POLITECNICO DI TORINO

Master degree course in Communications and Computer Networks Engineering

Master Degree Thesis

Study of Anti-Congestion algorithms for autonomous and connected vehicles

Supervisor
Prof. Carla Fabiana Chiasserini
Prof. Francesco Paolo Deflorio
Ing. Giuliana Zennaro

Candidate
Carla Amatetti

Academic year 2017-2018

Contents

List of Figures

Ringraziamenti

Desidero ringraziare la Prof.ssa Carla Fabiana Chiasserini, relatore di questa tesi, per la disponibilità dimostratemi, e per l'aiuto fornito durante lo svolgimento del lavoro. Un sentito ringraziamento all'Ing. Giuliana Zennaro, responsabile in azienda della tesi, per la disponibilità, la pazienza e il tempo dedicatomi quotidianamente, facendomi crescere professionalmente e umanamente nell'esperienza lavorativa. Desidero inoltre ringraziare l'Ing. Paolo Bande, che mi ha aiutato nella risuluzione di tutti i problemi quotidiani. Un profondo ringraziamento ai miei genitori, che, con il loro incrollabile sostegno , mi hanno permesso di raggiungere questo traguardo. Un sentito ringraziamento al Centro Ricerche Fiat, tutti i membri del "laboratorio", Andrea, Giuseppe, Francesco, Alex e Mirko, per tutto quanto hanno fatto per me durante il periodo di stage. Un ultimo ringraziamento ai compagni di studi, per essermi stati vicini sia nei momenti difficili, sia nei momenti felici: sono stati per me più veri amici che semplici compagni.

Abstract

Modern vehicles are becoming smarter thanks to the continuous development of new Intelligent Transport Systems. For example, with some new features is possible to detect pedestrians on the road and automatically come to a stop avoiding a collision. With some other is possible to obtain information about traffic congestion through the cellular network and suggest the driver another route to save time. Nevertheless, drivers are always looking for a safer, cleaner, and more efficient way of travelling . In particular, the applied research industry is investigating on opportunities and benefits that can be brought by the introduction of autonomous vehicles (at whatever level of automation), because are considered safer than the guide of a human being. Accidents due to men are mostly caused by distracted driving, undecided performance and speeding. Keeping in mind the potential of autonomous and connected vehicles and the new needs that society faces, the thesis consists of the development of an algorithm, which is used in event of accident on highway with a fleet of connected and autonomous vehicles. It has the purposes: 1) to optimize flow and speed of cars on the lanes avoiding other accidents or slowdown and 2) to decrease the waiting time of the cars behind the obstacle without a sacrificial approach, when cellular vehicle-to infrastructure (C -V2I) is adopted as a communication technology. The vehicular mobility is obtained thanks to Simulation of Urban MObility (SUMO), a traffic simulator. The infrastructure and the vehicles are represented in the application level of the open source OMNeT ++ network simulator, through the simple modules client and server. In particular, it is considered an LTE -A network with an eNodeB (eNB) located at the center of the topology. The server, hosting the developed algorithm, is located inside multi-access edge computing (MEC), which is in the eNB. The LTE-A network runs thanks to SimuLTE, simulator for LTE/LTE-Advanced cellular networks in Frequency Division Duplexing (FDD), representing the Radio Access Network and Evolved Packet Core. It is based on OMNeT + + and exploits the INET framework. The vehicles send CAMs (Cooperative Awareness Message) messages to the server which takes decision about the single car with the aim to improve the traffic conditions. To verify the goodness of the algorithm, the results obtained by its application are compared with those given by the application of ALINEA, a ramp metering algorithm, and with those obtained by studying the behavior of the simulators only, in the same traffic conditions. The results show, that in the case of algorithms application, the possibility of accidents and other slowdowns is significantly reduced.

Chapter 1

Communication technologies

1.1 LTE-Adavenced

LTE – A is not new technology but it is an evolution of the LTE standard. The first phase of LTE is define in Rel.8, providing the first version of LTE network with its initial set of functionalities on both network and user equipment side. The second phase is represented by Rel 9, which contains a set of improvements and developments. LTE-A appears for the first time in Rel. 10, offering an improved performance of the previous releases, such as increased capacity and bit-rate, enlarged spectral efficiency, enhanced performance at edge of the cell and number of subscriber which are active at the same time.

Going into details, below are described the characteristics of the release 10:

- Carrier Aggregation: the goal is to increase the bit rate, enlarging bandwidth. The components of the carrier aggregation are called component carrier (CC). It allows the aggregation of maximum five CC, making a maximum band of 100 MHz. The are two ways to achieve the carrier aggregation: the simplest one is obtained using the contiguous stream of channels within the same band and it is called intra-band contiguous.The second options considers non-contiguous allocations: the channels belong to the same operating bandwidth or to different bands. It can be used with FDD and TDD versions of LTE.

- Higher order MIMO: with Multiple Input Multiple Output the whole bit rate rises thanks to multiple transmission and receiver antennas. LTE can support 4x4 MIMO, but the most common configuration is 2x2. The major change of LTE-A is the introduction of 8x8 MIMO in DownLink (DL) and 4x4 in UpLink (UP). With higher order MIMO, throughput and spectral efficiency are enhanced.

- Relay nodes and Heterogeneous networks: the Rel 10 is characterized by the presence, in the same network, of small and large cells with the goal to increase the capacity and the coverage at the edge of cells.This is possible through the introduction of relay nodes, which are base station with low power and behave like repeater to improve the quality of the signal. They are connected with eNB via wireless interface and are cheaper than the installation of a new eNB. The concept of Relay nodes leads to the idea of heterogeneous network (HetNet).

> HetNets make possible networks of varying cell sizes, output power and radio access technologies to work together with the goal of boostinng network coverage and capacity. [9]

- Enhanced Inter-Cell Interference Coordination: is the first procedure to reduce and manage the interference in HetNet.
- Coordinated Multipoint transmission: is introduced in Rel. 11 in order to enhance the performance of the network at edge of the cell. In CoMP several eNBs work,coordinating with each other, in order to avoid interference with other transmission signals, both in transmission and in reception.

1.1.1 Realise 12

The main enhancements of Rel. 12 can be divided in four groups:

- enhanced small cells, such as higher order modulation, dual connectivity, self configuration and cell discovery.
- Carrier aggregation, which includes 2 uplink carriers, 3 downlink carriers and FDD/TDD aggregation.
- MIMO
- New and enhanced services

as it is possible to see in the following picture:

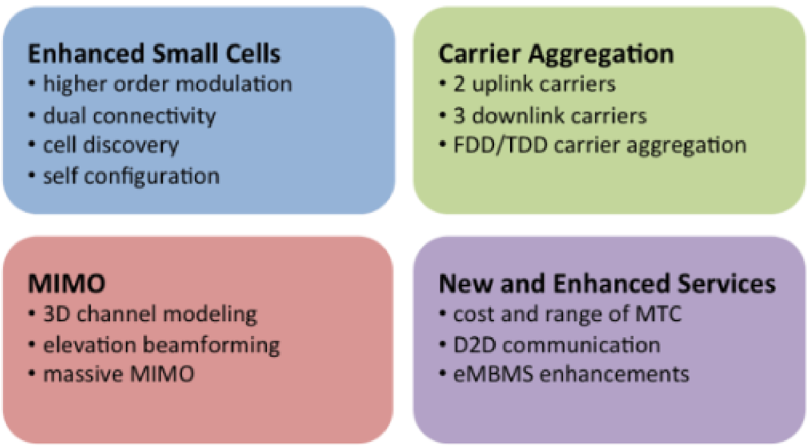

Figure 1.1: Rel.12 features [5]

One of the strengths of Release Rel-12 is the development of the small cells, because they allow the massive reuse of the spectrum. Going into detail, thanks to the small cells the mobile user is closer to its base station and the close proximity provides higher Signal-to-Noise Ratio (SNR). This means that is possible to introduce higher order cardinality, such as 256 QAM. In this way there is a significant spectral efficiency improvement. The basic idea of the enhanced small cells is represented in Fig. 1.2:

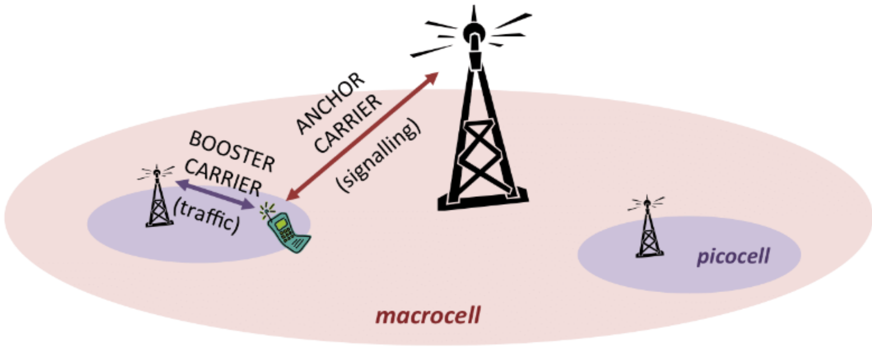

Figure 1.2: Enhanced small cell's principle [4]

In Fig. 1.2 the are both the macrocell and the picocell. The first one supplies an anchor carrier, which provide to the user a strong connection in the wide area. Instead the latter gives a booster carrier, which makes possible a reliable delivery of the user traffic in the local area. The presence of the two types of cells provides that a UE has an efficient connection as it moved in the area of the network coverage and if it is required of period of high communication, traffic can be supported by the local booster carrier.

This solution can be used both when the macro and pico cells are deployed in different frequencies (causing a decreasing of the interference problem) and when they are used in the same band. If the range in not important, picocells are optimally suited to new spectrum at higher frequency [15].

Another important enhancement of the Rel. 12 is the evolution of MIMO antennas, focusing on the elevation of beamforming and massive MIMO.

The latter implies the use of large numbers of elements of base station antenna, in order to have a narrow beams of radiation. It maximizes the SNR, experienced by each mobile terminal. The massive MIMO is a very useful technique when systems move into higher frequency, with the goal to compensate for the bigger radio signal path loss at those frequencies. For what concerns the elevation beamforming, it implies the use of active antenna arrays at the base station. In this way antenna beams are directed both in vertical and horizontal planes. This technique is useful in the urban environments, where the users are located in tall buildings, by allowing the network operator to increase the number of users served by a cell and to improve the quality of the services given.

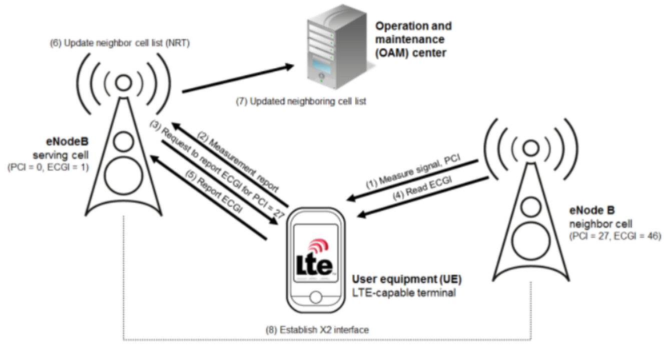

Figure 1.3: Example of Self-Optimizing Networks [16]

For what concerns the new and enhanced services, they include:

- machine type communication, that cover all the low-cost, low power and simple devices which use the mobile network to send and receiver data;
- Proximity Services, through which the closest users communicate in a direct way;
- UE Receiver Enhancements, which identifies Network Assisted Interference Cancellation and Suppression (NAICS) receivers, a new type of receivers, that exchange semi-static cell configuration between the eNB neighbors;
- Self-Optimizing Networks (SON) (in Fig.. 1.3), which represents an attempt to construct and manage mobile networks through interoperability among characteristics already present and new additions features;
- HetNet Mobility, that gives offload benefits for the UE mobility and it is applied both in a single carrier and in multi carrier conditions;
- Multimedia Broadcast/Multicast Services, which provide maintenance to all the interface and node of MBMS, in case of node or interface fails;
- Local Internet Protocol Access and Selected Internet Protocol Traffic Offload (LIPA/SIPTO) sited in the local network, allow offloading of internet traffic from the RAN node through an embedded Public Data Network Gateway into private network;
- Enhanced Interference Management and Traffic Adaptation, used to match the traffic exploiting the two different duplex modes, TDD and FDD;
- FDD-TDD Carrier Aggregation provides an increase of the user throughputs, because the UEs can control the TDD and the FDD spectrum together.

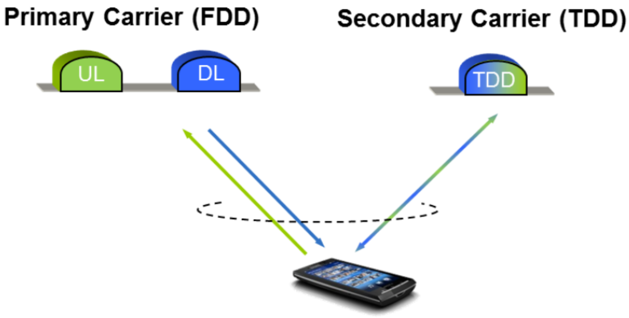

Figure 1.4: FDD-TDD Carrier Aggregation (CA) [3]

1.1.2 Release 13

The main enhancements of Rel. 13 relate to features and services of the previous releases. In particular, this release pays attention to the LTE in unlicensed spectrum. This term means the permission granted to the cellular network operators to offload some of their data traffic by accessing the unlicensed 5 GHz frequency band, such as the 5 GHz band used by 802.11a and 802.11ac compliant Wi-Fi equipment. Other Improvements concern the Carrier Aggregation, where the number of carrier aggregations increases by 32 CC in order to provide higher data rates and to add up a large number of carriers in different band. Then they concern Machine-Type Communication, defining a new class of UE with low complexity, low transmit power and low bandwidth. Other key points of this release are the enhancement of D2D, supporting new requirements for the definition of the Mission critical Push-to-Talk (MCPTT), and the full-dimension MIMO, which will support the future capacity demand.

1.1.3 Release 14

This release is the starting point of 5G in the 3GPP. Hence, beyond the LTE improvement, a new radio access technology is developed. These two technology together will be part of 5G radio access.

The Rel. 14 is characterized by a high level of ambition, since it must meet the 5G requirements. Examples of these enhancements are the introduction of new use cases, such as the Intelligent Transport Systems, including vehicular-to-vehicular and vehicular-to-infrastructure communications with a latency reduction. The latter is important for an improved end-user experience and to fully exploit the high data rates and can provide better support for new use cases. In addition to this, other improvements relate to Cellular Intern of Things, Machine-Type of Communications and Locating reporting.

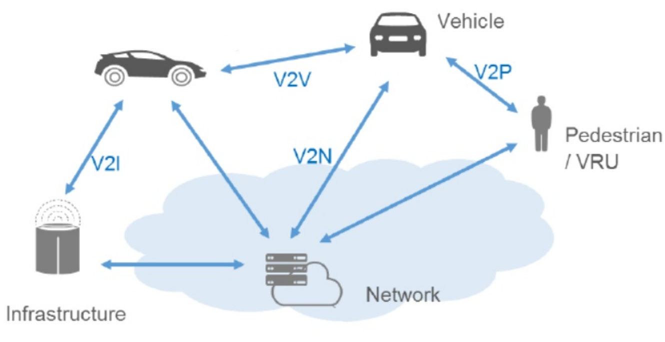

Figure 1.5: V2X [1]

1.2 5G

With the enhancement of 5th Generation of mobile communications system there will be huge changes in the industry and the society.

> Networks present until today have provided mobile broadband connectivity for smartphone, tablets and laptops [8].

With their relative architecture are offered services and content, such as video, web browsing and voice. But the are new applications and services with different characteristics to be delivered and for this reason architecture evolution is necessary. The first version of 5G is expected to be available in 2018 and the complete solution present in the market in 2020. For this year and beyond are defined three scenarios, each of which requires different network services. These scenarios are:

- Enhanced Mobile BroadBand (eMBB): it characterizes services with high data rates, like new application related to multimedia consumption and collaborative working and social communications (e.g video in all its various forms).
- Massive Machine Type Communications (mMTC): it is a class that includes applications with high demand for connection density, like services for smart city.
- Ultra-Reliable and Low Latency Communications (URLLC): it groups all the applications characterized by ultra-low latency, including all the real time services, which require end-to-end network latency of the order of milliseconds and with a single digit.

The first phase of 5G started in Rel. 14 with the study of key issues and requirements. 3GPP has introduced Non-Standalone and Standalone architectures. This latter architecture would be composed by 5G radio (5G NR) and 5g core (5GC), while non-standalone architecture would exploit the existing LTE radio and core network (EPC) as an anchor for mobility management and coverage while adding a new 5G carrier. The initial 5G deployments are likely to be based on 5G-NSA architecture, which is also called 5G architecture option-3. In order to support 5G-NSA, the UE would be Dual Connectivity with New Radio (DCNR) capable. In the Non-Stand-Alone (NSA) are specify only the radio and associated enhancements.

> The 5G radio will work with 4G core network and is expected to give hotspot coverage with 4G providing the overlay network. This is also known as E- UTRA-NR Dual Connectivity (EN-DC) [13].

So two distinct phases can be identified:

- Phase 1:

  > as agreed by International Telecommunications Union (ITU), in the first phase of complete 5G specification will be covered the radio, core, security and all associated specifications. Specifications will be completed in June 2018 [13].

- Phase 2: in the second phase will be specified technology specifications for Massive Machine-Type Communication (MMTC) and Ultra-Low Latency Communication (URLLC).

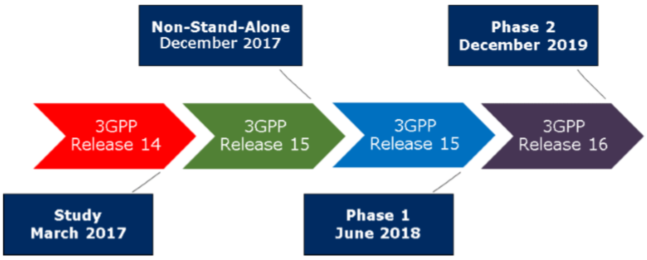

Figure 1.6: Roadmap of 3GPP activities

The goal of New Radio (NR) is to bring much higher data rates with lower latency and the base station, called gNB, is divided in two: central unit, named gNB-CU, and distribute unit, named gNB-DU. The strength of NR is that it will support both high and low frequency bands (high $\geq$ 6GHz and low $\leq$ 6GHz). 3GPP identified the key performance indicators (KPI) for both the Phase 1 and the Phase 2 and they require that for eMBB the peak data-rates is 20Gb/s for downlink and 10Gb/S for uplink. For what concerns the latency of UTLCC it will be 0.5 ms and for eMBB 4ms. MMTC solutions should a battery life of 15 years.

1.2.1 The NG-RAN Architecture

The key element in the 5G Radio Access Network (also called Next Generation RAN or NG-RAN) architecture is the distributed concept from the beginning. NG-RAN includes gNBs and can include ng-eNB (evolved 4G base station). The gNB is divided in gNB-Central Unit (CU) and gNB-Distributed Unit (DU), where the Central Unit can be located in the cloud infrastructure. The gNB-CU hosts higher layer protocols, like Radio Resource Control (RRC), Service Data Adaptation Protocol (SDAP) and Packet Data Convergence Protocol (PDCP). While gNB-DU hosts Radio Link Control (RLC), Medium Access Control (MAC) and Physical (PHY) layers. The fact that it is closer to the end users implies lower latency, which is the major requirement for the application related to URLLC, such as autonomous vehicles. The NG-RAN architecture is made up by several logical nodes gNB and ng-eNB. The firsts supply termination of NR- User-plane (UP) and Control-plane (CP) towards the User Equipments, while ng-eNB brings LTE UP and CP protocols termination to UEs. The difference between LTE base-station and ng-eNB is that the latter can work with 5G core. In the NG-RAN architecture the are three different interfaces:

- NG interface: provides many functions, such as handover and bearer management used for communication in the radio interface [13].

