POLITECNICO DI TORINO

Master's degree in Mechatronic Engineering

VEHICULAR COMMUNICATION FOR ROAD SAFETY: SIMULATION AND ANALYSIS OF EMERGENCY VEHICLE PREEMPTION USING TRAFFIC LIGHT-BASED V2I TECHNOLOGY

Supervisors

Prof. Claudio Ettore Casetti

Dott. Angelo Borneo

Candidate

Chiara Deriu

Academic Year 2023/2024
October 2024

ACKNOWLEDGEMENT

Before presenting this thesis work, I would like to express my gratitude to MCA for giving me the opportunity to carry out my thesis project with them. A special thanks goes to Dr. Angelo Borneo, who supervised me throughout the process with great professionalism, allowing me to develop my project idea and always providing the right support when needed.

I would also like to sincerely thank my supervisor, Prof. Claudio Ettore Casetti, for his constant availability and quick responses. His support, especially during challenging moments, was crucial to the success of this work.

Lastly, I would like to thank the Politecnico di Torino, which has accompanied me throughout my academic journey, providing me with a top-level education and preparing me for future challenges.

INDEX

TABLE OF FIGURES

ABSTRACT

In an era of growing urbanization and complex of transportation networks, emerging technologies in the fields of the Internet of Things (IoT) and Intelligent Transportation Systems (ITS) are revolutionizing the way vehicles communicate with each other and with surrounding infrastructure. This thesis explores the evolution of these new technologies, outlining the continuous progression from assisted driving, enabled by Advanced Driver Assistance Systems (ADAS), to autonomated driving made possible by vehicular to everything (V2X) communication.

The research begins with an overview of intelligent transportation systems, followed by a detailed analysis of Vehicular Ad Hoc Networks (VANETs). It examines various communication protocols, such as Dedicated Short-Range Communication (DSRC) and Cellular-V2X (C-V2X), highlighting their advantages, challenges, and potential future developments, such as the introduction of new standards like IEEE 802.11bd and 3GPP releases 18 and 19.

A simulation was conducted to evaluate the integration of these protocols in a realistic scenario, focusing on the preemption of emergency vehicles through traffic light control. By utilizing the Simulation of Urban Mobility (SUMO) software for traffic network creation, combined with Omnet++ and Vehicles in Network Simulation (Veins) for communication modelling, the scenario illustrates how vehicle-to-infrastructure (V2I) communication can enhance road safety, optimize traffic flow, and improve emergency response times. The simulation demonstrates that, when emergency vehicle preemption is activated, the system significantly reduces delays for emergency vehicles at intersections, allowing them to navigate through in as little as 7 seconds compared to 20 seconds in non-preemptive scenarios. This leads to a marked improvement in traffic flow and emergency response efficiency.

The thesis also explores the expanding role of smartphones as potential communication nodes within vehicular networks. With these devices now playing a central role in daily life, their integration with existing infrastructure offers promising opportunities to further enhance road safety and traffic efficiency.

The result of this research demonstrates how the integration of ADAS systems and, subsequently, V2X technologies can not only increase road safety but also promote the development of smarter and more sustainable mobility.

INTRODUCTION

In the context of increasing urbanization and the expansion of transportation networks, the need to improve road safety, traffic efficiency, and environmental sustainability has become a global priority. Emerging technologies in the field of the Internet of Things (IoT) and Intelligent Transportation Systems (ITS) are revolutionizing the way vehicles communicate with each other and with surrounding infrastructure, paving the way for new possibilities for the future of mobility. Vehicular communication has emerged as a key component in the development of advanced solutions to address challenges related to safety, traffic management, and emission reduction.

This thesis delves into the evolution of vehicular communication technologies, with a specific focus on Vehicle-to-Everything (V2X), Advanced Driver Assistance Systems (ADAS), and the communication protocols that enable interaction between vehicles and infrastructure. The analysis concentrates on the capability of these technologies to support assisted and automated driving, highlighting how these innovations are transforming the automotive sector, leading to a convergence between telecommunications and transportation.

The first part of the thesis offers an overview of intelligent transportation systems, followed by a detailed analysis of Vehicular Ad Hoc Networks (VANETs), which form the foundation of vehicular communications. Subsequently, various communication protocols, including DSRC and C-V2X, are examined, highlighting their advantages, challenges, and potential future directions, such as the development of the IEEE 802.11bd standard and the evolutions in 3GPP releases 18 and 19.

The second part of this thesis focuses on the simulation of realistic scenarios aimed at understanding the importance of integrating these communication protocols. Using the SUMO simulation software, the Omnet++ simulation framework, and the Veins connectivity software, the scenario of emergency vehicle priority was selected for analysis. The choice of this scenario is motivated by the high frequency and severity of incidents involving emergency vehicles such as ambulances, fire trucks, and law enforcement vehicles. For instance, in Germany, between 2014 and 2019, nearly 600 accidents involving ambulances occurred, resulting in over 1,100 injuries and dozens of deaths. Similar situations are recorded in other European countries such as Austria and Switzerland, and in major cities like London, where the number of incidents is steadily increasing.

In the United States as well, statistics highlight the severity of the issue: road accidents are one of the leading causes of death for firefighters and police officers while on duty. These data not only underscore the danger of emergency situations but also emphasize the urgency of developing innovative solutions to improve the safety of those on the front lines.

This work aims to explore the potential offered by simulation in analysing and optimizing the dynamics related to emergency vehicle priority. Simulation is a crucial tool for modelling and predicting the behaviour of vehicles in emergency situations, thereby helping to reduce the risk of accidents and improve response times. Moreover, the integration of advanced driver assistance systems and intelligent infrastructure offers new opportunities to make these operations even safer and more efficient. The analysis of the interaction between emergency vehicles, road units, and traffic lights allows for the identification of solutions that can be implemented in real-world contexts to safeguard human lives and optimize available resources.

Finally, the thesis explores the growing role of smartphones in vehicular communications, highlighting how these ubiquitous devices can integrate with existing infrastructure to improve traffic safety and efficiency.

Through this research, it is intended to demonstrate how the integration of V2X and ADAS technologies can not only enhance road safety but also foster the development of smarter and more sustainable mobility. The implications of such developments are profound, with potential benefits that go beyond merely reducing road accidents, encompassing a comprehensive optimization of the urban and interurban transportation system.

1. VEHICULAR COMMUNICATION

1.1 INTELLIGENT TRANSPORTATION SYSTEM

Intelligent transportation system is one of the main applications in the Internet of things (IoT).

What is IoT? It refers to a network of interconnected physical devices, vehicles, appliances, and other items embedded with sensors, actuators, software, and connectivity capabilities that enable them to collect, exchange, and act on data [1].

ITS refers to a sophisticated network of technologies, infrastructure, and applications aimed at enhancing the efficiency, safety, and sustainability of transportation systems. It integrates various information and communication technologies to manage and optimize traffic flow, reduce congestion, improve road safety, and provide travellers with real-time information and services [2].

