolute difference, between one CAM message and another sent by the same ITS-S, overpasses 4 degree;
- the distance between the current position of the originating ITS-S and the distance previously collected in a CAM, exceeds 4 m, and the absolute difference between current and previously sent speed exceeds 0.5 m/s

CAMs shall be generated also by RSU ITS-S. Another important constraint for CAM is that the CAM generation, namely the absolute difference between the instant in which CAM generation is triggered and the time at which CAM is passed to the Network and Transport Layer, shall not be greater than 50 ms.

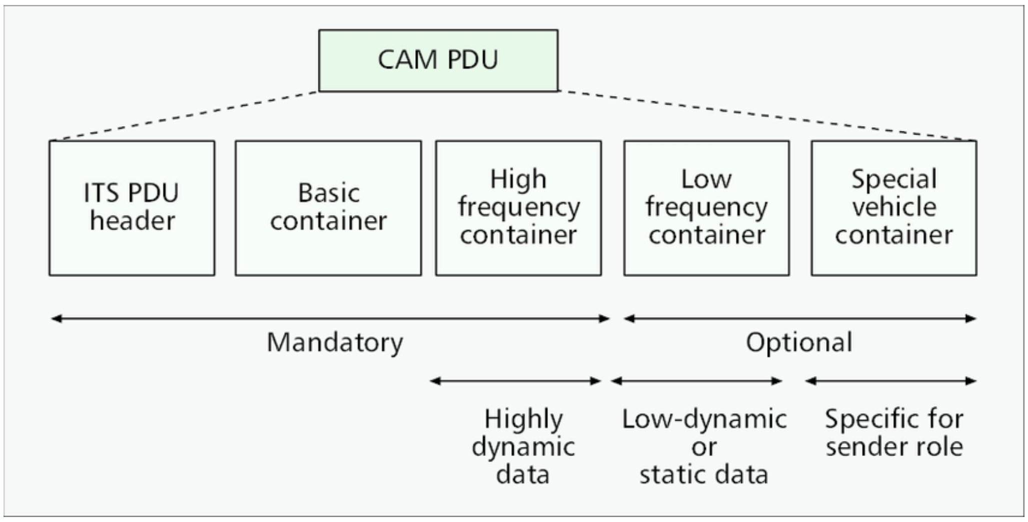


Figure 2.6: CAM general structure

CAM PDU composed by:

- Header: includes information about the protocol version, message identifier, which allow to discriminate between CAM and DENM and ITS-S ID;
- Payload: is composed by several containers, that are data structures encapsulating parameters:
  - The `basicContainer` is mandatory and contains very elementary information like the station type (whether the vehicle is a passenger car or an emergency vehicle, for instance) and the reference position.

  - The `highFrequencyContainer` is mandatory and includes two other containers, the former with the most important vehicle high dynamics parameters like speed, heading and vehicle length, the latter including information about the RSU.
  - The `lowFrequencyContainer`, contains low frequency information like exterior light status or the vehicle role.
  - The `specialVEhicleContainer` is an optional container, dedicated to vehicles different from traditional passenger cars, like ambulance, dangerous good transportation vehicles and others.

ETSI DENM

Whereas CAM message, as we have just discussed, represents an interface to the other vehicles and they are used to advertise vehicle information in the surrounding environment, Decentralized Environmental Notification (DEN) basic service has been introduced to support Road Hazard Warning (RHW) applications. It is located at the Facility Layer, as for CA basic service, and manages event-driven generation of messages called DENM [17], exchanged by ITS applications to inform the others about hazardous traffic conditions or dangerous events.
DEN basic service has the main objective to trigger, generate and deliver DENM messages to Network and Transport Layers, to be transmitted over a channel towards the destination.
To make a real example of their usage, we can imagine a road environment, populated by many vehicle ITS-Ss (e.g. car passengers) and a single stationary ITS-S (e.g, RSU), collecting CAM messages from this group of vehicles. When the RSU, processing a group of CAMs from other vehicles, foresees something is going to happen, for example a vehicle suddenly stops on the lane, it generates a set of DENM to inform the other road users that there is a potential dangerous vehicle frozen on the way. Depending on the application developed on the vehicle ITS-S, the obstacle can be for instance avoided or an alert message can be shown on the display of the on board computer.
In the DEN basic service, events are characterized by an ID, so that a DENM, related to a single event, can be repeated or terminated depending on the type, on the position and on the time duration, and can be sent either by a vehicle and by an RSU. We use to define four types of DENM, depending the criteria just discussed:

- New DENM: it is generated by DEN basic service when and event is detected for the first time;
- Update DENM: it is generated by DEN basic service when, for the same event, some information has changed. For instance we could imagine a pedestrian crossing an intersection during the night, who moves slowly along the cross walks and is detected by sensors located in the traffic lights;

- Cancellation DENM: it is generated by DEN basic service to inform the termination of an event;
- Negation DENM: it is generated to inform about the termination of an event for which it has been received a new DENM. For example, if an ITS-S detects that an event, issued by a vehicle which is no more in the reference zone, is no longer happening, can inform other ITS-Ss with a negation DENM.

Because of the multiple nature of DENM messages, each ITS-S shall include a data structure called ITS-S message table to collect information about received and transmitted DENM.
DEN basic service consists in sub functions which are similar to what we have seen for CAM, with the only addition of an optional DENM Keep Alive Forwarding (KAF) mechanism, which will not be discussed in this seat.
DEN Basic Service is in charge of encoding DENM, decoding DENM, performing DENM transmission management and DENM reception management. In particular, DENM transmission management triggers generation of the four DENM types described just before. DENM reception management updates the ITS-S message table, discards the invalid DENM and DENM data passage to ITS-S applications of the receiving ITS-S.
DENMs must be disseminated within a relevance area, namely the area in which the ITS application, sensing the event, considers it as the action area of the event, for example the area surrounded a crashed car. This concept is connected with the one of destination area, that is the shape of the area the DENM must be issued to. It can be circular, rectangular or elliptical, depending on choice made by DEN basic service of originating ITS-S and passed to network and transport layer.

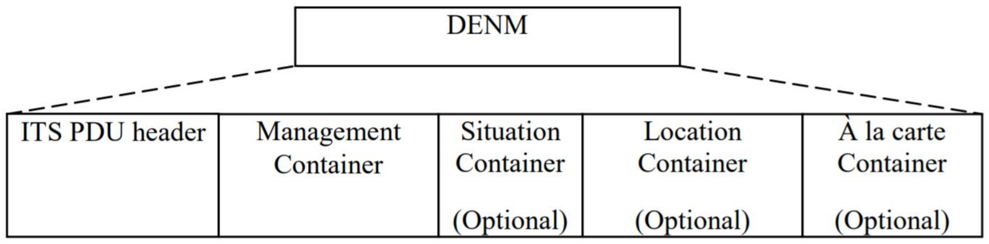

DENM
ITS PDU header
Management Container
Situation Container (Optional)
Location Container (Optional)
À la carte Container (Optional)

Figure 2.7: DENM general structure

DENM structure is similar to CAM since it is divided into:

- PDU Header: contains the same fields as CAM, but the message ID is used to identify the DENM;
- Payload: consists in four containers, the first one is mandatory, while the other three are optional. We present them in this order:

  – `ManagementContainer`: includes information about DENM management, like the action ID, the detection time and reference time of the event, the station type and others;
  – `SituationContainer`: contains information related to the type of the detected event, with specific attention to the cause that led to the mentioned event;
  – `LocationContainer`: contains traces ,the eventual speed and heading of the event and the quality index with which information is provided;
  – `ALaCarteContainer`: includes additional information, which are related to the use case and are not inserted in the previously mentioned containers.

As we will see in Chapter IV, CAM messages plays a central role in the applications we have developed at Centro Ricerche FIAT, therefore we will go deeper in the details of their structure in the next chapters.

2.1.4 Towards the NGV standard: 802.11px

ITS-G5 Release 2 is upcoming. Effort started in November 2016, by Car2Car (C2C) Communication Consortium. The evolution concerns mainly, as reported here [18] in:

- Channel usage in terms of modulation and congestion control,
- Information exchange,
- PHY and MAC layer modifications,
- Capacity increasing, since one of the objective is to exploit mmWave bands at PHY layer.

We can state that, one of the primary importance topics discussed in the document, in sight of Next Generation Vehicular (NGV) networks, is the enhancement of PHY layer, by means of IEEE 802.11px. This work-in-progress specification, which will be addressed to as *dot11px* in the next lines, started as a joint effort between C2C Communication Consortium, a community which counts many cars manufacturers, and ICT companies, in order to collect results to make a standardization proposal to IEEE between 2018 and 2019.
The requirement for an evolution of already existing vehicular wireless technology, came from the fact that traditional 802.11p is based on 802.11a, a very old version of the standard. In fact, Wi-Fi standards are evolving rapidly, exploiting new or renewed Modulation and Channel coding Schemes (MCS), that allow to increase the bandwidth and throughput, principal commodities in NGV networks.
According to what C2C has released, since some documents have limited access to the non participant to the Consortium, dot11px will rely, in describing IEEE 802.11px PHY layer, on the two most recent IEEE Wireless LAN standards, namely:

- 802.11ac, working at 5.x Ghz,
- 802.11ad, working at 60 Ghz (mmWave bands).

In this paragraph, we will give an overview of the two technologies, thanks to the contribution from two white papers [20] and [21], focusing more on the PHY layer, and relating their features with what is expected and planned for 802.11px.

802.11ac and 802.11px

The IEEE 802.11ac standard was developed from 2008 to 2013, when it was published, on December. The specification is also known as VHT (Very High Throughput), since it is primary aimed at extending dot11a, from which dot11p was derived, and dot11n standard, and offering higher throughput for devices operating in the 5 GHz band. In particular, in Europe, channel allocation sees the spectrum frequency for dot11ac ranging from 5.17 to 5.33 GHz and from 5.49 to 5.71 GHz.
The main features introduced are summarized in the Table below.

Bandwidth	optional 160 MHz, mandatory 80 MHz
Maximum data rate	up to 3.5 Gbps
MIMO	from 2 up to 8 spatial streams, MU-MIMO support
Modulation	up to 256-QAM
Guard interval	400 ns
Error recovery and coding techniques	LDPC STBC

Channels are wider than preceding Wi-Fi standards. It supports optional bandwidth, with size up to 160 MHz, translating in maximum achievable data rate of 3.5 Gbps, and mandatory minimum width of 80 MHz consists in two adjacent 40 MHz channels. Dot11ac supports also the 20, 40 and 80 MHz bandwidths.
OFDM access is used, as it happened for dot11p, and the number of sub-carriers varies, depending on the channel width. For instance, if the channel size is 80 MHz we will have 256 sub-carriers, 160 MHz doubles to 512.
Before proceeding in our description of 802.11ac, there are two channel techniques that is worth to introduce, in order to understand it better: STBC and LDPC.
Space-time Block Coding (STBC) is a PHY layer technique, that exploits the natural diversity of the wireless channel to improve the quality of the received signal, by transmitting multiple copies of the same data stream, multiplied by an opportune code. In fact, because of the obstacles encountered during the path towards the receiver, signal is scattered and reflected; thus multiple copies of the same signal arrive at the receiver. The redundancy can be used to compensate for the impairments on the channel and rebuild a strong signal at the receiver [23].

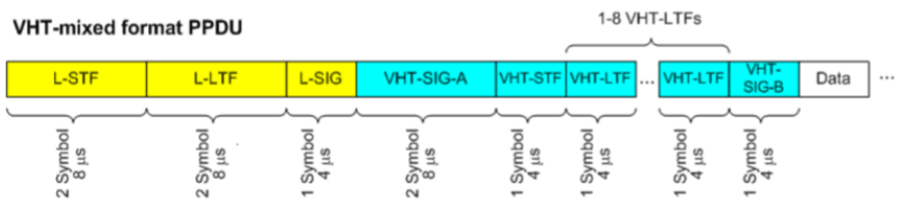


Figure 2.8: IEEE 802.11ac Frame format. Taken from [20]

Low-density Parity Check (LPDC) coding technique is used to transmit and receive correctly a message through a noisy environment [22]. The Frame format in 802.11p consists in a six-field header, partly containing legacy fields, inheritance from previous versions, partly new ones, and they are:

- Preamble: composed by three so-called *Legacy fields*, which were also included in dot11n and dot11a specifications: Legacy Short Training Field (L-STF), Legacy Long Training Field (L-LTF) and Legacy Signal (L-SIG). They are followed by VHT fields, which have been inserted specifically for dot11ac;
- VHT-SIG-A: this field is very important in perspective of dot11px. In fact, it consists in two OFDM symbols, that are modulated respectively with BPSK, to allow back compatibility with dot11n devices, and a $\pi/2$ rotated BPSK, to allow decoding of the packet from each dot11ac device. It contains CRC information, MCS, including an enabling LDPC bit and STBC option;
- VHT-STF (VHT Short Training Field): it is used to improve the estimation of the gain during a MIMO transmission;
- VHT-LTF (VHT Long Training Field): these 8 fields are used by the receiver to estimate the MIMO channel and equalize the received signal;
- VHT-SIG-B: provides information on data length and eventually support for Multi User MIMO (MU-MIMO).

Backward compatibility with legacy 802.11p devices, very important KPI for 802.11px, is guaranteed, since the header, as we have seen, is composed by a non-VHT data part, encoded in Basic Convolutional Coding (BCC), and a VHT-data part, encoded with LDPC and STBC. Each device, supporting 802.11ac, sends two times the message, once with the legacy header, the second time with the VHT header. IEEE 802.11px, in the 5 GHz band, as C2C is intended to propose it to IEEE, will address the following characteristics:

- OCB mode is activated, therefore 5.9 GHz band for Europe, because the OCB mode has been thought to work in that range of frequencies;

- It exploits dot11ac at the PHY layer, which we have just described, with two streams STBC codes and the LDPC enabled in the VHT-SIG-A field;
- Data is modulated with 64-QAM.

802.11ad and 802.11px

IEEE 802.11ad [19] represents more than and extension of dot11ac and it has been conceived to exploit the potentiality of mmWave technology.
Millimetre band (or millimeter Wave) is the commercial name for the range of unlicensed frequencies from 30 to 300 GHz, thus presenting very low wavelength, from 10 to 1 mm. They are characterized by very short range (1 km), since they experience high attenuation over the channel. For their huge bandwidth and achievable throughput, they are retaining special interest in telecommunication companies and communities, for their impact in near future communications systems, like 5G, in which urban area, where they will be massively deployed, are characterized by very small serving cells to support high devices density. The reserved bands in Europe for this technology are 58-64 GHz.
The dot11ad PHY is also reffered to as Directional Multi-Gigabit (DMG), to underline the focus of such a technology in guaranteeing high bit rates.

Bandwidth	2.16 GHz
Maximum data rate	up to 8 Gbps
MIMO	from 2 up to 8 spatial streams, MU-MIMO support
Modulation	up to 64-QAM
Error recovery and coding techniques	LDPC/STBC

According to [21], even if the frame format has a general structure, there are three physical layer modes:

- Control PHY: it is mandatory for each device supporting the *ad* variant, since is in charge to carry control information for establishment and management of the connection. Control data (40 bits long) are scrambled, LDPC encoded, modulated $\pi/2$ - DBPSK and then spreaded.
- Single Carrier PHY: it is the most used of the three types we are describing, and allows to reach, depending on the MCS, up to 8 Gbps. Data (64 bits long) are scrambled, LDPC encoded and modulated according different modulation schemes, depending on the rate and robustness desired. A guard interval sequence is then inserted at the end of the payload, in order to allow the receiver to collect a reference signal if the size of the data packet is important.

- OFDM PHY: it is considered as obsolete and is foreseen to be removed in the next versions of the standard. Thus it will not be discussed in this seat.

The general structure contains different fields:

- Preamble: it is the same in every dot11ac packet and is composed by the STF and Channel Estimation (CE) fields, very useful for synchronization and initialization operations and to recognize the PHY type from the three above-mentioned;
- Header: it varies depending on the specific type of PHY used (Control header, Single Carrier Header, OFDM Header) and includes length and MCS related information;
- Data,
- Training for beamforming: optional field used for optimization of the beamforming parameters. In fact, since at 60 GHz the free-space loss has greater impact on channel conditions, this technique assumes importance, and optimized beamforming sequence may be sent over the wireless channel.

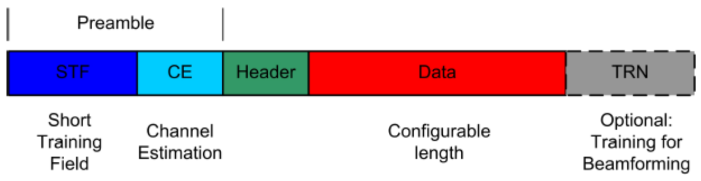


Figure 2.9: IEEE 802.11ac Frame format. Taken from [21]

The design guidelines, dictated by C2C, are given to optimize IEEE 802.11ad for mmWave VHT PHY OCB and, only in the long term, focusing on MAC layer.
It is relevant that the next standards for vehicular networks seem to follow the same technological path, at PHY layer, undertaken by mobile broadband networks. In fact, as we will discuss in the next Chapter, below 6 GHz and mmWave bands are the two frequency ranges in which 5G will work with. May be possible, in the near future,a meeting point between these two, opponent and debated solutions for many time, to join their forces and cooperating for a safer road environment?

2.2 C-V2X

The practiced effort, by 3GPP, to provide a specification for V2X support, exploiting cellular communication, converged in the generic denomination of *C-V2X*.

Such denomination changes, depending on the cellular technology supporting it, like LTE-V2X or the next 5G-V2X. C-V2X was firstly discussed in Release 14 [24], whose work started in March 2016, after completion of Release 13, and has been completed in June 2017.
Cellular V2X is nowadays based on the most recent LTE-Advance and LTE-Advanced Pro technologies.
In the next paragraph, main concept and components of LTE network architecture are revised, since are connected to the network architecture we work on in this Thesis, then LTE-V2X is explored.

2.2.1 LTE

Long Term Evolution (LTE) is the fourth-generation of mobile communication technologies. The standardization effort started with Releases 8 and 9, continued with Release 10 to 12 (LTE-Advanced) and has ended with LTE-Advanced Pro, in Releases 13 and 14. It is the direct evolution of UMTS (3G) technology. The main characteristics of an LTE architecture is to contains a fewer number of nodes, with respect to UMTS, involved in the handling of all the traffic. It is traditionally divided into the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN), by means of which user terminals, User Equipments (UE), get access to the network, and the Evolved Packet Core (EPC).
In this Chapter, we will limit in a brief overview of the traditional architecture described before: the Packet Core and the Radio Access Network, concluding with a description of bearers and protocols. Briefly speaking, LTE supports very high data rate, up to 300 Mb/s is downlink and 75 Mb/s in uplink. Channel access is Orthogonal Frequency Division Multiple Access (OFDMA).
In LTE, we have three different modulation schemes adopted: QPSK (2 bpsymb), 16-QAM (4 bpsymb) and 64-QAM (6 bpsymb), allowing to increase data rate with the increasing of the modulation symbols and, at the same time, decreasing resilience to errors and so making the transmission robust. Frequency channels range between 5 and 20 MHz width. LTE supports vertical hadover seamlessly towards previous technologies. Frequency bands belong to the portion of the spectrum below 6 GHz: 800 MHz for large radio cells called pico-cells, then 1.8 GHz and 2.6 GHz for smaller radius cells called micro-cells.

EPC

The EPC network handles user data and control traffic, establishing bearers for transmission of information and allowing users to access external networks, like Internet.
The main components of the EPC [25], sometimes defined System Architecture Evolution (SAE), are the following:

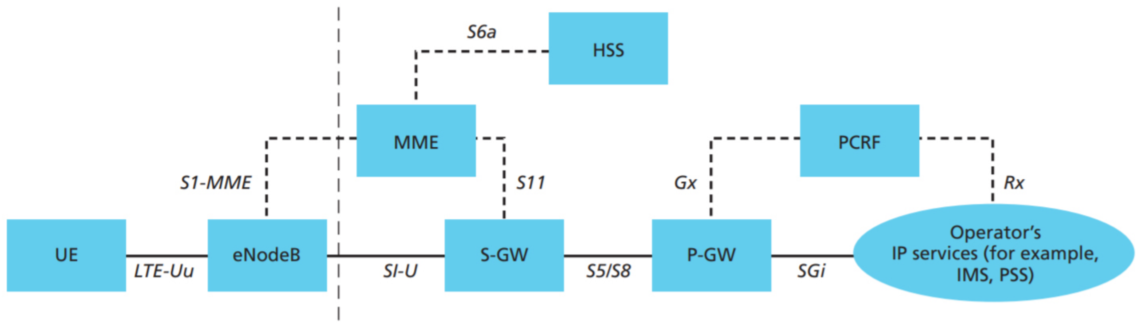


Figure 2.10: LTE architecture. Taken from [27]

- Mobility Management Entity (MME). It is the device dealing with several CN functionalities, among which excel control traffic management, handling the signalling related to mobility (handover and roaming) and security (NAS signalling security) for E-UTRAN access.
  It is also responsible for authenticating and authorizing the user, by interacting with the HSS and choosing a S-GW and P-GW for the UE. It manages and stores UE contexts. It communicates also with the eNB in the E-UTRAN, through the S1-MME reference point.

- Home Subscriber Server (HSS). It is a database containing subscriber-related information and connected to the MME. It provides mobility management support functions, call and session set-up, user authentication and access authorization.

- Serving Gateway (S-GW). It is unique for each UE. It serves all data. traffic, routing and forwarding ingoing and outgoing IP packets. It is also the main player for inter-eNB handover [26] bridging the UE context from an eNB to another one. When downlink data traffic arrives for the UE, it triggers paging to retrieve the UE which the information is for.
  Together with the MME, this device is the main touching point between the E-UTRAN and the EPC. It is connected to the eNB through the S1-U interface and is logically connected to the P-GW through the S8 interface.

- Policy and Charging Rules Function (PCRF). It performs decision-making for two main functions: flow based charging for service data flow, and application traffic and policy control, for QoS purposes. It supports the P-GW Policy and Charging Enforcement Function (PCEF), imposing the rules for charging according to the flow, providing QoS authorization, that decides how a certain data flow will be managed in the S-GW, in accordance with the subscriber's profile.

- Packet Data Network Gateway (PDN-GW), also known as Packet Gateway (P-GW). It interconnects the EPC with external IP networks (like Internet and the IP Multimedia Subsystem (IMS)), that supports delivery of voice and data services. Thus, there can be more than one P-GW per user. It assigns the IP address to the UE for exiting mobile network, eventually offering QoS service to the UE and flow-based charging, thanks to PCEF and PCRF. It can be combined by vendors in the same box, together with a S-GW, and this choice has been done also for this Thesis project. It is connected to the PCRF through Gx interface, to the S-GW through the S8 interface and accesses external networks through SGi reference point.

E-UTRAN

The E-UTRAN consists in User Equipments (UE) and evolved Node Bs (eNB).
A UE can be any device which is attached, wireless, to one or more eNBs, through a radio interface. An eNB is the edge hardware which connects the EPC with the UE. It embeds all the functionalities of the previous generation Radio Network Controller (RNC) and exploits OFDMA in downlink and Single Carrier FDMA (SC-FDMA) in uplink on an interface called Lte-Uu. An eNB interfaces with EPC and other eNBs through the following bearers:

- S1 interface: from eNB to the EPC. It includes the S1-MME interface for EPC control traffic, where S1-AP protocol defines communication procedure, and the S1-U interface for data traffic, with the S-GW and GTU protocol describing communication.

- X2 interface: from eNB to eNB and over which communication is done through the X2-AP protocol.

An eNB consists in a Remote Radio Head (RRH) and a Baseband Unit (BBU). The former is the unit which transmits and receives wireless signals. It converts, from one hand, signals received from the BBU into radio frequency (RF) signals, then power-amplified and transmitted to the UE, and from the other hand it converts signals received from UEs into digital baseband signals and transmit them to the BBU. This implies that packets received from the core by the BBU are then modulated into digital baseband signals and transmitted to the RRHs and the baseband signals received from the RRH are demodulated and sent to the EPC.
The latter is responsible for digital baseband signal processing, called processing and monitoring control processing. The two of them are today connected with an optical interface, compliant with the Common Public Radio Interface (CPRI). The link between RRH and BBU is called fronthauling, whereas the backhauling is the

connection between the eNB and the Core Network. Thanks to these two components the eNB, differently from UMTS Node B, embeds very advanced control functionalities, like interconnection to the EPC, header encryption, security and radio resources management, and it follows from that than, whenever a user needs to move into another cell, all its context, defined by the previous eNB, must be transferred to the new one.

Protocols: data and control plane

Users traffic is distributed between control plane and data plane. Control plane is the functional part of a network that takes decision about where traffic must be sent. The data plane is instead responsible of forwarding traffic to the next hop along the path to the selected destination network, according to control plane logic. Orthogonally to the distinction between user and control traffic, there are two groups of protocols for intra-LTE communication and they are called NAS and AS:

- Non-Access Stratum (NAS) protocols define procedures for the correct communication between the UE and the EPC and are marked in green in the figures.

- Access Stratum (AS) protocols define communication layer between UE and eNB and are marked in blue in the figures.