- Xn interface: provides handover between gNBs and dual connectivity among different radio technologies
- F1 interface: inside gNB, brings UP in the GPRS Tunneling Protocol for User Data (GTP-U)

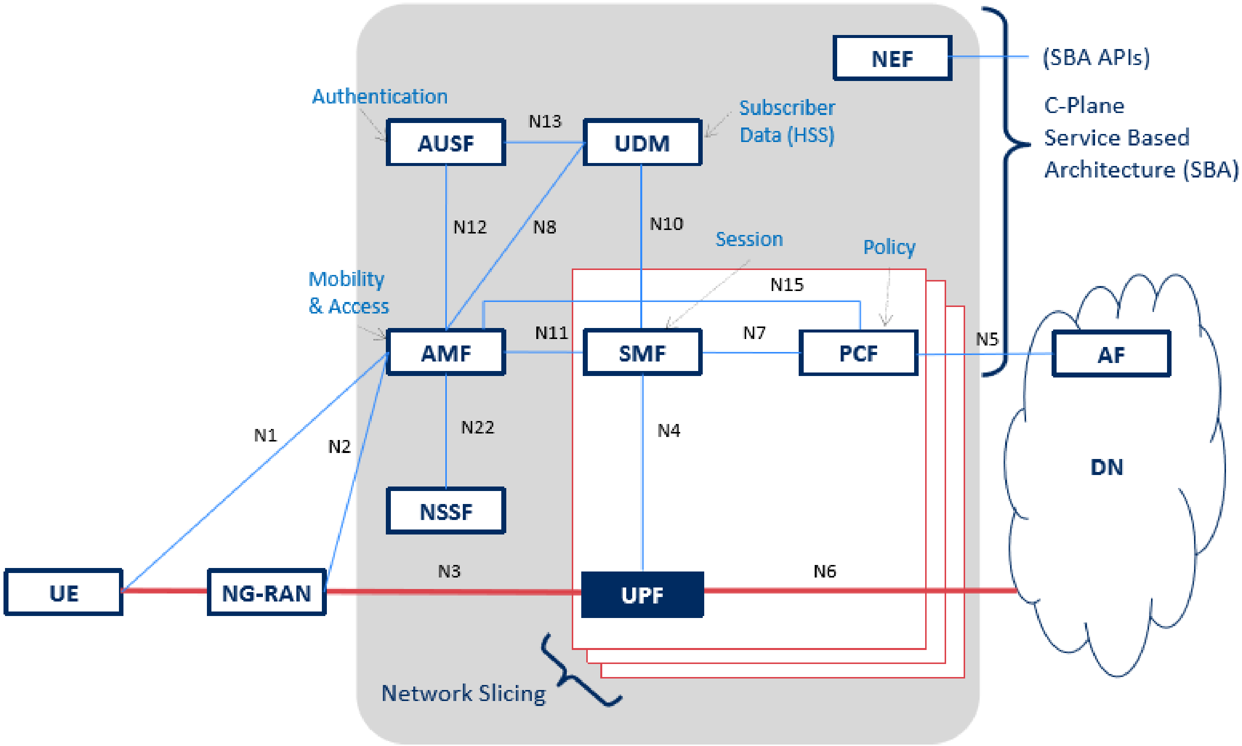

Figure 1.7: 5G system's architecture (5GS) [13]

In Fig. 1.7 in grey it is indicates the core network (CN). The Access and Mobility Function (AMF) supports mobility within CN and RAN and supports authentication of the UEs with the Authentication function (AUSF) and Unified Data Management (UDM), which stores the subscriber data. The Session Management Function (SMF) manages every session in 5G and receives policy from Policy Control Function (PCF). The Network Slice Selection Function (NSSF) provides network slicing. Through the APIs, used to interact with the network functions, the network is exposed by Network Exposure Function (NEF). The Control Plane is made up by AMF, AUSF, UDM, SMF, PCF, NEF and AF. Instead the User Plane include only the User Plane Function (UPF), with the task to connect the external parties to the Data Network. The CN has the following features:

- Virtualization;
- Unified service based architecture and interface;
- Control plane and User plane are separated;
- Mobility management and session management function decoupling;

- New quality of services (QoS) architecture;
- Network slicing in order to support new business field.

The control plane architecture is enhanced as compared to one of LTE. The 5G core (5GC) includes the Service Based Architecture (SBA), which allows the control plane nodes to access services provided by the other logical nodes. Since all the control plane nodes are connected with an interface bus type, called Service Based Interface, each control plane function can communicate with each other. This type of network topology makes easier the signaling of the control plane, introducing new functionalities obtained by the coordination of control plane functions.

1.2.2 Migration and Interworking

The introduction of the 5G network cannot be achieved in the day one providing 100% coverage. It obeys a migration step from 4G to 5G, this means that 5G should work with 4G network through an interworking solution. The starting step is known as Non-StandAlone (NSA) or E-UTRAN-NR Dual-connectivity (EN-DC).

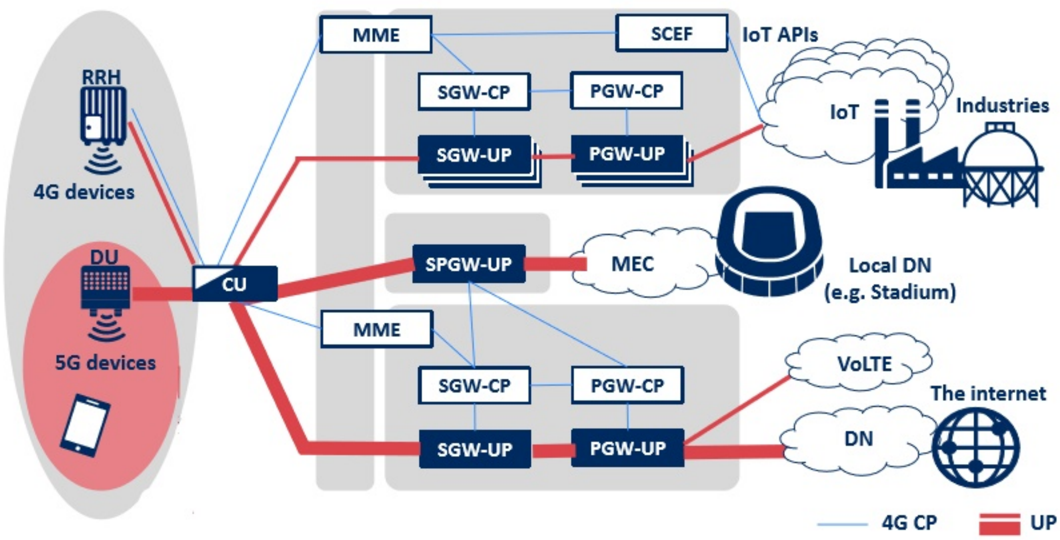

Figure 1.8: EN-DC architecture [13]

Reuse of existing networks

The feature of NSA is that it makes 5G based-service available without network replacement, because it use the existing LTE and Evolved Packet Core (EPC). The latter requires an upgrade in 3GPP Rel.15 in order to support EN-DC. The EPC provides low latency communications as well as enhanced mobile broadband and massive machine type communications. Since the objective is to guarantee low latency communication, the EPC is improved with new capability to handle new standardized QoS class identified.

The EN-DC uses LTE as the primary radio access technology and the new radio access technology (i.e NR) is used as secondary radio access technology with UE connected to both radios thus dual connectivity [13].

Radio Access Network

There are two types of migration path through which the 4G system can be evolved. The first one is that exploits the LTE radio and gradually to introduce the NR, i.e. EN-DC. The 5G can be available only in the areas where NR is available. The first migration path consist of replacing the EPC with 5GC, i.e the NG-EN-DC.

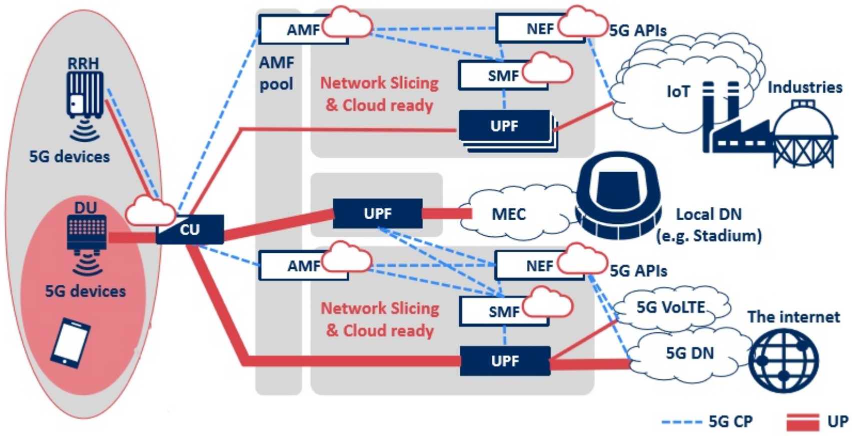

Figure 1.9: NG-EN-DC architecture [13]

In this way are used the network slicing and the QoS handing. This solution will add better user experience, flexibility ad more business opportunity for the verticals. During the transition phase two types of RAN-CN connections will be at the same time in the network: the LTE eNB connected to the EPC and the LTE eNB connected to 5GC. Sevice continuity can be guaranteed through the seamless mobility between EPC and 5GC and this system can be evolved to the standalone NR. The second type of passage is represented by the introduction of the NR with 5GC, the so called NR standalone. In this way the 5G services achieve excellent performances thanks to the network slicing and the new QoS handling. This path can be evolved by using the LTE radio, as happens in the case of NR-E-UTRA (NE-DC).

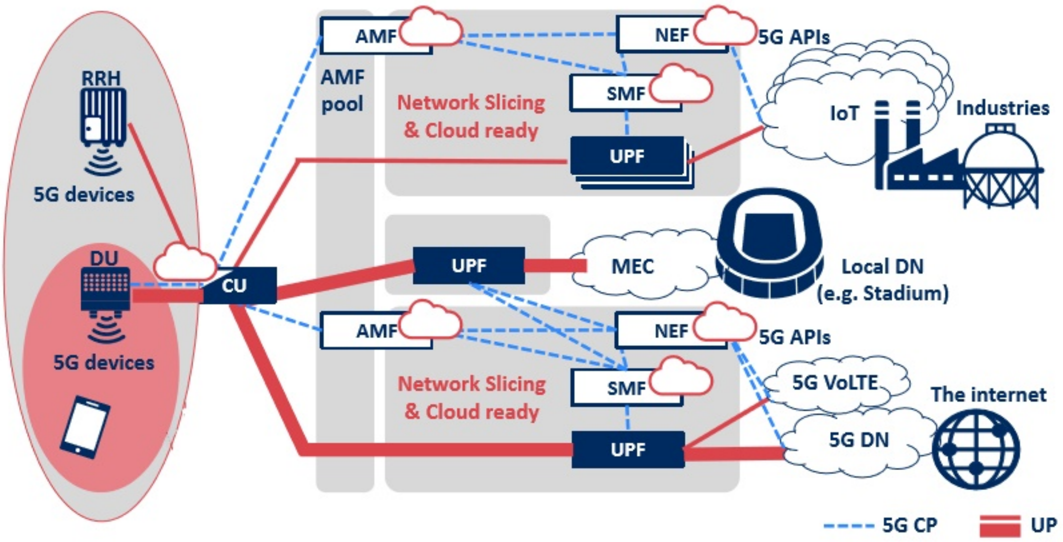

Figure 1.10: NE-DC architecture [13]

Core Network

For what concerns the migration of the core network, it follows the migration path of the RAN. With the aim to decrease unnecessary deployment during the migration, the 5GS entities are integrated with the EPC entities.

1.3 Multi-Access Edge Computing (MEC)

Multi-Access Edge Computing (MEC) is an emergent architecture, where cloud computing services are extended to the edge of networks. It offers an IT service environments at the edge of the network, bringing ultra-low latency, high bandwidth and real-time access to radio network information. It allows several types of access at edge, such as wireline. In this way the operators can open their networks to new ecosystems. The main purpose of the MEC is to reduce the congestion of the network and increase application performance by achieving related task processing closer to the user. In addition, the delivery of content and application to the users is reduced thanks to MEC. It gives new vertical business slices and services for consumers. MEC offers cloud computing capabilities within the Radio Access Network (RAN) and connects the user directly to the nearest cloud service-enabled edge network. Deploying MEC at the base station enhances computation and avoids bottlenecks and system failure.

The following use cases have been identified:

- Augmented Reality (AR);
- Virtual Reality (VR);
- connected cars;

- location services;
- Internet-of-Things (IoT);
- optimized local content distribution and data caching.

As standardized by [6], The MEC framework consists of the following parts, shown in Fig. 1.11:

- host, made up by:
  - mobile edge platform;
  - mobile edge applications;
  - virtualisation infrastructure;
- system level management;
- host level management;
- network level entities.

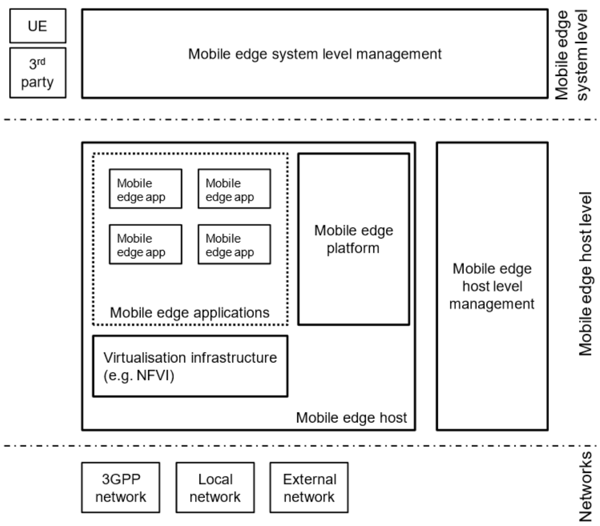

Figure 1.11: MEC framework [6]

It is a common idea that MEC is 5G-only feature, but it still present in the LTE networks. In this way a MEC host deployed in a 4G network can be reused to support 5G services as well.

Deploying MEC in 4G networks

There are four scenarios, identified by [7], where the MEC platform can be placed:

1. Bump in the wire: includes all the scenarios where the platform installation is located between the base station and the mobile core network.

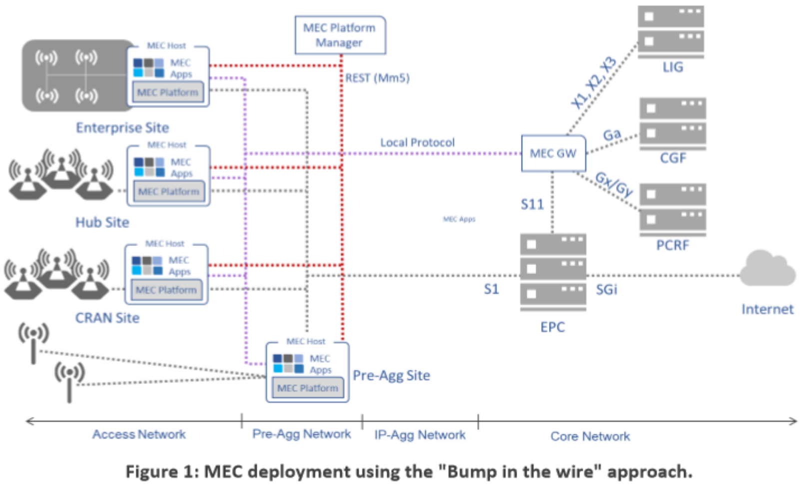

Figure 1.12: Bump in the wire [7]

2. Distributed EPC: in this deployment the MEC host incorporates all the parts of the EPC and the MEC data plane is located in the SGi interface.

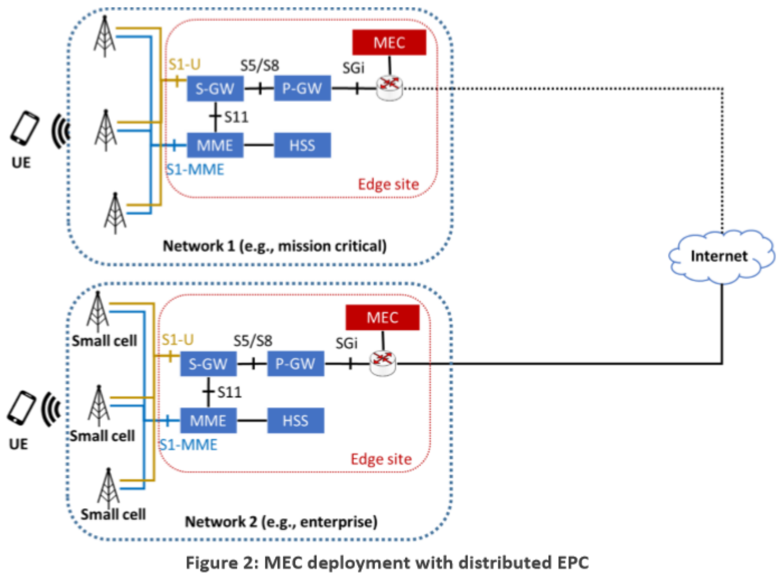

Figure 1.13: Distributed EPC [7]

3. Distributed S/PGW: it is similar to the previous one, but with the exception that the SGW and PGW entities are located in the edge site and the control plane functions are sited in the core site of the operators.

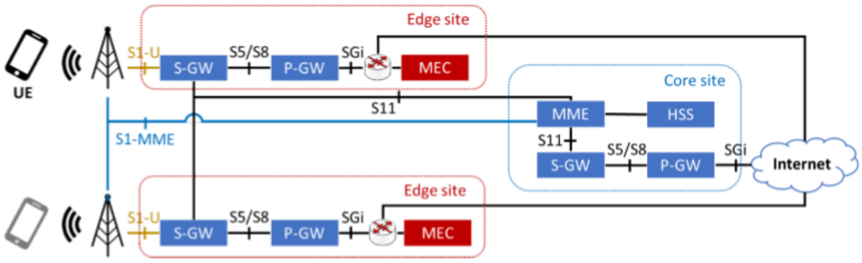

Figure 1.14: Distributed S/PGW [7]

4. Distributed SGW with Local Breakout (SGW-LBO): in this scenario the MEC host is co-located with the SGW. It is a new architecture for MEC, born from the need of operators to control better the granularity of the traffic that needs to be delivered. The basic idea is to make the users able to get to the MEC applications and operator's core site application in a selective way over the same APN.

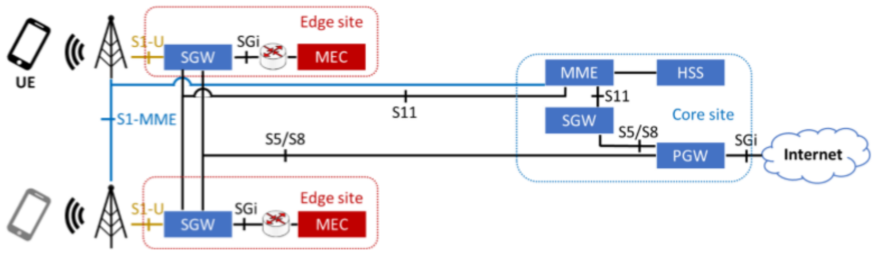

Figure 1.15: SGW-LBO [7]

Deploying MEC in 5G networks

As said before, the basic idea is to reuse the existing deployed systems for the transition to 5G networks. This, of course, is also valid for the MEC. In the 4G MEC deployment, the choice of the position of MEC at the edge site is left to the operator, it is the same for what concerns the 5G MEC. There are several architecture for the 5G MEC migration. For example the MEC host, which includes the 4G core network functions, can be transformed to support 5G by software upgrading the relevant network functions.

In the transition to 5G the MEC functionalities introduced with the 4G technology are preserved, fulfilling key requirements such as [7]:

- reusing the edge computing resources;
- interaction with 5G control plane;
- integration with the 5G network.

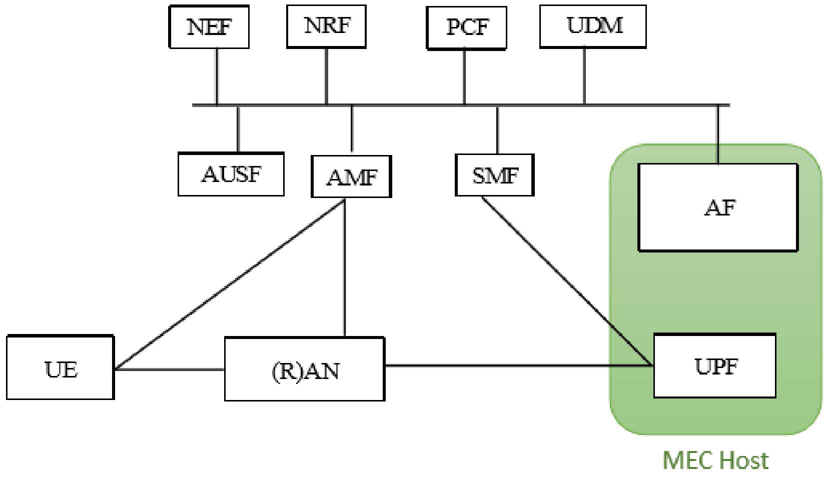

Figure 1.16: Example of the mapping of the MEC with 5G architecture [7]

1.4 Vehicle-to-Infrastructure (V2I)

Vehicle-to-infrastructure (V2I) is a communication technology, which allows vehicles to share information with the infrastructure, represented by components that support a highway system of a country, such as overhead RFID readers and cameras, lane markers, traffic lights, signage, streetlights and parking meters. This type of communication technology is typically wireless and bi-directional: the vehicle send operational data and critical safety to the highway infrastructure over an ad hoc network and vice-versa. The interaction between vehicles and infrastructure is intended to avoid or mitigate accidents but also to enable a great variety of other mobility, environmental and safety benefits.