The major components of ITS [2] include vehicle and user applications, road infrastructure, and a communication network (VANETs). Vehicle and user applications encompass various technologies and services designed to improve the travel experience and enhance safety for drivers and passengers. These may include navigation systems, driver assistance features, and real-time traffic information accessed through mobile apps or in-vehicle displays. Road infrastructure refers to the physical elements of the transportation network, such as roads, bridges, traffic signals, and toll booths. These components play a crucial role in facilitating the movement of vehicles and ensuring the safety and efficiency of the transportation system. The communication network (VANETs) serves as the backbone of ITS, enabling seamless connectivity between vehicles, infrastructure, and control centres. Through this network, vehicles can exchange information with each other and with roadside infrastructure to support applications such as traffic management, emergency response, and vehicle-to-infrastructure communication. Together, these components form a comprehensive system aimed at optimizing transportation operations, enhancing mobility, and improving overall transportation efficiency and safety.

1.2 OVERVIEW OF VANET

As we have seen before a vital component of ITS is the communication network (VANET). A VANET falls within the category of Mobile ad Hoc Networks (MANETs), constituting a specific instance thereof. MANET systems aim to establish connections between mobile devices, typically those operated by human power, to form a network. These networks are categorized as ad-hoc due to the absence of managing infrastructure devices such as routers or phone towers.

VANETs consist of two primary entities: On-Board Units (OBUs) and Road-Side Units (RSUs). OBUs are internal devices installed in vehicles, facilitating communication and information exchange within the VANET network. They enable vehicles to communicate directly with each other (Vehicle-to-Vehicle or V2V) and with roadside units (Vehicle-to-Infrastructure or V2I). RSUs, on the other hand, are stationary access points strategically deployed along roadways. They provide connectivity and support communication between OBUs and the broader infrastructure, enabling Infrastructure-to-Vehicle (I2V) communication as well.

VANET architecture allows three communication possibilities [3]:

- *Vehicle-to-vehicle (V2V) ad hoc network:* direct wireless communication between vehicles without relying on a fixed infrastructure support. The data transmitted encompass speed, location, direction, traffic updates, driver conduct, road conditions, and other pertinent details essential for travellers.
- *Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Vehicle (I2V) network:* communication between vehicles and roadside infrastructures and viceversa. *It is* aimed at enhancing safety applications such as collision avoidance, emergency assistance, and improving overall mobility.
- *Hybrid Architecture:* both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication are combined. In this setup, vehicles have the capability to communicate with roadside infrastructure either directly or through multiple hops, depending on the distance. This enables connections over longer distances, facilitating access to the Internet or communication with vehicles that are farther away.

While VANETs share some characteristics with MANETs, they also possess distinct features [3]:

- *Mobility*: vehicles exhibit higher speeds, spanning from urban streets to highways, while their movements, though swift, retain a degree of predictability, facilitating the design of anticipatory networking protocols. Moreover, the network accommodates both macro mobility, encompassing inter-area travel, and micro mobility, which pertains to localized movement, thereby shaping communication protocols and handover strategies.
- *Network topology*: highly dynamic, characterized by frequent network partitions and intermittent connectivity between vehicles. Multi-hop communication and adaptive routing protocols are employed to cope with these challenges and ensure efficient message delivery amidst the ever-changing vehicular environment.
- *Resources:* Unlike some traditional wireless networks, VANETs are not typically constrained by limitations in energy, memory, or computation capabilities. Vehicles are often equipped with sufficient resources to support communication and processing tasks. However, efficient resource

management strategies are still important to optimize network performance and prolong the lifespan of devices.

- *Localization:* Vehicles in VANETs are typically equipped with satellite positioning systems such as the Global Positioning System (GPS) to determine their geographical location accurately. GPS enables various location-based services and applications in VANETs, including navigation, routing, and localization of nearby vehicles or infrastructure elements.

Vehicular Ad-Hoc Networks (VANETs) have emerged as a transformative technology within the realm of Intelligent Transportation Systems (ITS), offering immense potential to revolutionize road safety, traffic management, and overall transportation efficiency. Their unique characteristics, including high-speed mobility, dynamic network topology, abundant resources, and precise localization capabilities, enable the development of innovative applications that can significantly impact the future of transportation.

VANETs hold immense promise for enhancing road safety by facilitating real-time information sharing among vehicles, enabling cooperative collision avoidance, and providing emergency alerts. By enabling vehicles to communicate with roadside infrastructure, VANETs can support traffic signal optimization, adaptive routing, and cooperative lane management, leading to smoother traffic flow and reduced congestion.

The deployment of VANETs presents both challenges and opportunities. Ensuring reliable and secure communication, addressing privacy concerns, and standardizing protocols across different regions are crucial aspects that need to be carefully considered. However, the potential benefits of VANETs far outweigh these challenges, paving the way for a safer, more efficient, and sustainable transportation system.

As VANET technology continues to mature, its integration into ITS will undoubtedly transforms the way we travel, making roads safer, traffic more efficient, and transportation more sustainable. The future of transportation is undoubtedly intertwined with VANETs, and the possibilities for innovation and improvement are limitless.

2. ASSISTED AND AUTOMATED DRIVING

Before proceeding with the description of ADAS and V2X it is necessary to describe the levels of driving automation defined by SAE.

In 2014 SAE released the first lunch of J3016. In 2021, in partnership with International Organization for Standardization (ISO), the latest version was published; With this collaboration a more consistent document is available for global community.

The classification [S1], [S2] depends on the features that are engaged in a vehicle, and it considers the three main characters in driving: human, driving automation system and other vehicle systems and components.

At Level 0, there is no automation, the driver is entirely responsible for all driving activities but there are limited features to provide warnings and momentary assistance. Level 1 entails basic driving assistance, such as cruise control or adaptive cruise control, but the driver remains the primary safety controller. At Level 2, the vehicle can control both steering and acceleration/deceleration simultaneously under certain conditions, but the driver must stay engaged and constantly monitor the surrounding environment. Level 3 allows the vehicle to handle most driving tasks under certain conditions, enabling the driver to disengage from driving duties but requiring intervention when prompted by the system. Level 4 denotes high automation, where the vehicle can perform all driving tasks under specific conditions without human input, though situations may arise where the driver needs to take control. Lastly, at Level 5, the vehicle can execute all driving activities under any condition without human intervention, eliminating the need for a human driver.

From the differences existing between level 2 and level 3 it is possible to distinguish between ASSISTED and AUTOMATED driving.

Advanced Driver Assistance Systems (ADAS) oversee assisted driving, while the integration of communication networks (V2X) catalyses the emergence of automated driving.

2.1 ADVANCED DRIVER ASSISTANCE SYSTEM

The percentage of car accidents made it necessary to introduce a roadmap for fully assisted driving. The first innovation was the Anti-Lock braking system (ABS) introduced in the latest 70s.

ADAS [S3] aimed to assist driver according to the variations in environmental situations. It should be capable of sensing, analysing, predicting, and reacting to the road context, this is a key factor in assisted guidance.

ADAS consists of the following sensors [4]:

- *Proprioceptive sensors:* to detect and react to dangerous situations.

- *Exteroceptive sensors:* Capable of reacting proactively and forecasting potential hazards at an earlier stage.
- *Sensor networks:* Utilization of multi-sensory platforms in conjunction with traffic sensor networks.

Existing technologies will be analysed in the following sections [5].

AUTONOMOUS CRUISE CONTROL SYSTEM

Cruise Control is an automated system designed to regulate a vehicle's speed. Autonomous or Adaptive Cruise Control goes a step further by adjusting the vehicle's speed to ensure a safe following distance from other vehicles. This adjustment relies solely on sensor data collected by onboard sensors. ACC employs either RADAR or LIDAR sensors to decelerate when needed and resume the preset speed when conditions permit.