In the EPC, the user plane is managed with the GPRS Tunneling Protocol (GTP), which describes communication layers between P-GW and eNB, respectively UDP/IP, data link layer and PHY. In the E-UTRAN, over the Lte-Uu interface, the AS protocol stack is divided into Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), MAC and PHY layer [27].
Control packets are exchanged between the MME and the UE, passing for the eNB. NAS protocol header is directly appended to control payloads and it is not removed by the eNB. The layer below, S1-AP, managed by the S1-AP protocol, provides an interface between MME and eNB, supported by lower layers of the stack that are SCTP (Stream Control Transmission Protcol), IP, data link layer and PHY. The Radio Resource Control Protocol (RRC) sets up the radio bearer at layer 3, between eNB and UE, and is supported by LTE lower layers, the same as for user plane.

2.2.2 LTE-V2X

In 3GPP specifications, Vehicle ITS-S are renamed User Equipments (UE) vehicles. The V2X applications supported by C-V2X are V2I, V2V, V2N and V2P.
LTE Release 14 envisages two alternatives for V2X communication:

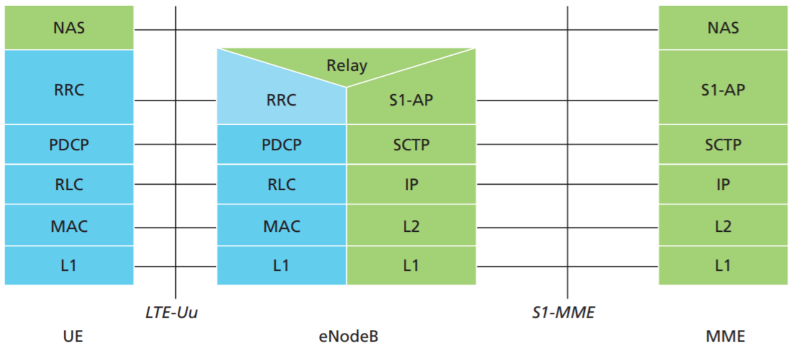


Figure 2.11: LTE control plane. Taken from [27]

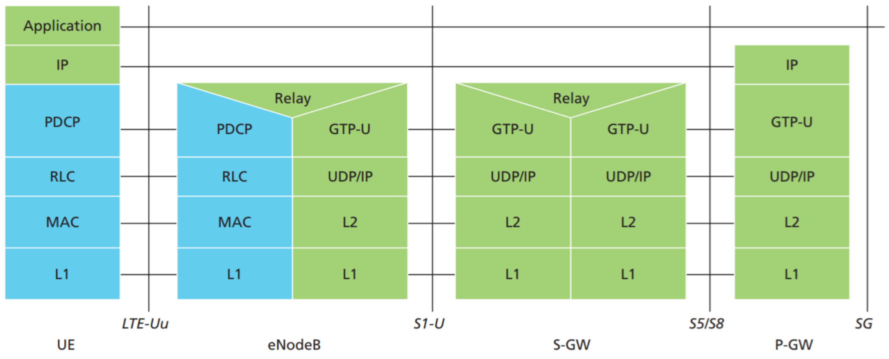


Figure 2.12: LTE user plane. Taken from [27]

- Lte-Uu interface: is the same defined in traditional mobile broadband LTE networks, established between the UE and the E-UTRAN and it is exploited by AS protocols.
- PC5 interface : is an interface for Device to Device (D2D) communication, sometimes referred to as *Sidelink*. Thanks to it, devices can immediately communicate, without additional latencies due to the communication link set up by the infrastructure. In this case, E-UTRAN support can be present or not.

In Lte-Uu, all the traffic is sent in uplink to the eNB, that, eventually, retransmits it in downlink to the destination. According to the standard, LTE-Uu transmission can be done in unicast or in broadcast, for example by making use of the evolved Multimedia Broadcast Multicast Service (eMBMS). As explored in [28], eMBMS is a transmission technique in which network is divided into MBMS Single Frequency Networks (MBSFN), so that a set of eNBs can send the same multimedia content to the user and it senses as if it were broadcasted in the same cell.

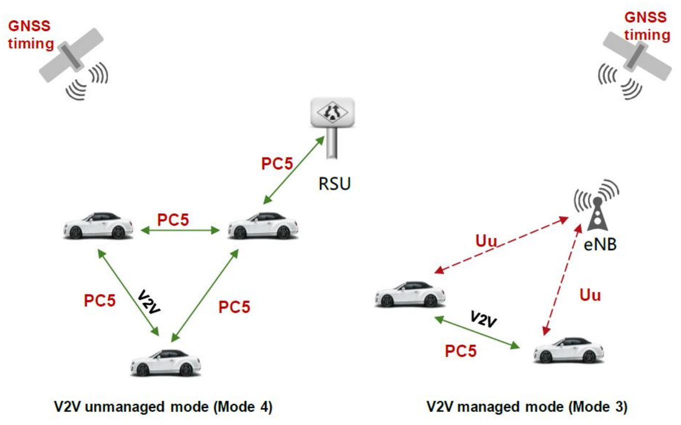


Figure 2.13: PC5 and Lte-Uu modes

A consistent difference, with respect to the ITS-G5, is that RSU is no more a mandatory architectural element, but can be implemented in two manners, as a UE running a V2X application or as a eNB-type UE, namely including eNB architecture and V2X application server.
Already in Release 13, two modes have been defined for Device to Device (D2D) communication. Mode 1, foresees that the network schedules UE transmission, so that UE asks for resources to transmit and the eNB configures them. In Mode 2 the UE operates independently from the network, selecting, in autonomous manner, which resources to use, from a pool signalled by the eNB.
In Release 14 two new modes have been defined, as an enhancement with respect to the previous one, for transmission over the Sidelink, with E-UTRAN service and without E-UTRAN service support enabled, that are defined at the authorization and provisioning phase:

- PC5 *mode 3* or managed, available exclusively when the UE is under mobile operator coverage and is authorized to the PLMN serving the cell. Optimized resources and parameters are provided to the UE for communication;
- PC5 *mode 4* or unmanaged, available when the UE is out of mobile operator coverage, is characterized by the fact that the UE operates independently from the network (E-UTRAN) support.

LTE-V2X exploits, as LTE does, OFDM at physical layer. Each transmission is composed by 10 ms frames, divided into 1 ms subframes, corresponding to 1 OFDM symbol. An OFDM symbol is 12 subcarriers of 15 kHz long and a Resource Element is the smallest transmission unit. LTE-V2X channels are 10 to 20 MHz long. Starting from Release 14, to the already mentioned Mode 3 and Mode 4, another operation modes for V2X communication has been specified, foreseeing both of them. This scenario is divided into two sub-scenarios:

- First scenario: a UE sends V2X messages, using Sidelink, to other UEs. The message is received by a UE-type-RSU, which forwards it in uplink via Uu interface. The message is taken in charge by the E-UTRAN, which pushes it in downlink to other UE(s).
- Second scenario: we have the reverse information flow. In fact, there is the reverse operation with communication via PC5 firstly and after that UE type RSU exploits retransmission in Sidelink.

In addition, further modifications to LTE have been applied, for Sidelink communication, as the insertion, in the LTE transmission scheme of extra Reference Symbols, to cope with higher Doppler Spread effect, due to high speeds, then additional techniques, have been introduced, to reduce latencies and allowing good performance also in a dense environment as the road.
For the LTE Uu interface, by the other side, in addition to the already mentioned eMBMS for V2X communication, QoS for C-V2X service has been introduced by Release 14.

LTE-V2X architecture

In this paragraph it is described the non-roaming architecture for V2X services. We have decided to introduce it, since it containes all the principal functional entities. The other are inspected in [29]. Main architectural elements in C-V2X are:

- V2X Control Function (VCF): a logical function used by the UE to retrieve the parameters to access V2X service in a specific PLMN, either when the UE is in coverage and when it is in off-coverage. V3 interface is used by UE to interact with VCF.
- V2X application server (V2X AS): is an entity designated mainly to receive UE unicast data and transmit them in unicast or in eMBMS modes. It is also in charge of signalling to the VCF the UE parameters for communication over PC5, when E-UTRAN service is enabled. V2X AS Discovery must be performed by the UE through the DNS service. V2X application servers can be used with many different purposes and deployed in different places, therefore, during years, standardization entities have purpose their versions. It seems that, nowadays, ETSI MEC is the best candidate for the role, and we will discuss it later on.
- MME: this architectural element, present in LTE networks, is in charge of performing additional functionalities with respect to LTE, obtaining user subscription information, related to the V2X service, and informing the E-UTRAN about the authorization status of the UE, concerning the V2X service.

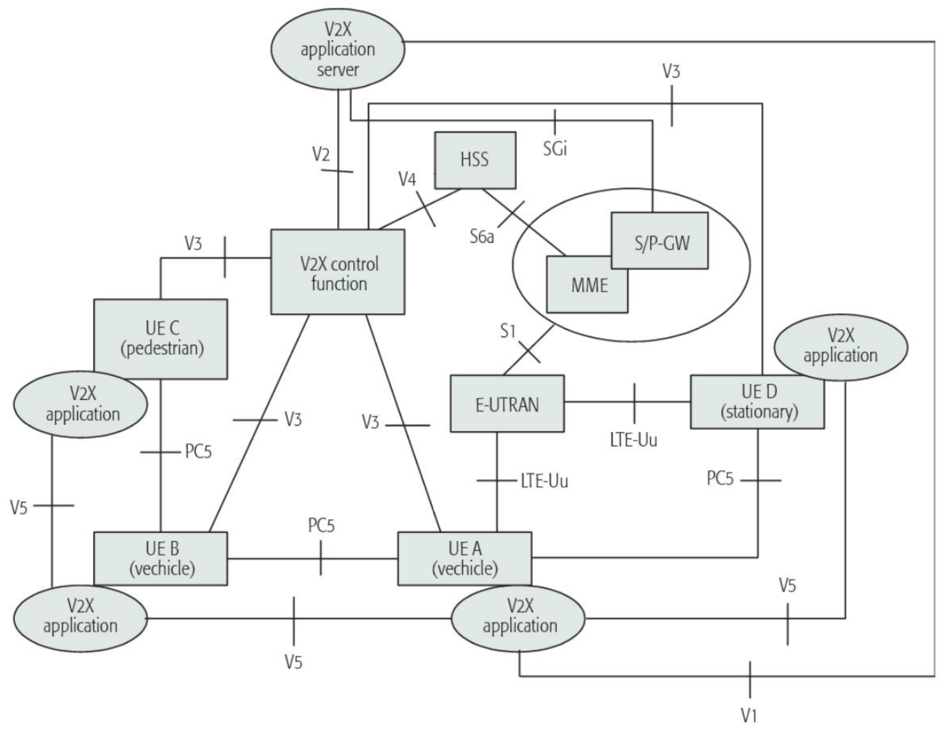


Figure 2.14: C-V2X non-roaming architecture. Taken from [29]

- UE can be a pedestrian (smartphone), a vehicle (passenger car, emergency vehicle) or an RSU. It mainly improves the characteristics of a traditional LTE UE by receiving from LTE-V2X entities, or by self-configuring, parameters for communication, like set of frequencies for V2X, V2X AS address, radio resources variables.

Main reference points, in addition to the ones derived from LTE, are:

- V1: interface between the V2X in a UE and the application server,
- V2: interface between the application server and the VCFs,
- V3: interface between V2X CF and UE,
- V4: interface between the VCF and HSS; it is used for authentication purposes (VCF can access the user data, update information related to a specific subscriber, removal of the subscription),
- V5: interface between two V2X applications running on two UEs,
- V6: interface between VCF located in different PLMNs. In fact, whenever a user leaves the HPLMN, the VCF, within the HPLMN, has to call the

Control Function in the VPLMN, in order to offer the user IMSI/MSISDN or to ask for authorization information in the Control Function of VPLMN. The CF in VPLMN shall check whether the user is present or not, and if it has authorization for V2X service.

In C-V2X, the transmission and reception of V2V and V2P messages (D2D), thus exploiting PC5 interface, can be done using Proximity Service (ProSe) [30]. In ProSe, each UE possesses an expression to communicate and, in order to establish a D2D communication, it asks to a ProSe APP server to retrieve the expression code of the UE, which it intends to establish direct communication with. Once it has retrieved it, air resources for the communication can be allocated by the UE itself or by the eNB [31].
The communication through Uu interface is conveyed by the V2X AS. V2X messages shall be delivered using LTE-Uu interface by a UE, the eNB receives the message and forwards it directly to the S-GW, that routes that to V2X Application server. The Application sever finally establishes how and to which area forwarding it in the network.
QoS is, ProSe Per Packet Priority (PPPP) mechanism, is used for PC5 interface, added at the application layer before transmitting V2X packets, instead, for Lte-Uu interface, QoS is assured to specific values of the QCI. A Home PLMN is the mobile operator network of which the user is a subscriber. When a subscriber moves out the geographical area covered by its own HPLMN carriers, its UE can attach to another PLMN, called Visited PLMN, in order to preserve seamlessly the connectivity.
Since V2X Application servers are different and may cover more than one cells, supporting different services, server discovery is one key issue for LTE-V2X. A first purpose has been done to exploit anycast connection. One-to-the-nearest communication, in fact, allows to identify the closest server to the GW. Also broadcast mechanism has been suggested.
Privacy should be assured, for subscriber accessing the V2X services, and neither the mobile operator or a third party entity can track the UE in a given region [20]. In order to keep privacy for the UE entity, 3GPP has identified three possible approaches:

- Exploiting a third party server, that assigns to the users an identifier, from a pool provided by mobile operator, which is encrypted (for instance an IMEI code), so that users can authenticate the systems without the operator knows their identity;

- Using a regional third party agency, that behaves as Home PLMN for the UE, which it means that there is a third party server that takes the place of the HSS and the other mobile operators are like Visited PLMN;

- Last solution is that the core network of the mobile operators is managed by an authorized agency. PLMN authentication procedure should be extended.

C-V2X, supported by LTE, promises very good performance thanks to the broadband mobile network technology service, scalability and, with respect to other V2X technologies, alike IEEE 802.11p, is already deployed in Europe and in the world. By the other side, such an architecture needs to be further deployed, to support V2X use cases, it requires a joint effort, especially in terms of economical expenses, between the telecommunications players and road operators, that will build the ecosystem. In this Chapter, we have overviewed main communication technologies for V2X in Europe, with a critical view on the future and how vehicular connected environment can evolve. 5G is going to pose new challenges and solutions for this new-born technology. In fact, services meeting KPIs and requirements, strictly connected with the ones related to vehicle and mobility, as low latency and high throughput, are mandatory in the new 3GPP Releases, starting from 15, as we will see in the next Chapter.

Chapter 3

5G: a network for services

The standardization effort towards 5G is already started in 2016, with Release 14 of 3GPP, concerning LTE Advanced-Pro. Commercialization of 5G is foreseen in the early 2019/2020 [32] and currently, Release 15, which defines the Non-Standalone 5G, is available, whereas the Standalone version is a work-in-progress and is planned to be completed in December 2019.
Yesterday, the concept of network was very clear and defined, something naturally confined into hardware and devices destined to specific uses and services. Billionaire telco companies, like CISCO, have built their economical empires designing purpose-oriented hardware. When we have discussed about MME and PGW in LTE, we could have immediately imagine them the first time we have heard about them as physical machines located in a central office.
In the last years, things have changed. *Virtualization* has made possible to imagine a network where instead of specific hardware we have multi-purpose machines with software defining what those ”empty” box should do or not. Virtual Machines (VM), direct consequence of virtualization, permit to emulate a computer Operating Systems, relaying on the hardware of a physical machine. On the same computer, more than one VM can be created, and the parallel execution of VMs is granted by an hypervisor software, that manages the access to hardware. Nevertheless, with just virtualization, still technological revolution was stacked at hardware, because, in order to build more powerful computers, we needed to increase hardware capabilities to the machine running a set of VMs. The deployment of Cloud Computing, started already in 1960, and which basically consists in accessing on-demand resources, for instance memory, applications and services everywhere and with just the support of an Internet connection, has made possible to decouple hardware and software, improving the system thanks to the availability, simplicity, reliability, efficient resource utilization, easy maintenance and scalability.
Software Defined Networking and Network Function Virtualization (SDN/NFV) [33] have finally breaks the border between physical devices, so that, the EPC of an LTE network is ready to become virtual [34] in proto-5G networks, supporting

all the functionalities described in the previous Chapter, all performed in software. Cloud Computing and its enormous promises are the main triggers for Multi-access Edge Computing and Network slicing, defined by many authoritative sources as pillars of new 5G communication. The former consists in moving, or dynamically allocating, network resources close to the access network, in order to serve users with lower latencies and the highest quality. For example, the most clicked 4K resolution video on Youtube and the most watched film on Netflix, can be placed in a docker close to the eNB, in order to be streamed to users with very small waiting time. Network Slicing is a form of virtual network architecture, which exploits NFV/SDN to create, over the same physical infrastructure, a set of virtual network called slices, with a great saving of economical resources. This network enhancement is subjected to interest from many Vertical industries, that can enter telecommunication market, offering services tailored to users and creating partnerships with mobile industry, to deliver those services. In 5G Transformer (5G-T), these industries are: Automotive, Entertainment, eHealth, eIndustry, MNO/MVNO. It is also very interesting for Mobile Operators, that need to diversify the range of their services because the Average Revenue Per User (ARPU) is flattening. The possibility, offered by 5G, to share network resources and the use of open-source platforms and Commercial Off The Shelf hardware, is the key for a saving resources and reduce costs.
In this Chapter, we will provide a view on 5G, describing the architectures, trying to understand what we have to expect from this new model and what are its key pillars. We will start with providing a description of 5G New Radio (NR), the new access network for the next generation network and the two architectures foreseen by 5G: Standalone and Non-Standalone. After that, we will focus more on 5G multi-service support, focusing on Network Slice, in relation with the architectures. Finally, we will explore the Ultra-Reliable Low latency Use case, that has more impact on our work in terms of requirements and features. Lastly, Multi-access Edge Computing will be envisioned.

3.1 5G

In 2016, a document published by 5GPPP, an European initiative aimed at exploring and developing 5G, with joint effort between companies and academic sectors, has presented a visionary overview for 5G future networks, supported by other associations and researchers.
In December 2017, 3GPP, the leading voice in mobile network standardization, has completed the first phase in 5G deployment; we will start from here to introduce the three steps 5G is going to follow to have completed standardization in December 2019:

- Non-Standalone (NSA): Release 15 introduces 5G New Radio (NR), deployed in coexistence with legacy LTE core network (E-UTRA-NR Dual Connectivity);

- Standalone(Phase 1): with Release 16, whose first phase has been completed in June 2018, the complete 5G specification has been terminated, including radio access, core network, security layer of one of the services supported by 5G, the so-called enhanced Mobile Broadband (eMBB) service;
- Standalone(Phase 2): Release 16 will be completed, including phase 2, in December 2019, considering, in addition to eMBB, also Massive Machine-Type Communication (MMTC) and Ultra-Reliable Low Latency Communication (uRLLC).

In this section, we will exploit, to review concepts of architectures introduced so-far, a very recent White Paper by NEC laboratory, in Japan [35], mentor company in telecommunication field.
5G has been announced for years as a network with very low latency, between 1 to 10 ms, 99.999 % reliability, 99.999 % availability, a user rate up to 20 Gbps, very high number of cells (tens of them for km^2) and a very high number of devices connected (up to 100 000 km^2). These KPIs depend and need to be tailored to the specific service supported. Will 5G be able to satisfy the expectations and win the challenge?

3.1.1 5G NR and Non-Standalone architecture

In [41], 5G NR access network is described. The new Radio Access Network aims at guaranteeing high data rates (20 Gbps in downlink and 10 Gbps in uplink), reliability, availability, flexibility, energy saving and cost reduction. Frequency spectrum will be distributed to 5G [36] and is divided into:

- High-band frequencies (between 24 up to 86 GHz), a portion of the spectrum where mmWave, available for market in 2020 [37], will play a central role. They can reach the most demanding 5G KPIs, in terms of bandwidth and data rates (theoretically, up to 100 Gbps). Nevertheless, on the other side, signals in mmWave bands experience more attenuation when frequency increases and thus, transmitting at higher frequencies, will comport smaller cells deployment. mmWave bands foresee maximum channel bandwidth up to 400 MHz. An urban environment is the best place for the deployment of such cells, that are also known as pico-cells.
- Low frequencies and mid-band frequencies (below 6 GHz), are the frequencies exploited by traditional wireless communication systems. They are very suitable for wide area coverage, in location where it is very difficult a large

frequency deployment, like rural area. It is foreseen a channel bandwidth between 5 and 100 MHz.

A difference with respect to LTE, is that in 5G, the spacing between carriers becomes variable, impacting on the length of the slot and thus the granularity of the transmission. Channel spacing can vary between 15 kHz and 60 kHz, while other numerologies have been purposed and are currently under study.
The possibility to support different sub carrier spacing poses benefits in terms of transmission. Whenever the spacing is higher, the cyclic prefix will be shorter, and thus the timeslot duration is shorter too, enabling the possibility for the so called transmission per mini-slots. In 5G, this characteristic allows a mobile station to wake up and transmit without being bounded as 4G by the timeslot granularity. This fact translates in a gain for all those systems, like mission critical ones, that require to send information whenever they have it, without waiting to be scheduled in a traditional slot.
For lower frequency bands, duplex scheme is Frequency Division Duplex (FDD); Time Division Duplex (TDD) is used in the highest part of the frequency spectrum. There exists a difference between such two schemes. In the first mentioned one, two frequency slots are allocated for transmission and reception, and thus a transmitter can exploit both of them to transmit and receive in parallel. In TDD, instead, just one frequency channel is allocated, containing timeslots for downlink and uplink transmission. TDD, on the basis of the good comparative in [40], are particularly suitable in case of small cell coverage and unbalanced traffic; on the other hand, FDD is useful in case of wide cell area coverage and balanced traffic. NR exploits both of them, with devices operating in half-duplex and others in full-duplex. It is particularly interesting that a flexible technique, called dynamic TDD can be exploited, in which, in a very dense urban environment, in order to sustain traffic variation, traffic scheduler selects uplink or downlink allocation for a given UE.
5G NR is based on OFDMA, either in uplink and downlink. New NR studies [38], in collaboration with important technology vendors like Samsung, DOCOMO, Hawei, Intel and Qualcomm, are testing the potentialities of Non Orthogonal Multiple Access (NOMA) schemes that have, according to several studies, conducted, in the last years, the capability to overpass current OFDMA technology in terms of some of the most stringent requirements for 5G, like spectral efficiency and low energy consumption. NOMA is an access scheme achieved in power-domain or in code-domain, giving the possibility to multiple users to access the channel using the same time and frequency resources, differently from Orthogonal Multiple Access (OMA) techniques like OFDMA. In power-domain NOMA, for instance, there are advantages in terms of superior spectral efficiency and mitigated interference through Successive Interference Cancellation (SIC) technique, reduced latency, support for an higher number of connected devices, maintaining user fairness. For counter, it introduces complexity in the system and more energy consumption, and arises the need for a lower number of users connected, in order to avoid error propagation that

is a systematic error in SIC technique, and, in order to maintain good performance, a reasonably high channel gain has to be maintained between devices [39].
Frame structure is very similar to that described in the previous Chapter for LTE and LTE-V2X. Time is splitted into 10 ms frames, divided into 1 ms subframes. Each of the subframe, subdivided in slots, transmits 14 OFDM symbols.
Internetworking is the answer of 5G NR to the well-know problem of channel unbalance in mobile broadband networks. The problem of unbalance is a matter of signal power; uplink channels are much more crowded and interfered, since there are more UEs transmitting, than downlink channels, where just base stations access the channels, especially to send control and data information to RAN devices. It is clear that an eNB will interfere at most with other eNB signals, whereas UEs interfere at least with devices in the same cells, that are much more. The base station transmits at high power and that make downlink transmission very suitable for higher frequency bands, whereas the UEs are power limited and require lower frequency bands. Through internetworking, 5G NR exploits a joint compound between NR and LTE frequencies to compensate for lack in coverage (typical of lower frequencies of the spectrum, that can reach lower data rates than higher frequencies). NR can cooperate in this way with LTE, without reducing its capabilities. In Release 15, in fact, there are presented two possibilities for coexistence between the two systems: the former is a full coexistence of either uplink and downlink transmission. The latter foresees just coexistence in uplink and, in particular, in the lower part of the spectrum, where NR can compensate for the unbalance.
Another gold requirement for 5G is the energy saving, specifically relevant for connected devices located in places far from urban areas, and even difficult to reach, as the top of a mountain or the seabed. In fact, the enormous growth of connected devices has increased the need to build up systems which are more sustainable and low energy. Terminal devices from the access network, nodes and Data Centers, are planned to be low energy consumptive. In relation to this, for NR, low-energy initial procedure has been conceived. In fact, the Radio Access procedure, that an LTE UE used to do, has been simplified; UEs receive the synchronization signal every 20 ms, four time less than LTE, allowing the UE to look for a carrier for more time and enabling a smarter energy reduction. Another way to save energy, in 5G NR, consists in reducing the always-on transmission, due generally to broadcast control information, reference signals for channel estimation and signals, to detect the base station, for instance non transmitting reference signals over the full bandwidth but just in self-contained transmissions, or distribute system information only by some cells.
Coexistence with LTE is another deal-breaker for 5G NR. In fact, LTE systems cover low frequency spectrum. The enormous capacity, offered by NR, consists in compensating, with high frequency carrier aggregation, the coverage unbalance between downlink and uplink transmission.