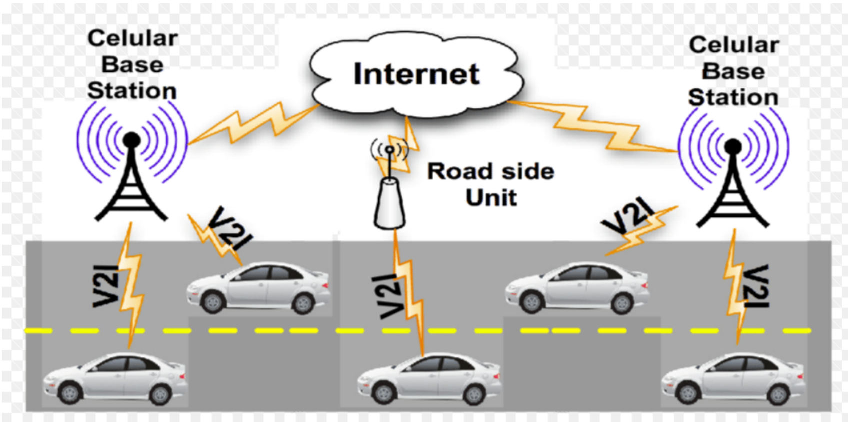

Figure 1.17: V2I

In the field of Intelligent Transport System (ITS), an important task is to capture infrastructure data, through V2I sensors, and to inform travelers about traffic congestion, road conditions, construction zones, accidents and parking availability through real-time advisories. One particularly crucial advantage of this technology is the ability to adjust traffic signal phase and timing (SPAT), in support of increasing fuel economy and traffic flow and delivering warnings to the travelers and safety advisories. All these cooperative services have common communication requirements:

- Periodic status exchange. With the periodic delivery of messages related to location, speed, identifier, etc the ITS services know the status of vehicle and roadside terminals.
- Asynchronous notifications, used to inform about a specific service event. With respect to the previous status messages, the reliable delivery of these messages to a single terminal or a group of them is usually a key requirement.

Examples of the usage of the first communication type are remote vehicle monitoring, including periodic status data from vehicles and safety services like cooperative collision avoidance. Asynchronous notifications are used in safety services, such as post-collision warning or slippery pavement. These two types of messages are defined by the ETSI as Cooperative Awareness Basic Servic, including the Cooperative

Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM). CAMs are periodically broadcasted by each vehicle to other closer vehicles or infrastructure to provide information of speed, position, temperature, and basic status. While, DENMs are event-triggered messages broadcasted to alert drivers or pedestrians of a dangerous event, delivered by the user of the road.

Figure 1.18: Example of V2I application

The V2I communications can be applied to all the types of vehicles and roads and make the infrastructure "smart". For example they can include algorithms which exploit the information exchange between vehicles and infrastructure to perform calculations. The latter need to know in advance if dangerous situations occur and to inform drivers with alerts and warnings with specific actions. The basic V2I structure should consists of the following parts:

- Vehicle On-Board Unit (OBU)
- Roadside Unit (RSU)
- Safe Communication Channel

The first one represents the vehicle side of the V2I system.

> An OBU is composed by a radio transceiver, such as DSRC, a GPS system, an applications processor and interfaces to vehicle systems and the vehicle's human machine interface (HMI). It provides communications between the vehicles and the RSUs and between the vehicle and other closer vehicles. The OBUs may regularly sends status messages to other vehicles in order to support safety applications between vehicles and also public applications. [14]

RSUs is the road side of the V2I and it can be located at intersections, interchanges and other locations, such as petrol stations. They provide the interface to vehicles within their range.

> Such as the OBU, the RSU is made by a radio transceiver, like typically DSRC or WAVE, an application processor, interface to the V2I communications network and GPS unit. [14]

The RSU is connected to the V2I communications network. Through the latter, it can send private data to and from the vehicles. An other important task of the RSU is to manage the different level of priority of messages to and from the vehicle. Also the OBU can identify the priority level, but it must also be set within the road-side unit to be sure that available bandwidth is not exceeded. Local and vehicle-to-vehicle safety applications have the highest priority; messages related to the different public and private network applications and entertainment messages have lower priority.

LTE has natural benefits in order to provide V2I communications thanks to its high penetration rate, comprehensive QoS supporting, high data rate and large coverage. But LTE faces challenges when applying in V2X communications:

- In dense areas or during peak hours, vehicles and infrastructure exchange informations, making the network heavy.
- Since the periodic message are broadcast, without the congestion control scheme, message latency is caused by the paylod of the messages. This challenges the capacity of the LTE.
- Rapid network topology change, frequent and fast handover, heavy broadcasting with QoS guarantee are challenges in the LTE network because downlink peak rate of LTE is 100 Mbps with speed of 350 km/h.

The basic idea is to combine LTE with direct communication capability between vehicles to become an integrated V2X solution. The Rel. 14 of the LTE supports more benefits to the V2X and with 5G telecom industry and vehicle manufacturers will be connected in order to work together to provide end-to-end solutions for future mobility and transportation services

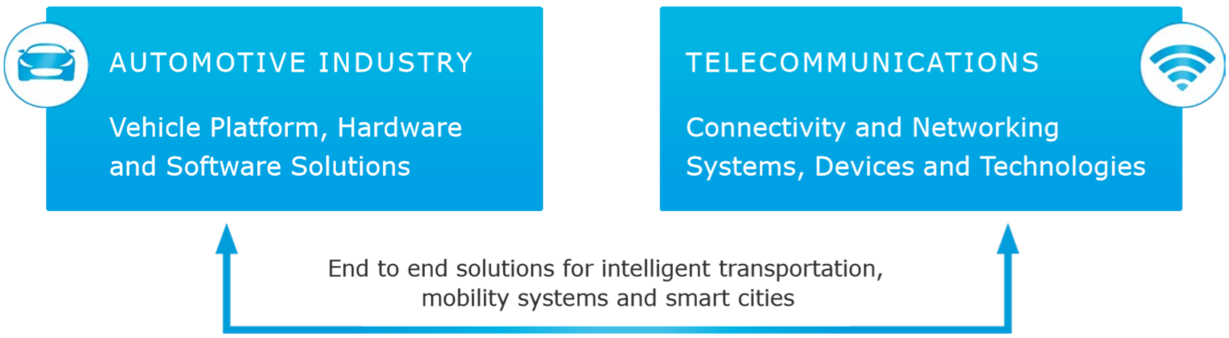

Figure 1.19: Interconnection of telecom industry and vehicle manufacturers

Chapter 2

Simulators

In these twenty years, two independent research topic have been developed largely: cellular communications and vehicular networks. The common goal of the two research fields

> are the increasing of the end-user bandwidth and providing ubiquitous coverage [12].

For this reason the research in the cellular networks identifies a growing number of use cases. This is clear in the 5G research, whose objective is to provide reliable performance to the largest possible numbers of users, regardless of their positions. According to this, the network needs to be developed in the denser way in order to support high speed mobility in reliable way, to perform fast handover between cells.

> These two fields have common points in the following applications: the vehicle-to-everything (V2X) communications, which include communications between vehicles (V2V) and communication among vehicle and infrastructure (V2I), such as scenarios in which are either or both endpoint of the communication [12]

and vehicular mobility can be a specific case for the user mobility. Vehicular mobility is provided by SUMO and vehicular networks and cellular systems are simulated with SimuLTE, which is a framework of OMNeT++.

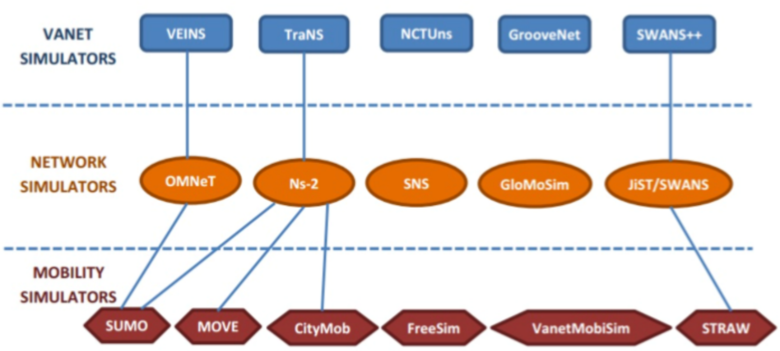

Figure 2.1: Type of simulators involved in the research

2.1 SUMO

Simulation of Urban MObility [2]

is an open source, microscopic simulator developed by the Institute of Transportation Systems at German Aerospace Centered and it is licensed under the Eclipse Public License (EPL) from 2001. There are two types of mobility modelling approach, the first one is the microscopic traffic, where every vehicle driver is defined through many parameters, such as positioning, speed, direction. The second one is the macroscopic traffic, where are used mathematical traffic models with the aim to define relationship among traffic flows feature like density, flow, mean speed etc. SUMO, as said before, is microscopic, because each vehicle running in the simulation is defined explicitly by parameters relative to the single vehicle in a specific file, in the format *.rou.xml*. In order to exploit the feature of this space-continuous and time-discrete simulator, the [2] shows the number of variables associated to each entity in the simulation such as car, people, bus and so on , and the possibility to set/ retrieve their value at each instant of the simulation. For what concerns the map where the vehicles drive, on the hand there is a useful tool NETEDIT, which allows to create your own personal map and other hand SUMO provides the possibility to import an existing map and customize it, if necessary.

SUMO is written in C++ and all the parameters needed to run a simulation are defined in the *xml* files, like *net.xml* and *rou.xml*, used in the configuration file, that load the map and routes with the different types of vehicles.

SUMO is a safe simulator, this means it avoid accidents between vehicles or pedestrians. As it is possible to read in [11] SUMO tracks gap between vehicles that are on the same edge or either fully or partially. There is a MinGap value, that if it reduced below a threshold (2.5 m) a collision is registered. It is possible to deliberately change the parameters that control safety in SUMO, with the possibility of creating accidents by modify the following parameters:

- car – following model: increasing the SpeedFactor the vehicles drives above the speed limit, higher value of acceleration and deceleration leads to abrupt speed changed. Tau is distance in time with the vehicle ahead. Decreasing this value induce a reduction of the gaps to the leader vehicles. If tau is lower than the simulation -step-length parameter, collision may occur.

- Lane-Changing Model: there are five types modes to change lane, such as *lcCooperative*, lower value causes less cooperation among the vehicles during the lane changing; *lcSpeedGainhigher* value implies more overtaking for speed; the reduction of *lcKeepRight* causes less driving on the right. If *lcPuschy* is set between 0 and 1 implies aggressive lateral encroachment and *lcAssertive* if set above 1 causes

    > acceptance of smaller longitudinal gaps in proportion to driver impatience [10].

- Junction Model

This simulator is supported by a strong community.

2.2 OMNeT++

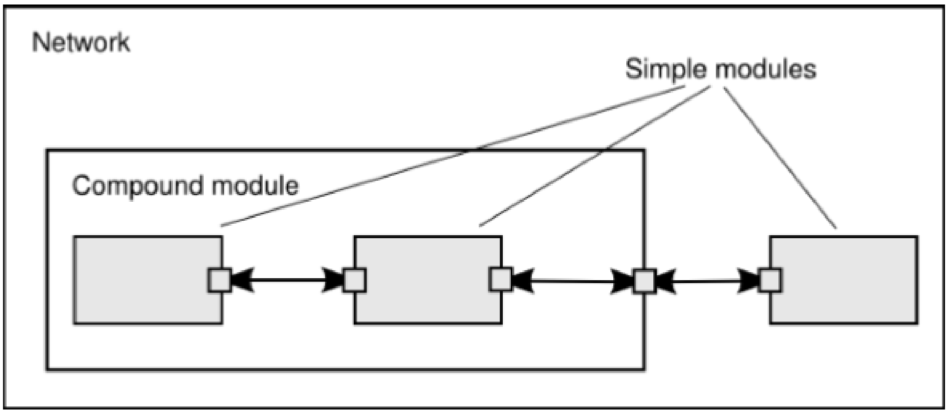

Figure 2.2: Module in OMNeT++ [17]

> OMNeT++ is an extensible, modular, component-based C++ simulation library framework, primarily for building network simulators [17].

It has a generic architecture and it can be used to simulate different situation, not only networks, such as:

- modeling of wired and wireless communication network
- protocol modeling
- modeling of queueing networks
- modeling of multiprocessors and other distributed hardware systems
- validating of hardware architectures
- evaluating performance aspects of complex software systems
- modelling and simulation of any system where the discrete event approach is suitable, and can conveniently mapped into entities communicating by exchanging messages.

Omnet++ is a Discrete Event Simulation framework, this mean that state change events occur only at a discrete time instances and they take zero time to completely execute. A key strength is the modularity of this framework: the basic building block are modules, that can be simple or compound . Modules exchange messages to communicate between each other. The message is sent and receive through a connection, which links the gates of the modules; so the connections acts as interface.

> Connections are characterized by bit rate, delay and loss rate and cannot baypass module hierarchy [18].

At the top of the hierarchy there is the network, a particular compound module, with no gates to the outside world. Model behavior is implemented by simple modules. When a module receives a message, simulation Kernel causes events to handle, which are entrusted to the modules. So each module can run as an independent co-routine. In addition to this, simple module have a function of initialization and finalization. OMNeT++ gives the possibility to define separately model of implementation, description and parameter values. The implementation or behavior is written in C++. The description, such as gates, connections and parameter definition, is expressed in files written in Network Description (NED) language.

> It is a declarative language, that exploits inheritance and interfaces [18].

The strengths of NED are that is possible to write parametric topologies and it can be edited both textually and graphically, switching between the two views at any times. The NED files are very important files and they are characterized by being:

- Hierarchical: each single module can be decomposed in more and simpler submodules.
- Component-Based: simple modules and compound modules are inherently reusable.
- Interfaces: module and channel can be used as placeholder.
- Packages: the whole structure is packaged to reduce the risk of name clashes between different modules.
- Inner types: channel types and module types used locally by a compound module can be defined within the compound module.

The parameters value are declared in initialization (INI) files. These models and studies are separated in OMNet++. INI files generates studies, making the cartesian product of all the parameter value and generating replicas of the same instance with different seeds for the random number generators [18].

2.3 SimuLTE

> SimuLTE [18]

is a simulator for LTE/LTE-Advanced cellular networks in Frequency Division Duplexing (FDD) , representing the Radio Access Network and Evolved Packet Core.

> It is based on Omnet++ and exploits the INET framework [12],

in order to implement all the layer of the IP stack and the main IP nodes for communication networks. The INET has the important task to provide the concept of Network Interface Card (NIC) modules, that can be included in the other modules in order to implement models for the different communication protocols between the devices in the network. In SimuLTE the compound modules are eNBs and UEs, which are connected with each other and with other nodes, such as routers, applications, etc, to create the network. These two modules communicate via message exchange.

> The transmission signals are not explicitly modeled [12].

In order to do this, the Binder module keeps tracks of the Resource Block (RB) used in each transmission slot of 1ms. The User Equipment uses a set of RB for uplink transmission different from the RB of the eNBs, which use it in downlink. All the LTE nodes in the network register to the Binder when the simulation starts and can access it by call in order to share or obtain information that are common.

> These two nodes contain the LTE NIC together with modules that implements upper layers protocols, taken from the INET Between eNB and Internet there is interface via PPP, which allows to send and receive data traffic coming from application servers, implemented as INET standard host .[12].

eNBs are connected together with the X2 interface. Inside both the eNBs and UEs there is the LTE NIC, which implements a complete LTE protocols stack simulating submodules for each layer, called Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), MAC and PHY With SimulTE is possible to simulate both communication between Ues and eNBs and among Ues. In case where two UEs are served by the same eNB, there are two possible types of communications between each other: using the latter as rely or in direct mode, named Device-to-Device (D2D) communication.

Chapter 3

Ramp metering algorithm

A ramp metering algorithm controls a traffic corridor, made up by the freeway and its entrance/exit ramps control, by getting into the main lane a vehicle or a platooning from the ramp. Ramp metering has many potentially advantages:

- to eliminate the congestion with the improvements of freeway mainline flow, due to access control;
- to reduce the travel time and the number of accidents, fuel consumption emission and vehicle operating costs;
- to regulate the input demand on freeway system in order to have a corridor system operationally balanced.

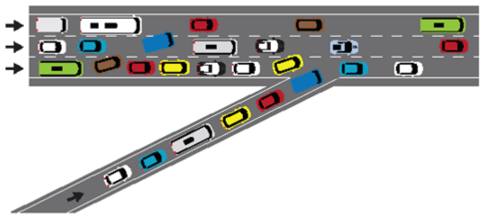

Figure 3.1: No ramp metering algorithm

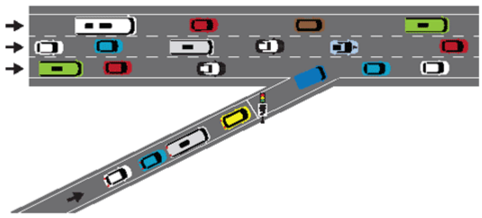

Figure 3.2: Application of ramp metering algorithm

Analyzing a freeway corridor system, it is composed by a hierarchical structure, where there is the mainline backbone, the freeway and the on-ramp and off-ramp.

> The dynamics of the traffic on-ramp affect traffic performance of the part of the mainline freeway downstream to the on-ramp,instead the part upstream to it, unless the on-ramp itself becomes a source of congestion [19].

The hierarchical structure of a corridor and the influence of the on-ramps to the main lane of the freeway determine the type of ramp metering algorithm to be used. There are three types of control systems that define many ramp metering control algorithm:

- isolated systems: control is applied to only a single ramp, taking into account traffic measurements only near a single ramp and does not take into account rates on other ramp.

- coordinated systems: take measurements and control several ramp meters as a system in order to optimize the traffic over the area;

- integrated systems: it has different types of control measures, such as ramp metering, signal timing and route guidance via variable message signs.

3.1 3.1 Isolated ramp-metering algorithms

The ramp metering rate of an on-ramp is determined based on its local traffic conditions, taking into account the following parameters: traffic flow, occupancy, travel speed and sometime queue over-flow on the metered ramp. The algorithms that belong to this category are: Zone Algorithm (Stephanedes, 1994), ALINEA (Papageorgiou et al, 1997) and Neural control algorithm (Zhang e al 1996).

ALINEA and the Neural control algorithm use feedback regulation in order to maintain the target occupancy, that is chosen to be the critical occupancy. Both the algorithms are flexible, effective and robust and easy to implement because the parameters taken into account are the control gain and the target occupancy. The drawback of these algorithms is that they do not consider queue spill-back directly, so it is difficult to balance freeway congestion and ramp queues when the traffic is highly congested. If on-line tuning is not implemented, the Neural control algorithm is limited in adaptive control.

3.1.1 ALINEA

ALINEA is a closed loop ramp metering strategy, which trays to maximize the throughput of the main lane by maintaining a desired occupancy on the downstream of the main lane of the freeway. ALINEA adjusts the metering rate to keep the occupancy downstream of the on-ramp at the desired occupancy, according to the following formula (Papageorgiou et al., 1997):

$$r(k) = r(k-1) + K_R[O - O_{out}(k-1)]$$

k is the discrete cycle index, K_R is the regulator parameter and it is the only adjustment parameter, r(k) and r(k-1) are the metering rate of the current and previous iteration, O_{out} is the measured occupancy, O is the target set point occupancy. The basic control law of the algorithm is that if the measured occupancy, at the cycle k, is lower than the target one, then the above equation becomes positive, this means that order ramp volume r(k) is increased with respect to its previous value r(k-1). Otherwise if O_{out} is bigger than O, the equation is negative. This control strategy is the same both in case of congestion and low traffic condition. Normally K_R is set to 70 [veh/h] in order to reach good results. As said before, it is the only parameter to be adjusted, because in the control law no threshold or constraints are fixed. The only indispensable thing in ALINEA is a detector station to measure the occupancy of the downstream of the merging area. For what concerns the location of the detector, it must be placed where it is possible to see if there is a congestion, caused by an excessive volume on-ramp. In case of steady-state, if the the upstream volume of the traffic (traffic entering on the merge area) is constant, then O_{out} will be zero. This means that if the upstream traffic is constant, the control law leads occupancy to its desired value.