ACC can be either radar-based or laser-based. Radar-based ACC systems often include a pre-crash system designed to alert drivers and offer brake support. Single radar systems are the most prevalent. Conversely, laser-based systems are incapable of detecting hazards in tough weather conditions. The sensors in laser-based systems are consistently exposed and typically situated in the lower grille, offset to one side of the vehicle.

The next-generation cooperative system (CACC) incorporates data transmitted from a leading vehicle within the same lane.

ADAPTIVE HIGH BEAM SYSTEM

An Adaptive High Beam System is a feature in automotive lighting technology that automatically adjusts the range and intensity of a vehicle's headlights based on surrounding conditions. This system utilizes sensors to detect oncoming traffic or vehicles ahead and adjusts the headlight beam to avoid glaring other road users while maintaining optimal visibility for the driver. It can switch between high and low beams seamlessly, providing maximum illumination without causing discomfort or danger to other drivers. It has been introduced firstly in Mercedes E-Class in 2009.

GLARE FREE HIGH BEAM & PIXEL LIGHT

"Glare-Free High Beam" refers to an automotive lighting technology designed to provide maximum illumination with high beam headlights while minimizing glare for other drivers. This system employs sophisticated sensors and control mechanisms to selectively adjust the distribution or dim the light, ensuring optimal visibility without causing discomfort or hazards to surround motorists.

"Pixel Light" describes an advanced headlight system that utilizes individual light-emitting diodes (LEDs) or clusters of LEDs to generate precise and customizable lighting patterns. These systems dynamically adjust the direction, intensity, and spread of light to adapt to changing driving conditions, such as curves, oncoming traffic, or weather variations. By offering more efficient and effective illumination, Pixel Light enhances safety and comfort for both drivers and other road users.

AUTOMATIC PARKING

This feature integrates comprehensive parking assistance systems that autonomously identify available parking spots and assist the driver during parking manoeuvres. By assuming control of the steering, these systems require the driver only to manage the vehicle's speed. Advanced systems require only 80 cm of space beyond the length of the car to park longitudinally, they can brake automatically in case of emergency. This functionality is made possible through the integration of a sophisticated network of ultrasonic sensors positioned around the perimeter of the vehicle.

The first prototype was developed at INRIA at a Lieger electric car in the 1990s.

BLIND SPOT MONITOR

This system monitors the car blind spot, that area where a passing vehicle hasn't yet come into view in the side mirror and is hidden by the car's rear pillar. Radar sensors installed at the back monitor the space behind and beside the car, picking up the presence of another vehicle. When a potential danger is detected, the system alerts the driver by activating an LED indicator in the corresponding side mirror. If the driver still activates the turn signal despite the warning, the LED starts flashing more intensely, grabbing the driver's attention to the potential hazard. Additionally, some systems intervene with the steering wheel.

CROSSWIND STABILIZATION

First featured in 2009 by Mercedes Benz. It is designed to help vehicles to maintain steer-ability and balance in the presence of strong lateral forces, particularly in highways or open roads. ADAS typically utilizes sensors to detect the intensity and direction of crosswinds in real-time. Based on this information, the system can adjust the vehicle's stability control systems, such as the Electronic Stability Control (ESC) or Active Steering systems. Advanced versions can regulate suspension forces based on the forces of the crosswinds, thus damping the vibrations influencing the vehicle body oscillations.

COLLISION AVOIDANCE SYSTEM

A collision avoidance system is a safety feature in vehicles aimed at minimizing the impact of accidents. It employs radar, laser, and camera technology to identify potential collisions before they occur. Upon detection, the system can either alert the driver to an impending crash or intervene automatically, independent of driver input, to mitigate or avoid the collision.

DRIVER ATTENTION ASSISTANCE

These systems oversee the driver's alertness level. While some analyse driving habits, others observe eye movements or head positions. In the event of potential danger, they emit an audible alert, often accompanied by a coffee cup symbol, prompting the driver to take a break.

DRIVER DROWSINESS DETECTION

Car technology to prevent accidents caused by driver drowsiness. Various technologies like steering pattern monitoring, vehicle position in lane monitoring, driver eye/face monitoring and physiological measurement are used to detect the issue and prevent dangerous situations.

EMERGENCY DRIVER ASSISTANT

It operates through the integration of four driving assistance systems. If sensors detect that the driver is unconscious, signalled by a period of inactivity involving neither acceleration, braking, nor steering, the system initiates coordinated countermeasures. Initially, the driver receives auditory, visual, and physical prompts (such as a gentle brake pulse). If there is no response, the emergency stop procedure activates: hazard lights flash, and the Adaptive Cruise Control (ACC) assists in automatic braking to prevent the vehicle from colliding with preceding vehicles. Even on multi-lane roads, the system steers in a controlled manner towards the rightmost lane until the vehicle stops, while simultaneously alerting emergency services.

HILL DESCENT CONTROL SYSTEM

Hill Descent Control enables a gradual and managed descent on rugged terrain without requiring the driver to engage the brake pedal. Once activated, the vehicle utilizes the Anti-lock Braking System (ABS) to regulate the speed of each wheel during the descent. If the vehicle starts to accelerate without any driver’s input, the system intervenes by automatically applying the brakes to reduce speed to the desired level.

TURNING ASSISTANT

It is a ‘new’ system introduced in 2015. It aimed to help drivers navigate intersections more safely and with greater confidence by assisting them to avoid a collision with a vehicle, pedestrian, or cyclist at a cross street.

WRONG-WAY DRIVING WARNING

Introduced in 2010 to avoid wrong-way guidance. Its operation relies on a camera installed on the windshield, which reads road signs (Traffic Sign Recognition), paying particular attention to no-entry signs.

INTELLIGENT SPEED ADAPTION

The system monitors the vehicle speed. If it overcomes the speed limit the system acts by regulating the velocity. This is done using an advisory system or an intervention system. It will be mandatory from July 2024.

COGNITIVE CARS

Cognitive cars represent an interesting project in ADAS, mixing artificial intelligence and machine learning algorithms with automotive technology to realize vehicles that can sense, learn and communicate with the environment.

2.2 VEHICLE TO EVERYTHING COMMUNICATION

V2X is a communication system that ensures the exchange of data among vehicles, pedestrians, infrastructures, and networks in their vicinity.

The types of communication are schematized in *Figure 1* and include:

- Vehicle-to-Vehicle (V2V) guarantees direct data exchange between vehicles due to the Wi-Fi based technologies.
- Vehicle-to-Infrastructure (V2I) admits data exchanged between vehicles and infrastructure (RSU).
- Vehicle-to-Pedestrian (V2P) guarantees the exchange of data between vehicles and vulnerable road users like pedestrians, bicycles, and scooter.
- Vehicle-to-Network (V2N) guarantees communication with a remote application server hosting a network-based service.