MIMO remains one of the key enabler for transmission in 5G. MIMO is the transmission which we obtain from Multiple Input Multiple Output antennas. The very basic concept is to exploit channel diversity to get at the receiver a stronger signal. In 5G NR, the number of antenna ports doesn't change with respect to LTE in Single User Mode (SU-MIMO), standing to 8. Nevertheless, 16 antenna ports are exploited by transmitters in Multi-User Mode (MU-MIMO). The main aspects, compared with aspects of LTE-A pro, are reviewed in the following table. In the next paragraph, the architecture of 5G NR is envisioned.

Feature	**LTE**	**5G NR**
Peack data rate	downlink: up to 1 Gbps uplink: up to 150 Mbps	downlink: up to 20 Gbps uplink: up to 10 Gbps
Carrier frequency	from 450 to 3.8 GHz	from 600 MHz to 86 GHz
Bandwidth	up to 20 MHz	below 6 GHz: up to 100 MHz mmWave: up to 400 MHz
Channel spacing	15 kHz	between 15 and 60 kHz
Carrier aggregation	up to 32 CC	up to 16 CC
Multiplexing scheme	OFDM	OFDM
Frame structure	10 ms radio frame 1 ms subframe 0.5 ms slot	10 ms radio frame 1 ms subframe slot duration variable, depending on the channel spacing chosen
Channel coding	Turbo coding (data) TBCC (control)	LDPC (data) Polar (control)
MIMO	8 antenna for SU-MIMO 2 antenna for MU-MIMO	8 antenna for SU-MIMO 16 antenna for MU-MIMO

EN-DC (Dual Connectivity) architecture, also known as Non-Standalone, is the hybrid architecture conceived by 3GPP as a smooth step towards full 5G support. It allows a reduced time-to-market for 5G and providing some services with the minimal upgrade to existing LTE architectures. It foresees the adoption of 5G NR, discussed so far, as secondary access technology, besides LTE.

NR architecture

5G evolution is a smooth change with respect to LTE talking about the the architecture. The main elements of the new 5G Radio Access Network or Next Generation RAN (NG-RAN) are:

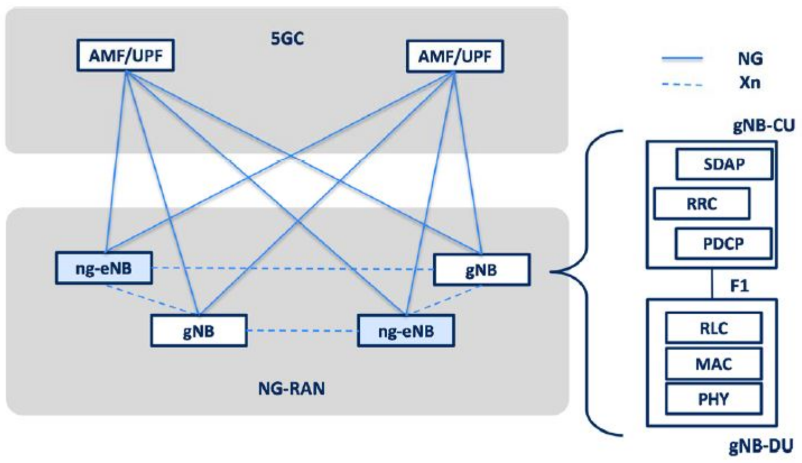


Figure 3.1: 5G NG-RAN architecture. Taken from [35]

- LTE eNB: is the radio access device, defined also just eNB in LTE network;
- NG eNB: it represents the evolution of the eNB, conceived to support the attachment with either the EPC and the NextGen core (called 5GC), the Packet Core of 5G architecture. It is divided into gNB-CU and gNB-DU:
  - gNB-CU (Central Unit): functional unit that must be placed in the base station,
  - gNB-DU (Distributed Unit): functional unit that can be placed in the Cloud.
- NR gNB: is the access device, a base station to 5G network, equivalent to the eNB for LTE networks;
- User Equipment (UE).

The main interfaces in NG-RAN are:

- NG: it connects the gNB with the NextGen core, similar to the S1 (S1-U and S1-C) in LTE. It is divided into NG2 and NG3, which are control and user plane reference points, respectively between NG-RAN and core network architecture;
- Xn: it is the interface connecting the NBs (NG and gNB) each other;
- F1: it is the interface connecting the two splitted parts of gNB architecture (CU and DU).

3.1.2 5G CN and Network slicing

In order to understand in the deep the importance and the disruptive features of 5G Core Network, with respect to legacy EPC network architecture, we shall introduce Network Slicing.
Network Slicing is a concept that dates back to 2015, when, in March, Next Generation Mobile Network (NGMN) published its famous White Paper [42]. Even in Release 13, something similar were introduced for the EPC, with a feature called DCOR (Dedicated Core Network), in which the UE could attach, at the authentication phase, to dedicated packet core elements, in order to improve network performance, for example in terms of latency.
Network Slicing is a paradigm that enables operators to create tailored services logical networks, called network slices, above a common and shared physical infrastructure. A network slice is a group of resources, either physically shared and virtually isolated, which are arranged together, in order that specific requirements of a given business case are fulfilled. This paradigm meets in a perfect way the requirement of configurability feature of 5G networks. Legacy mobile networks were bounded to the foundation of a (physical) network that *fits all services*, idea that becomes obsolete if we think of what we said about 5G: a set of *multi-purposes* technologies destined to support many services.
A single network slice is going to be tailored and mapped, in 5G, to a given service, which possesses different requirements, for instance in terms of latency, bandwidth, thoughput and reliability. A slice has to harmonize different characteristics to return the service to the user, without losing too much in performance [43]. In fact, the main goal of Network Slicing, has been deliberated by 5GPPP, is to accommodate different vertical sectors under a unified infrastructure view (5GPPP White Paper).
A Vertical is any industry, not necessarily belonging to ICT, that is focused on a specific business field or sector. For instance, 5G-Transformer project, a 5GPPP initiative, has investigated four verticals: Automotive, Entertainment, eHalth (evolved Health), eIndustry (evolved Industry) and Mobile Networking. Due to different histories and necessities, these sectors are looking out for 5G, in order to better provide their services, introduce new ones and reduce costs. Examples of why different services could have different requirements, are the vehicular environment and the next generation mobile broadband service. In fact, while for vehicles is of primary importance the immediate reception of a collision warning message, and thus very small latency KPI is important, a 5G UE will require very high throughput to receive in Ultra High Definition located in content delivery networks video servers.
The enabling technologies for Network Slicing are to be found in Softwarization. Softwarization is the network approach aimed at moving and running, in software, programs and functions that in traditional networks were performed in hardware, and in defining ways to easily and efficiently manage these resources. Two products

of Softwarization are Network Function Virtualization (NFV) and Software Defined Networking (SDN). They are very extended models and to let the reader deepening in this topics, we will leave references during this brief introduction.
NFV is a framework standardized by ETSI [64], in which network functions, which traditionally run over single-purpose hardware, we may think, for example, to firewalls, load balancer, DHCP servers or, to stay closer to our research, MME and HSS, are turned into virtual network functions (VNF), namely they are executed over one or more VM, running above a shared hardware and software infrastructure, and managed by a supervisor software entity called Orchestrator. The SDN orchestrator relies also on a concept introduced in October 2012 by the OpenFlow software project, under the name of Software Defined Networking [45] .
In SDN, the two functional entities, control and user plane, are kept separated in a network device. Control plane is, in fact, extracted from the physical device and the decision-making process is demanded to an external controller, in charge of setting rules, to address the traffic to the requested destination. By means of Softwarization approaches, companies can benefit an evident economical advantage in terms of and Capital Expenditure (CAPEX), the investments that a major can do to purchase network components or services, that in this case is reduced to the purchase of common COTS devices. Also Operational Expenditure (OPEX), the costs sustained for running daily this business (research and development, maintenance), are reduced, because of software platforms that can be managed in remote locations.
Slices are allocated as Instances (NSI) by a control entity to the UE that asks for a specific service, Service provisioning is currently being inspected in 3GPP Release 15-16 study groups, which is discussing about key points in network slicing. These problems can be listed in:

- Slice selection by a user. The literature identifies two possible approaches for the procedure: either could be the user that selects the slice by asking for it to the CN or can be the network itself that, on the basis of a service request, selects the best service to satisfy the user basic requirements;
- Slice isolation. In fact, even if the slices draw from common resources, provided by the network operator, they are logically independent networks, separated in their operations one by the other;
- Multiple-slice allocation. This is done in case, that could eventually happen, when a user requires usage of two slices at the same time;
- Resource sharing. Last but not least, there is this very critical issue. In fact, resources must in some way be shared, the most efficient way as possible, between virtual network functions, that may belong to different tenants. There are two ways in which resources can be shared, either in a static way (over-provisioning them to the peak usage) or in a dynamic way (network allocation

is checked and resources are flexibly re-allocated on the basis of service requirements variations). Several recent studies as this one [46], have shown that this second approach results theoretically in best performance.

For completeness, network slicing is not just a matter of core network services but is referred also to radio access network slicing or RAN slicing, in which resources are spectral and access. It is well explained in [47] and successive, but we will not cope with it, since it fells out of the scope of this Thesis.

3.1.3 Network slicing and 5G Stand-Alone architecture

As we stated in the paragraphs before, 5G system will be a network of functions and services. To this purpose, in Release 15, two 5G options for Standalone CN architectures are presented as available:

1. Traditional reference points architecture;
2. Service-Based Architecture (SBA), in which control plane functions can access directly, through an interface called Service Based Interface (SBI), to services provided by other functions.

ETSI indicates this second one as the [48] most promising, especially in sight of MEC deployment, which is our interest and will be introduced in the following paragraphs.
The main Network Functions supporting it are:

- Authentication Server Function (AUSF): authentication,
- Access and Mobility Management Function (AMF): it is directly interfaced with 5G base station (5GBS) and is the termination point of NAS messages (N1) and RAN CP (N2). It is main devoted to registration, connection and mobility management. There exists one AMF for each UE.
- Data Network (DN): it consists in Internet and services of the external third party providers.
- Session Management Function (SMF): it divides, with AMF, the functionalities performed in LTE by MME, managing sessions and providing the UE, at the attachment phase, of an IP address. It has direct interface towards the 5G Gateway, that performs UPF. It also interfaces with PCF.
- Network Exposure Function (NEF): is the function in charge to exchange information between other network functions, and allow to access network services, for example MEC functional entities.

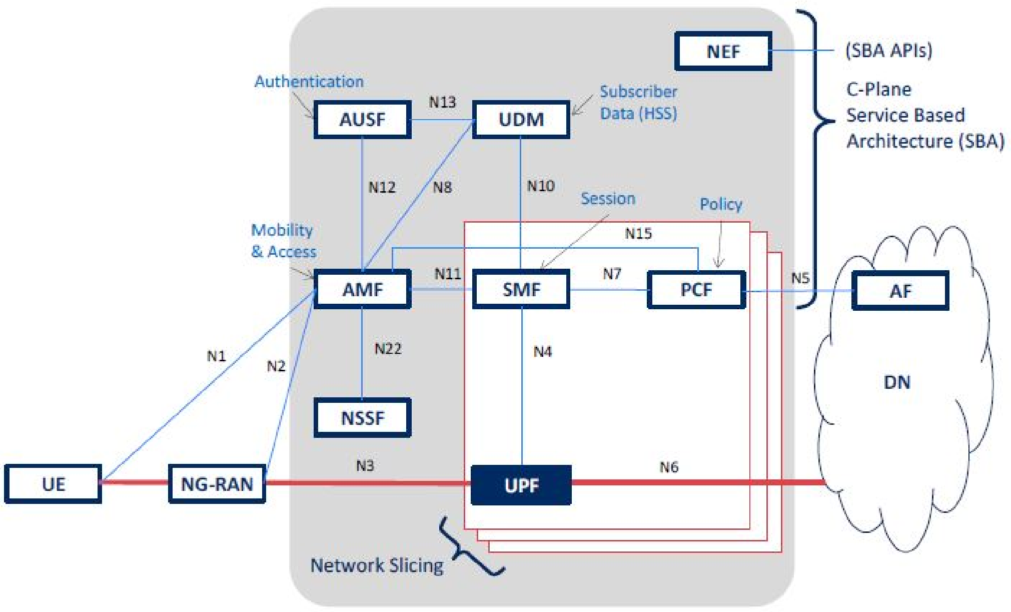


Figure 3.2: 5G Core Network (5GCN) architecture. Taken from [35]

- Network Repository Function (NRF): it stores context and profile for a specific network function and the services it produces. When more network slices are present, it can be more than one.
- Policy Control Function (PCF): it provides the 5G network with a framework to police traffic.
- Unified Data Management (UDM): it supports many functionalities in user identification and subscription, similarly to what HSS was claimed to do in LTE networks.
- Unified Data Repository (UDR): stores and retrieves information by the UDM, the PCF and NEF.
- User Plane Function (UPF): performs forwarding and routing functions, including access to external networks like Internet or 3rd party services.
- Application Function (AF): provides application services to the 5G CN.
- Security Edge Protection Proxy (SEPP): policing of inter PLMN control plane interfaces.
- Network Data Analytics Function (NWDAF): it provides to a network function status analytic info.

In 5G, the most granular network slice information is contained in the S-NSSAI (Single Network Slice Selection Assistance Information), composed by a mandatory field, called Service Type, namely the requirements the user expects to be fulfilled. Many S-NSSAI form a NSSAI, in other word the set of requirements a UE expects to have satisfied.
Whenever the UE accesses the network and is authenticated, the AMF, which takes in charge that UE, on the basis of the S-NSSAIs, contained in the NSSAI, can address the UE to an AMF that can provide the service tailored to that user. In Release 15, each UE is indicated to support at most 8 Service Type in the S-NSSAI identifier, but at the current time just three of them have been defined. Although this Thesis concerns vehicular communication, we will show an overview of all of them and then we will concentrate more on the class of services we are more interested in:

- Massive Machine Type Communication (mMTC). It is a network slice realized for the very massive Internet of Things (IoT) traffic that, for the forthcoming years, is going to overcome the number of smartphones connected to Internet. In fact, IoT devices, connected to cellular network, are foreseen to be 3.5 billion in 2023, according to a report by Ericcson [32]. Such type of service is generally characterized by a huge number of connected devices per km^2, like sensors for smart home or environmental analysis, with lower latency, data rate and mobility requirements. Because of these points, core network resources supporting it will be likely located outside the access network, in a centralized location.

- Evolved Mobile Broadband (eMBB or extreme Mobile Broadband xMMB). This is the most natural evolution of current cellular systems, since represents an enhancement of the current mobile cellular technology. It has already been standardized in Release 15. eMMB has been thought to enhance the data rate and support a much denser environment, increasing user experience, even in very crowded scenarios, like within a stadium or during a concert, giving, for instance the capacity of 10 Tbps to stream video of the event to 30000 devices at 50 Mbps [49].

- Ultra-reliable Low latency Communication (uRLLC). This service slice requires very low latency and bandwidth efficiency. Physical distance between network subscribers and network resources are the basic conditions in order to offer good service. Core network devices need to be placed, in fact, very close to the UE, up to 100 km between eNB and PGW, in order to serve it in the proper way and to guarantee the user continuity. uRLLC has been designed to support very sensitive and fundamental services, like connected vehicle communication, which is now called evolved V2X in new 3GPP Releases. The URLL requires applications which are placed very close to the users.

Many services need to be supported, tight requirements need to be reached, and seems just a technology paradigm can help to fulfil many of them. It is called Multi-access Edge Computing, as we will discover in the next section.

3.2 Multi-access Edge Computing for uRLLC services in 5G netwoks

One of the applications that will likely find concrete realization in 5G network is MEC. ETSI has devoted an Industry Specification Group (ISG), in 2014, to the standardization of this concept, originally called Mobile Edge Computing, because it was aimed to extended Edge Computing paradigm to mobile networks, but it was renamed some years later Multi-access Edge Computing, since the interest of the Group was extended to many types of access networks, including Wi-Fi. ITG focuses on providing an open environment across a cloud environments from multiple vendors and that could be accessible from by third party applications or services.
Main goal of this peculiar architecture is to exploit Cloud Computing features to run application services at the edge of the network, on the top of a virtualized infrastructure, very close to user and subscribers, to achieve latency reduction, high bandwidth, efficient spectrum utilization and reducing the amount of traffic exchanged between the RAN and the core network, hence saving bandwidth.
This is one of the main focus in the Thesis, we have chosen therefore to dedicate more attention and time to detail it. The framework is presented here [50]. Entities of MEC are virtual and categorized in: system level, host level and network level entities.

3.2.1 MEC Architecture

We can imagine the MEC architecture divided into ME management entity and one or more ME hosts.

ME host

It is the component containing a ME Platform. It is the unit that offers ME applications to UEs, instancing them above a set of compute, network and storage resources called Virtualization Infrastructure. Applications are provisioned to the access network user, namely UE applications or 3rd party players (enterprises) through Costumer Facing Service portal (CFS), on the basis of requests done to the ME management. Typically, these functions, run on VMs, strong of the isolation guaranteed by virtualization, and offer specific network services.

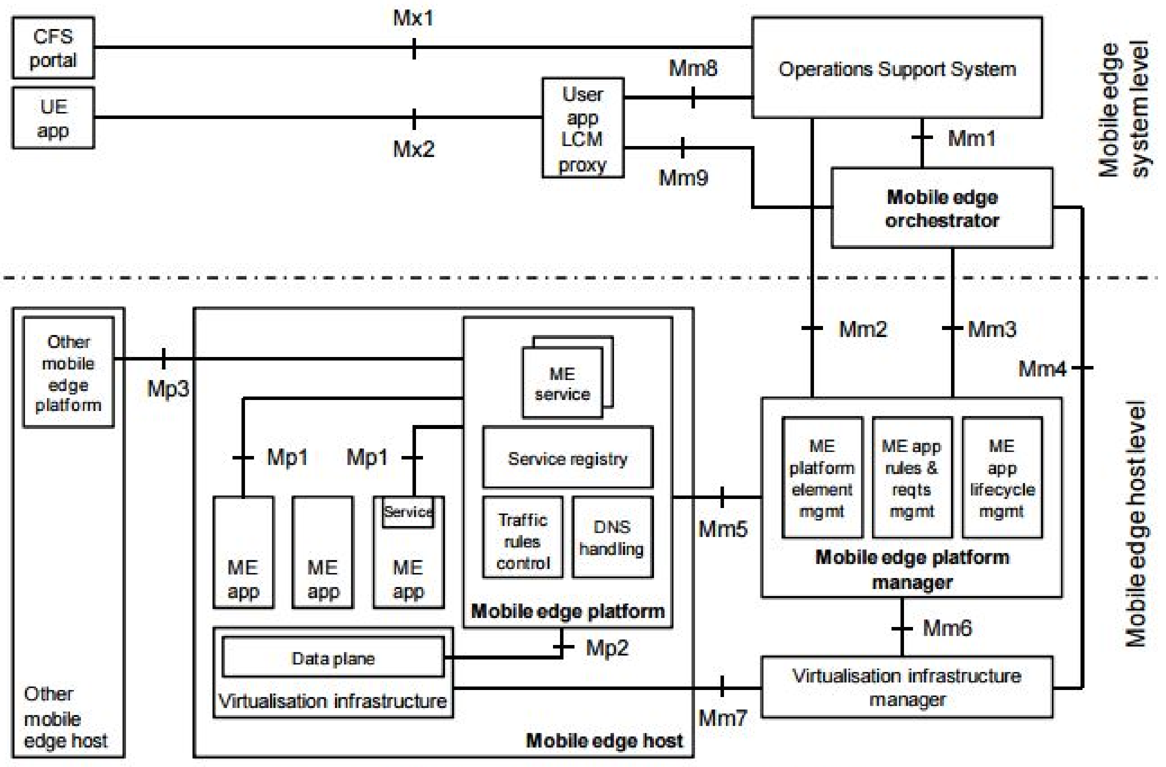


Figure 6-1: Mobile edge system reference architecture

Figure 3.3: MEC architecture. Taken from [50]

ME Platform includes also a service registry, used to take trace of the services enabled on ME applications, it receives traffic rules issued by the ME management and configures DNS proxy and server, according to received records form the management part. Each ME host can be connected to one or more ME host blocks, depending on the type of architecture and type of services involved.

ME management

It is divided into two logical components, the ME host level management and the ME system level management. The *system level* interfaces with access network and the ME host level management. It is constituted by:

- ME Orchestrator (MEO): it is the brain of the MEC, since it possesses the complete view of the architecture topology, selects the most appropriate ME host on the basis of the user requirements, directs the application life-cycle (instantiation, termination or relocation). It performs system level tasks thanks to the OSS and User application life-cycle management proxy.

- Operation Support System (OSS): receives instantiation, termination of relocation requests from access network applications and decides whether to accept or not them, relying on the orchestrator for further processing, even if it has accepted them.
- User application life-cycle management proxy: it is a function which can be accessed just from the mobile network, and cooperates together with the OSS to manage life-cycle of UE applications.

The *host level* directly interfaces with ME host and ME system level management entities. It consists in:

- ME Platform manager: performs full management of functions performed by the ME platform, including spreading important information to the orchestrator concerning the ME platform, overviewing the life-cycle of ME applications, issuing rules, configuring DNS, authorizing services and managing ME functions.
- Virtualization Infrastructure Manager (VIM): deals with the life-cycle of resources including allocation, management and release of resources for ME applications, preparing the virtual environment to host a VM and eventually managing Cloudlet instantiation [53].

MEC architecture reference points are divided into Mp, the set of interfaces concerning the ME Platform functionality, the Mm interfaces, related to the management functionality and Mx, towards external entities. MEC reference architecture presents many similarities with new SBA (Service Based Architecture) 5G, presented in the previous paragraph, network architecture and ETSI highlights these analogies in [48]. It is the part of the system which interfaces with the access network, in our case a mobile network.

3.2.2 Application life-cycle management

The procedures for life-cycle management are included in [51] ETSI specification. Such operations involve the exchange of information between the OSS, real dividing line between the access network and the MEC entities, principally directed by the orchestrator, about on-boarding, enabling, querying or deleting application packages, namely the set of files, received from an application provider exploited by the ME system to instantiate ME Applications.
The service instantiation is very important phase for service provisioning to the users. It is triggered by the OSS upon requests received from a UE application or from CFS; then the instantiation of such an application, for instance a Vulnerable Road User application, is granted and forwarded to the MEO. The ME host is selected by the MEO and corresponding ME Platform Manager, which issues the

resource allocation to the VIM. After resources are allocated, VM tailored to that service is loaded with the image application. The VIM sends back response of successful resource allocation to the ME Platform Manager, that communicates at its own time the positive outcome of the procedure. After that, the ME Platform Manager requests to the ME Platform configuration, inserting DNS rules and traffic rules and service-related information. The difference between traffic and DNS rules is the following:

- Traffic rules are filtering and routing operations that redirect the data traffic to a desired MEC application or towards external networks,
- DNS rules are dictated from the ME Management to the ME Platform, that consequently configures the mapping between IP address into DNS, based on the received rules.

Traffic and DNS rules are configured and, when the application jumps into running state, rules are activated. From now on, traffic flowing into the MEC is going to be re-routed into the VM, providing the required function. Finally, ME Platform confirms the response for configuration completed to the ME Platform Manager, that sends it to the orchestrator. Finally, orchestrator gets in contact again with OSS and returns result of instantiation, together, if successful, with an instance ID. The termination procedure is symmetrical to the previous one. In fact, it begins with the OSS, that waits for the MEO, to get in contact with ME Platform Management, VIM and ME Platform, in order to deallocate the resources and return positively to the OSS. Influential interfaces in life-cycle application management are the Application Life-cycle Management Interface, that makes possible the management information change between the OSS, the MEO and the ME Platform Manager, and the Application Life-cycle Change Notification interface, for the correct notification process that leads to modifications in the life-cycle process.

3.2.3 ME services

After a service has been instantiated, the Mp1 interface, which we have already referred to as belonging to the reference points for management purposes, and connecting ME Platform and ME applications, plays a fundamental role in the interaction between the two entities, as explained in [52]. A ME application, after resources are allocated for it and the instantiation phase is terminated, sends a message to inform the Platform that it is currently up and running. ME Platform, by its own side, configures it with parameters describing the application's access profile (for instance: what are the required information to access the service). Through the Mp1 interface, ME Platform returns activation, deactivation or update state for the DNS and traffic rules, that are requested by the application. Then the ME Platform registers the application and all its context, included services

supported by the application itself in the ME service registry.
Each application, in MEC, is in fact set-up in order to provide one or more services that can be mandatory or optional. A service may be offered, used, advertised to other applications, or discovered by a single ME App.
Finally, when the ME Platform gets a request for the termination of an application, it removes all the application profile stored in the starting phase, including rules and services, whose non-availability to applications that consumed that services, are sent in for of Availability notification. The Platform asks the ME application to terminate, eventually starting from a timer set by the ME Platform itself. Exploring more in particular Service Availability messages, ETSI describes two types of them:

- New service available: firstly the ME application selects, through a query to the Platform, a transport, which is irrevocable if the application intends to distribute the service to other applications that consume it. The message is sent to the ME Platform to inform it that a new service, offered by the sending application, is available, and can be added to the service registry, the memory location in which the ME Platforms stores current available services;
- Update for an already existing service: this message is sent to the ME application to inform it that a service, already present in the service registry, has changed in some of its part, and thus the service registry shall be updated.