3.1.2 Zone algorithm

In the Zone algorithm, the the main lane is divided into different zones, assigning to each one an entry ramp. This algorithm is based on traffic conservation, this means that the metering rate is computed in order to level the traffic volume entering and coming out each zone. In this way the traffic of each zone has a specific pace. Other adjustments to the rate can be done taking into account environments factors and other considerations. The goodness of the algorithm depends on the proper division of zones, the accurate estimation of the bottleneck capacity and the accurate measurements of all in and out from a zone. But the difficulty is to tune the parameters algorithm in order to suite the local traffic and freeway characteristics and it is not possible to consider incident conditions when fast changes of traffic flow occur, because this algorithm does not consider the fact that the traffic flow is dynamic.

3.2 3.2 Cooperative ramp-metering algorithms

The cooperative ramp metering algorithms are improvements of the isolated one. As the latter, they compute the metering rate for each on-ramp. Through the system-wide congestion, needed to avoid both congestion at the bottleneck and spill-back at the critical ramp, further adjustments are done. The drawback of this type of algorithms is that a traffic instability may occur because because it is reactive to critical condition.

3.2.1 Helper ramp algorithm

The Helper ramp algorithm expects that a freeway corridor is split in six groups, each one includes one to seven ramps. Based on the occupancy of the local upstream mainline, each meter selects one of six metering rates. In this way, if a ramp has a long queue and it becomes critical, its metering rate its distributed to its upstream ramps. This algorithm is a good strategy, but the controller has not an accurate models of the traffic flow and information about origin and destination.

3.2.2 Linked-ramp algorithm

The key element of the Linked-ramp algorithm is the concept of demand-capacity. The local metering rate is computed as follow:

$$meteringrate = targetflowrate - upstreamflowrate$$

[19]

Once this rate is calculated, if it is in one of its lowest three metering rate, then the upstream ramp has to use the same rate or less. The drawback is that if the traffic is more congested, the lower the flow rate of the upstream and the higher metering rate is produced by the algorithm. This is the opposite of the desired situation.

3.2.3 Competitive algorithm

Competitive algorithm takes into account two types of traffic conditions: local and global. According to this, two sets of metering rates are computed and the more restrictive rate is chosen.

3.2.4 Compass algorithm

Also the Compass algorithm considers local and global traffic conditions. For what concerns the first one, it selects the metering rates through an ad hoc look-up table. In this table there are seventeen levels for each ramp, selected by the following parameters:

- local mainline occupancy;
- downstream mainline occupancy;
- upstream mainline occupancy;
- upstream mainline volume;
- threshold for local and downstream occupancy;
- threshold for upstream volume.

Globally, metering rates, based on system-wide information, are generated by off-line optimization. The actually implemented rate is the more restrictive of the two rates. If the detector of the ramp queue notices that the occupancy is bigger than its threshold value, the algorithm increases the metering rate by one rate level until the excess occupancy is smaller than the threshold level. The main drawback of this algorithm is that it is not robust, because it is based on look-up tables and predetermined metering rates.

3.2.5 Bottleneck algorithm

The bottleneck algorithm considers local level and general level. In the first case it is based on real time capacity, so the control strategy does the difference between the upstream demand with the downstream supply and it is the locally metering rates. For what concerns the global level, the coordinated control strategy identifies the bottleneck and through the flow conservation decides the volume reduction of the bottleneck itself. After this, based on the weights, previously identified, the control strategy distributes the reduction volume to upstream ramps. The realized rate is the more restrictive of the locally and globally rates.

3.2.6 System wide adaptive ramp metering (SWARM)

SWARM algorithm also has a two-level structure. The local control identifies the rates based on local density. At global level, the control, considering the bottleneck, decides the volume reduction from the ramps upstream. Then, based on predetermined fractions used to have new ramp metering rates, it distributes the volume reduction to upstream ramps. Also in this case, the most restrictive rate is used.

3.3 Integral ramp metering algorithms

This category of algorithms has the feature to have control objective, which is explicitly or implicitly linked to the control action. The goal can be the travel time or the throughput of the whole system. The ramp metering rate is decide through the objective optimization and the the consideration of system constraints, like bottleneck capacity and maximum allowable ramp queue.

Unlike the previous class of algorithm, integral ramp metering algorithms are most appreciated for their capability to deal different types of metering and system constraints and their solid theoretical foundation. Their performances depend on the goodness of the input data and traffic model used.

3.3.1 Fuzzy logic algorithm

The Fuzzy logic analyzes analog input value, such as speed, flow rate, ramp queue and occupancy, in terms of logical variables,like small, medium and big. Then it develops rules in order to link metering levels and traffic conditions.

3.3.2 Linear programming algorithm

This type of algorithms work under congested road conditions, considering constraints on the length of the ramp queue and metering bound.

In linear programming algorithm, users select weights,which are used to do the weighted sum of ramp flow, maximized by the algorithm. The purpose of the weight is to reflect the user's belief in the different ramps importance. Then a real time capacity of every road is computed by the algorithm.

The main limitations of linear programming algorithm are the fact that it does not consider the variation of the travel time in its computations of rates and its performance depend on the quality of the input data.

3.3.3 METALINE algorithm

METALINE algorithm derives from ALINEA and it is its extension. The algorithm is governed by the following equation:

$$r(k) = r(k-1) - K_1(o(k) - o(k-1)) - K_2(O(k) - O^c))$$

It says that the metering rate is computed based on the change in measure occupancy of each freeway segment under METALINE control, and

> the deviation of occupancy from critical occupancy for each segment that has a controlled on-ramp [19].

$r(k)$ is the vector of metering rate of the m-th controlled ramp at time step k, $o(k)$ is the vector of the n-th measure occupancies at the time step k, O and O^c are the measured and desired occupancy downstream of the controlled ramps. K_1 and K_2 are two gains matrices. The performance of this algorithm depends of the correct choice of the matrices K_1 and K_2 and the vector of the target occupancy O^c.

3.3.4 Coordinated metering using artificial neural networks

These types of algorithms use artificial neural intelligence to acquire and remember the metering rate generated by a ramp control system and traffic generation model. They do what a ramp control expert system does.

Chapter 4

Algorithm description

The present work consists of the development of an algorithm that allows to enhance mobility on an highway, in the event of an accident along a given lane. The focus is to prevent vehicles to be stuck behind damaged vehicle(s) while waiting the possibility to change lane and therefore overtake the obstacle.. To verify the goodness of the algorithm, the results have been compared with those obtained from the application of the ALINEA algorithm (which is chosen for a matter of simplicity), as well as with those obtained with no applied algorithms, but just with the applications of the simulators.

The simulated scenario includes the highway, 970 meters long, with two lanes, where there is a fleet of autonomous and connected vehicles (CAD) and there is a damaged car in the middle of the lane, hereafter represented by an obstacle. The simulation lasts 100 seconds. To avoid boundary effects, the transient has been deleted (the first 10 seconds of the simulation).

The algorithm has the purposes to optimize flow and speed of cars on both lanes and to decrease the waiting time of the cars behind the obstacle without a sacrificial approach, when cellular vehicle-to-infrastructure (C -V2I) is adopted as a communication technology.

The basic idea to achieve these goals is to reverse the cruising speeds of the two lanes: considering "not Commonwealth countries" normally the right lane is the one dedicated to ordinary marching, whereas the left one is used by cars to overtake and travel at higher speed. In the study, the speed of the cars on the right lane has been increased, with the aim of accessing the other lane so as to avoid the accident. Vehicles on the left will encourage their entry by slowing down. The algorithm can be applied specularly, in the event of accident occurring on the left lane.

4.1 Reference scenario

The reference topology considered (in Fig.. 4.1) is an highway. The entities moving in the topology are vehicles. Each of them is connected to the cellular infrastructure and is equipped with on-board units for C-V2I communications. Vehicles periodically send CAMs toward the application server.

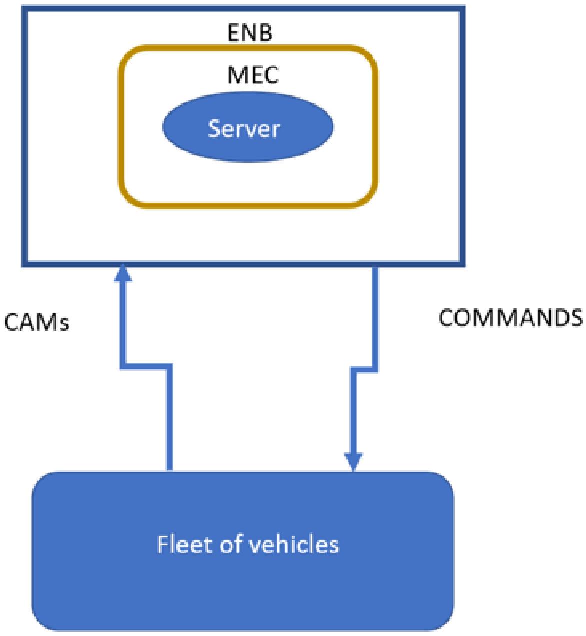

Figure 4.1: Network architecture

In particular, we consider an LTE -A network pro with an eNodeB (eNB) located at the center of the topology. The server, hosting the developed algorithm, is located inside multi-access edge computing (MEC), within the eNB, as shown in Fig. 4.1.

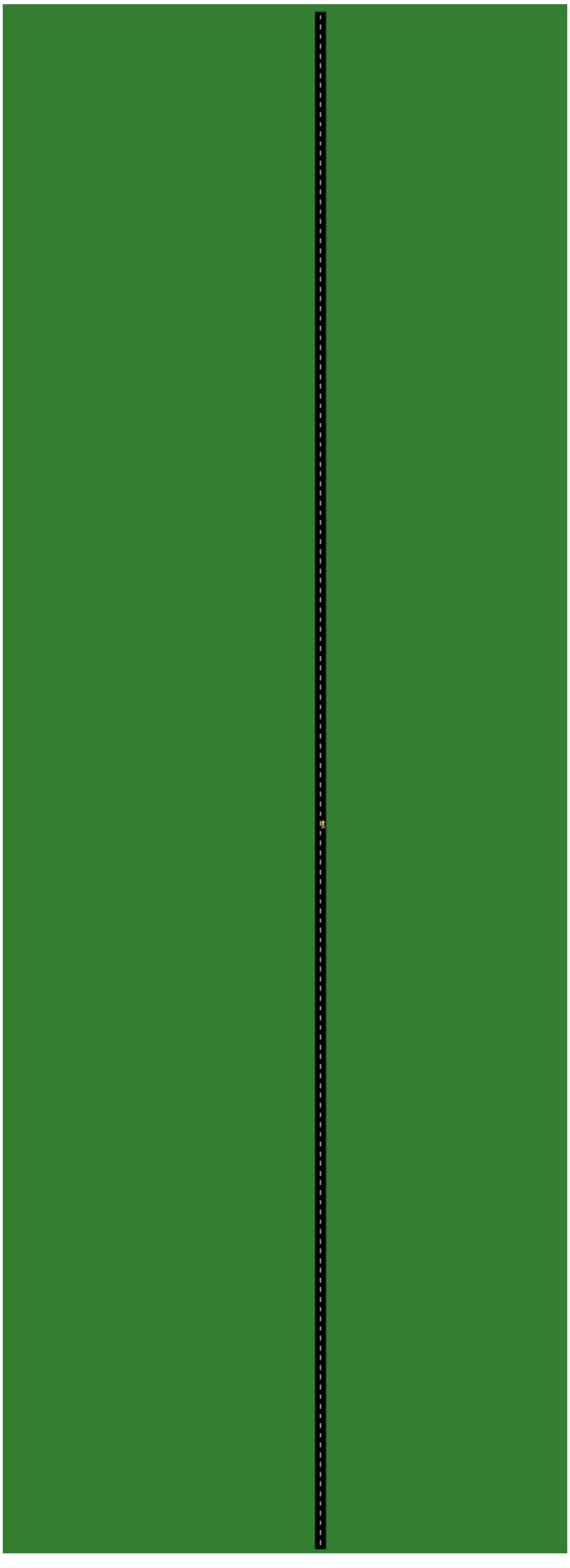

Figure 4.2: Map

4.2 Traffic stability

After generating the simulated road with SUMO, represented by the *.map.xml* file, it is necessary to decide the vehicular density of the simulation. It is the number of vehicles, in a given timestep, that are present in the *.map.xml* file. The vehicular density depends on the generation rate, that is the number of vehicles that every second enter the simulated system. The higher is the generation rate, higher will be the number of vehicles enter the simulation and higher will be the vehicular density. Large values of generation rate increase the congestion of the scenario: the first vehicle created is the obstacle at 470m from the beginning of the system and the other cars entering the simulation in the same lane of the obstacle create a long queue and cannot change the lane because also the number of vehicles in the free lane is high. This is, precisely, the situation to avoid, so it is important that the vehicular density remains below congestion a threshold in order to work under medium traffic condition to avoid congestion. In order to do that, the generation rate has to be properly set and through the study of the system traffic stability the threshold of the traffic stability is found.

For the traffic stability inspection, the generation rate grows from 0 veh/s (vehicle per second) to 1.5 veh/s. For each value, five simulations are generated. Then the outputs were mediated in order to have the following plot:

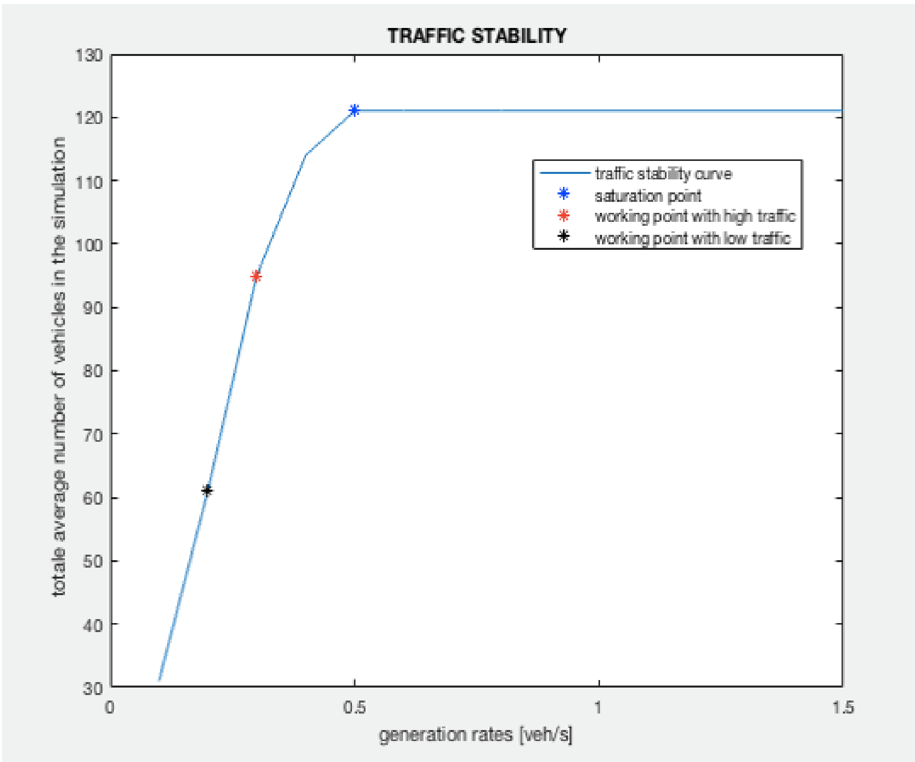

Figure 4.3: Traffic stability

As it is possible to see from Fig. 4.3, from generation rate 0.5 [veh/s] the system is saturated and in the range 0 [veh/s] and 0.49 [veh/s] the vehicular density grows linearly. So in this condition, no more than 121 vehicles can enter the simulation.

This is due to the fact that if there are few vehicles per second in the map, the cars entering the system are less than the exiting ones. Whereas for large value of generation rate, only a few number of vehicles can exit the simulation, because are stuck in queue waiting to change lane. After 0.5 [veh/s] no more car can enter the simulation. In order to simulate low traffic conditions and high traffic conditions, two generation rate thresholds are set, respectively 0.2 [veh/s] and 0.3 [veh/s].

4.3 Simulation parameters

After establishing the vehicular density, the parameters that come into play in the simulation are defined. From a mobility point of view, each car is described in the *.rou.xml* file of SUMO with the following parameters:

- id;
- length;
- width;
- maximum speed allowed;
- depart time;
- depart position;
- depart speed;
- depart lane;

The depart lane is casual, so it is not possible to know a priori how many cars are present in the two lanes.

For what concerns the network, OMNeT++ defines the simple modules client and server, in the application layer of the protocol stack of LTE-A pro. The client is represented by each car and the server is the server in the eNB. Both the client and the server are represented by two *.ned* files,where are specified

- the local port, i.e the port from which the module sends and receives messages;
- the destination port, i.e the port to which to send messages;
- input and output gates, which indicate that the module can actively both send and receive.

The type of communication is defined in the *.ini* file of OMNeT++; since this is a Vehicle-to-Infrastructure communication type, all the vehicles send messages to the server, which, in turn, transmits a command to the specific car, whenever necessary.

4.4 Interoperability between simuLTE and Veins

The development of this algorithm requires that, from the time of the order imposed by the server in the application layer, the vehicle performs the command. SimuLTE deploys the INET framework, which provides TCP and UDP connections to the modules.

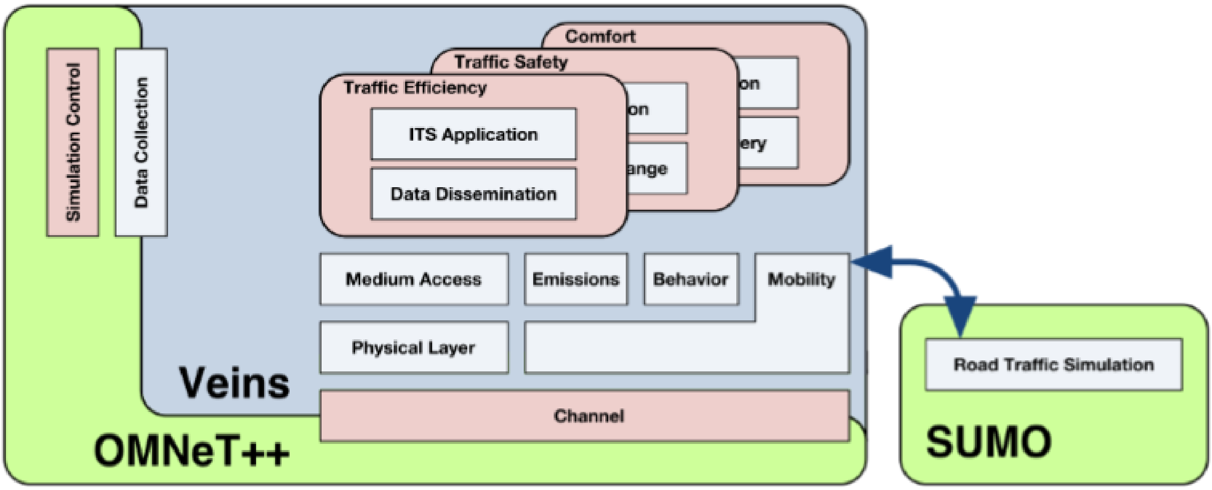

Figure 4.4: Veins Simulators

As a result, in addition to network-level mobility, vehicular mobility is required for the vehicles. In order to achieve this, it has been necessary to define an interface implemented by *TraCIMobility* module defined by Veins. *Veichular in Network Simulation* (VEINS) is an open source framework for running vehicular network simulations, while *The Traffic Control Interface* (TraCI) allows to develop online retrieving and editing of vehicle parameters, such as changing a particular route as a consequence of an event occurred in proximity or for a command received.

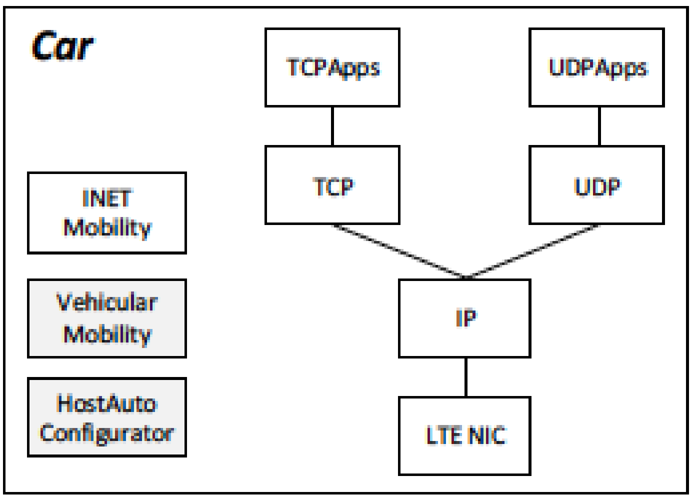

Figure 4.5: Module of the car [12]

As it is possible to see from the above picture, in the module car there are both the *INETMotbiity*, to be backward compatible, and the *vehicularMobility*. The car cannot use the two mobility modules at the same tine in the simulation, so they are defined in the *ned* file as conditional modules. Since both the mobility modules exploit signals, provided by OMNeT++, they can be activated by mobility events.