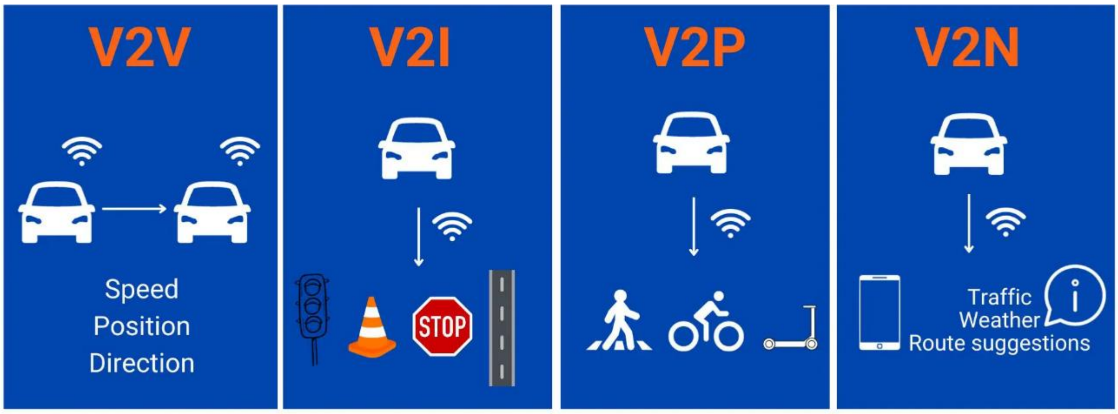

Figure 1 - Types of V2X communication.

The utilization of V2X technology within automotive and transportation sectors aims to enhance roadway safety, optimize traffic flow, minimize energy usage, and collaborate with ADAS systems to achieve the realization of Automated Driving.

Compared to the VANET introduced in the previous chapters is the origin and the point of view to change. It emphasizes its operation in Ad Hoc communication mode and primarily supports road safety applications through V2V and V2I connections, predominantly using DSRC communication mode. At the other side, V2X communication encompasses a broader range of communication technologies, including DSRC, C-V2X and other emerging technologies to satisfy the requirements on safety and efficiency.

Nowadays, two communication protocols are used for V2X: DSRC and C-V2X. These will be described and analysed in the next chapter.

3. COMMUNICATION PROTOCOLS

In this chapter, we enter the intricate field of communication protocols, specifically exploring the dynamic interaction between Dedicated Short-Range Communications (DSRC) and Cellular Vehicle-to-Everything (C-V2X). Communication protocols serve as the pillar of vehicular communication systems, facilitating the exchange of critical information among vehicles and infrastructure. DSRC, a mature technology rooted in IEEE 802.11 standards, has long been considered the frontrunner in vehicular communication. On the other hand, C-V2X, exploiting cellular networks, has emerged as a promising alternative with its potential for enhanced scalability and compatibility. This chapter aims to dissect the strengths, weaknesses, and comparative features of DSRC and C-V2X protocols, highlighting their suitability for diverse vehicular communication applications and opening the way for informed decision-making in the domain of intelligent transportation systems.

3.1 DSRC

This section aims to describe to the reader a group of protocols designed for vehicular communication, which nowadays represent one of the most researched and utilized V2X technologies.

These standards have been introduced by IEEE and ETSI, with IEEE employing the Wireless Access in Vehicular Environment (WAVE), which stands as the predominant standard in America, while ETSI has introduced ITS-G5 in Europe.

3.1.1 WIRELESS ACCESS IN VEHICULAR ENVIRONMENT

IEEE WAVE system delivers radio communication that is interoperable, efficient, and dependable, serving applications that enhance safety and convenience within an Intelligent Transportation System (ITS).

It includes 802.11p standard for the physical and the MAC layer (channel access) and the 1609.x standards for the MAC layer (channel coordination) and the upper layers.

Physical layer

In 2010, the IEEE released a specific version of the 802.11 standard known as 802.11p. It stands as the predominant DSRC standard for vehicular communication.

At the physical layer, 802.11p incorporates several features borrowed from the 802.11a standard. These include Orthogonal Frequency Division Multiplexing (OFDM), which divides the available spectrum into multiple subcarriers to enhance data transmission reliability and efficiency. Operating within the 5.850/5.925 GHz band, 802.11p allocates seven 10 MHz-wide channels for communication, enabling robust and interference-resistant connections in vehicular environments [6]. Data rates ranging from 3 Mb/s (utilizing BPSK modulation) to 27 Mb/s (utilizing 64 QAM modulation) are supported.

Modulation	Coding rate	Data bits per OFDM symbol	Data rate
BPSK	**1/2**	**24**	**3 Mbit/s**
BPSK	3/4	36	4.5 Mbit/s
QPSK	**1/2**	**48**	**6 Mbit/s**
QPSK	3/4	72	9 Mbit/s
16-QAM	**1/2**	**96**	**12 Mbit/s**
16-QAM	3/4	144	18 Mbit/s
64-QAM	3/4	192	24 Mbit/s
64-QAM	3/4	216	27 Mbit/s

Table 1: 802.11p data rates with 10 MHz channels

BPSK (Binary Phase Shift Keying) modulation is a digital modulation technique commonly used in telecommunications and digital communication systems. In BPSK modulation, the carrier signal phase is shifted to represent binary data. Specifically, in BPSK, there are two possible phase states: 0° and 180° (or π radians), corresponding to the two binary symbols 0 and 1.

64-QAM (Quadrature Amplitude Modulation) is a digital modulation scheme used in telecommunications to transmit data over radio waves. It is an extension of QAM, which combines both amplitude and phase modulation. The name "64-QAM" indicates that there are 64 different possible states (or combinations of amplitude and phase) that can be used to represent data.

MAC layer

To understand how data are exchanged, it's important to introduce the MAC layer, which serves two main functions: *channel access* and *channel coordination*. The former is regulated by 802.11p, while the latter by the IEEE 1609.4.

The *channel access* is based on Distributed Coordination Function (DCF). It operates without the need for a Basic Service Set (BSS) establishment, enabling vehicles and roadside infrastructure to communicate directly in an ad-hoc manner. This decentralized approach facilitates rapid deployment and flexibility, essential for dynamic vehicular scenarios. Additionally, WAVE utilizes a contention-based mechanism for channel access, allowing devices to compete for channel access based on priority levels defined by the Enhanced Distributed Channel Access (EDCA) mechanism. EDCA defines four access categories (ACs), each assigned a specific priority level [6]:

- Voice (AC_VO): This category is assigned the highest priority and is typically reserved for real-time voice communication, such as emergency calls or vehicle-to-vehicle (V2V) communication involving urgent safety alerts.

- Video (AC_VI): The video category is prioritized next, suitable for streaming applications that require timely delivery with relatively low latency, such as video surveillance or traffic monitoring.
- Best Effort (AC_BE): This category is for non-time-sensitive data traffic, providing a balance between throughput and latency. It is commonly used for general data exchange, such as internet browsing or software updates.
- Background (AC_BK): The lowest priority category is background, intended for non-urgent or bulk data transfers that can tolerate longer delays. Examples include file downloads or software backups.

The access to the categories is obtained after an *Internal Contention*. How it works? Each AC has a unique AIFS, representing the time interval before a station can start contending for the channel. Lower-priority ACs have longer AIFS values, introducing a delay to prioritize higher-priority traffic. Stations use a backoff mechanism with a contention window (CW) to randomly select a transmission slot. The size of the CW varies based on the AC and the number of consecutive failed transmission attempts. Higher-priority ACs typically have smaller CWs, allowing them to contend more aggressively for channel access. After selecting a backoff value, stations contend for channel access by monitoring the channel for idle periods. When the channel is idle for the duration of the selected backoff interval, a station initiates a transmission attempt.