Both these two messages are followed by a Service Availability notification to other ME applications, requiring that service, namely those application which have subscribed a Service Availability Notification service to the ME Platform. An application can eventually asks for availability of a specific service by performing an apposite Query to the ME Platform.
Synchronization an timing is possible thanks to the Time of Day (TOD), information that is shared by the ME Platform with all the ME App requiring it and allow common reference time.
All the resources we have discussed before, like traffic and DNS rules, services, timing and transport, are reachable from all the entities exposed to the Mp1 through Application Programming Interfaces (API). They can be retrieved, deleted or updated through the HTTPS protocol. All of them can be reached from a root folder, like the one showed in the figure below. For instance, whenever a ME Application wants to retrieve a DNS rule, it sends an HTTP GET to the ME Platform, and receives a response that can be either positive (200 OK) or negative (400 Bad Request, 404 Not Found, 403, Forbidden). API MEC concept will be resumed in the next paragraph.

3.2.4 MEC APIs

An Application Programming Interface (API) is a set of tools, subroutine definitions and communications protocols that are used to build software. Representational

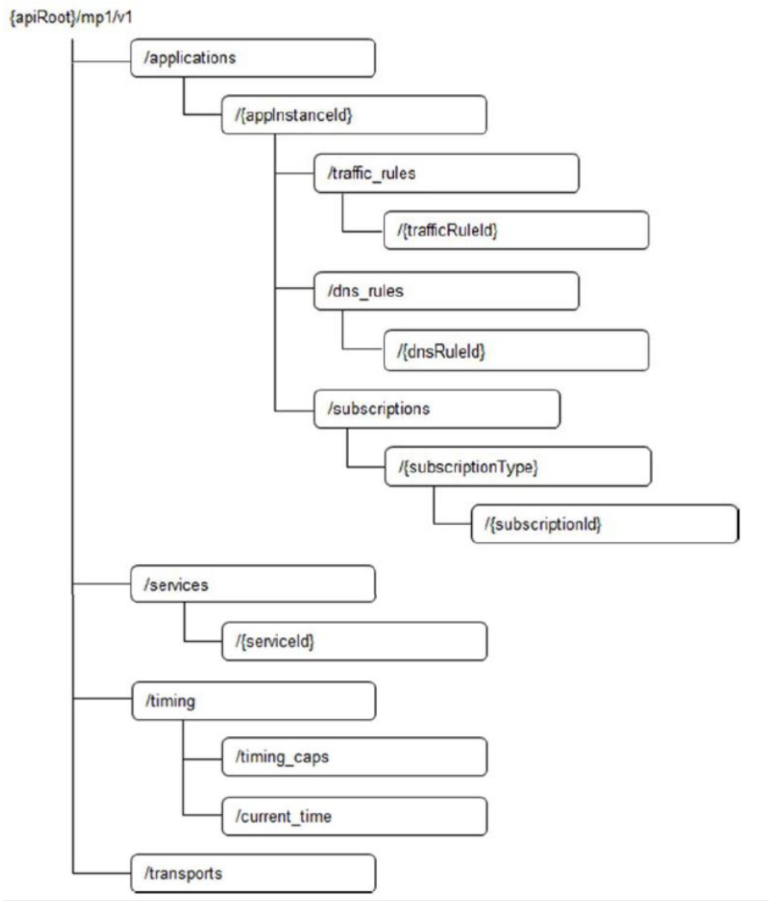


Figure 3.4: MEC API. Taken from [52]

State Transfer (REST)ful APIs are APIs that are used to create web services. Particularly suggested is the HTTP protocol, therefore, all the calls to create, delete or update a service can be done through HTTP and the most important are the following:

- GET: call for requesting resource information, with no intent to modify it,
- POST: call to create a new resource in a collection of resources,
- PUT: call for modifying or update an already existing resource,
- DELETE: call to remove a resource.

These APIs can be used for many different services operations in the MEC context. In particular there exists four types of APIs that are useful to create, delete, modify or notify services. Information are exchanged between ME applications and

ME Platform. **Radio Network Information (RNI) API** [55] are used to access updated information related to the radio access condition, either from the applications and the ME Platform. The aim is to optimize the current services running on the applications consuming these services.
We may think, for example, to applications that require to get how much bit rate (or quality) allocating for a certain content or that have the goal to understand how many users, at the current time, are pinged off the cell. The service can ask for are Radio Access Bearer, PLMN and S1 bearer information.
REST APIs offer the possibility to the service consumer to subscribe a notification service for RNI. Service subscription request is done commonly through a POST message, so that the RNI Service (RNSI) can understand to which type of subscription the consumer service wants access to. The subscription could have a duration time-out, established by the RNIS, sent back as response after the requests for subscription and can be also be updated, by means a PUT call request by the consumer service to the RNIS, or cancelled, through a DELETE call. Furthermore, a consumer service can also get notifications about the Radio Access Bearer establishment, changes or release of the bearer itself and notification about the S1 bearer. Finally, there are available information about the UE (UE measurement reports), timing advance or whether the carrier aggregation changes settings.
Location API, described here [55], are used instead to provide geo-referenced information about UEs in the network. Location services, associated with them, exploit the Zonal Presence service, built on basis of OMA specification [54]. Network subscribers privacy is primarily important, in fact, no UE ID information is collected, in order to avoid matching between a specific user and the statistical data for analysis. Types of information, collected by this APIs, are position information related to a single user, generic information belonging to a group of UEs in a specific area. There exists, as already seen for the other APIs, subscription services, to allow the Location Service to report to a consumer service information related to a UE or a set of users. The consumer service can decide to cancel the service notification.
The **UE Identity API** [57] are build upon the concept of UE tag. Whenever a ME application is set up by the Platform, it can provide to this last a UE Identifier, a tag, that the ME Platform maps on the UE, so that, when the rule application phase arrives, the UE tag enables the traffic rule activation associated with it. The tag can be either registered or de-registered by the ME Application to the ME Platform, through specific PUT requests.
Bandwidth Management APIs are related to the allocation of bandwidth to ME applications, in order to save resources and, at the same time, save more disadvantaged applications. More in the deep, as reported in [58], on each ME application, like Content Delivery Network owned by Netflix, to distribute its products, or gaming applications, can run more than one session at the same time and bandwidth

needs for a scaling up or down with variations of requirements. Therefore, Bandwidth Manager Service (BWMS) can be exploited by the applications. Through the RESTful APIs, they can in fact get in contact with the service and register or unregister from/to it. When service is set-up, applications can then interact with it, requesting for more bandwidth allocation.
UE application interface API are the last set of APIs standardized by ETSI [59]. These are in fact used over Mp2 interface, between the UE and the User application LCM proxy, to correctly manage application life-cycle. In fact, when a UE interacts with MEC to receive a service, it shall communicates to it its own context, in order to be registered by the Mobile Edge system and obtain a serving ME application. A UE application can start asking for the list of currently available applications to the LCM proxy. This list is characterized by the number of user applications saved in the proxy and a list of application information. It provides with application-related information like unique identifier (URI), name, provider and description and technical, like maximum amount of RAM and storage the application is expected to consume, required bandwidth, RTT supported by the ME system and whether the service continuity must be supported or not for the specific case. Some of these data coincide with the UE application context that is sent by a UE application, whenever it wants to join to an already existing user application or create a new one.

3.2.5 ME applications and Mobility

Mobility is a critical problem to support many services for which MEC has been ideated. In fact, dealing especially with mobile networks, an user that moves from a cell into another one is a common scenario. At this point we have to distinguish between the concept of cell and the concept of service area:

- Mobile network cell: the basic area unit of a mobile cellular network, covered by a single base station, in LTE by an eNB
- ME Service area: the area covered by a ME host. It could or not corresponds to the mobile network cell

This arises handover problems related to the transfer of contexts, information related to some applications, deleting or updating application instances (Virtual Machines providing services to one or more users) and services. The [60] is aimed at describing it in the details. In this paragraph we will limit to give hints in mobility since our tested concerns a static scenario, in which mobility and UE context and instance transfer is not considered. Let's introduce a common scenario in order to better clarify which are the players. A UE is a vehicle moving inside a mobile network. It requires very critical low latency service in order to support either security and Quality of Experience (QoE) to its passengers and driver, therefore, it has opted for a MEC service. When the UE moves out of the coverage of the current MEC host, into another ME service area, depending on the type of application and for each type of application whether the type is stateless or stateful. A stateless application is an application that does not keep in memory anything about the current UE information, whereas stateful application does.

- Dedicated application: it is a ME application instantiated and tailored to a specific user, in our example it could be a Bird's Eye view application in the UE vehicle. In this case, when the application is stateless , the instance must be relocated, when instead it is stateful the UE context and or the instance must be transferred from a source ME host to another destination ME host.

- Shared application: When the application is stateless, the application instance could be relocate, but it is not mandatory, whereas for stateful ones, the UE context and/or the application instace could be delivered to the destination ME host, but the source ME host can decides if it is possible to maintain active ME application up to serve other host. It is the case in which several vehicles from the same car vendor are served by an application that warns them about collision from other vehicles.

From this we can conclude that a ME application is not just an entity that serves one single user, but it can use services to transmit information to many road users. Incisive problems interesting relocation are three: when the relocation is done too early, to late or the chosen ME host is the wrong one. When the point is to serve the connected vehicle, timing results to be the tightest constraint. The steps in user mobility are the following:

- User is detecting to change the bearer, for instance because a change of IP address or thanks to the RNIS.

- If the user is no more in the ME host service are, service relocation management is activated and, depending also on the type of application we have discussed before, it can decide specific relocation phases involved, for example if the full instance must be transferred from a Source (S) ME Host to a Target (T) MEC Host.

- In order to activate new services at the T-ME Host, traffic rules need to be changed in relation with the position of new ME Host.

Prediction algorithms shall be adopted in ME systems, before relocation processes takes place, in order to guarantee high QoE, even when the user is in handover from a service area towards another one. For mission critical applications, the ones that demand the lowest latency, in order to reduce to zero the probability that the chosen MEC is wrong to sustain a service a group of MECs, sharing the user application information, called relocation group, shall be identified, so that, whenever the user moves from a service area into another one, service is guaranteed. In both these cases LS and RNIS are fundamental in the capability of the system to understands when the user's hadover will take place.

3.2.6 MECinNFV

A very recent document, published in 2018 by ETSI [63], breaks the thin boundary between NFV and MEC architectures, providing operators with an overview of MEC architecture related with NFV framework presented firstly by ETSI in 2012, and opening the path towards possible 5G network slicing service integration. The complete name is *MECinNFV* architecture, but we will referred to as MEC for simplicity, and traditional MEC referring to the so-far discussed architecture. The main goal of the aforementioned release, is to present an architecture in which entities described in previous standards are read in NFV key, and then to illustrate the main issues and possible solutions connected to them. In particular, either ME Platform and ME applications becomes VNF, simple software resources that run over a commonly shared physical infrastructure, and the Virtualization Infrastructure turns into a Network Function Virtualization Infrastructure (NFVI). The ME Platform Manager functionalities are splitted into:

- ME Platform Manager NVF (MEPM-V),
- Virtual Network Function Manager (VNFM): which is delegated to LifeCycle Management (LCM) of the ME platform or more LCM of ME App.

The entire Mobile Edge Orchestrator (MEO) is divided into the Mobile Edge Application Orchestrator (MEAO) and Network Fuction Virtualization Orchestrator. This splitting orientation is leaded by the objective of incorporating NFV and MEC; in fact the MEAO uses the functionalities of the NFVO to allocate resources for ME applications and manage them as a group of NFV Network Services (NFV NS). The ME Platform is redefined as a NVF and basically it interacts with two units, already existing in the traditional MEC infrastructure, now virtualized: the Network Function Virtualization Infrastructure (NFVI) and the related manager entity Virtualization Infrastructure Manager (VIM).
In addition to Mp, Mx and Mm interfaces, there are present reference points derived from NFV infrastructure [64] and reference points created ad hoc for the integration process called Mv are used to allow permit interaction between MEAO and NFVO (Mv1), MEPM-V and VNFM (Mv2) aimed at LCM of ME applications, that are connected to it with another one (Mv3).
Differently with respect to traditional MEC, the management Mm3 interface is redefined with a subset of functionalities performed by traditional Mm3, since the traditional MEO is composed by MEAN and NFVO.
Even in the traditional architecture, data plane is a critical and largely discussed

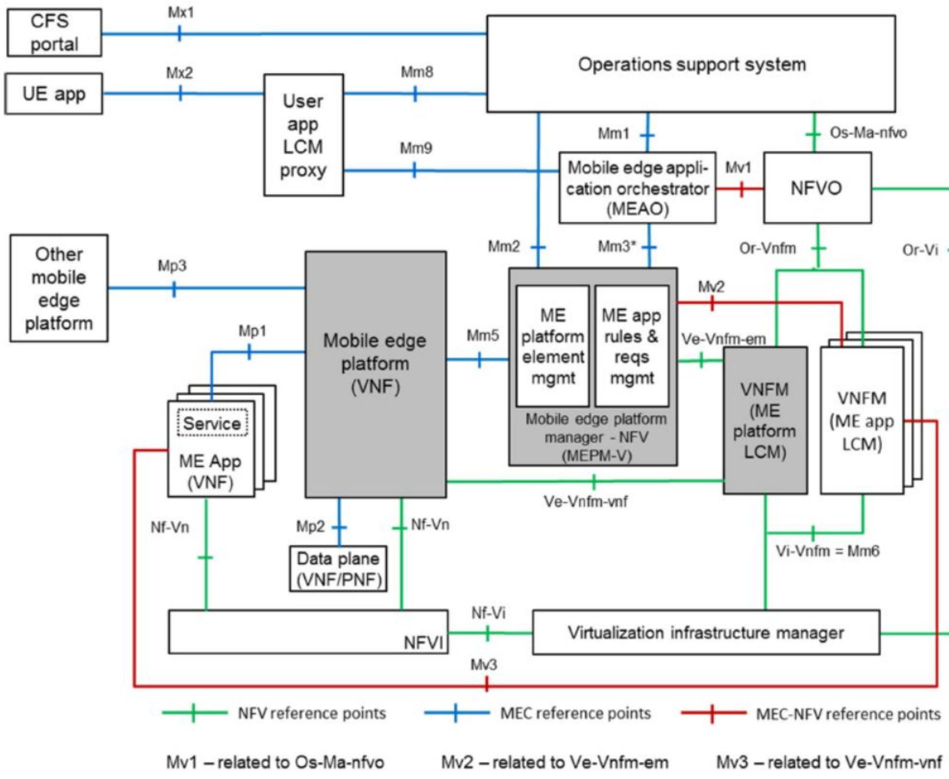


Figure 3.5: MECinNFV architecture. Taken from [63]

problem, since interests how to deliver traffic to correct application instantiated in the ME host. Adding NFV capabilities to the system, data plane could be simply be a mix between virtual and physical infrastructure, communicating with the ME Platform VNF, in a very transparent way with respect to the previous design, or rethought in the sense that the ME Platform doesn't control directly user data traffic, but it waits for instruction from MEAO and NFVO, that instructs the VIM about traffic rules.
The document discussed before is an introduction to a series of issues, possible solutions, that will be discussed further and in the deeper, like the mapping between new ME Application VNF to Network Services (NS), the utilization of a NS and the communication exchanged by functional units like the new functional repartition MEAO/NFVO or MEPM-V/VNFM and ME Application Instance Relocation, but they will not be discussed in this seat. What is very important to remember is that NFV/SDN paradigms are disrupting the way in which we conceive the network and that the passage from a network of entities to a network of functions is interesting many players and standardization companions. MEC and 5G are becoming more and more close each other.

3.2.7 MEC for connected vehicle

MEC virtual functions seem very appropriate to support vehicular communication, that requires uRLL services. In [61], ETSI has announced a series of standard group of services that can be supported by MEC entities.
ETSI underlines the central role that mobile network connectivity plays for V2X communication. Generally speaking, in a vehicular environment, MEC servers, colocated with the base stations, close to vehicles, receive messages that they send to them and provide very fast response to user requests. Service that can be served in a vehicular scenario vary from the intersection collision avoidance, to parking applications to video streaming for car passengers.
It is very interesting the new business models which can arise for automotive ecosystem that sees involved Car makers, Cloud Service Provider and Mobile Network Operators, to take echonomical advantages from their reciprocal support. And therefore, to more traditional business models like B2B (Business to Business) and B2C (Business to Consumer) in which these three stakeholders earn money from the user or the other stakeholders, are associated B2B2C (Business to Business to Consumer), in which Operators benefit from hosting network infrastructure, Application provders from the final users and Car Makers monetize the users that sell their cars; or C2C (Consumer to Consumer) models, where passengers or drivers share their road information with other road users [62].
Once the overall MEC architecture has been defined in ETSI, future works of the standardization group are focused on defining the characteristics that MEC services should have in order to support V2X communication. Four main use cases have been introduced in [61]

- Convenience: it includes car software updates and telematics features;
- Advanced Driving assistance: services related to augmented driving experience systems like see-thorugh systems, in which each vehicle in a platoon can share ahead camera vision with vehicle behind;
- Vulnerable Road users: exploits the information shared by cyclists and pedestrians smartphones and tablets, to make the road environment safer;
- Safety: it is a use case feature related to security in road environment, exploiting V2I communication. In the document, ETSI propose the Intersection Movement Assist, in which vehicles approach to an intersection and can avoid accidents, thanks to collision warning messages instructed by the infrastructure. This represents our final destination, the aim of this Thesis, as it will be discussed in the next Chapter.

5G KPIs announced for years a network with very low latency, between 1 to 10 ms, 99.999 reliability, 99.999 availability, a user rate up to 20 Gbps, very high number of cells, 10ns for km^2 and a very high number of devices connected (up to 100 000

km^2). These KPIs depend on the specific service to provide and need to be tailored to the specific services supported. Nowadays, MEC paradigm proposes as a viable solution to many issues concerning timing requirements and high throughput.
Will 5G be able to satisfy the expectations, integrate with MEC and win this challenge? We believe that it can, and the project of this Thesis is one of the answer to this question, as the reader can explore in the next Chapter.

Chapter 4

Cooperative applications for CAM and DENM generation

Cooperation is a redundant concept for what concerns ITS deployment towards 5G. Release 15, published in 2017, has supported the trend in developing enhanced and cooperative V2X use cases, supported by new generation of mobile communication technologies. This new connected vehicles technology, defined evolved V2X (eV2X) or 5G V2X, has introduced new use cases in [65] to those described by LTE-V2X. What is a cooperative system? C-ITS represent a consistent evolution with respect to the ITS of the past. Traditional ITS are systems in which the decision-making intelligence is placed directly within vehicles or road infrastructures, with consequent limitations in terms of computational capabilities and energy consumption. Cooperative ITS point of success, instead, is the communication between road entities, in other worlds vehicles, network infrastructure and pedestrians. Cooperative systems are, for instance, vehicle platooning, in which a group of vehicles, arranged in a train configuration, exchange and share information (like position, map, visual), or Cooperative collision avoidance, in which a group of vehicles exchanges relevant cooperative awareness messages in order to notify the others about their position and, possibly, avoid car accidents. Such use cases are studied by important standardization entities and research groups like ETSI and 3GPP.
Besides, MEC and Network slicing concepts are going to assume an important role in cooperative vehicle services provisioning, initially supported by LTE-A pro, that is playing a fundamental role as a bridge technology supporting 5G in the early deployment, guaranteeing the coverage since 5G frequency spectrum is not ready yet.
During Mobile World Congress 2014 [66], was officially announced an initiative, named 5G Infrastructure Public Private Partnership (5GPPP), born by the willing of European Commission. It grew thanks to the collaboration of a wide range of stakeholders from Vertical industries and academies, to ensure the leading role of

Europe in the Telecommunication market. The Phase 2 of 5GPPP has incorporated 21 projects. 5G-Transformer (*aka* 5G-T) distinguished for the range of partners involved, coming from the different Verticals, and the number of topics dealt.

4.1 5G-Transformer

5G-T objective, presented in the *Deliverable* available here [67], is to transform the mobile transport network available nowadays into a distributed SDN/NFV-based 5G Mobile Transport and Computing Platform (MTP), capable to support Verticals needs to meet the requirements imposed by the market, in order to deliver their services.
5G-T is the promoter of new technologies, like Network Slicing and Multi-access Edge Computing, largely discussed in the previous chapters, that are essential means to achieve those objectives.
The general architecture foreseen by the project consists in three main entities that share similarities with 5G and network slices:

- Vertical Slicer(VS): is the entity in charge of managing the requests that Verticals made to the infrastructure, to provide access to networking and computing resources. High level requests enters the systems in the form of very simple blueprints or templates, and, after that, that are mapped by the VS into a set of Virtual Network Functions, tailored to the required service;

- Service Orchestrator (SO): is the brain of the 5G-T system, an entity which coordinates the physical resources utilization and allocation to slices requested by different Verticals;

- Mobile Transport and Computing Platform (MTP): this entity has control on the physical mobile transport network architecture, integrating MEC resources coming from multiple domains and supporting network slicing concept.

Why is Automotive Vertical so much invested by the revolution in connectivity? Automotive industry is interested by many evolution trends, that require fast research and development. Driving automation, road safety, infotainment and traffic efficiencies are the main use cases families that require immediate attention.
Centro Ricerche FIAT, in Orbassano (TO), is working together with Politecnico di Torino and Sant'Anna School of Advanced Studies in Pisa (PI), at a delicate use case of 5G-T called Intersection Collision Avoidance (ICA). Intersection is the convergence, in the same point, between two or more streets segments. Intersections represent a critical situation by the traffic point of view and, many times, sources of potential accidents, even if they are controlled by traffic lights, because of drivers negligence and poor visibility conditions. Here comes into play ICA. ICA is a service that a car maker, for instance FCA, may want to add over its proprietary

vehicles. A driver, that is using a branded vehicle, has the opportunity by his own side to enable or disable the service. It is based on cooperation and communication between vehicles of mobility-related data, in proximity of an intersection, where an entity is in charge of:

- computing real-time the collision probability of two vehicles,
- taking a decision on the basis of exchanged road users information sending alerts messages to interested vehicles and eventually activating the emergency braking system.

The KPIs for this user case are the following table:

KPI	**Requirement**
Reliability and availability	20 ms
Ultra-Low Latency	< 20 ms
Security level	high
Priority level	high

4.2 Description of the testbed

ICA module is going to be ready in the early 2019 and shall be built upon very strong milestones. In fact, core part of the system, namely communication and cooperation between vehicles and network infrastructure and testing of the communication itself on a real network testbed, is central topic of this Thesis.
Communication takes place between two C-V2X applications, that the candidate has developed during six months at Centro Ricerche FIAT and that exchange two types of basic safety messages: ETSI CAM and ETSI DENM. The applications have been called `OAICodeCar`, from now on called `oaicc`, and `OAICodeCAlert`, which we will call `oaica`.

- `oaicc`, behaving as an application running on the on board computer of a vehicle, is in charge of transmitting CAM and receiving CAM and DENM. The point of strength of this application, is the possibility to extract mobility-related values from two different sources: real vehicle information sources, the CAN bus and the GPS receiver, and simulated information sources, a file containing data extracted by SUMO simulator traces. This central feature make this application flexible and adaptive to the context in which it is used.
- `oaica`, behaving as the road infrastructure, ideally located within a RSU, receives CAM and generates DENM.

Both these applications are written in C++ programming language and made-up to run on a Linux-based system.

One of the most interesting and innovative feature of the ICA service is that 5G vehicles supporting it shall be equipped with 5G modem in the final demo, meaning that in the final implementation of the project a full working 5G architecture will support connectivity.
5G commercially, as we discussed before, is not ready yet. Fortunately, in Europe, there is great turmoil in creating open-access software, interoperable with closed-access commercial wireless systems components and devices, in order to allow easy development and testing of newest network features and encouraging development and research projects.

4.2.1 OpenAirInterface: open-source software for 5G development

The research center EURECOM [68] has launched, in 2009, the OpenAirInterface (OAI) Software Alliance, in charge of developing a software that were a standard-compliant implementation of 3GPP mobile network components. Such powerful tools has been called OAI. OAI [69] is a Linux-based (Intel x86 PC architectures) software package, that provides an implementation of a subset of Release 10 of 3GPP, namely LTE-A. Network functions supported by the package are the UE, the eNB, MME, HSS, S-GW and P-GW, namely many of the entities we have described talking about LTE architecture.
The framework, which consists in a set of Linux packages that can be downloaded from the main site and installed on one or more machines, can be used together with Radio Frequency equipment (like Ettus USRP) and RF platforms customized by EURECOM.

4.2.2 The testbed

The testbed, set-up at Politecnico di Torino, consists in:

- 3x PC Notebook (DELL), equipped with Intel i7 processors, Linux v14.04 OS: on each of them run respectively a UE, an eNB, and EPC plus MEC platform;
- 2x USRP B210 board + 2x Ettus enclosure kit: these two components have been assembled together forming a complete Software Defined Radio device for transmission and reception of messages;
- 4x SMA-SMA cables: connecting the two USRP board described below, in order to build up a stable communication channel.

The specifications of the machines used for the OAI network testbed are collected in Table below.