> Depending on the activated mobility modules, the car subscribes itself to the signals generated by the corresponding module [12].

Always in the same image, *HostAutoConFig.urator* module can been observed. This allows the new vehicle created to obtain an IP address to communicate and to be associated to the eNB.

When the network is defined, it is important to include the *Binder* and *TraCIScenarioManager*; the latter needs to know the type of vehicles that it has to add to the simulation. An example of simulation scenario is available in the following plot:

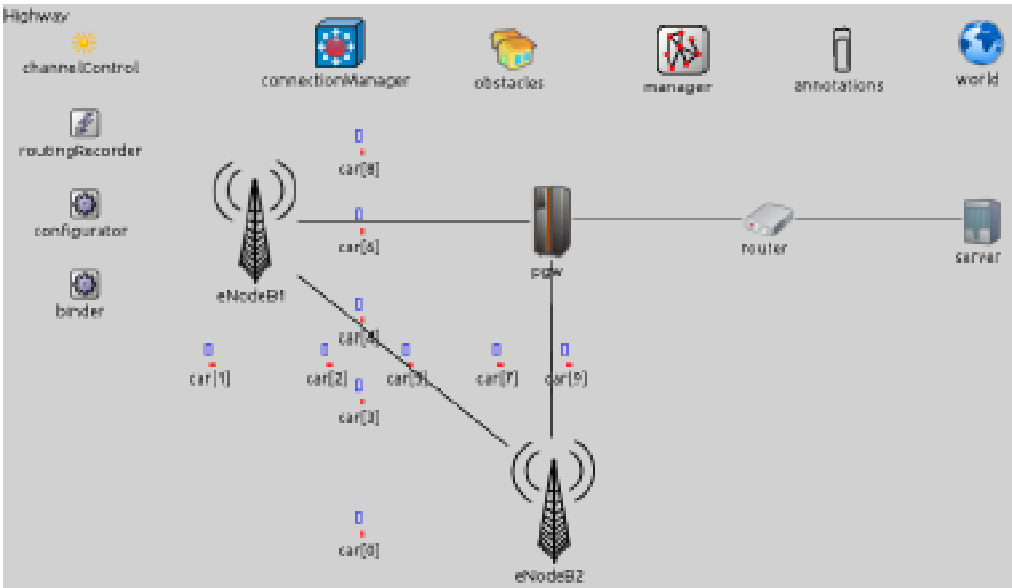

Figure 4.6: Scenario of the simulation [12]

4.5 Algorithm development

The algorithm has been developed from the point of view of the server. For a matter of convenience, the lane where the accident is located, is called lane 0 and the other one lane 1. This has been summarized in the block diagram below:

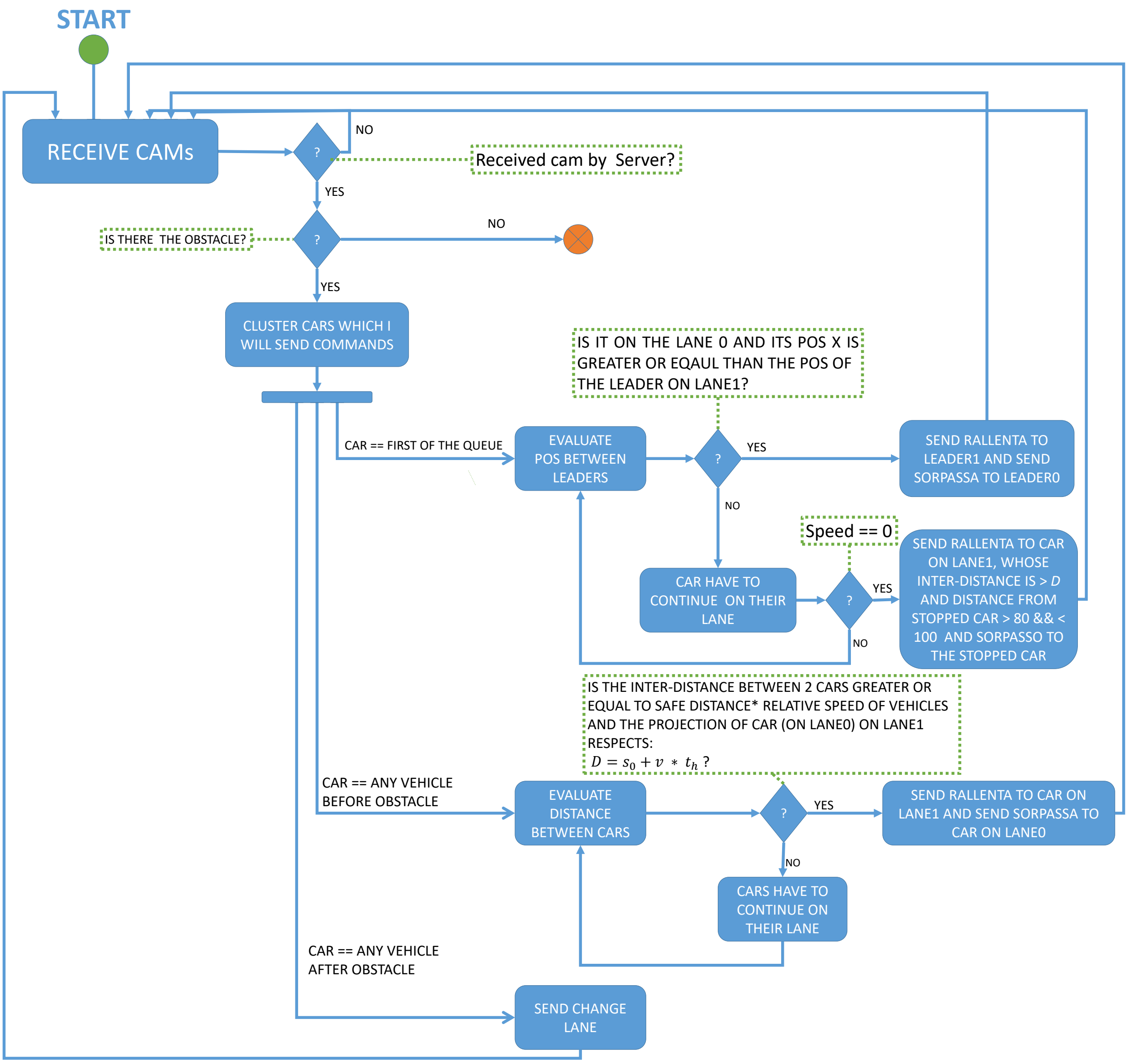

Figure 4.7: Block diagram of the algorithm

Every 10 Hz, the cars entering the simulation send CAMs (with information about ID, speed, position, time at which the message has been sent, lane, ID of the vehicle ahead and the distance between the vehicle sending the message and the car ahead) to the server and modify their speed profile in order to not to have any type of speed control since SUMO is a safe simulator. The server, once received the CAMs, memorizes each type of data in structures, as shown in the image below:

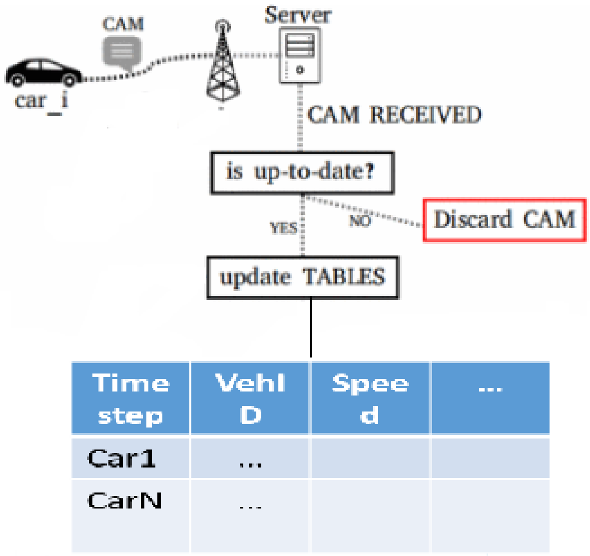

Figure 4.8: Memory structure model

The server compares the current speed of each vehicles, extracted from the current CAM, with its previous speed, coming from the CAM received 0.1 seconds before. If these two data are equal to zero, the server identifies the obstacle, sending to all cars a message with obstacle position; then an other message is transmitted to the cars containing various commands, depending on the lane the vehicles belong to. If the message is received by the car on lane 1, it must set $LaneChangeModel$ in such a way as to favor overtaking by the cars that are in the lane 0. If the message is received by vehicle in the lane 0, the car must set $LaneChangeModel$, so that it has to change lane. This is done because SUMO is a safe simulator and avoids lane changes if there are cars in the other lane with high speed and the inter-distance between vehicles is low.

Only if the obstacle is identified, for each vehicle, the server classifies its status, i.e if is the leader of the queue, or any vehicle before the obstacle or any vehicle after the obstacle; in parallel it slides the memory structure of the position of the vehicle in lane 1 and identifies the first one on this lane. This is performed through the function:

$$checkDistance(vehID)$$

If the current car is the leader in lane 0 and its position is bigger or equal than the leader in the lane 1, it sends a message to the latter. Through this message, with the function $sendMessage(vehID1)$, the server orders the leader in lane 1 to slow down at speed equal to 20 [m/s], for 2.5 [s]. However, if the server understands that current vehicle is any vehicles before the obstacle verifies the cars positions. Keeping in mind the position of the car on lane 0, it slices the map of the inter-distance of vehicles in lane 1 and finds if there is a couple of cars, whose positions include the one of the car 0. Then it checks if the inter-distance of this couple of vehicles is equal to the threshold:

$$threshold = s_0 + v * t_h$$

where s_0 is a safe distance equal to 3 meters; v is the speed of the second vehicle; and t_h is the time to collision on highway equal to 1.2 seconds. The threshold represents the safe distance between the projection on lane 1 of the vehicle in lane 0 and the second vehicle.

The server verifies if the position of the current car is included between two vehicles in the lane 1 with an inter-distance equal to the threshold. If this is the case, the server applies the function $sendMessage(vehID1)$, where in this case $vehID1$ is the second vehicle.

The function $sendMessage(vehID1)$ calls an other function that is

$$sendSorpasso(vehID0)$$

With $sendSorpasso(vehID0)$ the server sends a message to the car in lane 0 with the command to change lane and restore a speed equal to 36.11 [m/s], which is the maximum highway speed legally allowed. This is done because, in SUMO, when a a vehicle starts from a standstill, it takes a long time to reach cruising speed, especially if the target speed that the car must track is not indicated. In this way, vehicles in the lane 0, whenever possible, change lane immediately without getting stuck behind the accident.

When the vehicle receiving the command *changeLane* implements the maneuver successfully, the server imposes on overtaken car to restore the cruising speed of 36.11 [m/s]. This is possible because the server detects the lane change from the new CAM, comparing it with the previous one. The function that implements this order is:

$$sendAccelera(vehID0, vehID1)$$

This function needs two input parameters, which are the vehicle overtaken on lane 1 and the vehicle that just changed the lane. The function verifies if the vehicle which received *sendSorpasso* actually changed lane. If this is the case, the server compares the positions of the two cars and checks if the position of $vehID0$ is bigger than the position of $vehID1$ and if their relative distance is bigger or equal to:

$$distance = t_s * v$$

distance is a safe distance comparable to the space to collision, where t_s is equal to 3 seconds and v is the speed of $vehID1$.

After the lane change and 10 meters overcoming of the obstacle by both the vehicles coming from the lane 0 and the ones coming from lane 1, the server sends a message, called *Alert*. With this communication, the server redistributes cars between the two lanes to avoid slowing down again.

Sometimes it happens that if the vehicle on lane 0 receives the order of overtaking this fails to implement it and remains stopped behind the obstacle. The order is not implemented because SUMO does not consider it feasible and safe, for example because the car on lane 1, despite slowing down, is not at distance accepted by SUMO to allow overtaking.

As a result, the car stops and waits for the change lane order again.

Because the purpose of the algorithm is to reduce waiting times of the car behind the obstacle and it is based on a non-sacrificial approach, the server will create "holes" between vehicles of lane 1. So it applies the function

$$checkDistanec(vehID0)$$

where the $vehID0$ is the car stopped behind the accident. With this function the server slides the memory structure of the distance between the vehicles in the lane 1, looking for a inter-distance that is bigger than the *threshold* and a couple of vehicles with this inter-distance, where the second vehicle is at a distance in the range of 100 and 80 meters from the stopped car. If this is true, the server sends a message to the this vehicles, causing it to slow down to 17 [m/s] for 4.5 seconds. In this way "holes" are created in lane 1 and the stopped car can start again and change lanes.

4.6 Simulators behavior and ALINEA interpretation

As said above, to verify the goodness of the algorithm developed in the thesis, its results are compared with those obtained with the study of the simulators behavior, so no algorithm is applied, and with outputs of the ALINEA algorithm's application.

For what concerns the first case, cars send CAMs to the infrastructure and then it identifies the obstacle, in the same way of the algorithm, and tells, through a message, where is located the accident. In this way vehicles do not receive instruction to improve traffic conditions. As a consequence, cars entering the lane 0 will remain on the same lane until there is the possibility to overcame the obstacle. A long queue is created behind the accident and the vehicles on lane 1 do not encourage the enter of cars on lane 0, but exit the simulation with highest possible speed. The stopped vehicles only pass when the cars in the lane 1 have overtake the obstacle or if between the vehicles of lane 1, before the accident, there is much space to allow the overcoming.

While for what concerns ALINEA, the lane 0 is considered as the ramp and it is applied a cycle of 4 seconds, where for 3 seconds the cars on lane 0 remain stopped (called red phase) and for 1 second they change lane (named green phase). The indispensable thing in ALINEA is a detector, represented by the server, which knows the traffic situation through the CAMs received from the vehicles. The number of vehicles, which can change lane during the green phase, is different in the various cycle. It depends on the capacity of the merging zone, which depends on the traffic upstream in the lane 1. During the green phase, the server slowdown the traffic on lane 1 to encourage the change of lane of vehicles stopped in lane 0. The algorithm is activated when the server detects a stopped car behind the obstacle. This is done in order to well interpret the algorithm, but also to counteract the fact that SUMO is safe and to give a correct interpretation of simulation's time in reality. When the car starts again from a standstill to reach the target speed takes a long time going very slowly at the beginning. This is the reason why if the cars on the lane 1 do not receive the order to slowdown they do not encourage the change lane of the cars in lane 0. After several attempts, the choice of the cycle length of 4 seconds proved to be excellent. Because 4 seconds of the simulation correspond to at least 60 seconds in the reality and the fact that the car on the lane 0 can change lane only for 1 second does not create congestion in the merging zone.

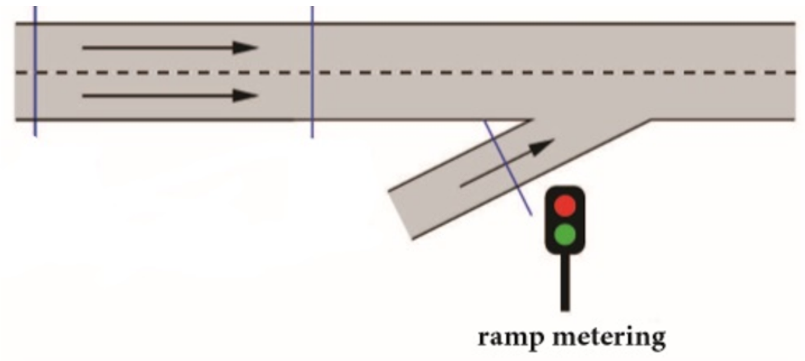

Figure 4.9: Model of ALINEA algorithm interpretation

Chapter 5

Results and Analysis

The last step of a simulation process is represented by the data analysis, it is the critical one. This allow to extract the results from the system and verify the goodness of the work carried out. The method used for the output analysis is named post-processing and it is a second-level of data elaboration. The first level is that one generates the output, through the algorithms discussed previously. In the post-processing, the raw data represent the input files for Matlab, which produces the final plotted results. Below is given a brief overview of the simulation process parameters:

- Inputs: files, parameters and variable to start the process.
- Simulation: a model of reality that provides for evaluating and predicting the development of a series of events or processes subsequent to the imposition of certain conditions. In this case the conditions imposed are represented by the algorithms developed at the application level and the events are the behaviors of the vehicles after the reception of the messages and the general trend of the traffic flow.
- Post-Processing:during this phase the results of the simulation are processed.
- Results: represent the final phase, where the outputs of the post-processing are represented graphically.

In order to monitor the progress of the algorithms and to compare them each other, it is necessary to define useful and comprehensible metrics. For this reason, the following $Key\ Performance\ Indicator$ (KPIs) are investigated:

- Waiting Time: intended as the time that the stopped cars spend behind the accident waiting to be able to change lane.
- Time of imprisonment, that is the time taken by the cars, in the same lane of the accident, to change lane.
- Percentage of cars that do not exit the simulation: represented by the vehicles that fail to finish the simulation or because they were stuck behind the accident or because they started too late the lane change maneuver.

- Mean speed: intended as the mean among the simulation of

$$\frac{1}{N}[\sum_{i=1}^{N}\dot{x}_{mean}(v_i)]$$

  where N is the number of vehicles and $\dot{x}_{mean}(v_i)$ is the mean speed of each vehicles then mediated over the number of vehicles.

- The mean speed in the lane 0 before the accident: mean speed computed in the same way as before, considering only vehicles in the same lane of the obstacle within 470 [m].

- The mean speed in the lane 1 before the accident: calculated in the same way as the previous KPI but taking into account vehicles in the free lane.

- The mean speed on lane 1 after the obstacle: takes into consideration only vehicles in the free lane after the accident, there are cars that come from the same lane 1 and others from lane 0.

- The mean speed before obstacle: it considers the mean speed of the two lanes before the obstacle.

- The mean speed after obstacle: that is the mean speed of the two lanes after the accident.

- The mean speed of the lane 1: it is the mean speed considering the free lane before and after the obstacle.

- The mean speed of the lane 0: it is the mean speed considering the same lane of the accident before and after this.

- The mean space on lane 0 before the accident, intended as the mean among the simulation of:

$$\frac{1}{N}[\sum_{i=1}^{N}x_{mean}(x_i)]$$

  where N is the number of vehicles and $x_{mean}(x_i)$ is the mean space of the each vehicles mediated over the number of cars

For each scenario (thesis algorithm , ALINEA algorithm, SUMO behavior) are done thirty simulations considering the generation rate for high traffic conditions and that for low traffic conditions.

5.1 Waiting time

The most important result to achieved the first purpose of the algorithm is the waiting time. It is computed with the following formula:

$$\frac{1}{N}[\sum_{i=1}^{N}t_{mean}(t_i)]$$

where $t_{mean}(t_i)$ is the mean waiting time of the those cars stopped behind the accident, then mediated over total number of cars stopped N. So at the end of this process, the output is a vector, whose length is equal to the number of simulations. In each cell there is the average waiting time of each simulation. Through this vector the *Probability Density Function* (PDF) is calculated as $(frequency\ event)/(total\ number\ of\ events)$. It represents the most probable waiting time spent by the stopped cars behind the obstacle.