Overall, EDCA plays a crucial role in enabling efficient and reliable communication in vehicular environments by providing differentiated access to the wireless medium, ensuring that safety-critical messages are delivered promptly while maintaining support for various other types of traffic.

As we said before, the *channel coordination* is regulated by IEEE 1609.4.

IEEE 1609.4 standardizes the multi-channel operations within vehicular communication systems, introducing essential concepts and protocols to ensure efficient and reliable data exchange. It defines two distinct channel types: the control channel (CCH) and the service channel (SCH). The CCH is primarily utilized for critical functions such as service advertisement, there is only one CCH on CH178, while the SCH accommodates various types of services and messages, there are 6 SCHs. Channel routing mechanisms are established to guide data transmission based on the service type and priority level, optimizing network performance. Time synchronization and coordination between the CCH and SCH are crucial for maintaining coherence within the system, with Coordinated Universal Time (UTC) serving as the common time reference. Additionally, IEEE 1609.4 defines three channel switching modes [6] to manage the transition between control channel (CCH) and service channels (SCH) in vehicular communication systems:

- *Continuous Mode*: it ensures that devices remain synchronized with the network and are ready to always handle service requests. It is suitable for scenarios where a constant presence on the CCH is required for critical control functions.

- *Alternating Mode*: Alternating mode involves periodic switching between the CCH and SCH. Devices alternate between listening to the CCH and engaging in service-related communication on the SCH according to a predefined schedule. Alternating mode is suitable for scenarios where intermittent access to the CCH is sufficient for coordinating communication activities.
- *Immediate Mode*: Immediate mode enables devices to switch channels instantly in response to specific events or triggers. When a device detects the need to transmit or receive service-related data, it immediately switches to the SCH without waiting for a predefined schedule. This mode prioritizes timely communication over channel access coordination, ensuring rapid response to critical events or service requests. Immediate mode is suitable for scenarios where quick channel switching is essential for maintaining connectivity and responsiveness in dynamic environments.

The standard also introduces primitives specifically designed for multi-channel operations, providing a foundation for efficient communication and coordination across diverse channels within vehicular environments.

Upper Layers

WAVE Networking and WAVE Security are standardized respectively by 1609.3 and 1609.2 [6].

IEEE 1609.3 introduces the Wave Short Message (WSM) and the corresponding WAVE Short Message Protocol (WSMP) to provide to the network and transport layers of DSRC safety applications without relying on IP-based transmissions. Alongside, it defines the WAVE Service Advertisement (WSA) message to signal the presence of DSRC messages at specific locations. Although non-safety applications can utilize WSMP, they primarily employ the legacy TCP/UDP and IP protocol stack. WSMP is engineered with minimal protocol overhead, making it ideal for single-hop collision avoidance messages, while multi-hop messages can leverage IPv6 routing.

IEEE 1609.2 describes the security framework for WAVE, it represents one of the most important components in ensuring the integrity and confidentiality of communications. IEEE WAVE security is underpinned by five core principles: authenticity, authorization, integrity, non-repudiation, and confidentiality. These principles collectively safeguard communication by verifying sender identity, ensuring sender privileges, maintaining data integrity, enabling third-party verification, and restricting message access to intended recipients.

When security is activated, a Secure Data Service (SDS) assumes responsibility for translating insecure packets into Secured Protocol Data Units (SPDUs) before transmission and vice versa upon reception.

A portion of the upper layers has not been standardized by IEEE but by SAE. This is the WAVE application layer, which consists of a set of messages defined by the SAE standard J2735 and includes 16 types of messages. Among these, it is important to highlight the following 5 messages [7]:

- *Basic Safety Message (BSM):* Periodic message transmitted with 10 Hz frequency, it is used both as a heartbeat to transmit position, speed, and direction and to report safety events.
- *Common Safety Request (CSR):* Unicast message transmitted by a vehicle engaged in the exchange of BSMs, aiming to request supplementary data from other vehicles. This request for additional information is prompted by the needs of on-board safety applications.
- *Emergency Vehicle Alert (EVA):* Broadcast message issued as a warning to all nearby vehicles regarding an emergency vehicle's presence. Upon receiving the Emergency Vehicle Alert (EVA), vehicles are expected to yield to the emergency vehicle and exercise heightened caution.
- *Intersection Collision Avoidance (ICA):* Event-triggered message broadcasted as a warning when a hazardous situation occurs near an intersection, heightening the risk of potential collisions. This message, known as a Collision Warning, can originate from either a vehicle's Collision Avoidance service (decentralized approach) or from infrastructure (centralized approach).
- *Road-Side Alert (RSA): E*vent-triggered message, spread by infrastructure, to notify travelers of nearby hazards. These hazards could encompass icy road conditions, an approaching train at a level crossing, or the presence of a construction zone.

3.1.2 ITS-G5

In [8] is introduced a new European standard for vehicular communications known as ITS-G5.

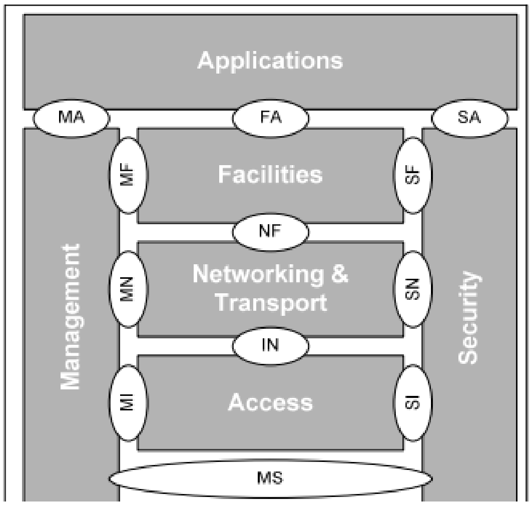

Figure 2 - ITS architecture.

This document analyses the architecture of the protocol designed to facilitate V2V communication in ad hoc network, specifically operating at the 5.9 GHz frequency band.

From the *Figure 2*, it is possible to highlight the introduction of a facilities layer between Networking and Applications.

In this chapter, we will mainly focus on the description and the analysis of the Access and the Facilities layers, with a brief overview of the remaining layers.

Access Layer

The access layer encompasses the data link layer and the physical layer.

The physical layer is addressed in IEEE 802.11, for this it uses OFDM operation with seven 10 MHz frequency channels as we have seen in the previous chapter, the novelly lies in the frequency allocation. ITS-G5 introduces a fine-grained service channel assignment divided in:

- *ITS-G5A*: it is set for road traffic safety applications.
- *ITS-G5B*: it is set for non-safety road traffic applications.
- *ITS-G5C*: it is referred to broadband radio access networks (BRAN), radio local area network (RLAN) and wireless local area network (WLAN).
- *ITS-G5D*: it is set for future usage of road traffic applications.

The data link layer comprises two sublayers: the medium access control (MAC) and the logical link control (LLC). While the MAC sublayer discerns between various network layer protocols, the LLC sublayer manages transmission scheduling to reduce interference among ITS stations. The main difference compared to the WAVE IEEE 802.11p lies in the way the messages are managed. While it used to rely on contention, now it is congestion-based, meaning on the network load management performed by the Decentralized Congestion Control (DCC) outlined in [9].