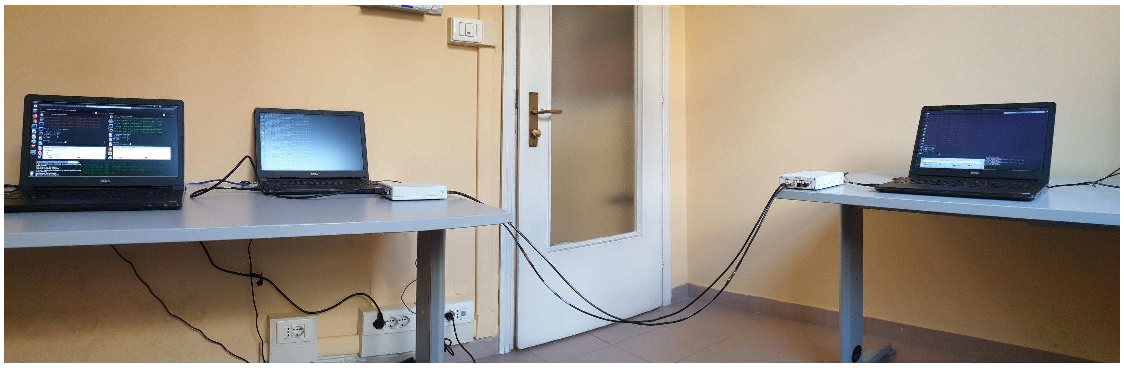
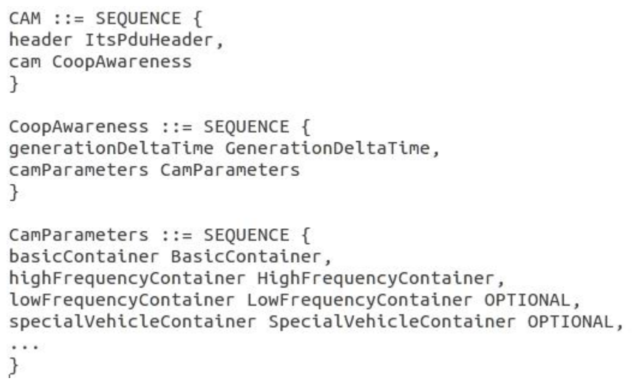


Figure 4.1: OAI testbed at Politecnico di Torino. From the left: EPC+MEC PC, eNB PC, UE PC

Hardware feature	**Requirement**
OS	Ubuntu 16.04 LTS
Memory	15.5 GB
Processor	Intel Core i7-8550U CPU @ 1.80GHzx8
Graphics	Intel Kabylake GT1.5
OS type	64-bit
Disk	967.6 GB

4.3 `asn1c` compiler

As reported here, ASN.1 (Abstract Sintax Notation One) is a pseudo-language that allows to describe complex data structures, independently of the programming language in which they need to be developed. The `asn1c` [75] is an open-access compiler, which takes in input one or more files written in asn1 notation (for instance, `example.asn1`). It generates a support code for serialization and de-serialization of these data structures, according to the most used encoding rules: BER (Basic Encoding Rules), CER (Canonical Encoding Rules), DER (Distinguished Encoding Rules), PER (Packed Encoding Rules) and XER (CML Encoding Rules).
CAM and DENM messages have been described by ETSI in [16] and [17], using such notation (see Figure 4.2). According to the specification, they must be encoded and decoded making use of Unaligned Packet Encoding Rule (UPER), one of the two variants of PER, which foresees no padding bit inserted between fields implying a saving in bit value for encoding.
On the website of `ans1c`, there is a guide explaining how to use it. In this Thesis, the candidate provides here the set of instructions and procedures used to carry out files and ad it to the Codeblocks project, in order to facilitate future usage.



```
CAM ::= SEQUENCE {
header ItsPduHeader,
cam CoopAwareness
}

CoopAwareness ::= SEQUENCE {
generationDeltaTime GenerationDeltaTime,
camParameters CamParameters
}

CamParameters ::= SEQUENCE {
basicContainer BasicContainer,
highFrequencyContainer HighFrequencyContainer,
lowFrequencyContainer LowFrequencyContainer OPTIONAL,
specialVehicleContainer SpecialVehicleContainer OPTIONAL,
...
}
```

Figure 4.2: CAM asn1 notation example

1. create three files with extension *.asn1*, containing DENM, CAM and generic ITS fields descriptions. Since, in fact, CAM and DENM share many data structures, ETSI has sketched them with basic containers and has specified another file including all the common structures. Assume the files are called *CAM.asn1*, *DENM.asn1* and *Its.asn1*

2. Enter Linux CLI, in the folder where the three files have been stored, and type the following commands to test the parser:

   ```
   $ asn1c -EF CAM.asn1 DENM.asn1 Its.asn1
   ```

   If the command returns no errors, we can proceed, otherwise we must open files and correct notation errors suggested by the error output.

3. Then, compiling process can be started with:

   ```
   $ asn1c -pdu=CAM -gen=PER *asn1
   ```

   The option *-gen=PER* provides additional file creation, specifically targeted for UPER coding.

4. Whenever files have been produced, compilation process ends with the following command:

   ```
   $ make -f Makefile.am.sample;
   ```

5. In order to import the project in the code editor (in our case, *Codeblocks*), a library file must be created with command:

   ```
   $ ar rc <libname>.a <file1>.o <file2>.o ..  <filen>.o
   ```

The *.o* files are the object files created after compiling and *¡libname¿.a* is the lib package just created. At this point, the library file is created, and it can be added into the directory of our project. It is very important that also all the header files (*.h*) are copied into the project too.

At this point, it becomes easier manipulating these structures. In `oaicc` and `oaica`, the data structures describing CAM and DENM are two very complex objects of type `CAM_t` and `DENM_t`. They can be filled exploring the structure in the deep, according to the standard, remembering that optional fields structures and variables are define by the compiler with pointers.
For example, taking in analysis `CAM_t`, in order to access speed value, which is a parameter belonging to the BasicVehicleContainerHighFrequency, the data structure is explored in this way:

```
CAMsg->cam.camParameters.highFrequencyContainer.choice.
basicVehicleContainerHIghFrequency.speed
```

`asn1c` provides a set of C functions for encoding and decoding of messages. We will report in the next lines these two functions, since in the compiler guide, available on the Web, is very well explained how to encode and decode a message, printing the results on a file; very few words have been spent instead for buffer transmission and reception. The two functions are:

- `uper_encode_to_new_buffer`: this function can be fed with an empty buffer and a `CAM_t` or `DENM_t` object and returns the serialization buffer size in bytes;

- `uper_decode`: this function accepts a received buffer and its size, returning a structure through which we can understand if decoding process has been successful or not.

4.4 OAICodeCar: a vehicular application

`oaicc` is an application developed in C++11, using the *Codeblocks* Integrated Development Environment (IDE).
The application behaves as a real vehicle and as a traffic simulator, as we will see later on.
In this section, firstly, a development view will present the packages, all the classes created and the correlation among them. After that, a logical view, describing the functionalities provided to the users, is given.
In order to understand the following parts, three concepts should be at least introduced: NMEA 0183, CAN and SUMO. In fact, as we will see, data used to construct CAM messages have been taken from three sources: GPS receiver, CAN

bus or simulation trace:

- NMEA 0183 (National Marine Electronics Association 0183) is a standard [70] that defines electrical signal requirements, data transmission protocol and time, and specifies sentence formats for data communication using Global Positioning System (GPS) system. Protocol is based on the principle that the source, namely *talker*, can just send data strings (called sentences) and receiver, said *listener*, can just receive them. All the sentences, each lasting at least 80 character, present the following structure:

  ```
  $PREFIX,data1,data2,..,data(N-1)
  ```

  The `PREFIX` is always `GP` followed by the sentence type. Sentences we have used in `oaicc` have been: GPRMC, GPGGA, GPGST, GPGSA.

- CAN bus (Controller Area Network). In modern vehicles, about 70 Electronic Control Units (ECU) are present to control audio systems, airbag, cruise control, electric power steering an many others. They need to exchange, in order to guarantee correct functioning of the on board equipment, many information, carried by means of electronic signals. In fact, the majority of the functions, performed by a vehicle nowadays, are supported by ECUs and their signals, travelling across a shared bus called CAN bus. CAN bus [71] is a vehicle bus message-based protocol for communication of micro controllers and devices to communicate with each other in applications without an host computer.
  The CAN specification, ISO 11898 [72] defines CAN network into three different layers, object, transfer and PHY layer. The PHY layers are two:

  – High Speed CAN: offers Baud rates from 40 Kbitps to 1 Mbps.
  – Low Speed CAN: offers Baud rates from 40 Kbitps to 125 Kbitps.

  A very powerful tool, used to analyse bus communication, is called CANalyzer, developed by Vector Informatik GmbH. This software allows to connect to a vehicle CAN bus and analyse data traffic or even to simulate a vehicle signals flowing within a bus.

- SUMO (Simulator of Urban MObility) [73], is an open-source road traffic simulator, designed to handle large road networks. It allows to simulate how a given traffic demand, consisting in single vehicles, moves through a given road network. It includes all the applications that are useful to perform traffic simulations. Data of the vehicles, which are moving during a simulation, can be saved in files with different extensions. Applications developed in this Thesis interface with files saved with *.csv* format.

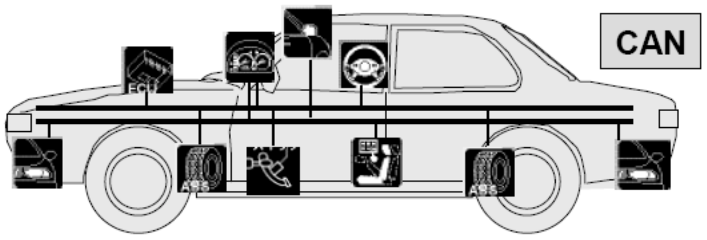


Figure 4.3: CAN bus

4.4.1 Development view

The UML diagram of `oaicc` and `oaica` is shown in figure 4.4, since the two application has several classes in common.

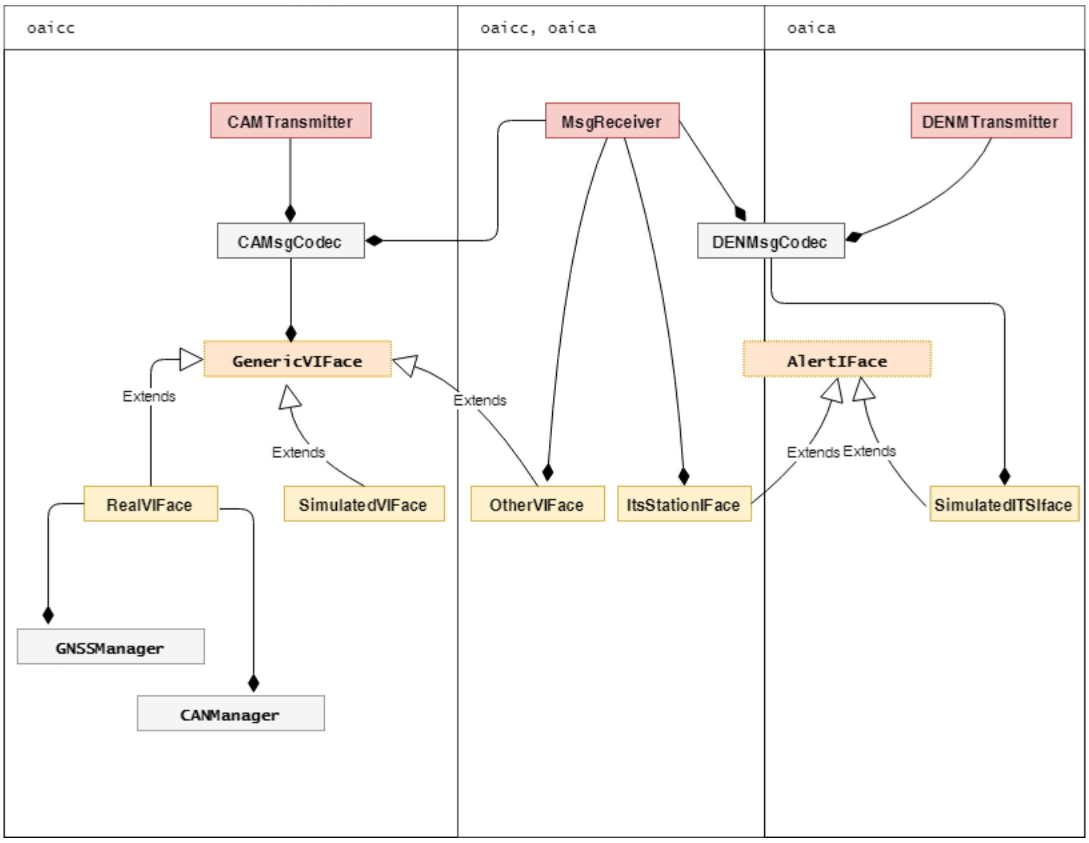


Figure 4.4: UML `oaicc` and `oaica` diagram.
In *red* are shown the top-level classes, in *orange* the two generic interfaces, in *yellow* all the specialized interfaces, in grey the other classes.

The project consists in twelve classes, namely:

- `GNSSManager`: this class is in charge of interfacing with a GPS receiver and extracting geo-referenced parameters from standard NMEA strings;

- `CANManager`: this class is in charge of interfacing with vehicle CAN buses, to extract information, related to vehicle system status and mobility, from the electronic signals;

- `RealVIFace`: this class is in charge of providing an interface either towards the vehicle GPS receiver (`GNSSManager`) and the CAN buses (`CANManager`), collecting all the information extracted from these sources;

- `SimulatedVIFace`: this class is in charge of providing an interface with a simulated vehicle trace, extracting the parameters gathered by a pre-compiled `.csv` SUMO file;

- `OtherVIFace`: this class is in charge of extracting data from received CAM messages and deliver them to the class user;

- `GenericVIFace`: this class is a generic vehicle interface, defining all the possible interfaces a vehicle can face with: the RealVIFace (the vehicle itself real data), the SimulatedVIFace (the vehicle simulated data), or OtherVIFace (data coming from other vehicles);

- `ITsStationIFace`: this class is in charge of extracting data from received DENM messages and deliver them to the class user;

- `AlertIFace`: this class is an RSU generic interface, defining interfaces that a vehicle can use to interact with the RSU and that a RSU can use to construct alert messages;

- `CAMsgCodec`: this class is in charge of encoding the information collected either from the real or the simulated interface, in order to convert values in an ETSI CAM compliant format. It provided also methods to encode (or decode) CAM messages;

- `DENMsgCodec`: in `oaicc`, this class is in charge of decoding DENM messages. It includes additional functionalities, related with `oaica`;

- `CAMTransmitter`: this class is in charge of encoding CAM messages, opening a UDP socket and transmitting CAM messages toward a destination IP address. Port and destination IP address are defined by the user and passed to the class constructor;

- `MsgReceiver`: this class is in charge of receiving CAM and DENM messages, listening on a specific UDP port. Port is defined by the user and passed to the class constructor;

During the development process, used libraries are the standard C and C++ libraries. In addition, two important packages, proprietary of CRF, have been exploited by the candidate:

1. The former is a package containing a set of useful classes written C++, for the management of memory buffers, signals and event, sockets, mutex and other operating system tools;

2. The latter is a package containing classes written in C++, that can be used to extract signals and correspondent values on the basis of the message FCA map. To describe the properties of the CAN network, in terms of ECU connected, CAN messages and signals, the *.dbc* file format has been adopted. A vehicle might have multiple CAN buses and every CAN bus is be represented by its own *.dbc* file. The *.dbc* files used are proprietary of FCA and due to the confidential information contained no other details will be provided.

In the UML diagrams, shown in the following subsections, classes from these two packages have been marked in *violet*, and invented names have been used to define them.
From the description depicted above, about the transmission of CAM messages, there are delineating two independent approaches for the CAM construction, that will be deepen in the logical view part. In the former, each CAM message is filled with parameters extracted from a real interface, in the latter instead is used a simulated interface.
Before proceeding further, we shall append an important annotation that holds either for CAM and DENM. The frame format of the two messages is well delimited by the data structure containers introduced in 2.1.3; for each container, some of the parameters are mandatory, some are optional. The discussion and planning about what parameters would have been added in CAM and DENM transmitted, has been done jointly with the Politecnico di Torino (for the simulation files) and CRF team (for the CAN bus and GPS vehicle side). Therefore, when in the following sections the candidate will refer to some values, like speed, or acceleration, it will assume the a priori decision to insert in CAM (and DENM) message that specific parameter.

GNSSManager

`GNSSManager` is a subclass class, namely extends, `ThreadManager` (from FCA library) class, and that works using `minmea` library, whose repository is freely available on Git [74]. Its objective is to provide the user with proper *getter* methods of

selected GPS parameters.
`minmea` is a GPS parser library, written in pure C, targeted for extracting GPS parameters from a set of supported NMEA sentences. The sentences we needed were:

- RMC: contains information about position, velocity and time,
- GGA: contains information about fix and altitude value,
- GSA: contains information about precision and active satellites,
- GST: contains pseudo-range noisy statistics.

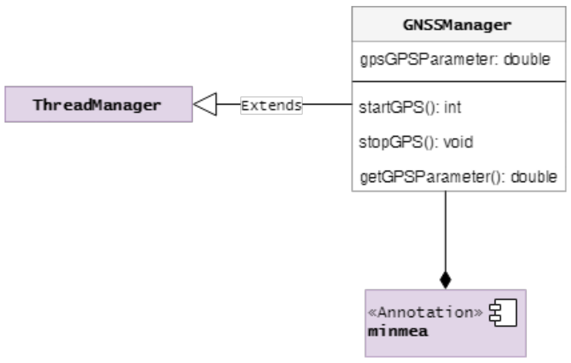


Figure 4.5: `GNSSManager` UML diagram

Note that the diagram, shown in figure 4.5, and all the following depicted in this Chapter, provide a summary view of the class architecture. Therefore, some of the methods have been not reported, to maintain complexity low, and other methods, for instance *getter* methods, have been summarized. For example, with the generic denomination of *getGPSParameter* is meant a set of *getter* methods used to retrieve useful values.
All the sentences needed are supported by `minmea`, therefore we have been able to retrieve GPS parameters reported in the following table:

Parameter	**Measure unit**	**NMEA used**
Latitude	deg	RMC
Longitude	deg	RMC
Altitude	m	GGA
Error ellipse semi-major axis sigma error	m	GST
Error ellipse semi-minor axis sigma error	m	GST
Error ellipse orientation	deg	GST
Heading	deg	RMC

CANManager

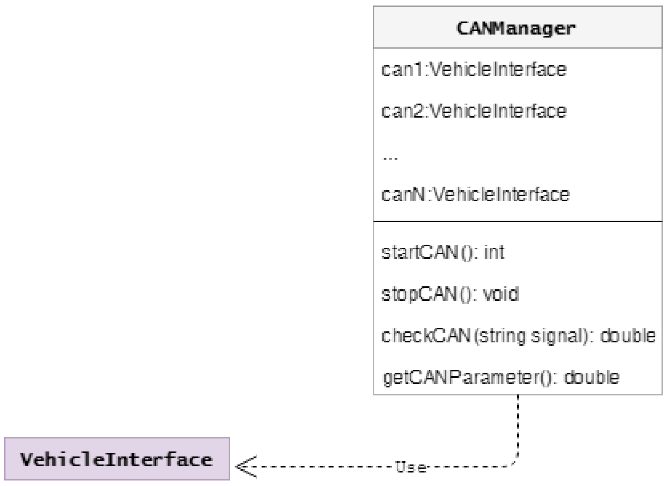


Figure 4.6: `CANManager` UML diagram

`CANManager` objective is to extract CAN parameters from CAN buses, using the CRF libraries.
In order to correctly perform all the operations, `VehicleInterface` objects have been defined. Each of these object is constructed with two components: a *.dbc* file and a network interface to a can bus. These network interfaces are connected to CANAnalyzer in a configuration we will discuss in the next subsections.
Once selected the signals related to significant data for the CAM construction, their names have been specified in a configuration file, according to a notation which allows to determine, in a unambiguous way, CAN signals and originating ECU.
This file allows the `CANManager` to determine, in a unique way, which signals are needed to fill the CAM and guarantees also an easy way to change signals collected, in case additional CAM parameters are required in the future.
The CRF library allows to get the correct signal values and opportune *getter* methods provide to the user an easy way to recall CAN parameters extracted. The values collected by the class, mapped to the CAM fields required, are reviewed in the following table:

Parameter	**Measure unit**
Speed	km/h
Longitudinal Acceleration	m/s^2
Yaw Rate	deg/s
Drive direction	NA
Brake Pedal On/Off	NA
Gas Pedal On/Off	NA
Emergency Brake On/Off	NA
ACC (Adaptive CC) On/Off	NA
CC On/Off	NA
Speed Limiter On/Off	NA
Steering wheel angle On/Off	deg
Lateral acceleration	m/s^2
Vertical acceleration	m/s^2
Low beam lights OnOff	NA
High beam lights OnOff	NA
Left turn light OnOff	NA
Right turn light OnOff	NA
Daytime running lights OnOff	NA
Fog lights OnOff	NA
Parking lights OnOff	NA

GenericVIFace

The `GenericVIFace` is the superclass for three subclasses: `RealVIFace`, `SimulatedVIFace` and `OtherVIFace`, that inherits its methods and states. The `RealVIFace` and `SimulatedVIFace` are used in transmission, the `OtherVIFace` is used in reception of messages.
Inheritance is one of the milestones of object-oriented programming and consists in defining a class, whose methods and states are inherited by a set of classes said children or subclasses. `GenericVIFace` provides `virtual` methods, then defined by its subclasses and logging methods (`dumpToFile`), though which each subclass, by specifying an header and a file name, can fill log files either while is transmitting, or receiving CAM messages.

RealVIFace

`RealVIFace` is subclass of `GenericVIFace`. It makes use of two objects of type `CANManager` and `GNSSManager` to provide the user of the class all the `getter` methods to access CAN bus and GPS parameters we have seen in previous paragraphs.

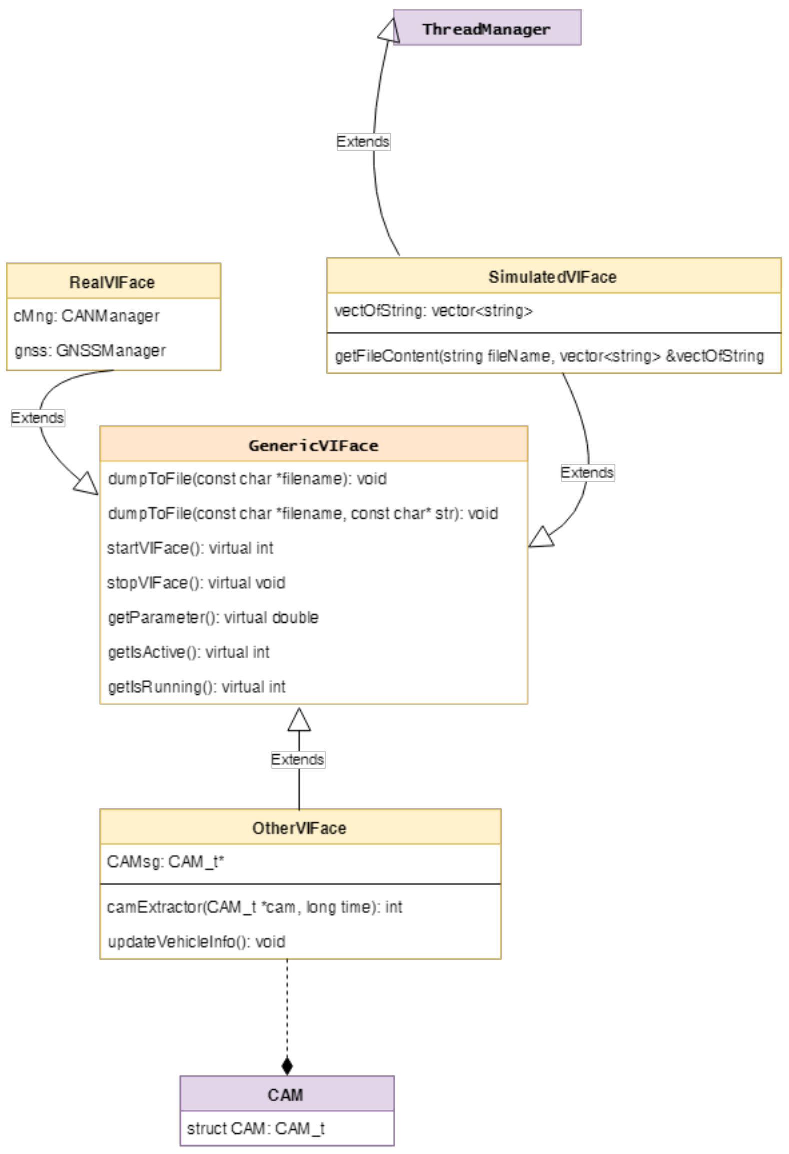

ThreadManager
Extends
RealVIFace
cMng: CANManager
gnss: GNSSManager
SimulatedVIFace
vectOfString: vector<string>
getFileContent(string fileName, vector<string> &vectOfString
Extends
GenericVIFace
dumpToFile(const char *filename): void
dumpToFile(const char *filename, const char* str): void
startVIFace(): virtual int
stopVIFace(): virtual void
getParameter(): virtual double
getIsActive(): virtual int
getIsRunning(): virtual int
Extends
Extends
OtherVIFace
CAMsg: CAM_t*
camExtractor(CAM_t *cam, long time): int
updateVehicleInfo(): void
CAM
struct CAM: CAM_t

Figure 4.7: `GenericVIFace` UML diagram

SimulatedVIFace

`SimulatedVIFace` is subclass of `GenericVIFace`. It is used to extract parameters read by a a *.csv* SUMO file and offer them to the user of the class. The parameters, selected in accordance between the Politecnico di Torino team and CRF are:

- Timestamp,
- Altitude,

- Longitude,
- Altitude,
- Heading,
- Speed,
- Acceleration (longitudinal),
- Yaw Rate,
- Reverse gear On/Off,
- Steering Wheel Angle,
- Acceleration (lateral),
- Acceleration (vertical),
- Acceleration control data (Brake Pedal On/Off, Gas Pedal On/Off, Emergency Brake On/Off, ACC On/Off, Cruise Control On/Off, Speed limiter On/Off),
- Lights related data (Low beam lights On/Off, High beam lights On/Off, Left turn light On/Off, Right turn light On/Off, Daytime running lights On/Off, Parking lights On/Off).