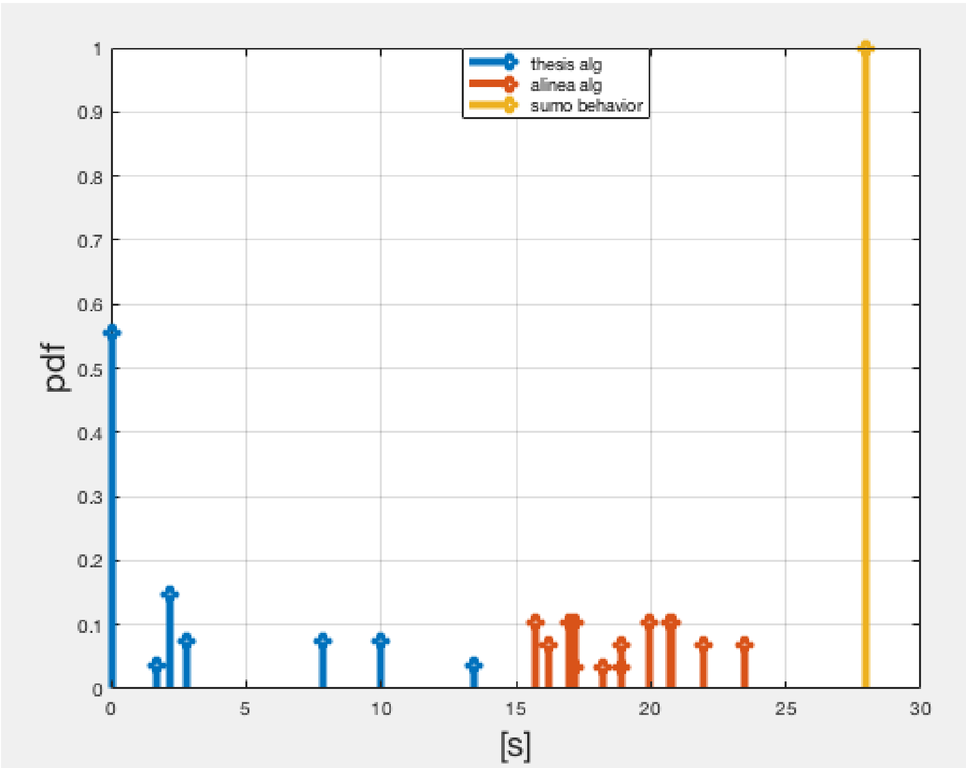

Figure 5.1: PDF of Waiting time with rate = 0.3

Fig. 1.1 shows the waiting time obtained in high traffic condition. In case where no algorithm is applied, but only the SUMO behavior is studied, the waiting time is 28 [s], while, in case of ALINEA algorithm, it is between 16 and 24 [s] with probabilities included in the range 0.1-0.03. The last waiting time is acquired through the application of the thesis algorithm. The values are included between 0 and 13 [s]. The most probable event is to wait zero seconds, this means that vehicles can immediately change lanes without stopping to wait for lane change. With a probability smaller that 0.04 vehicles can wait 13 [s] to to carry out the maneuver.

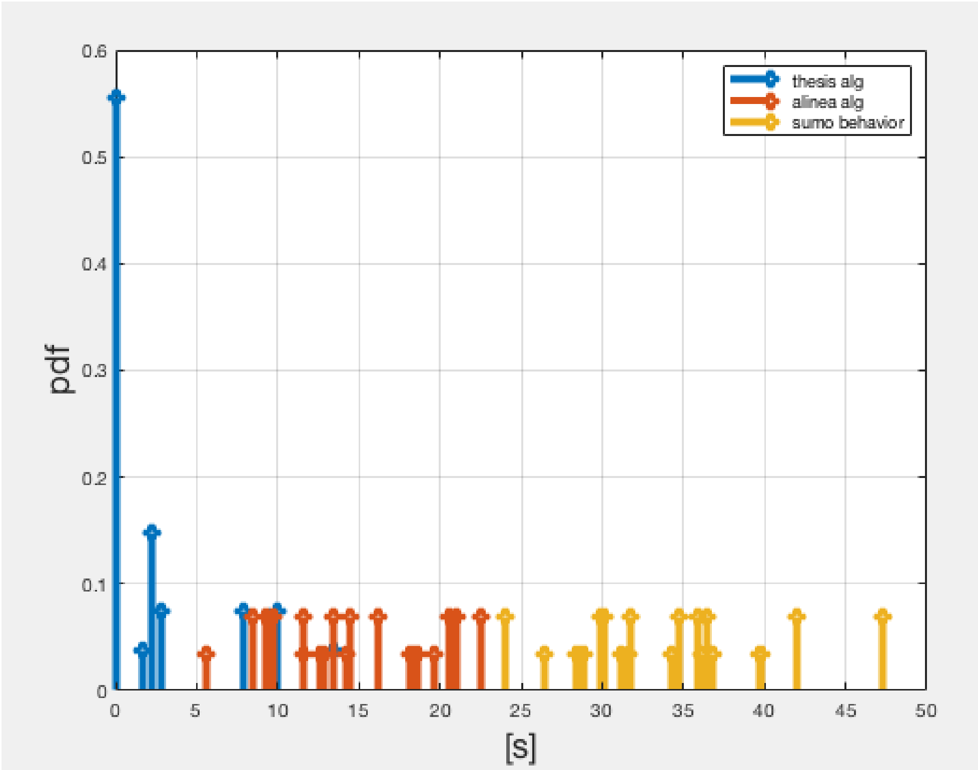

Figure 5.2: PDF of Waiting time with rate = 0.2

The above image displays the waiting time in low traffic conditions. Since there are less vehicles, the waiting time is lower in all three cases, especially improved with the application of both algorithms. In the case of the thesis algorithm, with a probability bigger than 0.5, the vehicles do not stop.

The application of both algorithms brings benefits to the simulated system both when the rate is high and when it is low. In the thesis algorithm the advantages are due to the fact that server forces car on lane 1 to slow down to allow the lane change both to the vehicles stopped behind the obstacle and vehicles travelling on lane 0. In ALINEA algorithm benefits are due to the lane change allowed during the green phase of the cycle without creating slowdowns in the other lane.

5.2 Time of imprisonment

The time of imprisonment is computed as the waiting time, but considering vehicles that are in the lane 0 before the accident, including both the cars entering the simulation in lane 0 immediately changing lane and those that remain stuck behind the obstacle. Also in this case the output is a vector, containing the average time of imprisonment of each simulation. With these values is computed the *Cumulative Density Function* (CDF). The CDF of a real random variable X, evaluated at x, is the probability that X will take a value less than or equal to x, given by:

$$F_X(x) = P(X \leq x)$$

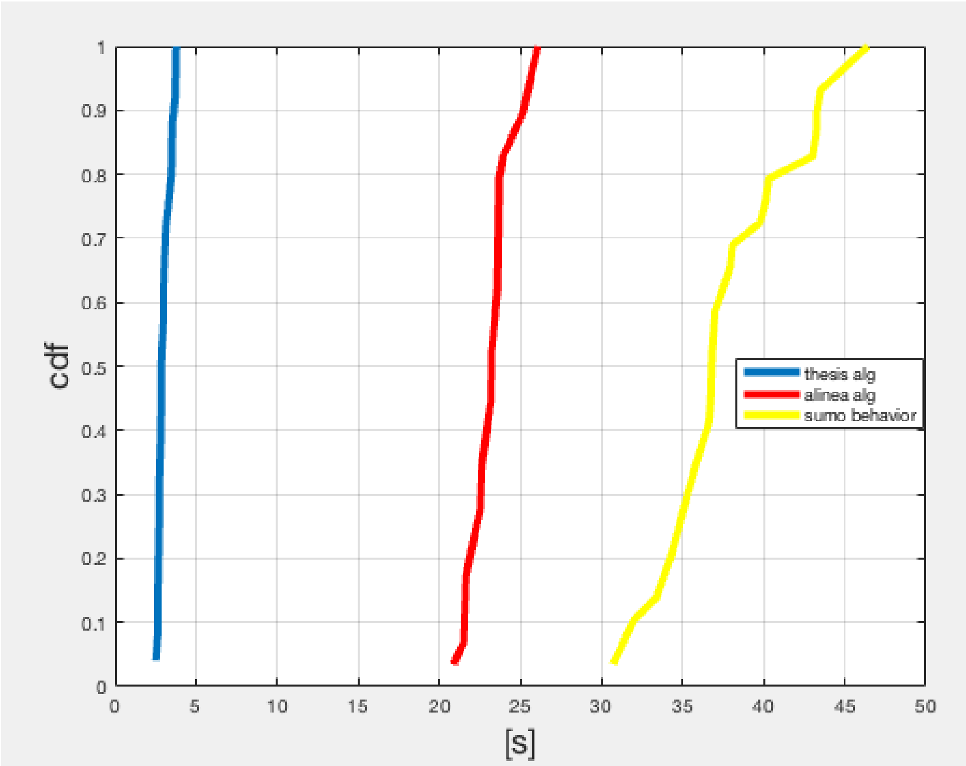

Figure 5.3: CDF of Time of imprisonment with rate = 0.3

As it is possible to see from the above image, the CDF of the algorithm (the blue line), is between 2 [s] and 4 [s] and with probability ≤ 0.5 the time spent in lane 0 before the obstacle is in the range 2 - 3 [s]. The red line represents the CDF of ALINEA algorithm in the range of 21-26 [s], while the last line (SUMO behavior) is in a longer interval than the previous CDF.

A completely similar situation but with lower values is that obtained in low traffic conditions as visible in the picture 1.4:

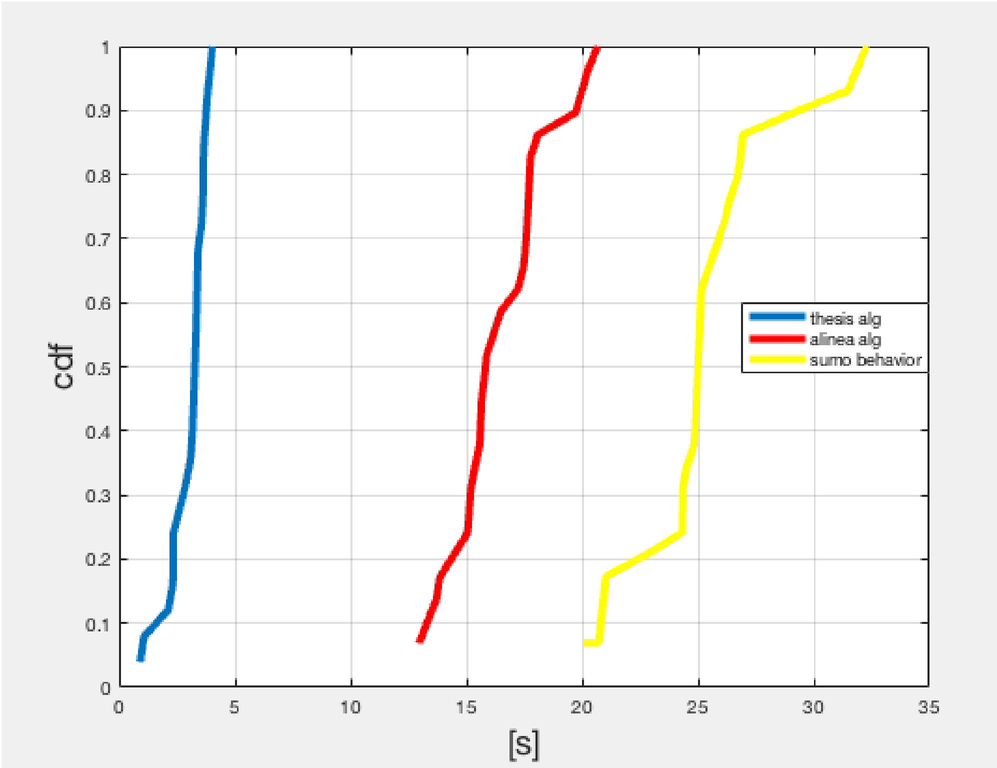

Figure 5.4: CDF of Time of imprisonment with rate = 0.2

The reason why the CDF in the case of the thesis algorithm is included in a low interval, is due to the fact that the server, after appropriate checks on the traffic situation, sends the message to vehicles on lane 0 with the order to change lanes immediately, without getting to create queue behind the accident. This is the strength of the algorithm, because the traffic is immediately checked upstream of the obstacle.

5.3 Percentage of vehicles which do not exit the simulation

The percentage of vehicles which do not exit the simulation is a significant indicator for two reasons:

1. because it affects some of the following results;
2. because the thesis algorithm is based on a non-sacrificial approach, which means that the cars stopped behind the obstacle will not wait for the other vehicles on the lane 1 overcome the incident, but all are encouraged to change lane.

The formula used is:

$$\frac{1}{N}[\sum_{i=1}^{N} m_i]$$

where N is the number of simulation and m_i is the number of vehicles which do not exit the simulation of each run.

Fig. 1.5 shows the results of this analysis.

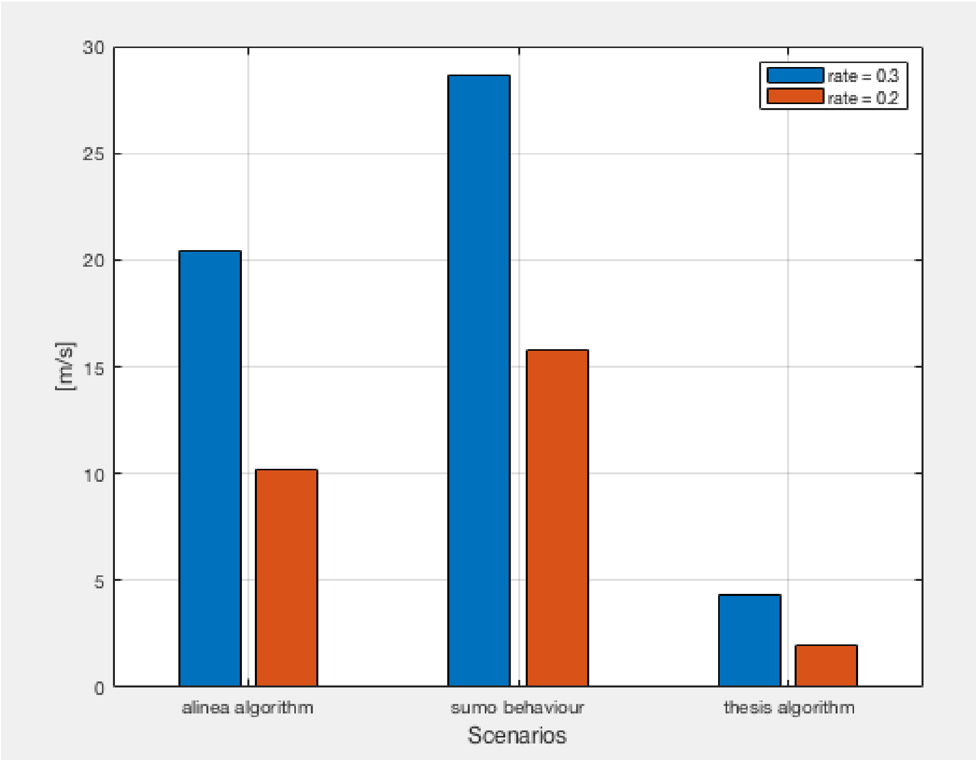

Figure 5.5: Percentage of vehicles which do not exit the simulation

As is clear from the histogram, in case of generation rate = 0.3 when no algorithm is applied the percentage is really high, almost 30%, because no car in lane 1 slows down to favor the change of lane of the cars remaining trapped in lane 0. Better results are obtained thanks to the application of ALINEA algorithm, with a percentage equal to 21%, which is even more reduced with the application of the thesis algorithm.

In the case of low traffic conditions the trend is the same, but obviously the percentages of vehicles which do not exit the simulation take on lower values, showing halved values

This is a useful KPI because it is important verifies how many cars finish the simulations, since it is a value that influences the average speed of the sections of the highway after the obtstacle

5.4 Mean speed

The mean speed is an other important result that highlights the effectiveness of the thesis algorithm: despite the decrease in speed in lane 1 to allow the change of lane to the cars behind the obstacle, the mean speed achieved by the thesis algorithm is bigger than ones obtained in the other two cases.

Also the mean speed of the whole system is represented by the *Cumulative Density Function*.

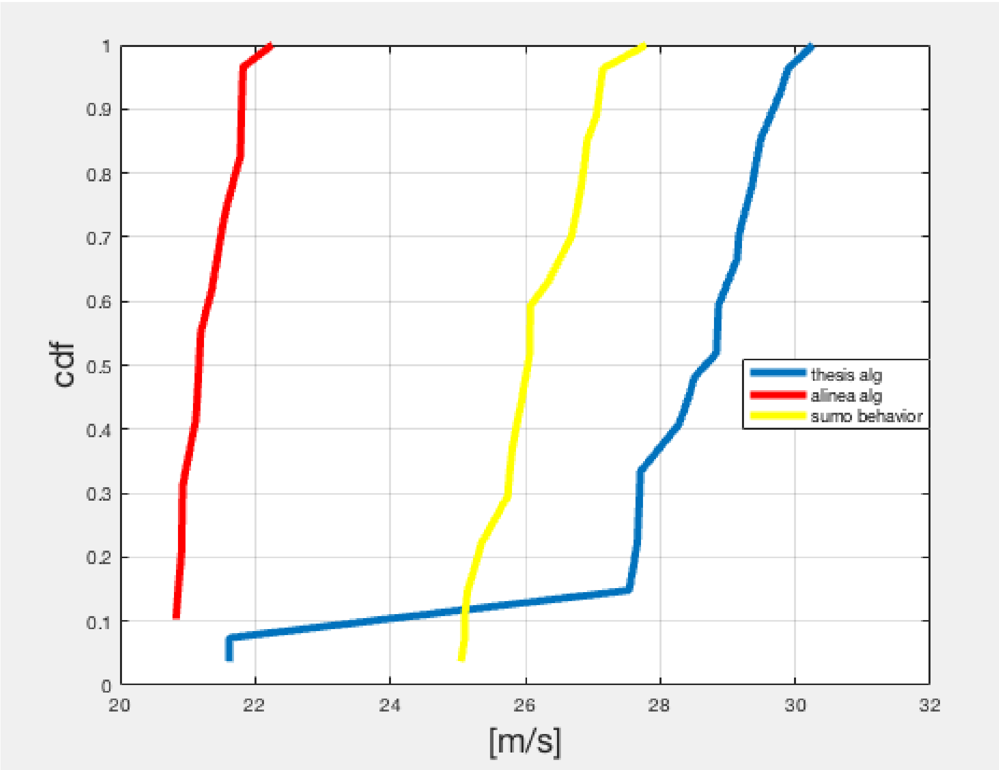

Figure 5.6: CDF of the mean speed of the system with generation rate = 0.3[veh/s]

In both images, the blue line represents the CDF of the mean speed of the thesis algorithm, the red one of ALINEA and the yellow one of SUMO behavior.

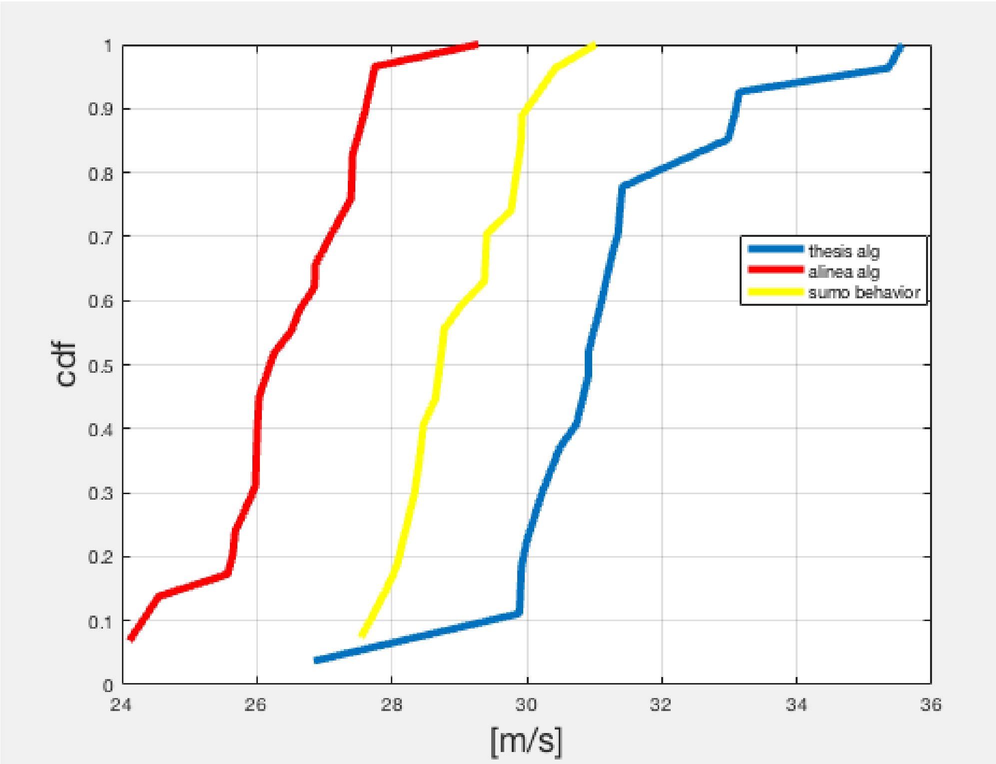

Figure 5.7: CDF of the mean speed of the system with generation rate = 0.2[veh/s]

The thesis algorithm increases the mean speed of 12% in case of high traffic condition and 15% when low traffic conditions are simulated. It is important to emphasize that with SUMO, if a car stops behind the obstacle and has the possibility to change the lane, it starts from 0 and to reach the cruising speed of 36.11 m/s takes a long time. In this way the mean speed of the whole system decreases. If this maneuver was not so slow the mean speed would be higher. It has been attempted to change this setting of SUMO in all possible ways, even removing all the controls on the speed. However we came to the conclusion that this is a feature of the simulator, that cannot be altered.