This method uses the Channel Busy Ratio (CBR) as a crucial metric for analysing the level of utilization of a communication channel by monitoring adjacent radio transmitters. It represents the percentage of time during which the channel is occupied by transmissions or signals, compared to the total time interval observed.

In its basic setup, as it is possible to see in *Figure 3,* a vehicle is categorized by three states: relaxed, active, and restricted. The shift between these states is controlled by monitoring the channel load within minimum and maximum thresholds over a designated time interval [10].

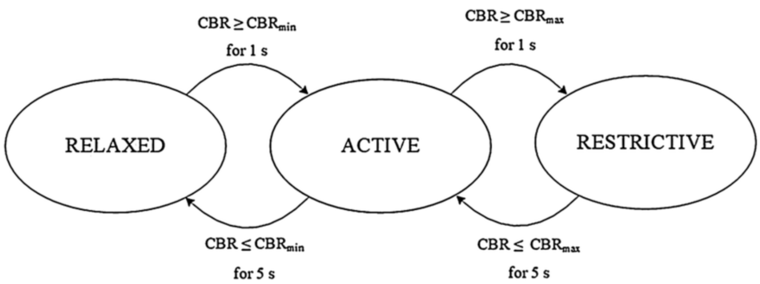

Figure 3 – State for DCC.

In the relaxed state, the vehicle operates with greater assertiveness on the channel, meaning it may employ higher transmission power or transmit messages more frequently. This state occurs when a low CBR is detected, indicating a situation of sparse connected vehicle density. Conversely, the restricted state is activated during high load conditions, requiring constraints on maximum transmission power or message frequency.

In network load management, four techniques are commonly employed. One such method is Transmit Power Control (TPC), which involves adjusting the output to regulate the current channel load. Another approach, Transmit Rate Control (TRC), governs the timing between consecutive packets, elongating the interval during periods of heightened utilization. Additionally, wireless systems offering multiple transfer rate options utilize Transmit Data Rate Control (TDC). Finally, the Sensitivity of Clear Channel Assignment (DCC) is instrumental in detecting a busy channel during packet reception. These techniques collectively contribute to the effective management and optimization of network resources in diverse communication environments.

Networking & Transport Layer

Within the Transport and Networking layers, two key protocols are featured: Basic Transport Protocol (BTP) and GeoNetworking.

BTP serves as a lightweight, connectionless transport protocol, facilitating the transmission of ITS messages to vehicles. It employs port numbers to determine the intended destination service, whether it's for managing periodic or event-based messages. ETSI outlines two variations of BTP headers: BTP-A, designed for interactive sessions requiring a reply from the recipient, and BTP-B, utilized for non-interactive message dissemination. BTP-A headers include both source and destination port information, whereas BTP-B headers solely feature destination port details along with an optional destination port info field, typically left unused and set to 0. Furthermore, the standard assigns a specific BTP port number for each type of vehicular message.

GeoNetworking [11] is a network layer protocol designed to facilitate packet routing within ad hoc networks. It leverages geographical positions for packet transport, enabling communication among individual ITS stations and facilitating packet distribution across geographical areas. Messages can be routed in three ways:

- *GeoUnicast*: used to facilitate the transmission of unicast messages by considering the destination location and the GeoNetworking address.
- *GeoBroadcast*: used for disseminating broadcast messages within a specified area indicated in the GeoNetworking headers. Messages can be forwarded as necessary until they reach the designed destination area.
- *Topologically-Scoped Broadcast*: employed for broadcasting messages to the nearest n hops within the network topology.

Facilities Layer

Figure 2 illustrates the addition of a novel layer termed Facilities Layer [12] positioned between the Applications Layer and the Networking & Transport layer. It assesses priorities among various messages and communicates available resources to regulate the channel load generated by each application. This layer can be divided into three sub-layers:

- *Application Support*: This sub-layer encompasses all services that offer message encoding/decoding functions and support for the ITS applications above it.
- *Information Support:* This sub-layer is responsible for storing and managing the Local Dynamic Map (LDM) used for cooperative perception. It stores all dynamic data received from other ITS nodes via CAM and DENM messages, allowing applications to access relevant information from the LDM as needed.
- *Communication Support:* This sub-layer provides future-proof lower layer independence by providing support for diverse access technologies (802.11p, VLC, LTE) and network protocols (IPv6 and non-IP).

This layer manages various standardized message types, among which the most significant are detailed below:

- *Cooperative Awareness Message (CAM):* it enables the provision of essential information regarding the presence, positions, and basic status of communicating ITS stations to neighboring stations within a single hop distance. All participating ITS stations in V2X networks can generate, transmit, and receive CAMs. By receiving CAMs, an ITS station gains awareness of nearby stations, their positions, movements, basic attributes, and sensor information. Upon receiving CAMs, efforts are made to assess the relevance of the information provided. This functionality allows ITS stations to obtain situational awareness and respond accordingly. CAM Management is considered a mandatory facility as the information it distributes is crucial for various use cases, such as Approaching Emergency Vehicle and Slow Vehicle Warning.
- *Decentralized Environmental Notification Messages (DENM):* DENM messages serve as event-triggered notifications primarily utilized by ITS applications to warn road users about detected events through ITS communication technologies. DENM messages encompass descriptions of various events detectable by ITS stations.
- *Service Announcement Message (SAM):* they are a type of message used within the ITS-G5 ecosystem to advertise available services over the control channel. These messages inform nearby ITS stations about the existence and details of services offered within the network. SAMs play a crucial role in facilitating communication and interaction between different ITS applications and stations by providing essential information about available services and their functionalities. Similar to WAVE Service Announcement (WSA).

- *Collective Performance Message (CPM):* Standardized in 2023. It facilitates the interoperable exchange of fundamental information concerning the broadcasting ITS-S, which is necessary for interpreting the transmitted data. It includes details about the sensory capabilities of the ITS-S, perceived objects, and road-related perception regions. CPMs are generated quasi-periodically based on CPM generation events.
- *Vulnerable Road Users Awareness Message (VAM):* Standardized in 2020, VAMs are messages sent from Vulnerable Road User (VRU) ITS-S to establish and sustain awareness of vulnerable road users engaged in the VRU system. They are similar to CAMs, and they are transmitted with a variable frequency between 0.2 Hz and 10 Hz.

In the following chapter, cellular-based protocols will be described, analysed, and compared to the protocols described thus far.

3.2 C-V2X

In recent years, the utilization of cellular networks for both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications has emerged as a highly promising technology. It offers a potential solution to the limitations of Wi-Fi-based protocols, particularly as vehicle density increases. However, it's essential to acknowledge that cellular communications alone may not fully meet the stringent requirements of V2X (Vehicle-to-Everything) communications. Challenges such as low-latency and high-reliability communication in scenarios with dense nodes, multiple concurrent transmissions and receptions remain unsolved by cellular networks alone.

Cellular networks have been effectively employed to support remote information and infotainment services, they boast advantages such as wide coverage, substantial capacity, high reliability, and scalability in industrial applications [13].

Currently, the debate over which protocol reigns supreme remains ongoing, but 802.11p continues to maintain its preference. It stands out as a more mature and extensively tested protocol, offering lower latency and superior performance compared to C-V2X.