As we can see, these information are almost the same collected by the `GNSSManager` and `CANManager`.
This class gets the lines of a SUMO file, which has been passed to the constructor, and place them within a `string` vector. After doing that, it reads the first timestamp value, available in the first line of the vector, corresponding to the time in which first CAM is transmitted in the simulation, and waits for that period to elapse. Thus, makes the first line of parameters available to the user, namely the `CAMsgCodec` class. Before starting to read the successive line of the file, it waits exactly the difference between the timestamp read from the new line and the previous one. This behaviour is very important, and is exploited by the user of the class, the `CAMsgCodec` and consequently by the class using it, the `CAMTransmitter`, that will be discussed in the next paragraphs.

CAMsgCodec

The `CAMsgCodec` is one of the core classes of the project. Its objective is to fill up a CAM message, namely a `CAM_t` structure, with parameters, opportunely converted in a compliant ETSI CAM format, got from an object of type `GenericVIFace`.
The class provides the user, which we will see are the `CAMTransmitter` and `MsgReceiver`

classes, with methods `encodeCAM` and `decodeCAM`, to perform serialization and deserialization of the message.
Furthermore, the class is defined by two constructors. The former, accepting in input all the parameters needed for a `RealVIFace`, like the network can interfaces, or the *.dbc* files, defines the `GenericVIFace` object as a `RealVIFace`. The latter, accepting as input the SUMO file, defines the `GenericVIFace` object as a `SimulatedVIFace`. The challenging parts, in developing such a class, are essentially two:

- All the parameters, independently from the source of extraction, real or simulated, cannot be inserted in the `CAM_t` object as they are. They need to be opportunely converted into ETSI CAM specification format. To this purpose, the candidate has studied this document [76], that provides a detailed definition of CAM and DENM data structures called *data frames* and *data elements*.

- The `CAM_t` structure, as seen in the section 4.3, is very deep, and its subdata structures are different from the standard C types (`int`, `double` etc.). In fact, they are built up by `asn1c` compiler, on the basis of indications present in ASN1 input files. For particular data structures of the ASN1, like `BIT STRING` or `SEQUENCE OF`, the management has not been easy, in terms of memory allocation and access.

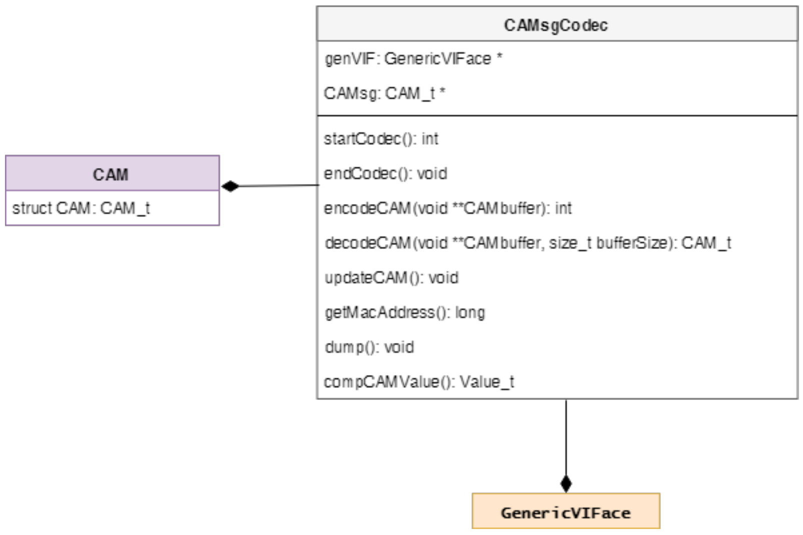


Figure 4.8: `CAMsgCodec` UML diagram

Example
`Acceleration Control` is a `CAM_t` data structure. It corresponds to an ASN1 BIT

STRING, which is a data structure representing an arbitrary string of bits. The possible values assumed by a BIT STRING fields are 0 or 1. Acceleration control, as defined in [76], is composed of seven bit fields, that define whether a specific in-vehicle acceleration control system, like brake pedal or emergency brake, is enabled or not.
In CAM message, `Acceleration Control` is present within the `highFrequencyContainer` and is an object of type `AccelerationControl_t`, consisting in:

Field	**value**	**description**
size	1	number of bytes
buf	buf[0]	the byte buffer
bits_unused	1	number of bits unused for each byte of BIT STRING

After properly defining the data structure, the Acceleration Control bits need to be filled up with the switch statement:

```
CAMsg->cam.camParameters.highFrequencyContainer.
choice.basicVehicleContainerHighFrequency.
accelerationControl->buf[0] |= <0or1> << (7-<positionOfTheField>)
```

Among all the fields, an important role is played by identifiers. The CAM identifiers are:

- StationID: is an INTEGER value, from 0 to 4294967295, identifying and ITS-S. To assign this identifier, we have extracted the MAC address of the `eth0` interface of the Linux machine in which the final application is executed, then converted into an integer number with a simple `atoi` function
- Vehicle Identification: is a SEQUENCE of two number, the World Manufacturer Identifier (WMI) and the Vehicle Descriptor Section (VDS), parts of the Vehicle Identification Number (VIN) code, a unique code used by automotive industry to identify individual motor vehicles [77]. Since either in CAM and DENM we haven't use the VIN field, we will refer in the next parts just to the WMI, calling it `VehicleIdentification`. This identifier is not present natively in ETSI specification, still, we have succeeded to push it into the CAM message structure, and this is one of the results achieved by this Thesis, as we will see in the next Chapter.

The CAM filling process is performed in two phases. At the beginning, when the `startCodec` is executed, all the default parameters, like CAM version, the messageID and all the CAM fields we have decided to set for all the transmissions, are put inside a void CAM structure. Then, each time an `encodeCAM` is executed, the method `updateCAM`, located just before the encoding function, fills `CAM_t` with all the parameters passed to this class by the `GnericVIFace` *getters*.

DENMsgCodec

`DENMsgCodec` is a class shared by `oaicc` and `oaica`. In `oaicc` it is used mainly to decode DENM messages. A more intense utilization is described in `oaica` section.

CAMTransmitter and MsgReceiver: common objectives and functionalities

`CAMTransmitter` and `MsgReceiver` are the objects that manage, more or less directly, almost all the classes seen so far, and, in fact, in the `main` function of the application, they are the two only classes called.
Since they are transmitter and receiver objects, they have some similarities. We will briefly overview them:

- they extend the `ThreadManager` class from the FCA package,
- they use `CAMsgCodec` class,
- they use `SocketManager` class from the FCA package.

These two classes are very important, since they provide, at least in simulation mode, statistical details on the number of packets transmitted and received, with methods as `getCAMCounter` and `getDENMCounter` and also information if some packet has been not transmitted, because of socket errors.
Furthermore they both offer the possibility, if enabled, to dump relevant information about transmitted and received messages writing real-time log files. Summary of log files that can be created, each time `oaicc` runs, and their relative description, have been reviewed in the table below.

Log file name	**description**
vehicle_self_log.csv	this file includes a table that collects relevant information for each transmitted CAM
car$< stationID >$.csv	this file is created for each ITS-S, thus possessing a given stationID, sending CAM to the `MsgReceiver`
alert_log.csv	this file collects relevant information for each received DENM

Finally, in order to compute the Facility Layer Latency (FLL), a value that will be detailed more in the next Chapter, the following methods have been defined:

- `rttCalculation` method is given to both of the classes,
- `dumpAnalytics` at the transmitting class and `dumpRTTFile` at the receiving class are used.

These three methods allow the creation of three log files that are fundamental for the FLL measurement.

CAMTransmitter

`CAMTransmitter` is one of the two top-level classes in the `oaicc` application. Its objective is to instantiate an encoder, by means of `CAMsgCodec` and a client socket, by means of `SocketManager`. While the encoder is running, that, we recall it, is when the `GenericVIFace` object is active, it encodes CAM messages and transmits the returned buffer using a socket composed by a pre-defined IP address and a transmission UDP port. These values are passed by the final user of `oaicc` to the constructor of the class.
It is also involved in dumping of log files related with FLL calculation.

MsgReceiver

`MsgReceiver` is the other top-level class composing `oaicc`. It exploits two different decoders: an object of type `CAMsgCodec`, to decode CAM messages, and a `DENMsgCodec`, to decode DENM messages. It declares also two interfaces, a `GenericVIFace`, specialized into `OtherVIFace`, and a `AlertIFace`, specilized as `ItsStationIFace`, to deal with received messages.

Whenever the receiving process is launched, the class sets up a UDP server, listening on a specific port, and then receives messages arriving on that port. Whenever a message is received, it decodes the header and looks for a particular field called *messageID*. If this value is:

- 2, then the message received is a CAM message. The `OtherVIFace` provides therefore a method to extract CAM parameters from their structure `CAM_t`, in order to return this value to the `GenericVIFace`, which can process and eventually log them;
- 1, then the message received is a DENM message. The `ItsStationIFace` provides a method to extract DENM parameters from their structure `DENM_t`, in order to return this value to the `AlertIFace`, which can process and eventually log them.

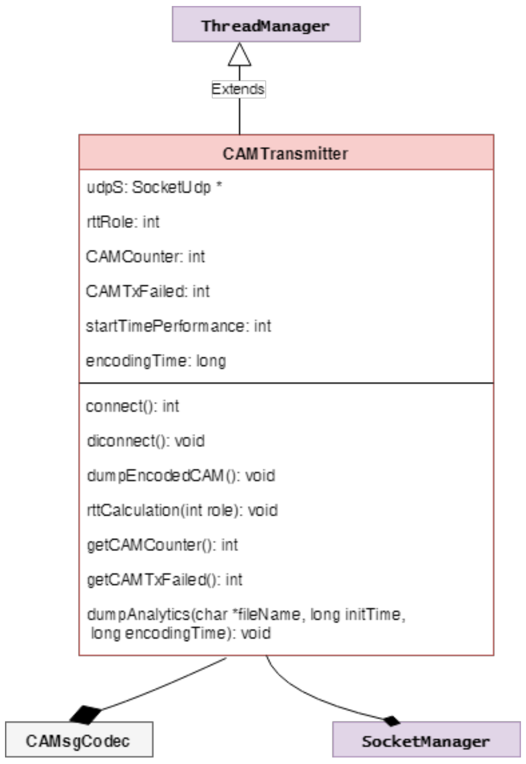


Figure 4.9: `CAMTransmitter` UML diagram

OtherVIFace

The `OtherVIFace` is a subclass of `GenericVIFace`. This class is in charge of taking a `CAM_t` data structure, with method `CAMExtractor`, and of providing a user with set of `getter` methods that return CAM parameters, converted in a non-standard ETSI format.
For instance the speed, read by a CAN bus in [km/h] and converted in [cm/s] by the codec for transmission, is converted back to [km/h] by the `OtherVIFace`, before being passed to the `GenericVIFace` for logging purposes.

AlertIFace

This class is the parent of two subclasses, named `ItsStationIFace`, used in reception of DENM and `SimulatedITSIFace`, used in transmission of DENM. It represents the interface exploited by an ITS-S to transmit and receive DENM messages. It performs similar functionalities to the one executed by `GenericVIFace`.
`AlertIFace` provides `virtual` methods, then defined by its subclasses, and logging methods (`dumpToFile`), through which each subclass, by specifying an header and a file name, can fill log files either in transmission, and reception of DENM messages.

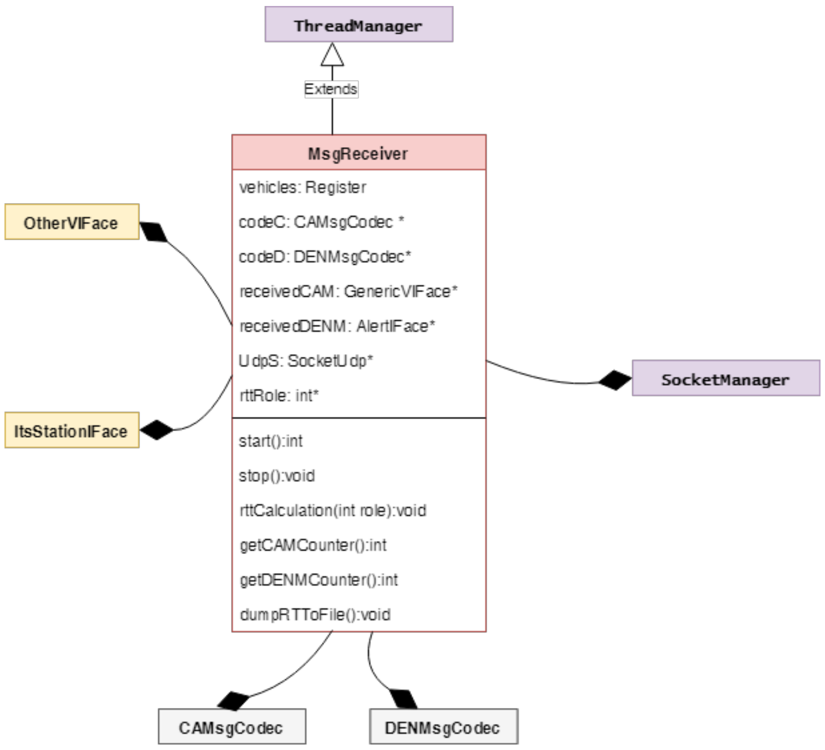


Figure 4.10: `MsgReceiver` UML diagram

Since the `SimulatedITSIFace` is not used in `oaicc`, we will describe it just in the relative part, now focusing on `ItsStationIFace`.

ItsStationIFace

This class is the subclass of `AlertIFace`, and its objective corresponds symmetrically to the `OtherVIFace` for the `GenericVIFace`. In fact, it is in in charge of taking a `DENM_t` data structure, with a method called `DENMExtractor`, and provides the user of the class with a set of *getter* methods that return parameters, eventually converted back to non-standard ETSI format, in the same way as `OtherVIFace` does for CAM.

4.4.2 User interaction and supplementary functionalities

`oaicc` is the vehicle application which performs four main functionalities: CAM encoding and CAM transmission, CAM and DENM reception and decoding. The user of the application can interact with it, by means of Linux terminal, namely the Command Line Interface (CLI).

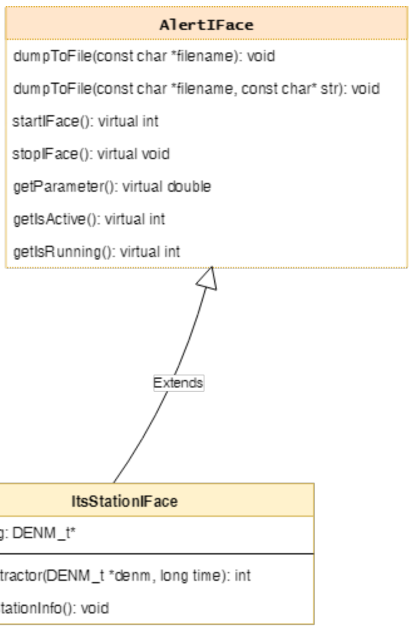

AlertIFace
dumpToFile(const char *filename): void
dumpToFile(const char *filename, const char* str): void
startIFace(): virtual int
stopIFace(): virtual void
getParameter(): virtual double
getIsActive(): virtual int
getIsRunning(): virtual int
Extends
ItsStationIFace
DENMsg: DENM_t*
denmExtractor(DENM_t *denm, long time): int
updateStationInfo(): void

Figure 4.11: `AlertIFace` UML diagram

option	**description**
-s	CAM are generated extracting data from a SUMO file. It is a mandatory alternative to *-r* option
-r	CAM are generated extracting data from CAN bus and GPS interface. It is a mandatory alternative to *-s* option
-tx	CAM are transmitted towards a specific destination address. It can be used with or without the *-rx option*
-rx	CAM and DENM are received. It can be used with or without the *-tx* option
-d	destination IP address, mandatory when the -tx option is enabled
-ptx	optional transmission UDP port option. If it is not specified, a default value is set
-prx	optional reception UDP port option. If it is not specified, a default value is set
-dev	optional addition of two CAN bus interfaces and GPS device. Can be enabled when the *-r* option is enabled
-v	optional additional log files of received messages are created
-rtt	this option allows to define the application as server (se) or client (cl), in order to perform FLL analysis

It is provided with a set of options, some of them mandatory, others optional, that can control and enable by launching the application using a Linux terminal.
All the options are reviewed in a helper file, that can be accessed typing *-h* besides the name of the program. The `main()` code manages the instantiation of the `CAMTransmitter` and `MsgReceiver`, depending on the options set by the user of the application.
Whenever is launched from the terminal, the application prints coloured lines to show the reception of messages. The color codes which have been chosen are *green* for received CAM and *red* for received DENM. We have decided to not show the transmitted messages on the `stdout` to save computational resources. Furthermore, `oaicc` is an application specifically thought for cooperation. Then, watching at the screens of two different machines, where two instances of it are running, we have an immediate feedback that messages reach or not the destination.
Besides this main functionalities, the application is equipped with analytic additional functionalities:

- FLL calculation. As we will explain in the next chapter, we have used this value to calculate the FLL. The candidate has equipped the application with an option called *-rtt*. This option allows to define the machine, hosting the application, as client or server. The client and the server construct three files, each of them can be used to compute the FLL and other statistical measures. File names and descriptions are collected in the table:

Log file name	**description**	**Role**
analytics_tx.csv	this file collects, for each CAM transmitted, the initial timestamp value and the CAM encoding time interval	cl
analytics_rx_se.csv	this file collects, for each CAM received, the decoding interval at the server and re-encoding interval	se
analytics_rx_cl.csv	this file collects, for each CAM received back, the final timestamp value and the CAM decoding time interval	cl

  We have decided to keep the files disjointed since the client and the server are not synchronized; they are not using, thus, the same reference time.

- Log file construction: as we reported before, the `CAMTransmitter` and `MsgReceiver` allow to build log files, particularly interesting in post processing to check whether the values transmitted are the same values received.

4.4.3 Physical view: CAN interfaces

This brief section concerns the physical configuration adopted when CAM messages extract real vehicle information from the CAN bus and the GPS receiver.
The full development of the applications has been made within a VirtualBox VM, running Linux OS. Thanks to CANAlyzer software, running on the Windows host machine, it has been possible to create an ad hoc CAN configuration with can buses used to extract the parameters then used to populate CAM messages. Such configuration has been enriched by a signal generator, pushing signals in a box, produced by the same Vector company, called CAN Case XL [78]. In this way, it has been possible to emulate the behaviour of a real vehicle using the FCA signal maps, the real CAN signals and ECUs involved.
To connect the Virtual Box VM linux to the CAN CASE XL, the KVaser transceiver has been used.
The constructor of `CAMTrasmitter` is able to recognize the channels used and set up the CAN parameters (baudrate, device, etc).

4.5 OAICodeCAlert: a Road Side unit application

`oaica` is an application developed in C++11, using the *Codeblocks* IDE.
The need for a second C-V2X application has arisen in order to test encoding, transmission and reception of DENM messages.
In fact, as we have described previously, the `oaicc` is compatible with the reception of either CAM and DENM messages. By one side, CAM correct reception can be tested just using two instances of `oaicc` on two different machines, eventually connected to the same LAN. On the other side, it missed an application in charge of showing the correct delivery of DENM, namely an application assuming the role of the Road Side Unit and emulating part of its functionalities. Such application, called `OAICodeCAlert` is capable of spreading alert messages (DENM), in unicast to vehicle applications (namely other `oaicc`). These messages are just simulated, thus there is not present the *real* option as in the `oaicc`.
Three classes have been added to the already described ones, thus, we will limit the development view to these classes. After that a logical view, describing the functionalities the application is capable of, is given.

4.5.1 Development view

The development view diagram of `oaicc` is shown in figure 4.4. As we can see, there is a partial overlapping of functions with the other application. Thus, we will focus on three of the seven classes composing the project: `DENMTransmitter`, DENMsgCodec and SimulatedITSIFace.

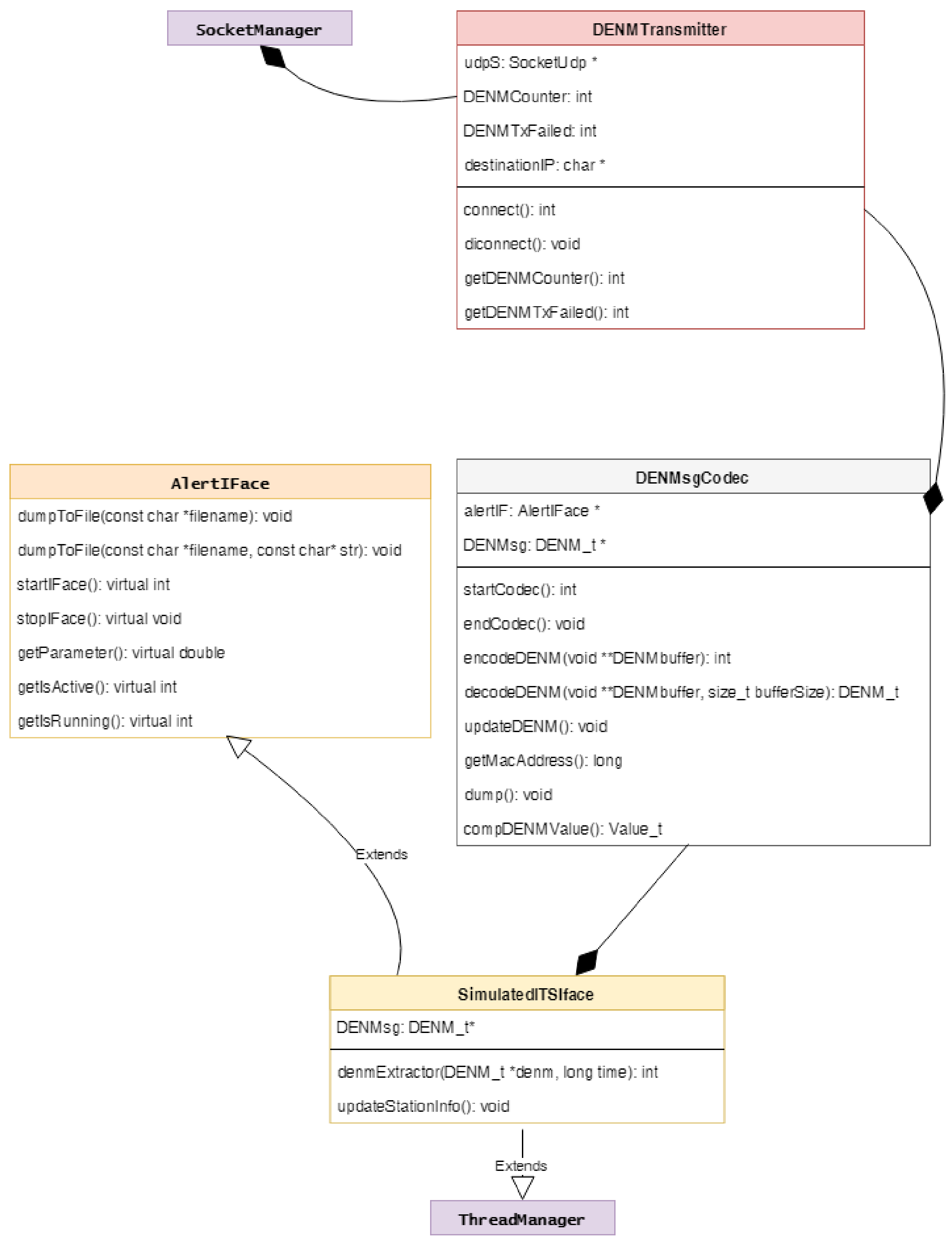


Figure 4.12: oaica UML diagram

SimulatedITSIFace

This class is subclass of the `AlertIFace` whose structure is described in the previous section, and the `ThreadManager`. Its structure and functions are similar to the `SimulatedVIFace`. In fact, it is capable of extracting parameters read by a *.csv* SUMO file and, through *getter* methods, offer them to the user of the class. The parameters, decided in accordance between Politecnico di Torino and CRF are:

- Action ID: is the identifier of an event (e.g a car accident) detected by an ITS-s,
- Detection Time: timestamp at which the event is detected,
- Reference Time: timestamp at which a DENM message is generated,
- Termination: parameter indicating the termination of an event,
- Latitude,
- Longitude,
- Altitude,
- Validity duration: duration of DENM validity,
- Station type,
- Information quality: is the quality level of the information provided by the ITS-S application of the originating ITS-S, namely the probability that the detected event, for instance a car accident, is really happening;
- Cause Code: is the direct cause of an event and provides an high level description of the detected event type. For instance if there are roadworks present on the lane, or if there have been a vehicle breakdown. It is a code from 0 to 255. We have decided to transmit DENM with CauseCode 97, which means *CollisionRisk*, suitable for the use case of ICA.
- Sub Cause Code: it provided more detailed information of the event reported by the Cause Code. In ETSI, each Cause Code is accompanied by a set of Sub Cause Codes. For the Cause Code 97, 5 Sub Cause codes have been defined. We have chosen number 2, meaning *crossing collision risk.*
- Event speed: it is the speed of the detected event. In our case we have considered it as the movement speed of the other car when a car accident has been foreseen by the RSU;
- Event heading: it is the heading direction of the event;

- Trace: this complex field represents all the possible geo-referenced traces leading to the position of the event;
- Vehicle Identification: SEQUENCE of two number, the World Manufacturer Identifier (WMI) and the Vehicle Descriptor Section (VDS), parts of the Vehicle Identification Number (VIN) code, a unique code used by automotive industry to identify individual motor vehicles [77].