5.5 Statistics on both lane before and after the obstacle

In this section are analyzed the the mean speed in the lane 0 before the accident, the mean speed in the lane 1 before the accident and mean speed in the lane 1 after the accident in order to verify the traffic conditions in each portion of the highway.

Fig. 5.8 shows a graph which simplifies the placement of these results on the map.

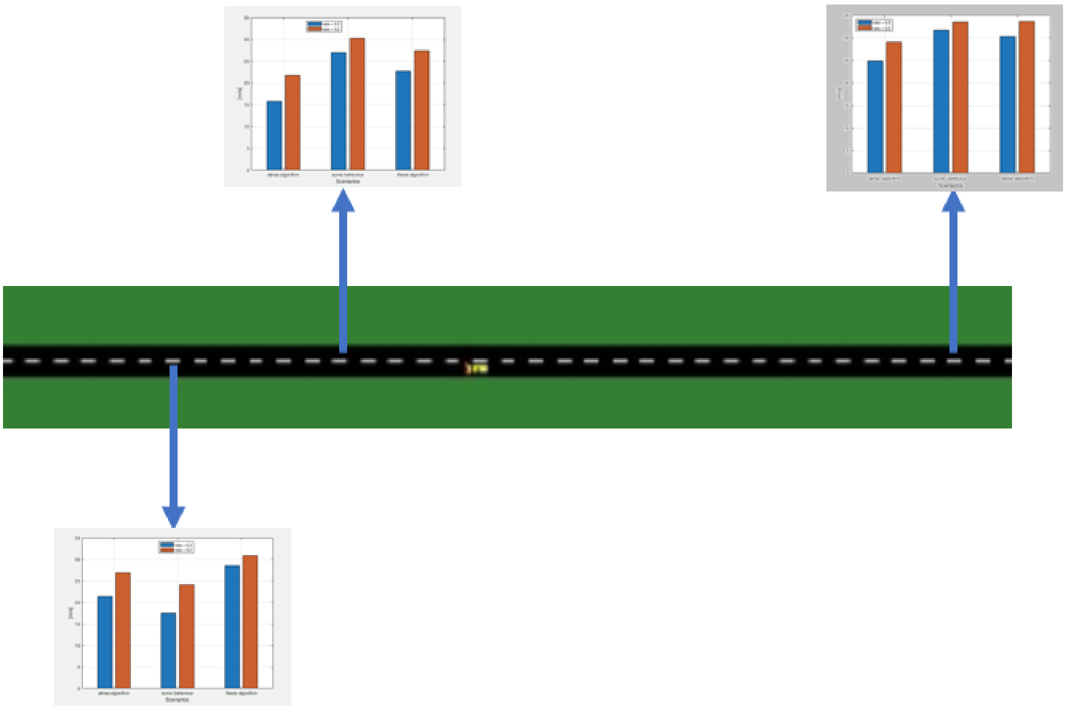

Figure 5.8: Placement of the statistics on both lane before and after the obstacle in the simulated system. From the bottom to the top: mean speed of lane 0 before the obstacle, mean speed of the lane 1 before the obstacle, mean speed of the lane 1 after the obstacle. Both the generation rates are simulated.

Mean speed in the lane 0 before the obstacle

The part of the zero lane before the obstacle represents the bottleneck of the simulated system. For this reason the mean speed of the vehicles here is a useful index for the evaluation of the algorithms.

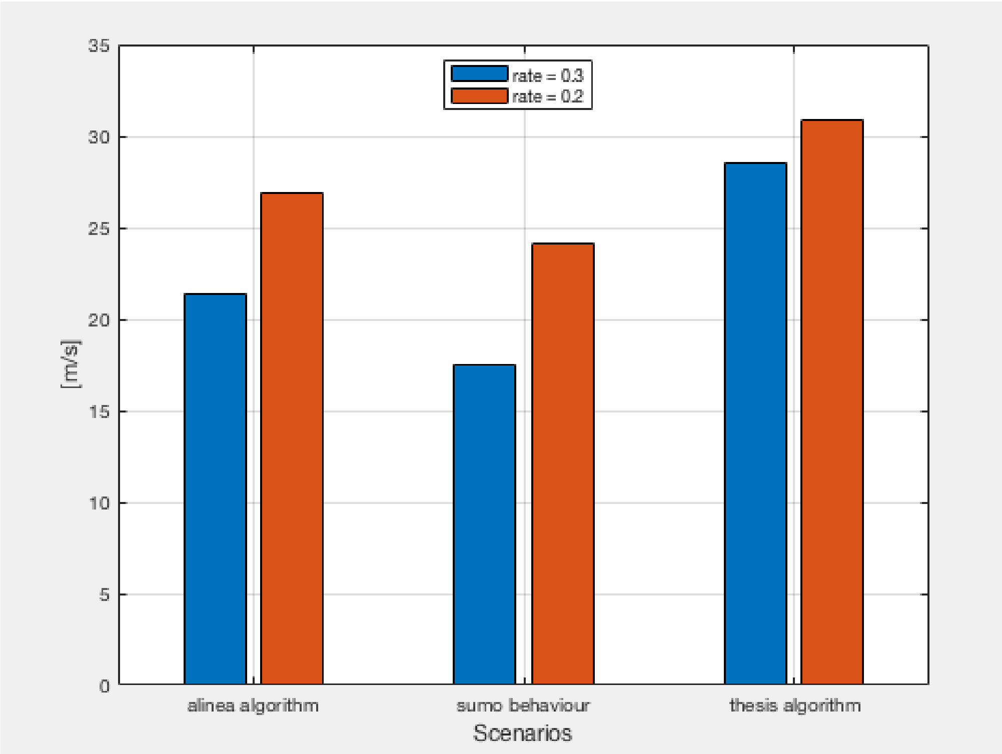

Figure 5.9: Mean speed in the lane 0 before the accident

As we have seen from the analysis of the time of imprisonment, both the algorithms bring benefits in this part of the system. The algorithm analyzes the upstream traffic and allows vehicles on lane 0 to change lane long before they get stuck behind the obstacle. This translates into an increase in the speed of 40% with respect to the mean speed of SUMO behavior. While, when ALINEA is applied, the entrance of the vehicles, stopped behind the obstacle, in lane 1 brings an increase of the mean speed of 20% compared to the average speed obtained by SUMO. These results are produced with a generation rate equal to 0.3 [veh/s]. When the traffic conditions are low, the average speed in this portion of the street is increased by 23% with thesis algorithm and by 10% thanks to ALINEA.

Mean speed in the lane 1 before the obstacle

In this portion of the highway the mean speed of SUMO behavior is higher than the ones acquired with the two algorithms. Even this result, which apparently may seem negative, actually highlights the effectiveness of the algorithms. Both the algorithms, through the server, slowdown the flow on lane 1 to allow the insertion of the vehicles behind the obstacle. While SUMO allows all the cars of the free lane to travel with the highest possible speed.

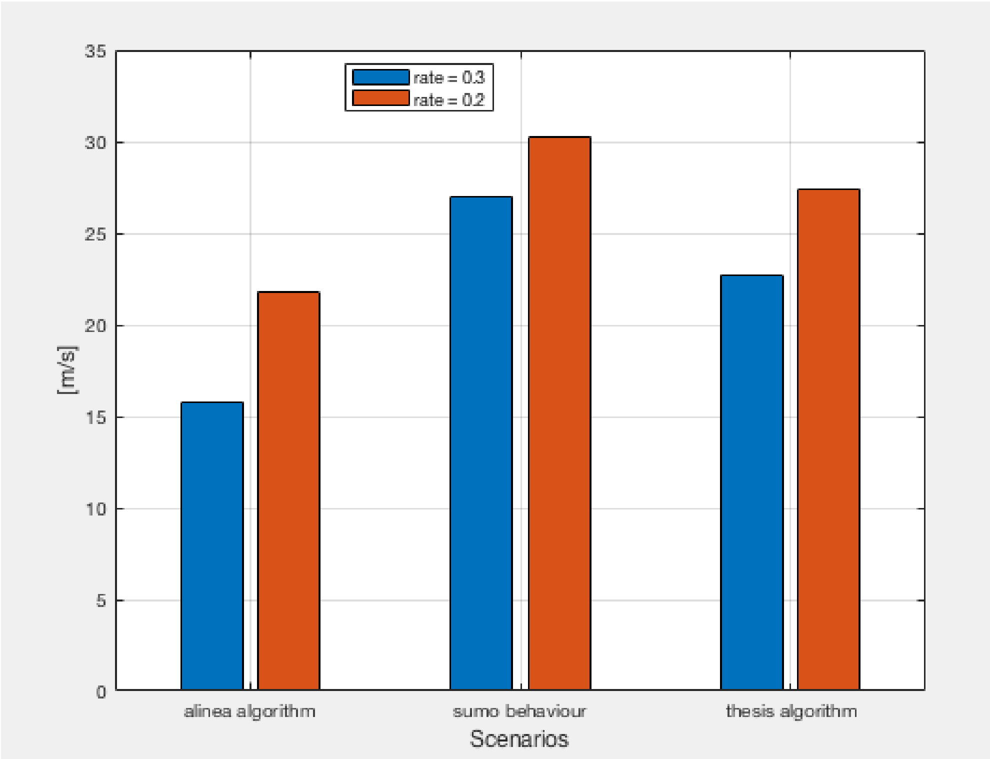

Figure 5.10: Mean speed in the lane 1 before the obstacle with the two generation rates

Mean speed in the lane 1 after the obstacle

The mean speed in the lane 1 after the accident is a parameter used to control the flow downstream of the obstacle and to verify if there are other slowdowns.

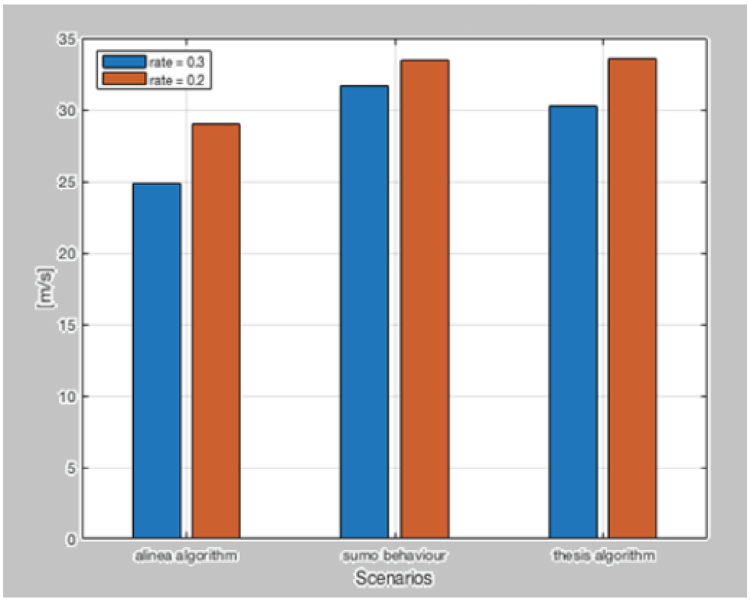

Figure 5.11: Mean speed in the lane 1 after the accident

As it is clear from the Fig. 5.11, seems that SUMO behavior produces the highest mean speed with both the generation rates. When the percentage of vehicles that do not exit the simulation has been analyzed, we said that this result would have influenced some of the following ones. The average speed in this part of the highway, shown in Fig. 5.11, does not consider, in fact, the stationary cars but only those that have overcame the obstacle. When only SUMO behavior is applied, only the cars entering the simulation in lane 1 are considered in this portion of the highway. This because, SUMO does not slowdowns vehicles on lane 1 and does not encourage cars on lane 0 to the lane change. So in the lane 1 after the obstacle there is 28% less than vehicles. Obviously the remaining 72% of the vehicles has a very high average speed having never slowed. Whereas the thesis algorithm is driven by a "non-sacrificial" approach, which translates into a percentage of vehicles that do not finish the simulation very low, as analyzed before. This means that vehicles on lane 1 slowdown to encourage the change of lane of cars on lane 0. So when it is applied the algorithm, the portion of the lane 1 after the obstacle includes both cars from lane 1 and lane 0, with a consequent decrease in the average speed. Same considerations apply to ALINEA. Hence it is mandatory to consider this percentage, which significantly influences the mean speed of the lane 1 after the obstacle, in order to know the real average speed and the benefits of the thesis algorithm

The following image is the plot of the mean speed of lane 1 after the obstacle, weighed on the percentage of cars which do not exit the simulation.

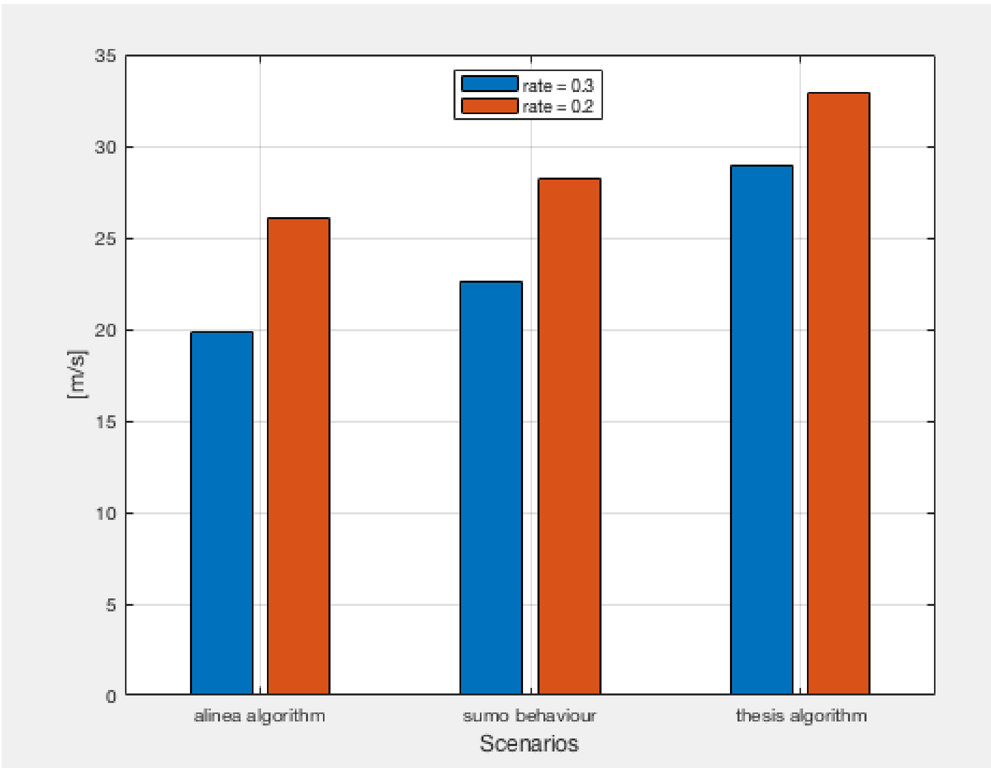

Figure 5.12: Weighed mean speed in the lane 1 after the obstacle with both the generation rates

So the lower the percentage of vehicles that do not end the simulation, the lower the effect on the average speed on this portion of the highway. In case of SUMO behavior, with rate equal to 0.3 [veh/s], taking into account this percentage, the mean speed is decreased by 30%. Also this result highlights the goodness of the thesis algorithm.

5.6 Statistics on the mean speed of the lanes before and after the obstacle

In this section are analyzed the the mean speed of the two lanes together before and after the obstacle, as it possible to see from the picture below:

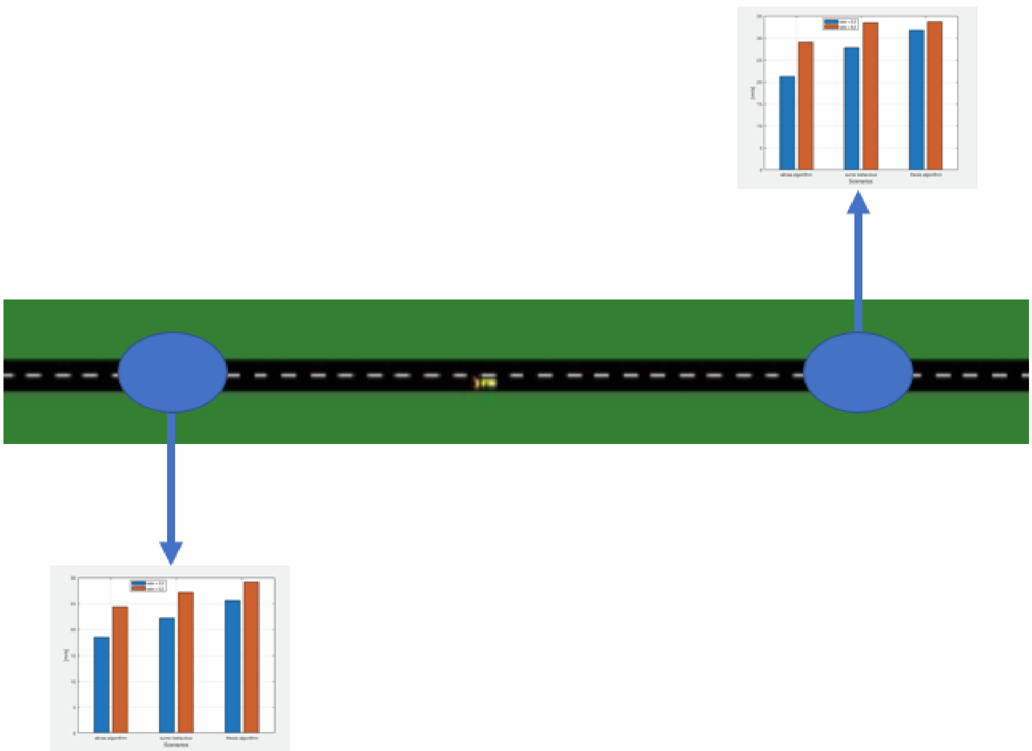

Figure 5.13: Statistics on the mean speed of the lanes before and after the obstacle. From the left to the right: mean speed of the two lanes before the obstacle and mean speed of the two lanes after the obstacle. Both the generation rates are simulated.

Overall, the average speed before the obstacle, shown in Fig. 5.14, is greater in the case of thesis algorithm than in the other two cases, thanks to the fact that the vehicles can change immediately lane and have a null or very low waiting time. In this way the vehicles of lane 0 can compensate the reduction of the mean speed of lane 1.

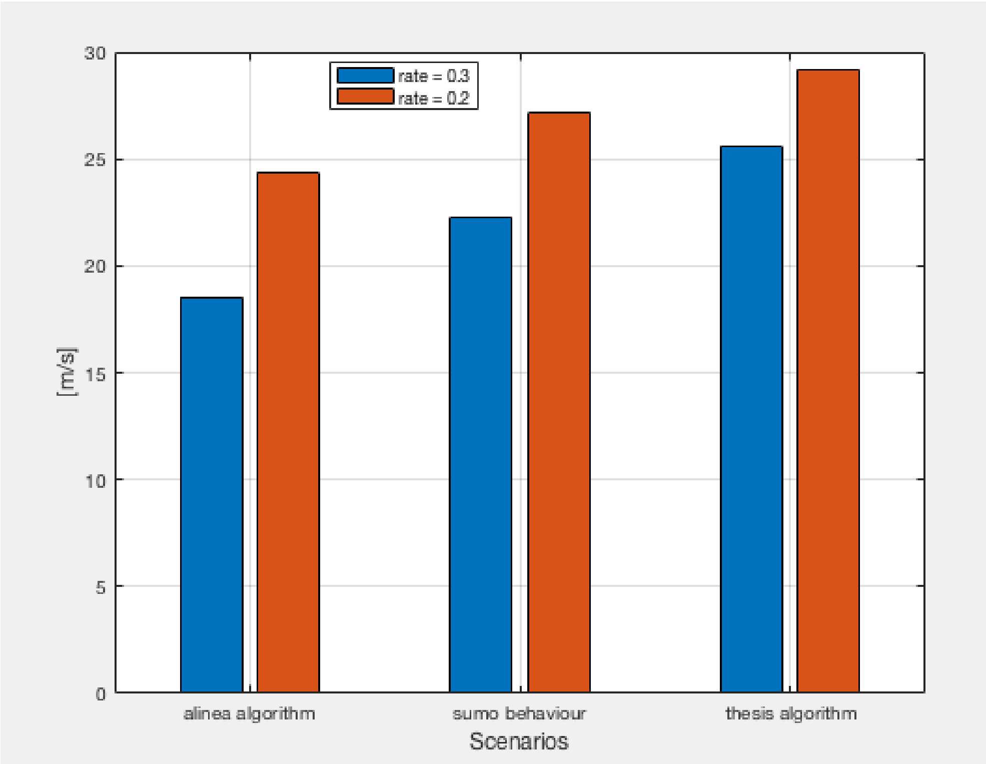

Figure 5.14: Mean speed of the two lanes before the obstacle produced with the two generation rates

The excellent result reported by the thesis algorithm in the two lanes after the obstacle (Fig. 5.15) is due to the consideration of the percentage of vehicles which do not exit the simulation and the redistribution of the cars in the lanes. The latter is another strength of the algorithm, because in this way other slowdowns or incidents are avoided.