The initial stride towards C-V2X was delineated by 3GPP within their Release 12 standard, which introduced Device-to-Device (D2D) communication under the Proximity Services (ProSe) framework. Initially tailored for mobile devices in public safety scenarios, this laid the foundation for C-V2X. The first comprehensive standardization of C-V2X technology within 3GPP emerged with LTE-V2X in Release 14. Presently, this technology is bifurcated into two phases: LTE-V2X and, as of Release 16, NR-V2X.

3.2.1 LTE-V2X

LTE-V2X serves as a proxy standard, encapsulating a subset of the 3GPP Release 14 specification, defining Cellular Vehicle-to-Everything (C-V2X) technology. It utilizes device-to-device communication (PC5) operating at 5.9 GHz, functioning independently of a base station presence. What is a device-to-device communication (D2D)?

D2D communication refers to direct communication between two mobile users, bypassing the need to relay through the Base Station (BS) or core network. Typically, D2D communication operates independently of the cellular network and can occur either on the cellular frequencies (inband) or in unlicensed spectrum (outband) [14].

Standardized by 3GPP in 2016 as part of LTE Release 14, LTE-V2X comprises two [S4] key interfaces:

- The wide area network LTE interface (Uu), facilitating connectivity between end-user devices, vehicles, mobile network base stations, and mobile core networks. This interface supports Internet access and enables Vehicle-to-Network (V2N) services.
- The direct communications interface (PC5), establishing connections between vehicles, roadside infrastructure, pedestrians, and other vulnerable road users. It enables low-latency and high-reliability vehicular services without necessitating mobile network assistance.

Concerning the Physical Layer, LTE-V2X utilizes Single Carrier Frequency Division Multiple Access (SC-FDMA) and supports 10 MHz and 20 MHz channels. In the frequency domain, each channel is partitioned into 180 kHz Resource Blocks (RBs), each containing 12 subcarriers of 15 kHz. In the time domain, channels are segmented into 1 ms subframes, with each subframe comprising 14 symbols featuring a normal cyclic prefix. Within a subframe, nine symbols are allocated for data transmission, while the 3rd, 6th, 9th, and 12th symbols are designated for transmitting Demodulation Reference Signals (DMRSs), enhancing link performance, and mitigating the Doppler effect, especially at high speeds.

A subchannel constitutes a contiguous group of RBs within the same subframe, with the number of RBs per subchannel being configurable beforehand. Subchannels serve as the fundamental units for transmitting both data and control information. Link control information is conveyed through Sidelink Control Information (SCI) blocks via the Physical Sidelink Control Channel (PSCCH), while data transmission occurs through Transmit Blocks (TBs) over the Physical Sidelink Shared Channel (PSSCH). Each TB is associated with an SCI and occupies 2 RBs. SCIs are transmitted concurrently with their corresponding TBs within the same subframe, containing vital information such as Modulation and Coding Scheme (MCS), ProSe Per-Packet Priority (PPPP) value, resource reservation interval, and an indication of RBs occupied by the TB. A TB encapsulates a complete ITS message packet and may span one or more subchannels based on factors like RBs per subchannel, MCS, and packet size. As of Release 14, TBs can be transmitted using either QPSK or 16-QAM, while SCIs are restricted to QPSK modulation due to their critical role in ensuring TB reception accuracy.

Release 14 [13] introduces two sub-channelization schemes: Adjacent PSSCH+PSCCH scheme and Nonadjacent PSSCH+PSCCH scheme. As it is possible to see in *Figure 4,* in the Adjacent scheme, TBs and their associated SCIs are transmitted over adjacent RBs. Conversely, the Nonadjacent (*Figure 5)* scheme divides the channel into two sections: one reserved for SCI transmission and the other for transmitting their associated TBs. Notably, the United States deployment has opted for the Adjacent PSSCH+PSCCH scheme.

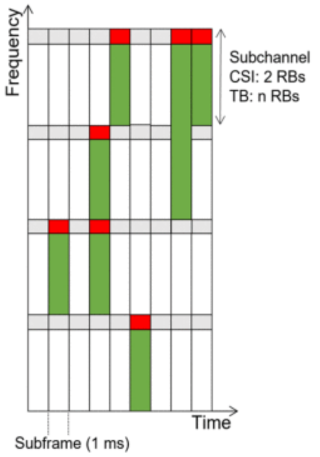

Figure 4 - Adjacent PSSCH+PS

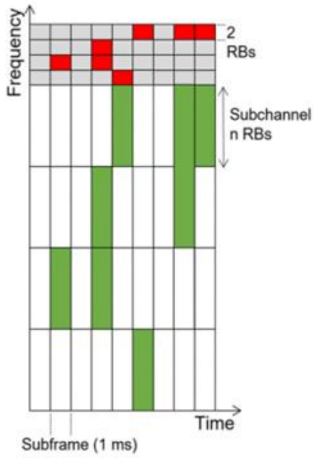

Figure 5 - NonAdjacent PSCCH+PSSCH.

V2V communication via PC5 in LTE_V2X can happen in two modes: Mode 3, which operates within network coverage, involves infrastructure managing sidelink D2D communication, including resource selection and configuration. On the other hand, Mode 4 is utilized when outside coverage, enabling vehicles to autonomously select and configure resources and sub-channels. Notably, Mode 4 operates independently of SIM cards or MNO subscriptions.

3.2.2 NR-V2X

In late 2017, 3GPP finalized Release 15 of the 5G standard [15], which focused on the New Radio (NR) technology, aiming to offer a more adaptable radio interface to provide to diverse business needs. Subsequently, NR-V2X technology was formally introduced during the second phase of 3GPP standardization efforts, spanning Release 16 (2020) and Release 17 (2022).

Before delving into the analysis of the features introduced by the two releases, it's worth highlighting the differences in the physical layer compared to LTE-V2X. In NR-V2X, the numerology of the PHY layer is no longer fixed, but rather a scalable parameter. As a result, subcarrier spacing in NR-V2X is not limited to 15 kHz anymore; it can now be increased up to 30, 60, or 120 kHz. This leads to varying OFDM symbol durations per numerology and a variable number of resource blocks (RBs) in each channel, given a fixed channel bandwidth (e.g., 10 MHz). Consequently, the number of sub-channels in a channel varies with the numerology.

Release 16 introduces notable enhancements across several features, particularly in multiple-input, multiple-output (MIMO) and beamforming, dynamic spectrum sharing (DSS), dual connectivity (DC), carrier aggregation (CA), and user equipment (UE) power conservation [15].

Enhanced beam handling and channel-state information (CSI) feedback, along with support for transmitting to a single UE from multiple transmission points (multi-TRP) and full-power transmission from multiple UE antennas in the uplink (UL), are introduced. These improvements boost throughput, reduce overhead, and enhance robustness. Mobility enhancements in Release 16 reduce handover delays, particularly beneficial for beam-management mechanisms in mm wave bands.

DSS enables cost-effective transitioning from 4G to 5G by allowing LTE and NR to share the same carrier. Release 16 expands the number of rate-matching patterns available in NR to facilitate spectrum sharing when CA is utilized for LTE. Moreover, Release 16 reduces latency for setting up and activating CA/DC, leading to enhanced system capacity and the capability to achieve higher data rates. Unlike Release 15, where measurement configuration and reporting occur only when the UE enters the fully connected state, in Release 16, connection resumption after periods of inactivity is streamlined without extensive signaling for configuration and reporting.