The last parameter extracted by the class is a destination IP address, corresponding to the destination to which the RSU application shall deliver a certain DENM. This solution has been adopted since the `oaica` doesn't run any intelligence inside it, and for this initial project step, allows to decide off-line the destination of a group of DENM.
The class save the content of a SUMO file in a vector, then reads the vector line by line, on the basis of the reference time written in the file and provides a set of *getter* methods to the user.

DENMsgCodec

The tasks of the `DENMsgCodec` are correspondent to the tasks performed by `CAMsgCodec` in `oaicc`. It has to fill up a DENM message, namely a `DENM_t` structure, with parameters extracted from an object of type `SimulatedITSIFace`, opportunely converted into a compliant ETSI DENM format. The class provides the users, the `DENMTransmitter` and `MsgReceiver` classes, with methods `encodeDENM` and `decodeDENM`, to perform serialization and de-serialization of the message.
The challenging parts, in building a class like this one, are the same seen in the `CAMsgCodec` part. In particular, management of *Traces* and *PathHistory* fields has been particularly laborious.

DENMTransmitter

`DENMTransmitter`, specular version of the `CAMTransmitter`, exploits an encoder and a Socket manager to transmit DENM messages. It also provides methods for statistical analysis and builds a log file about DENM transmitted.

4.5.2 User interaction and supplementary functionalities

`oaica` is the Road Side Unit application which performs four main functionalities: DENM encoding and DENM transmission, CAM decoding and reception. The user of the application can interact with it, by means of Linux CLI.
Among the options offered by the application to the user, some of them mandatory, others optional, that can control and enable by launching the application using a Linux terminal.
All the options are reviewed in a helper view, that can be accessed typing *-h* besides

the name of the program. The `main()` code manages the instantiation of the `DENMTransmitter` and `MsgReceiver`, depending on the options set by the user of the application:

option	**description**
-s	CAM are generated extracting data from a SUMO file. It is a mandatory option
-d	destination IP address
-ptx	optional transmission UDP port option. If it is not specified, a default value is set
-prx	optional reception UDP port option. If it is not specified, a default value is set
-v	optional additional log files of received messages are created
-extra	since the application runs just a simulated trace and, after the trace have been all read, context and application is terminated, extra time before application ends can be given by the user, for instance to listen for a longer period CAM arriving from another source

Whenever is launched from the terminal, the application prints coloured lines to show the reception of messages. The color codes which have been chosen are the same seen for `oaicc`.
Besides this main functionalities, the application is equipped with log file construction. The `DENMTransmitter` and `MsgReceiver`, if *-v* option is enabled, build log files, particularly interesting in post processing to check whether the values transmitted are the same values received.

Log file name	**description**
its_self_log.csv	this file includes a table that collects relevant information for each transmitted DENM
car$< stationID >$.csv	this file is created for each ITS-S, thus with a given stationID, sending CAM to the `MsgReceiver`

4.6 Integration with OAI

A second, important part of the Thesis has been conducted at the Politecnico di Torino, where a team, actively involved in connected mobility, is working jointly

with CRF in meeting the guide lines of 5G-T.
In this part of the activity, the candidate has touched closely the OAI interface repository, and installed all the components needed to run the OpenAirInterface LTE Modem on a OAI UE PC. Furthermore, it has been done an integration of the `oaicc` and `oaica` applications with the environment.
The network of OpenAirInterface can be set up by installing files from a repository, freely available here [79]. The tutorials, used to install it, are available here [80]. Thanks to Virtualization, that permeates the software architecture of OAI, network functions, for instance EPC or eNB, can be installed in one, two or three machines, depending on the objective and computational resources of the testbed. The testbed consists in three separated machines. We will call them OAI UE, OAI eNB and OAI EPC.

- OAI UE PC: hosts an LTE modem and performs UE procedures like attach, authentication, service access and radio bearer establishment;
- OAI eNB PC: hosts an other LTE modem and performs eNB functionalities,
- OAI EPC PC, hosts:
  - Packet Core functionalities. It contains a group of Virtual Machines that exploit functions of MME, HSS and SP-GW
  - MEC functionalities. The candidate will refer to a generic MEC architecture, since the MEC project, in which tests have been made, has not been released yet.

In the following subsections we will look closely to the UE and EPC laptops, in order to understand how the integration with the two applications described this chapter has been done.

4.6.1 OAI UE

The OAI UE PC represents the UE vehicle, namely the on board device used by a vehicle to communicate within a 3GPP network.
It consists in a laptop PC (Ubuntu v16.04) connected to a USRP B210 board with USB 3.0 cable. The LTE connectivity is given by OAI thanks to a low level application called `LteSoftModem`. Its role in the testbed is to transmit and receive from another vehicle or from ME Applications in the MEC platform, basic safety messages (CAM) and event-driven messages (DENM).
Whenever the LTE modem, installed on the UE, is running, IP service is offered by the OAI network. At the application layer runs an instance of `oaicc`. The Linux installation of the application consisted in moving the executable file into the folder `/urs/local/bin`.

4.6.2 OAI EPC and MEC

The EPC consists in a set of network functions, running on two different VMs, one for the HSS and MME, the other one for the SP-GW. The OAI SP-GW is the border router between two networks:

- 172.16.0.0/16: this network is the mobile operator network. The SP-GW hosts a DNS server containing a pool of addresses and, whenever a UE attaches to the the eNB, and for this purpose it sets up a network interface called *oip1*, it is configured by the EPC with an address from that pool.

- 192.168.0.0/24: this network is the MEC network. It is located beyond the SP-GW;

In addition, for implementation reasons, the 10.10.10.0/24 network is designated to control purposes. By means of it, administrators can perform the management of EPC and MEC VMs. As we described in 3.2.1, ME platform interacts directly with the data plane of the mobile network, in this case the SP-GW. It manages a service registry, that contains the set of supported services and controls Virtualized resources, turning on VMs to serve a specific service targeted on the user needs.

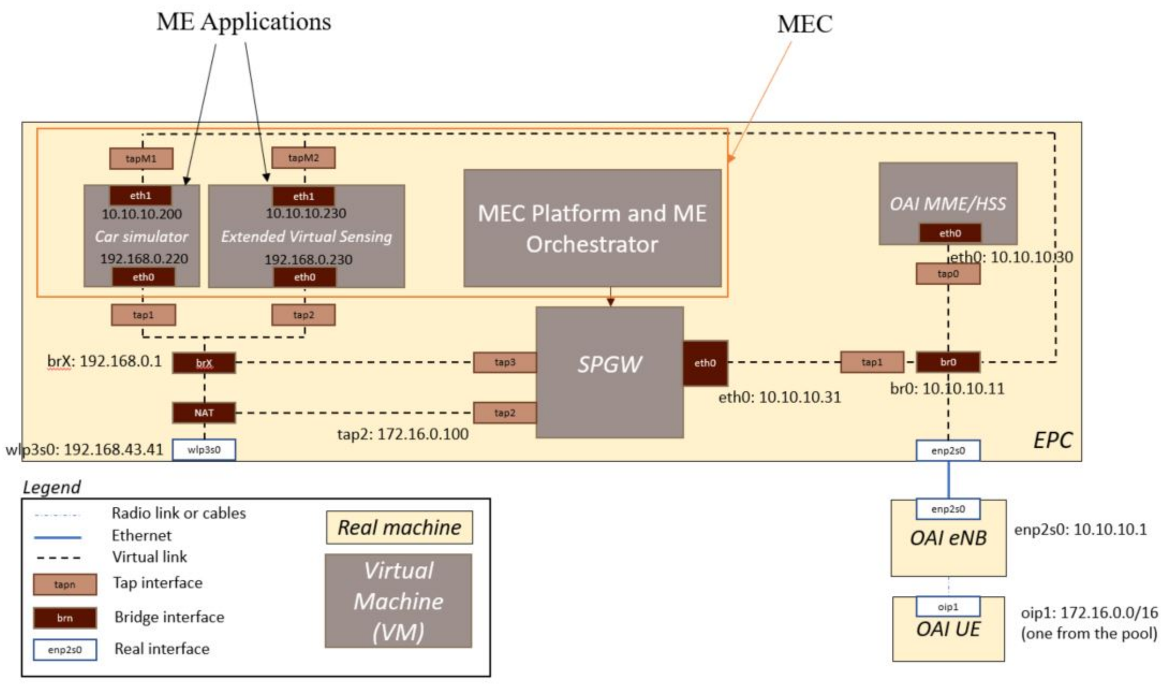


Figure 4.13: OAI network architecture testbed and MEC

In Figure 4.13, the overall network architecture is described in the details, except for MEC Platform. By means of the ME Orchestrator, traffic redirection rules are defined, that forward traffic coming from the UE to a set of service VMs. Such VMs are ME applications. In theory, these VMs shall be allocated on demand on the basis of user requests, as the result of the interaction between the Vertical Slicer described in 4.1 and the ME Orchestrator. For now, VMs are set up statically when the MEC platform is launched.
We are interested in two ME applications, both created by the candidate, depicted in Figure 4.13

- Car simulator: is a VM deployed using VirtualBox, equipped and configured to sustain `oaicc`;
- Extended Virtual Sensing (EVS): is a VM deployed using VirtualBox, equipped and configured to sustain `oaica`.

Many VMs in OAI presents two network interfaces, namely *eth0* and *eth1* in the Figure, which belong to two different networks. The reason of such configuration is connected to access an management purposes. In the current configuration, whenever a network administrator wants to access that machines from remote terminal (ssh), it uses the *br0* interface, because it is directly connected to the Ethernet network interface of the PC (*enp2s0*), thus exploiting 10.10.10.0/24 network.
On the MEC platform, traffic arriving from the UE PC, directed towards a generic address, for instance 8.8.8.8, can be forwarded to the EVS machine. This is exactly the test we have performed to show the correct reception of bunch of CAM and DENM, in the following Chapter.
In this Chapter have been described the two C-V2X vehicular applications that, on a network testbed deployed using OpenAirInterface, have been made able to exchange packets over a 5G network exploiting MEC architecture.
In the next Chapter, the network have been tested with two communications scenarios, to compute the FLL and prove the correct reception of all DENM messages to all the vehicle applications.

Chapter 5

C-V2X applications results

In this Chapter are presented some of the most relevant results achieved by the applications described in Chapter 4. Tests have been executed in different scenarios, exploiting the OAI network testbed located in the Department of Electronics and Telecommunications(DET) at Politecnico di Torino. Therefore, for each test, it has been exploited the simulative mode of `oiacc`, namely populating CAM with a SUMO file.
Channel used for the computation is the *wired* OAI testbed channel and all the references to IP addresses and network interfaces are related to Figure 4.13.

5.1 Facility Layer Latency calculation

The Facility Layer Latency is the latency in the transmission at the Facility Layer, that, according to ETSI ITS-G5 protocol stack, is the layer just below the Application. This value is equal to:

$$FFL = \frac{Tf - Ti}{2} \tag{5.1}$$

where *Ti* and *Tf* are respectively the instant in which a message is generated, namely the moment in which the CAM is populated using SUMO file parameters, and the instant in which the same message is received and its values have been extracted, after decoding process.
Double objective of this measurement is to evaluate of the performance of the proto-5G OAI network, powered by the low latency service ensured by MEC platform, and to weigh the processing impact of ETSI messages generation process in future 5G networks. We have computed the FLL between:

- UE Physical Machine (PM) client application (`oaicc`), which transmits a CAM message of fixed size (57 bytes);

- MEC VM server Application (`oaicc`), that receives CAM messages and retransmits them back to the client.

There have been used 30 simulation files, representing the transmission, for about 60 seconds, of CAM messages from a vehicle. For each of them, 579 CAM messages have been transmitted, and average, maximum and minimum FLL have been computed.

Application	**implemented in**	**interface:IP address**	**Log file created**
`oaicc` client	UE PM	oip1:172.16.0.4/24	analytics_tx.csv analytics_rx_cl.csv
`oaicc` server	MEC VM	eth0:192.168.0.220	analytics_rx_se.csv

Since the channel is asymmetric, and the machines have not been synchronized, the FLL can be approximated to this calculus:

$$FLL = EN1 + lat1 + DEC1 \approx EN2 + lat2 + DEC2 \approx \frac{Tf - Ti}{2} \tag{5.2}$$

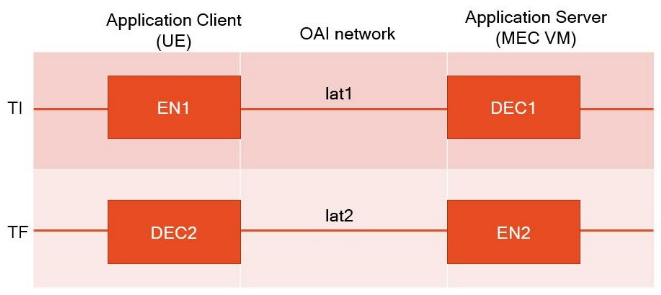


Figure 5.1: FLL

The log files, mentioned in table above, and constructed by the `oaicc` in client or server mode, are created separately by the two entities, and we wanted to merge them for analysis. Thus, we have created two cooperative scripts:

- the Sever script must be launched first, in the MEC VM. It starts listening the channel on the port specified by the user. When the client `oaicc` has concluded transmission of CAM in the SUMO file, the user can type `ctrl + c` to terminate `oaicc` server execution. The script transmits the file it has produced to the client machine with `scp` service. Finally, the server stands by, waiting for another reception;

- the Client script must be launched after the server, in the UE PM. It waits until the user has correctly instructed the server to send the log file to the client. Then, the client processes all the log files together and returns a file containing all data needed to compute FLL and FLL for 5G.

The two scripts are available in Appendix A.1.
Results have been collected in a Table and plotted. The FLL of thirty experiments has been computed in [μs], but, in order to make the graph more comprehensible at a first sight, we converted it into [ms].

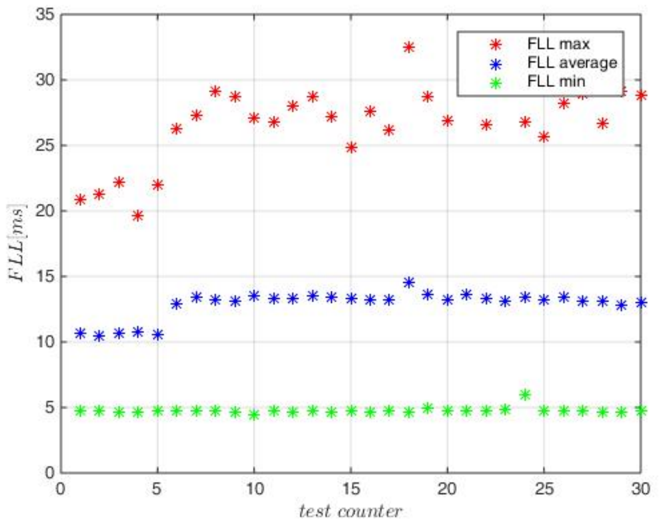


Figure 5.2: FLL

The graph shows a linearity of average and minimum values, respectively the blue and green dots in Figure 5.2. The highest variance interests the maximum values, that still don't overpass 32.66 ms. The minimum value registered is instead 4.49 ms.
Taking the average values in the graph and further averaging them, we got:

$$FLL_{final} = 12{,}87\ ms \tag{5.3}$$

In order to verify this result, and compare it with the probability distribution of all the tests, we have considered all the FLL values, namely 17370 measurements and we have constructed the Cumulative Distribution Function (CDF) of the FLL values, in [ms]. From the graph, we can observe that most of the values, namely the 55%, fall in the range between 10 and 15 ms. This observation perfectly reflects the mathematical computation showed before about the FLL_{final}.

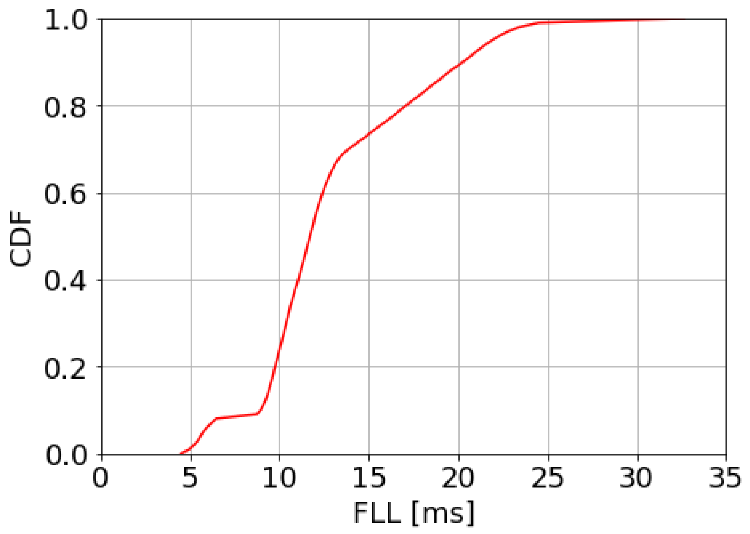


Figure 5.3: FLL Cumulative Distribution Function

To give an idea of what a similar results means in a real vehicle scenario, still keeping in mind that the testbed is in its early stages, we can consider a motorbike, travelling at high speed, from 80 to 130 km/h, sending messages to the other vehicles. The following table gives an idea of the precision, in terms of meter, guaranteed by the network in terms of delivery of messages, considering a FLL of 12,87 ms.

Speed	**Precision[m]**
80	0.29
90	0.32
100	0.36
110	0.43
120	0.42
130	0.46

This FLL value has been computed for a wired channel and a network architecture that is not yet supported by 5G functionalities, like the NR or the 5G Core Network. Thus, we have decided to compute, in the same conditions as FLL_final, a function showing the impact of extra processing time, due to CAM generation, encoding and decoding time, on tightest 5G latency requirements, at all layers. The latency values considered have been from 2 to 5 ms; they are very strict and related to the most demanding use cases and 5G applications. The $FLL_{5G}(x)$ is a linear function computed with the following formula:

$$FLL_{5G}(x) = (EN + DEC) + x, \ x \in [2,5] \tag{5.4}$$

Considering the Figure 5.1, EN and DEC are respectively:

$$EN = \frac{EN2 - EN1}{2}, \ DEC = \frac{DEC2 - DEC1}{2} \tag{5.5}$$

Then, since EN1, EN2, DEC1 and DEC2 value precision is in the order of μs, thus very small, we have decided to take their average on the 17370 measures.

$$FLL_{5G}(x) = \frac{28.46 + 36.54}{1000} + x = 0{,}06535 + x, \ x \in [2{,}5] \ [ms] \tag{5.6}$$

The computational time due to encoding and decoding has a relatively small impact on latency imposed by 5G. The order is 10^{-2} ms.

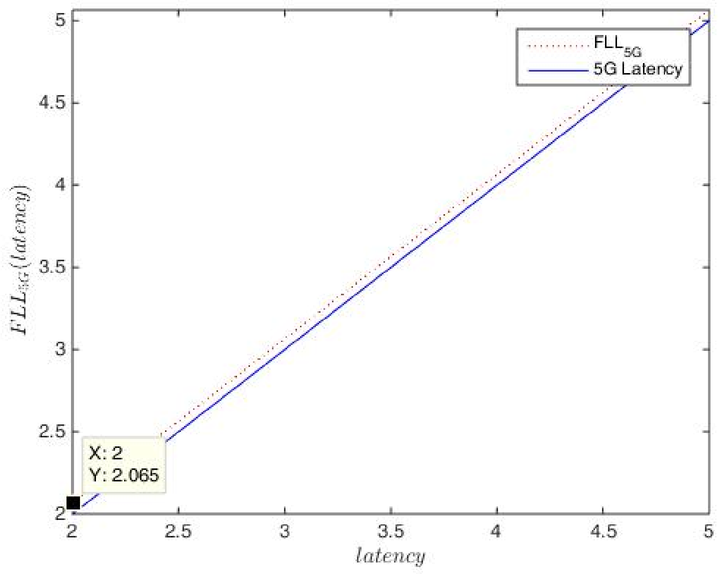


Figure 5.4: FLL for 5G

5.2 DENM Packet Success Rate

The DENM Packet Success Rate is the percentage of DENM messages that, transmitted by a source, correctly arrive to the destination.
We have computed it in the following scenario, depicted for clearness in Figure 5.5:

- UE PM application and MEC VM application, instances of `oaicc`, transmit CAM messages of fixed size (57 bytes) to another MEC VM Application (running `oaica`);

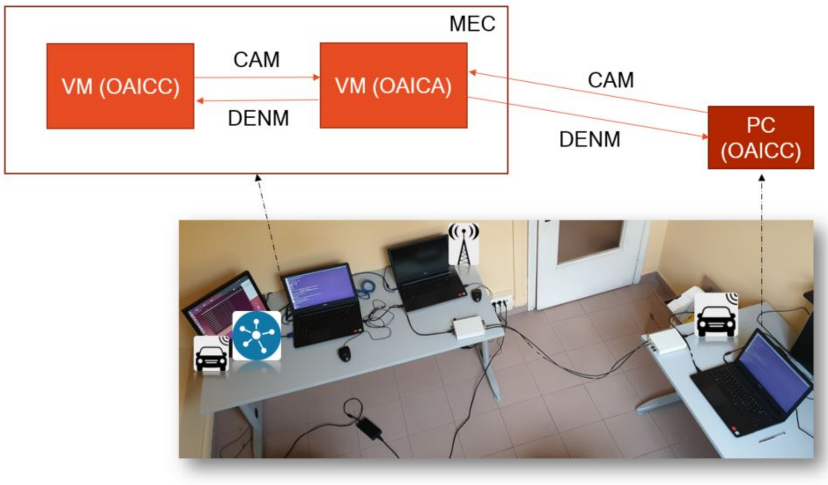


Figure 5.5: PSR scenario

- MEC VM Application (`oaica`), transmits a set of DENM messages, spreading them among the two destinations which are transmitting CAM messages at the same time.

App	**implemented in**	**interface: IP address**	**Task**	**Log file created**
`oaicc`	UE PM	oip1:172.16.0.4/24	Tx: CAM Rx: CAM and DENM	alert_log.csv vehicle_self_log.csv
`oaica`	MEC VM	eth0:192.168.0.230	Tx: DENM Rx: CAM and DENM	car$< ID1 >$.csv car$< ID2 >$.csv its_self_log.csv
`oaicc`	MEC VM	eth0:192.168.0.220	Tx: CAM Rx: CAM and DENM	alert_log.csv vehicle_self_log.csv

The test envisages that a redirection rule is set by the MEC platform. In fact, the UE vehicle transmits CAM towards a fixed address, for instance 8.8.8.8. The SP-GW, instructed by the Orchestrator to forward all the traffic destined to that address to a serving VM running `oaica`, diverts CAM messages the the EVS.
The rationale beyond this test is to reproduce a common situation that could verify in the successive versions of the testbed. When the ICA service will be ready to process CAM and generate tailored DENM, on the basis of information received by vehicles, such intelligence will be co-located in the same VM on which runs the

`oaica`. Vehicles subscribing the ICA service will likely receive an unicast address to which forwarding all the traffic and packets directed to that address will be intercepted by the SP-GW that will reroute them to the interested VM.
There have been executed 30 tests of 30 prepared SUMO traces. These SUMO traces come from different simulations, run with SUMO, in which two vehicles approach each other and the RSU sends DENM. Each simulation lasts about 60 seconds. According to the statistics, printed out by the applications at the end of their process, we have been able to compute the DENM PSR. An immediate way to prove the correct reception of all DENM is to take the log files of the two instances of `oaica` and counting the number of rows on each file, then compute the average, with a bash script.
On the basis of the equality:

$$DENM_{TX,\texttt{oaica}} = DENM_{RX,\texttt{oaicc}} + DENM_{RX,\texttt{oaicc}} \tag{5.7}$$

it has been built up a graph which shows that the number of transmitted DENM by `oaica` application results the same to the sum number of received DENM at the two receivers instances of `oaicc` applications.