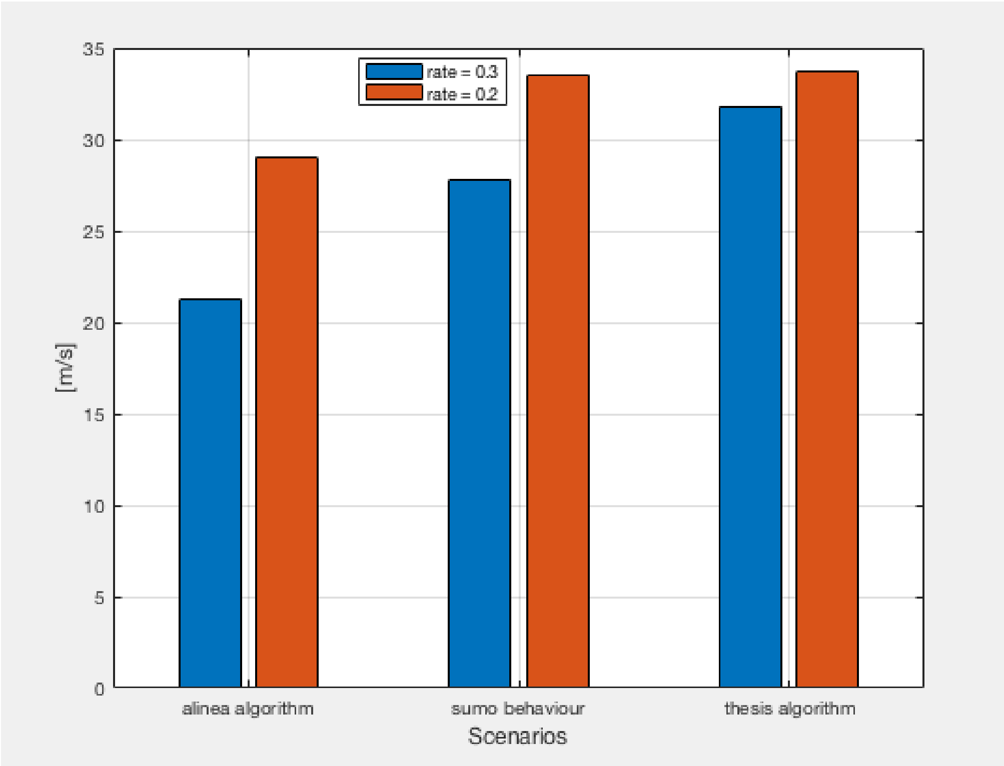

Figure 5.15: Mean speed of the two lanes after the obstacle with both the generation rates

5.7 Statistics on the mean speed of lane 0 and lane 1

In this section are considered the mean speed on the lane 0 before and after the obstacle and the mean speed of the lane 1 before and after the obstacle, as shown in Fig. 5.16

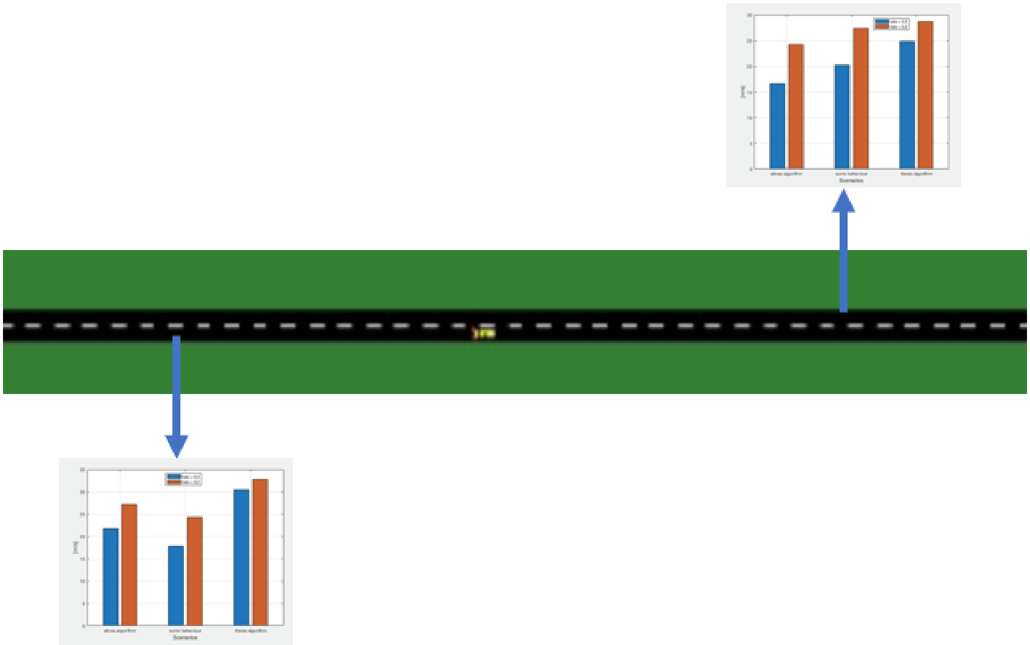

Figure 5.16: From the left to the right: mean speed of lane 0 and mean speed of lane 1. Both simulated with the two generation rates

In the whole lane 0 (Fig. 5.17), thanks to the application of the algorithms, in high traffic condition, the mean speed has increased by 45% with the thesis algorithm and by 23% with ALINEA. In low traffic condition with thesis algorithm the mean speed is 27% bigger than one of SUMO behavior and 12% with ALINEA algorithm.

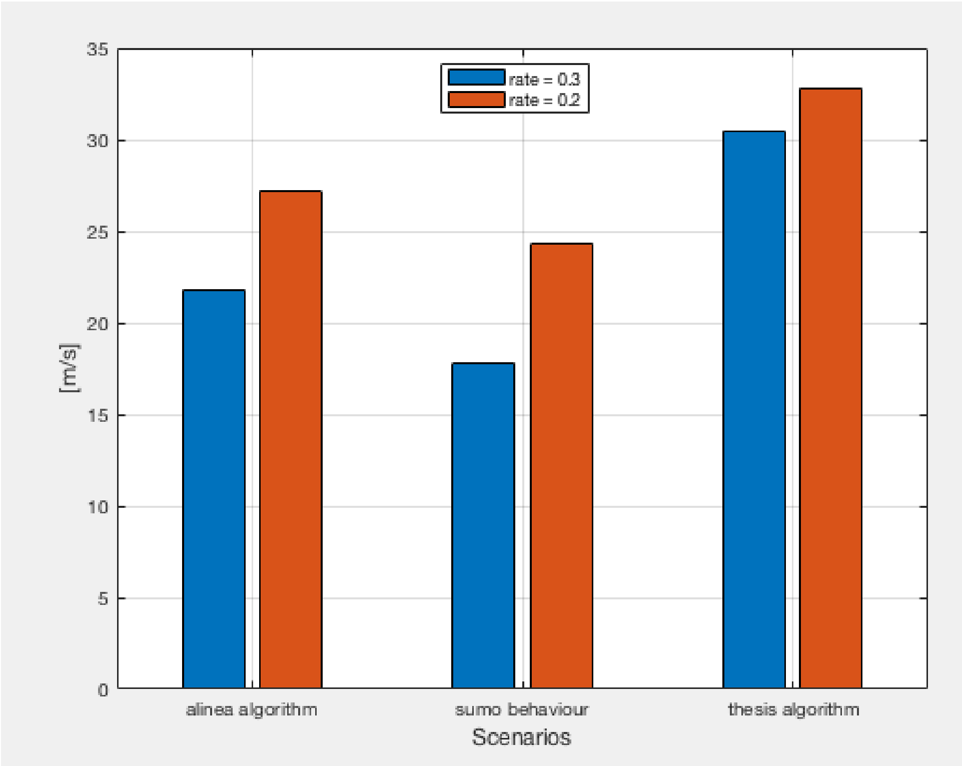

Figure 5.17: Mean speed of the entire lane 0 with the two generation rates

In the lane 1 (Fig. 5.18) the mean speed depends on the percentage of the vehicles fall to exit the simulation. Since the smallest percentage is achieved with the thesis algorithm, the latter produce the highest mean speed in the lane 1.

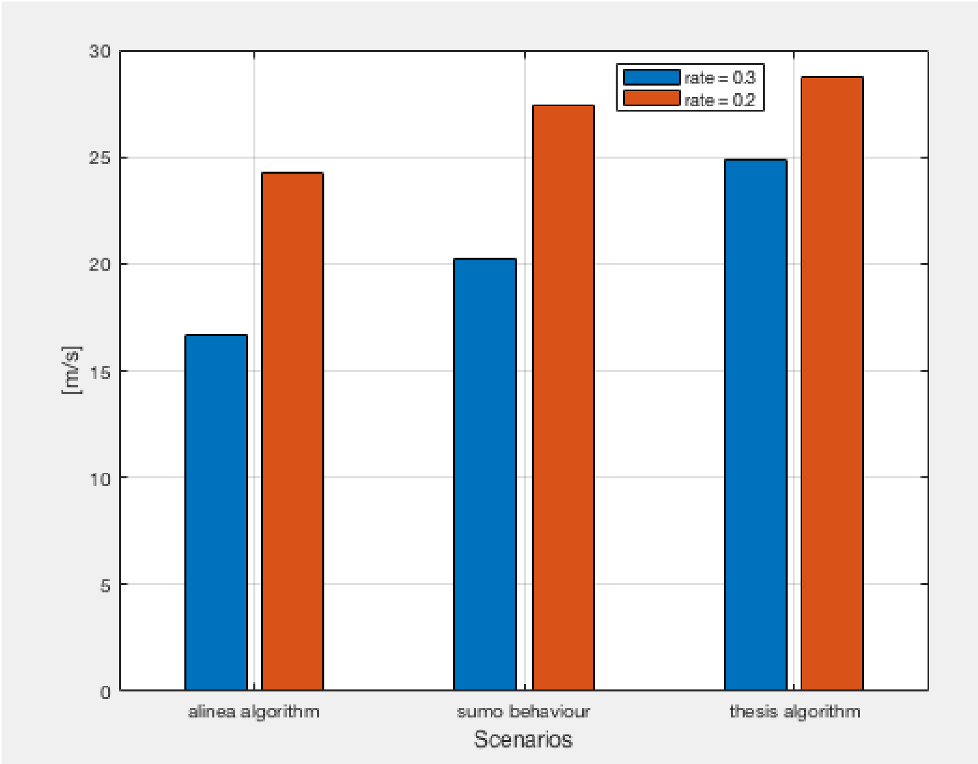

Figure 5.18: Mean speed of the entire lane 1 with both the generation rates

5.8 Mean space on lane 0 before the obstacle

The last KPI shows the average number of meters traveled in lane 0 before the obstacle.

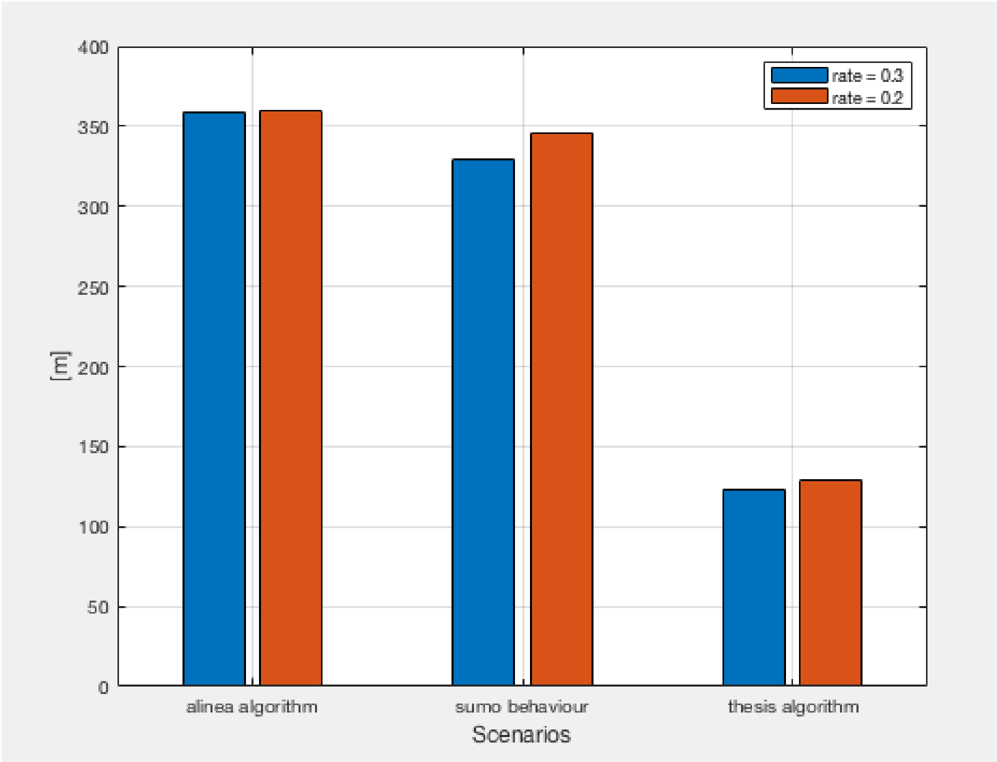

Figure 5.19: Mean space on lane 0 before the obstacle

As can be seen from Fig. 5.1, with the application of the algorithm, the average distance traveled is little thanks to the fact that the vehicles can immediately change lanes without creating the queue. As regards the behavior of SUMO, the average distance traveled is slightly lower than that of ALINEA. This is because the cars in SUMO create a long queue on lane 0 and the last ones that access the simulation can only travel a short distance, due to the length of the queue, while ALINEA tries to dispose of the latter during the green phase of the cycles, thus shortening it. Therefore when a car enters the simulation on lane 0, it travels more space before stopping.

Chapter 6

Conclusion

In this thesis, an efficient anti-congestion algorithm server based, tested in different condition of vehicles distribution on the two lanes, has been proposed. By exploiting the transmission of CAMs trough the server, the latter can determine which is the best maneuver to do and which vehicle can apply it. After analyzing all the KPIs, it can be concluded that the thesis algorithm has achieved the established purposes:

1. The waiting time behind the obstacle has been reduced by 80%, compared to the one obtained with SUMO behavior and by 35% compared to the waiting time of ALINEA, which is translates into a 0.5 probability, cars never stop behind the obstacle with both generation rates. In addition, no other slowdowns are created after the accident.

2. The mean speed of the whole system has been increased by 12% in high traffic conditions and by 15% in low traffic conditions compared to the one generated by SUMO behavior; also, the mean speed of each lane has increased.

Through the communication on cellular network it is easier to monitor the traffic conditions, so the server is able to make right decisions. This is also due to the fact that the server, hosting the developed algorithm, is located inside multi-access edge computing (MEC), whithin the eNB. Given the space proximity between the server and the users, the system is characterized by low latency, real-time access to the radio network information and high bandwidth. In this way the server receives and processes CAMs quickly.

The main strengths of this algorithm are:

- a clear overview of the traffic upstream of the incident;
- the management of a change of lane before a car reaches the obstacle.
- the generation of "holes" in the traffic of lane 1 when there is a car stuck behind the obstacle.

For what concernss ALINEA, it benefits the system in terms of waiting time, time of imprisonment, percentage of vehicles which do not exit the simulation and performance of speed before the obstacle. Interpreting the zero lane as an access ramp to the motorway leads to the creation of a queue which is then disposed of thanks to the alternation of the green and red phases and the right choice of cycle length, as well as the distribution of time in the two phases.

However, SUMO combined with the two network simulators is not the best way to simulate this algorithm, since it does not guarantee the application of the maneuver instantaneously. The latter is essential when one is working with cycles in which the duration of the action is of the order of one second.

The technical data, such as time to collision and space to collision, are provided by Centro Ricerche Fiat (CRF).

Chapter 7

Future work

The work in the thesis is only a possible development of this type of algorithm. Even within the same project can be made changes and new results can be studied. For example it is possible to decrease the penetration rate, i.e the percentage of vehicles connected, and study the behavior of the algorithms in the event of mixed scenario with autonomous and connected vehicles and vehicles driven by human beings only. An other type of modification is to change the processing time of the command received by the server, in order to verify if there are difference in the application of the maneuvers. Then it is possible verify the behavior of the algorithm when the damaged car starts again to move. Hence it is necessary a different traffic management.

In addition, it would be interesting to work with priority levels, for examples inserting police cars or ambulances. In this case, the server must prioritize the latter, changing commands sent to cars. Or it is possible to add vehicles which have to meet lower speed, such as coaches and trucks.

Another study can be carried out by changing scenario moving in an urban environment, where there are also pedestrians. An accident in the city is more difficult to manage, because the roads are narrower, there are pedestrians crossing the road, there are crosses or roundabouts. So it is possible to test the vehicle-to-pedestrian (V2P) communication.

For what concerns the communication technologies there are several directions for future research that can be envisioned:

- a study on the integration of the information obtained by CAMs and by sensors of the vehicles;
- a study of the integration of the communication on 5G network and artificial intelligence on vehicles;
- a study on benefits that $Vehicle\ to\ Vehicle$ (V2V) can provided.

Obviously, with the integration of 5G the latency will be considerably reduced, hence the commands will be received immediately, thus leading to a clear improvement in traffic conditions.

Bibliography

[1] The 3G4G Blog 3G4G.CO.UK. *The 3G4G Blog: Release 14*. URL: https://blog.3g4g.co.uk/search/label/Release%2014.

[2] Michael Behrisch et al. "Sumo–simulation of urban mobility". In: *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*. Vol. 42. 2011.

[3] Ericsson Research Blog. *Boosting Data Rates in LTE*. URL: https://www.ericsson.com/research-blog/boosting-data-rates-lte/.

[4] Alastair Brydon. *Enhanced small cells are the key to traffic boom*. URL: https://www.unwiredinsight.com/2013/enhanced-small-cells.

[5] Alastair Brydon. *Highlights of 3GPP Release 12*. URL: https://www.unwiredinsight.com/2014/highlights-of-3gpp-release-12.

[6] MECISG ETSI. "Mobile edge computing (mec); framework and reference architecture". In: *ETSI, DGS MEC* 3 (2016).

[7] Fabio Giust et al. "MEC deployments in 4G and evolution towards 5G". In: *ETSI, White Paper, under publication in February* (2018).

[8] GTI. *GTI 5G Network Architecture White Paper*. URL: http://www.gtigroup.org/d/file/Resources/rep/2018-02-22/06608ce6dbe32673ac1611359e11f794.pdf.

[9] Gunjan Indrayan. *LTE Advanced – Key features and differentiators*. URL: https://wirelesstelecom.wordpress.com/tag/hetnet/.

[10] C Krajzewicz and D Rössel. *SUMO User Documentation Rev. 1.24*. 2006.

[11] Daniel Krajzewicz et al. "Sumo (simulation of urban mobility)". In: *Proc. of the 4th middle east symposium on simulation and modelling*. 2002, pp. 183–187.

[12] Giovanni Nardini, Antonio Virdis, and Giovanni Stea. "Simulating cellular communications in vehicular networks: making SimuLTE interoperable with Veins". In: *arXiv preprint arXiv:1709.02208* (2017).

[13] NEC. *Making 5G a reality*. URL: http://www.nec.com/en/global/solutions/nsp/5g_vision/doc/wp2018ar.pdf.

[14] Gáspár Péter, Szalay Zsolt, and Aradi Szilárd. "Highly Automated Vehicle Systems". In: *BME MOGI* (2014).

[15] JS Roessler. "LTE-Advanced (3GPP Rel. 12) Technology Introduction White Paper". In: *München: Application Note-1MA252-Rohde & Schwarz International* (2015).

[16] Rohde Schwarz. *SON(Self-Organizing Networks,)* URL: https://www.rohde-schwarz.com/kr/solutions/test-and-measurement/wireless-communication/wireless-5g-and-cellular/lte-lte-advanced/in-focus/self-organizing-networks_229096.html.

[17] Andras Varga. "OMNeT++". In: *Modeling and tools for network simulation.* Springer, 2010, pp. 35–59.

[18] Antonio Virdis, Giovanni Stea, and Giovanni Nardini. "SimuLTE-A modular system-level simulator for LTE/LTE-A networks based on OMNeT++". In: *Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), 2014 International Conference on.* IEEE. 2014, pp. 59–70.

[19] Michael Zhang et al. "Evaluation of on-ramp control algorithms". In: (2001).