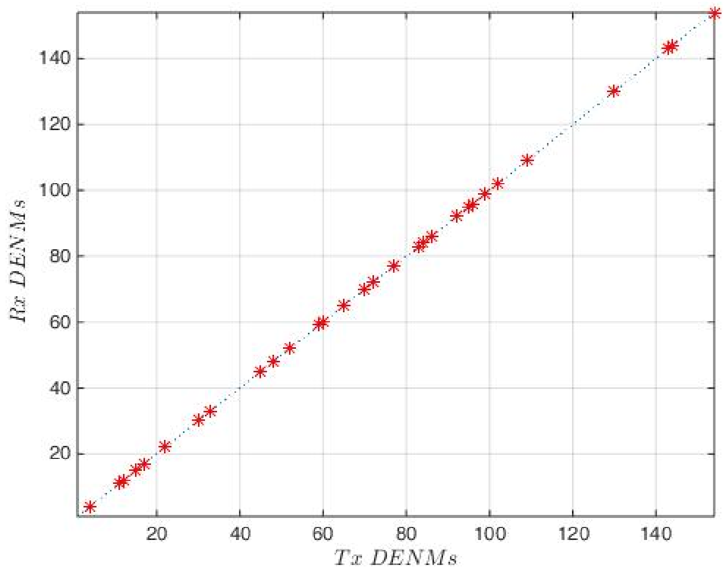
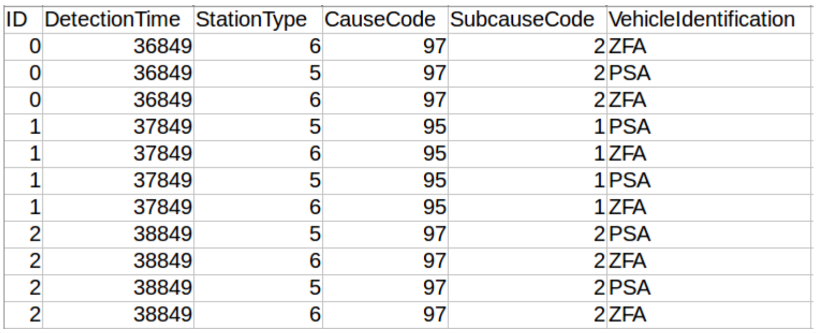


Figure 5.6: DENM Packet Success Rate

As we can see the number of transmitted DENM is resulted to be always the sum of messages transmitted and messages received. Points thicken to form a line with gradient of 45. From this observation we can conclude that:

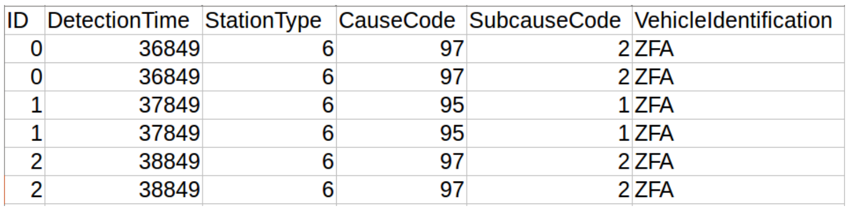
$$DENM\ PSR = 100\% \tag{5.8}$$

Still, even if we have proven that DENM messages are correctly delivered at the destination, we want to show that all values extracted by simulation file, like detection time and Cause Code, are correctly filled up and received coherently.
In order to demonstrate it, we have considered, in the same scenario, a single alert simulation file passed to the transmitter and observed, after transmission, log files of the two instances of `oaicc` and log file of `oaica`. Furthermore, to make easier the file comparison, five fields from the first eleven DENM transmitted have been chosen, and compared in the three applications. Before transmitting, we have opportunely modified fields in the simulation file, in order to understand what has been really transmitted and received.
The chosen fields are: detection time, Station Type, Cause Code, Sub Cause Code and Vehicle Identification.
In the following images, we report the log file created by the `oaica` VM MEC (Figure 5.7), UE PM (Figure 5.8) and UE VM (Figure 5.9).



ID	DetectionTime	StationType	CauseCode	SubcauseCode	VehicleIdentification
0	36849	6	97	2	ZFA
0	36849	5	97	2	PSA
0	36849	6	97	2	ZFA
1	37849	5	95	1	PSA
1	37849	6	95	1	ZFA
1	37849	5	95	1	PSA
1	37849	6	95	1	ZFA
2	38849	5	97	2	PSA
2	38849	6	97	2	ZFA
2	38849	5	97	2	PSA
2	38849	6	97	2	ZFA

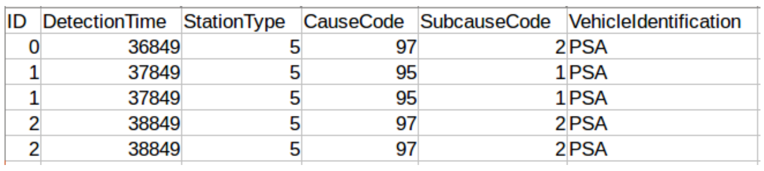
Figure 5.7: ITS log



ID	DetectionTime	StationType	CauseCode	SubcauseCode	VehicleIdentification
0	36849	6	97	2	ZFA
0	36849	6	97	2	ZFA
1	37849	6	95	1	ZFA
1	37849	6	95	1	ZFA
2	38849	6	97	2	ZFA
2	38849	6	97	2	ZFA

Figure 5.8: UE PM log

A first indicator of correct reception of DENM messages at the two receivers is the *VehicleIDentification* field. In fact, the UE PM is a ZFA vehicle, the UE VM MEC is a PSA machine. Checking line by line the three files we can see that DENM fields have been correctly filled up and transmitted to the right destination.



ID	DetectionTime	StationType	CauseCode	SubcauseCode	VehicleIdentification
0	36849	5	97	2	PSA
1	37849	5	95	1	PSA
1	37849	5	95	1	PSA
2	38849	5	97	2	PSA
2	38849	5	97	2	PSA

Figure 5.9: UE VM log

5.3 Time To Provision of ME Applications in MEC

Time To Provision (TTP) is a metric of the time taken by MEC platform to setup service VM. We have considered, in order to not detail too much the MEC Platform system used, just Car simulator and the EVS VM.
In future 5G network this value could play a central role, since a low latency service may require a low latency set-up of applications in the MEC Platform. TTP has been computed by means of a script which can be found in Appendix A.2. Lines of

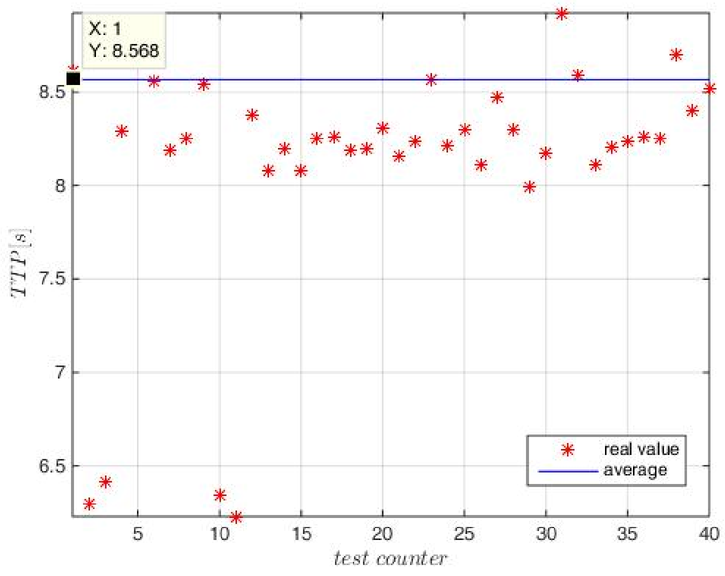


Figure 5.10: MEC Time To Provision

this script have been inserted in the MEC set-up code, before the VM have been instantiated and after the VM instantiation, in order to get the precise time at which VM are available to the user. Results are reported in Figure.
The TTP oscillates between 6,23 s and 8,92 seconds. The average value is:

$$TTP = 8{,}56\ s \tag{5.9}$$

5.4 New CAM Vehicle Identification field

This result is relevant in sight of new CAM specifications and concerns the addition of a `VehicleIdentification` field into the CAM message. As we have already discussed, ETSI messages present a structure composed by optional and mandatory fields, populating the containers. In order to make the CAM (and DENM) architectures flexible and modifiable for the future, ETSI has inserted a third field type: the extension, which in the ASN1 description is graphically indicated with three dots `...`.
The `VehicleIdentification`, according to CAM ASN1 description, is a SEQUENCE of two numbers:

- World Manufacturer Identifier (WMI)
- Vehicle Descriptor Section (VDS)

They are parts of the Vehicle Identification Number (VIN) code, a unique code used by Automotive industry to identify individual motor vehicles.
The extension field allows to add new fields in the CAM containers observing this notation. The modified message, maintains back compatibility with decoders not supporting new fields, and at the same time can empower service provisioning for vehicles and ITS-S stations supporting it. We have decided to push the identifier inside the CAM `BasicContainer`, hosting general CAM information like the Station Type and the location information. The creation of the new CAM message has been done using the `asn1c` compiler.
The compiling process has been repeated and the ASN1 file description of the CAM has been modified from:

```
BasicContainer ::= SEQUENCE {
stationType StationType,
referencePosition ReferencePosition,
...
}
```
to:
```
BasicContainer ::= SEQUENCE {
stationType StationType,
referencePosition ReferencePosition,
...,
vehicleIdentification VehicleIdentification,
...
}
```

In order to prove back compatibility of this new CAM for devices not supporting this field, we have inserted the message, encoded by `oaicc` in hexadecimal notation, as input of a free message codec available online [81] and tried to decode it. The coding tool is very flexible, with many functionalities. In our case, receiving in ingress an hexadecimal encoded string and an *.asn1* file description of CAM, is capable of decoding the message using the most used decoding rules, included UPER. The result has been what we expect. In fact, passing to the compiler the most recent CAM version, ITS CAM v1.3.2, the new `VehicleIdentification` field is not detected, but still CAM has been correctly decoded. Using the CAM description modified, the decoder detects and decode the new field.
In conclusion, in this Chapter we have seen four main results achieved in this Thesis project. At the beginning, FLL value has been computed and impact of extra time CAM processing has been evaluated. Then, in a well-known configuration for the 5G-T use case, DENM PSR has been measured and comparison between log files of the transmitting and receiving application has been done to demonstrate the coherent reception of all information transmitted in CAM at the receivers. Finally the CAM has been enlarged by a new field, the `VehicleIdentification`, useful for the Automotive Vertical and opening the path to larger CAM messages in the higher bandwidth, future 5G networks.

Chapter 6

Conclusions

Real and *reality* are redundant concepts in this Thesis project, which has consisted in the development of two applications, `oaicc` and `oaica`, that use C-V2X communication network, offered by OAI, to transmit and receive, encoding and decoding CAM and DENM messages.
The CAM messages are compliant with the ETSI standardization, and are filled with parameters collected either from a simulated interface, namely a SUMO file trace, and from a *real* interface, namely CAN buses and GPS receiver. DENM messages are populated with parameters obtained from a simulation file. Furthermore, the applications offer to the user a CLI Linux-based interface, showing received messages and additional functionalities to dump log files about transmission and reception of message and to compute FLL.
In the activity, pursued in collaboration with CRF and Politecnico, the candidate has proven the potentiality, on a *real* network testbed, of a MEC system, which offers very low latency for the most demanding applications in future 5G networks. In particular, the FLL=12,87 ms, found in the results Chapter, meets, in this phase of the 5G-T project, the latency requirements for the ICA use case, being lower than 20 ms. The DENM PSR shows as, in a scenario similar to the *real* one and that the project is going to face, when final implementation will be ready, all DENM transmitted by the Road Side Unit Application are actually received by the vehicle applications. The time of availability of the MEC service is a point in which the project needs to evolve and become more efficient, since timing is very important for future applications and VMs in the MEC should be ready in the shortest time as possible. Finally, the insertion of a new field inside the CAM message represents not just the possibility for car makers to discriminate between CAM coming from branded vehicles, but also the promises for new sized CAM, customizable by car makers and whose transmission allowed by the higher throughputs guaranteed by 5G technology.
The results collected represent a good starting point for the next phases of the 5G-T project. During this Thesis, things continued to move in Europe and in Italy.

Associations and companies work tirelessly to deliver 5G, as soon as possible, to their custumers. The conclusion of the project conincides with the start up phase of the early 5G deployment. Vodafone, in fact, has announced the conclusion of 4G coverage in Italy, which will allow to access in advance some 5G functionalities [83]. In the 2nd of October, has been closed, in Italy the auction for 5G frequency spectrum concerning 700MHz, 3.7 GHz and 26 GHz bands, to Italian operators [82]. In parallel, from the V2X front, realities like Politecnico and Città di Torino are working together to bring autonomous driving in the streets of our cities. 5G and Automotive are central players in next Generation Networks and projects, as 5G-T, represent an important turning point in cooperation between the industry and research world in order to make 5G a reality.

Bibliography

[1] McKinsey&Company, *Automotive revolution - perspective towards 2030*, January 2016.

[2] https://www.pwc.com/gx/en/industries/automotive/publications/eascy.html

[3] European Commission, *Annual Accident Report 2017*, 2017.

[4] NHTSA, *National Motor Vehicle Crash Causation Survey - Report to Congress*, July 2008.

[5] European Commission, *Framework for the Deployement of Intelligent Transport Systems - Directive 2010/40/EU*, 2010.

[6] Jiang et al.,*IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments*, 2008.

[7] ETSI EN 302 571 v1.1.1 (2008-09),*Intelligent Transport Systems (ITS) - Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band - Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive* , 2008.

[8] Hannes Hartenstein and Kenneth Laberteaux ,*VANET Vehicular Applications and Inter-Networking Technologies - Vehicular Applications and Inter-Networking Technologies*,Hohn Wiley & Sons,2009.

[9] IEEE Standards Association,*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 29 March 2012.

[10] IEEE Standards Association,*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6: Wireless Access in Vehicular Environment*, 2012.

[11] Sebastian et al.,*Performance evaluation of IEEE 1609 WAVE and IEEE 802.11P for vehicular communications*, Ubiquitous and Future Networks (ICUFN) 2010 International Conference, 2010.

[12] Stevens et al.,*Benefits and deployment opportunities for vehicle/roadside cooperative ITS*, IET, 2013.

[13] ETSI ES 202 663 v1.1.0 (2010-01),*Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of ITS operating in the 5GHz frequency band*, ETSI, 2001.

[14] Eckhoff et al.,*A performance study of cooperative awareness in ETSI ITS G5 and IEEE WAVE*, IEEE, 2013.

[15] ETSI EN 302 665 v1.1.1 (2010-09),*Intelligent Transport Systems(ITS); Communications Architecture*, ETSI, 2010.

[16] ETSI EN 302 637-2 V1.3.2 (2014-11), *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, ETSI, 2014.

[17] ETSI EN 302 637-3 V1.2.1 (2014-09), *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*, ETSI, 2014.

[18] CAR 2 CAR COM/ARCH, *IEEE 802.11p Extension Roadmap*, November 2017.

[19] IEEE Standards Association,*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band*, IEEE, 2012.

[20] Rohde&Schwarz,*802.11ac Technology Introduction - White paper*, 2012.

[21] Rohde&Schwarz,*802.11ad - WLAN at 60 GHz - A technology Introduction - White paper*, 2017.

[22] Amin Shokrollahi,*LDPC Codes: an introductionm*, Digital Fountain Inc., 2003.

[23] Santumon et al.,*Space-Time Block Coding (STBC) for Wireless Networks*, International Journal of Distributed and Parallel Systems Vol.3, No.4, July 2012.

[24] 3GPP TS 22.185 v14.4.0 (2018-06),*3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for V2X services; Stage 1 (Release 14)*, 3GPP, 2018.

[25] 3GPP TS 23.401 V14.9.0 (2018-09),*3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 14)*, 3GPP, 2018.

[26] 3GPP TS 23.009 V14.0.0 (2017-03) ,*3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Handover procedures (Release 14)*, 3GPP, 2017.

[27] Alcatel-Lucent,*Strategic White Paper: LTE Network Architecture - A comprehensive tutorial*, 2009.

[28] Mitrofanov et al.,*eMBMS LTE usage to deliver mobile data*, IEEE, January 2015.

[29] 3GPP TS 23.285 V14.7.0 (2018-06) ,*3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Architecture enhancements for V2X services(Release 14)*, 3GPP, 2018.

[30] 3GPP TR 36.843 V12.0.1 (2014-03),*3rd Generation Partnership Project;Technical Specification Group Radio Access Network; Study on LTE Device to Device Proximity Services; Radio Aspects (Release 12)*,3GPP, 2014.

[31] Udit Narayana Kar et al.,*An overview of device-to-device communication in cellular networks*, ICT Express, 2017.

[32] Ericcson ,*Ericsson Mobility Report - June 2018*, June 2018.

[33] Yashpalsinh et al.,*Cloud computing - concepts, architecture and challenges*, IEEE, 2012.

[34] Jeon at al.,*Virtualised EPC for on-demand mobile traffic offloading in 5G environments*, IEEE, 2015.

[35] NEC,*Making 5G a Reality*, NEC Corporation , 2018.

[36] Qualcomm,*Spectrum for 4G and 5G*, Qualcomm Technologies, December 2017.

[37] Sharma et al.,*5G Millimeter Wave (mmWave) Communications*, IEEE, pp. 3630-3634, 2016.

[38] 3GPP TR 38.812 V0.1.0 (2018-08),*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on Non-Orthogonal Multiple Access b(NOMA) for NR;(Release 15)*, 3GPP, 2018.

[39] Riazul Islam et al.,*NOMA in 5G Systems: Exciting Possibilities for Enhancing Spectral Efficiency*, IEEE 5G Tech Focus: Volume 1, No 2, June 2017.

[40] https://www.radio-electronics.com/info/cellulartelecomms/cellular_concepts/tdd-fdd-time-frequency-division-duplex.php.

[41] Parkvall et al.,*NR: The New 5G Radio Access Technology*, IEEE, December 2017, pp 24-30.

[42] NGMN Alliance,*A Deliverable by the NGMN Alliance - NGMN 5G White Paper*, 17 February 2015.

[43] Ericcson,*5G Radio Access Network Architecture ? Design Guidelines and Key Considerations*, IEEE, November 2016.

[44] ETSI GS NFV 002 V1.2.1 (2014-12),*Network Functions Virtualisation (NFV); Architectural Framework*, ETSI, 2014.

[45] McKeown et al.,*OpenFlow: Enabling Innovation in Campus Networks*, ACM SIGCOMM Computer Communication Review, Volume 38, Issue 2, April 2008.

[46] Raza et al.,*Dynamic slicing approach for multi-tenant 5G transport networks*, IEEE/OSA Journal of Optical Communications and Networking, Volume 10, pp A77-A90, January 2018.

[47] 3GPP TS 38.300 V15.2.0 (2018-06),*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; NR and NG-RAN Overall Description; Stage 2 (Release 15)*, 3GPP, 2018.

[48] ETSI,*ETSI White Paper No. 28 - MEC in 5G networks*, June 2018.

[49] 5GPPP,*5G Vision - The 5G Infrastructure Public Private Partnership: the next generation of communication network and services*, February 2015.

[50] ETSI GS MEC 003 V1.1.1 (2016-03),*Mobile Edge Computing (MEC); Framework and Reference Architecture* , ETSI, 2016.

[51] ETSI GS MEC 010-2 V1.1.1 (2017-07),*Mobile Edge Computing (MEC); Mobile Edge Management; Part 2: Application lifecycle, rules and requirements management*, ETSI, 2017.

[52] ETSI GS MEC 011 V1.1.1 (2017-07),*Mobile Edge Computing(MEC); Mobile Edge Platform Application Enablement*, ETSI, 2017.

[53] Ha et al.,*openstack for Cloudlet Deployment*, School of Computer Science, Carnegie Mellon University, August 2015.

[54] https://www.etsi.org/news-events/news/1204-2017-07-news-etsi-multi-access-edge-computing-group-releases-a-first-package-of-apis.

[55] ETSI GS MEC 012 V1.1.1 (2017-07),*Mobile Edge Computing(MEC); Radio Network Information API*,ETSI, 2017.

[56] ETSI GS MEC 013 V1.1.1 (2017-07),*Mobile Edge Computing(MEC); Location API*, ETSI, 2017.

[57] ETSI GS MEC 014 V1.1.1 (2018-02),*Mobile Edge Computing (MEC); UE Identity API*, ETSI, 2018.

[58] ETSI GS MEC 015 V1.1.1 (2017-10),*Mobile Edge Computing(MEC); Bandwidth Management API*, ETSI, 2017.

[59] ETSI GS MEC 016 V1.1.1 (2017-09),*Mobile Edge Computing (MEC); UE application interface*, ETSI, 2017.

[60] ETSI GR MEC 018 V1.1.1 (2017-10),*Mobile Edge Computing (MEC); End to End Mobility Aspects*, ETSI, 2017.

[61] ETSI GR MEC 022 V2.1.1 (2018-09),*Multi-access Edge Computing(MEC); Study on MEC Support for V2X Use Cases*, ETSI, 2018.

[62] 5GAA,*Toward fully connected vehicles: Edge computing for advanced automotive communications - White Paper*, December 2017.

[63] ETSI GR MEC 017 V1.1.1 (2018-02),*Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment*, ETSI, 2018.

[64] ETSI GS NFV 002 V1.2.1,*Network Functions Virtualisation (NFV); Architectural Framework*, ETSI, 2014.

[65] 3GPP TR 22.886 V15.3.0 (2018-09)*3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on enhancement of 3GPP Support for 5G V2X Services (Release 15)*, 3GPP, 2018.

[66] European Commission, *5G Infrastructure PPP: The next generation of communication networks will be ?Made in EU*, 2014.

[67] 5G-T, *Report on vertical requirements and use cases*, 2017.

[68] http://openairinterface.eurecom.fr/openair5g-lab.

[69] http://www.openairinterface.org/?page_id=72.

[70] https://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp.

[71] https://en.wikipedia.org/wiki/CAN_bus#Frames.

[72] ISO 11898-1:2015, *Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling*, ISO, 2015.

[73] http://sumo.dlr.de/userdoc/Sumo_at_a_Glance.html.

[74] https://github.com/kosma/minmea.

[75] http://lionet.info/asn1c/basics.html.

[76] ETSI TS 102 894-2 V1.2.1 (2014-09), *Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary*, ETSI, 2014.

[77] https://en.wikipedia.org/wiki/Vehicle_identification_number#Vehicle_descriptor_section.

[78] https://vector.com/portal/medien/cmc/manuals/CANcaseXL_Manual_EN.pdf.

[79] https://gitlab.eurecom.fr/oai.

[80] https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirUsage.

[81] http://asn1-playground.oss.com/

[82] https://www.repubblica.it/economia/2018/10/02/news/asta_5g_incasso_da_superenalotto _lo_stato_avra_6_7_miliardi_per_le_frequenze-207969038/.

[83] https://finanza.repubblica.it/News/2018/10/02/vodafone_completa_la_copertura _narrowband_iot_in_italia-149/.

Appendices

Appendix A

Results script

A.1 FLL scripts

Server script:

```
#!/bin/bash
#server
cd /home/carsim/Desktop
for i in {1..30}
do
oaicc -s -d 172.16.0.6 -rx -prx 5006 -rtt se
mv analytics_rx_se.csv  rwdc${i}_analytics_rx_se.csv

scp /home/carsim/Desktop/rwdc${i}_analytics_rx_se.csv
ue1@172.16.0.4:/home/ue1/results_UErealv2/rwdc${i}

rm rwdc${i}_analytics_rx_se.csv
echo "Waiting .."
read -n 1
done
```

Client script:

```
#!/bin/bash
#client
for i in {1..30}
do
cd /home/ue1/Desktop

oaicc -s  /home/ue1/Desktop/PACKET_LOGS/simulation_2/car4.csv
-tx -d 8.8.8.8 -ptx 5006 -rx -prx 5007 -rtt cl
```

```
mkdir /home/ue1/results_UErealv2/rwdc${i}

mv analytics_rx_cl.csv
/home/ue1/results_UErealv2/rwdc${i}/rwdc${i}_analytics_rx_cl.csv

mv analytics_tx.csv
/home/ue1/results_UErealv2/rwdc${i}/rwdc${i}_analytics_tx.csv

echo "Waiting 1.."
read -n 1
cd /home/ue1/results_UErealv2/rwdc${i}

paste rwdc${i}_analytics_tx.csv rwdc${i}_analytics_rx_se.csv
rwdc${i}_analytics_rx_cl.csv  > rwdc${i}_rtt_test.csv

rm  rwdc${i}_analytics_tx.csv rwdc${i}_analytics_rx_cl.csv
rwdc${i}_analytics_rx_se.csv

echo "Waiting 2.."
read -n 1
done
```

A.2 TTP

```
#!/bin/bash

init=$(($(date + %s%N)/1000000))

until sshpass -p "carsim" ssh -o StrictHostKeyChecking=no
carsim@10.10.10.220 "exit"
&& sshpass -p "evs" ssh -o StrictHostKeyChecking=no
evs@10.10.10.2203 "exit";
do
sleep 0.001
done
echo
ttp=$(( $(($(date + %s%N)/1000000)) - $init ))

echo "Time to provision is: $ttp"
```