Additionally, Release 16 introduces aperiodic triggering of CSI reference signal transmissions for carriers with different numerology aggregation to enhance efficiency. To curb UE power consumption, Release 16 incorporates a wake-up signal alongside improvements to control signaling and scheduling mechanisms.

In Release 17 [S5], 3GPP has introduced several key achievements that lay the groundwork for enhanced network and service management, as well as charging capabilities, to support new features in 5G networks, setting the stage for Release 18 advancements. Extensive enhancements have been made to the 5G network resource models (NRM) to facilitate provisioning and data collection. This includes extending the NRM to accommodate new features of the 5G Core (5GC) and Next Generation Radio Access Network (NG RAN), as well as incorporating transport requirement information models for end-to-end network slice management. Release 17 introduces autonomous network (AN) capabilities, intent-driven management, and closed-loop service level agreement (SLA) assurance. These capabilities aim to reduce complexity in network management through automation and analytics, ensuring continuous alignment with user expectations and efficient resource utilization. Energy efficiency (EE) key performance indicators (KPIs) have been defined across various levels of the 5G network to address environmental concerns and optimize resource consumption. Additionally, service level agreement requirements for different network domains have been standardized to meet vertical industry needs, along with flexible management capabilities for non-public networks. Release 17 introduces proximity-based services charging, enabling operators to monetize services based on the proximity of 5G-enabled devices. Charging frameworks for edge computing services have also been extended, allowing for the monetization of edge application deployments and usage. Further enhancements include charging solutions for roamers with 5G data connectivity, simplified charging options for network slice usage, and ongoing studies to explore additional charging mechanisms and optimizations. These achievements in Release 17 pave the way for more efficient, intelligent, and monetizable 5G networks, setting the stage for continued innovation and evolution in Release 18 and beyond.

V2V communication in NR-V2X can happen in two modes: Mode 1 and Mode 2, which replaced Mode 3 and Mode 4 of LTE-V2X. The communication methods of these two protocols will be analysed in the next paragraph.

3.2.3 RESOURCE ALLOCATION

In contrast to 802.11 DCF, which relies on contention, V2X communication operates on resource allocation principles. Here, a resource refers to a collection of OFDM subcarriers within an OFDM slot, forming a Physical Resource Block (PRB). Effective communication between the eNB and the UE hinges on mutual agreement regarding the specific Physical Resource Blocks allocated for data exchange.

PRBs are allocated in either Scheduled or Autonomous mode [14]. In Scheduled mode, termed Mode 3 in LTE-V2X and Mode 1 in NR-V2X, both transmission (Tx) and reception (Rx) must occur within coverage areas, minimizing conflicts and collisions. Conversely, Autonomous mode, known as Mode 4 in LTE-V2X and Mode 2 in NR-V2X, allows Tx and Rx even outside coverage (provided they were within coverage at least once), potentially leading to collisions.

Scheduled Mode

In scheduled mode, there are two communication modes: dynamic grant and configured grant.

With Dynamic Grant, UEs are required to request resources from the base station for each Transmission Block (TB) transmission. UEs achieve this by sending a Scheduling Request (SR) to the gNB via the physical uplink control channel (PUCCH). In response, the gNB communicates the resource allocation information using Downlink Control Information (DCI) over the physical downlink control channel (PDCCH). The DCI specifies the allocated Shared Link (SL) resources, including the slot(s) and sub-channel(s), designated for transmitting a TB and allows for up to two possible retransmissions of the same TB.

Requesting resources for each Transmission Block (TB) increases delay. Configured grant scheduling option to mitigate this delay by pre-allocating Shared Link (SL) radio resources. Under this scheme, the gNB can allocate a set of SL resources to a UE for transmitting multiple TBs, referred to as a configured grant (CG). Initially, the UE sends a message to the gNB containing assistance information regarding expected SL traffic, including TB periodicity, maximum size, and QoS requirements such as latency, reliability, and priority. This information enables the gNB to create, configure, and allocate a CG that meets the SL traffic's demands. The CG is defined by parameters including the CG index, time-frequency allocation, and periodicity of the allocated SL resources.

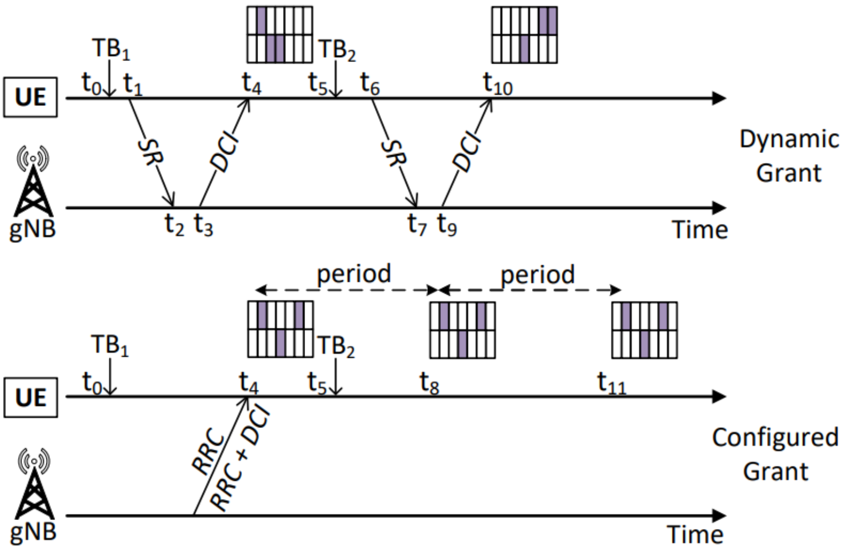

Figure 6 - Dynamic and Configured Grant

Autonomous Mode

In autonomous mode, UEs operate without direct communication with the gNB. While there is no active connection, UEs must have been previously authorized during their time within coverage. Within this mode, UEs autonomously select their Shared Link (SL) resources from a designated pool, enabling flexibility in resource allocation. The details of these chosen resources, along with parameters like Reselection Counter and Resource Reservation Interval, are communicated via Sidelink Control Information (SCI). Additionally, to ensure synchronization among devices, one UE is designated as the synchronization source, facilitating coordinated communication among all participating UEs. This autonomous approach to resource allocation empowers UEs to make efficient use of available resources while maintaining connectivity in scenarios where direct network control is limited.

3.2.4 SYNCHRONIZATION

The synchronization mechanism for NR-V2X sidelink is similar to LTE-V2X sidelink, operating synchronously. Each user equipment (UE) within the system must maintain consistent time and frequency references and, it tries to read PSCH and the SSCH (Primary/Secondary synchronization source). UEs determine frame, subframe, and slot timing based on a common time reference, derived from GNSS or the cellular network. Like LTE-V2X sidelink, NR-V2X introduces a similar mechanism to synchronize sidelinks, accommodating various V2X deployment scenarios, whether within or outside cellular coverage, and in areas with or without GNSS signals. When one or both UEs are out-of-coverage, their timings become unrelated. In such cases, the receiving UE requires supplementary information from another UE serving as a reference, known as the SyncRef UE [14].

A UE aspiring to be the SyncRef UE broadcasts the following: A Sidelink Synchronization Signals ID (SLSS ID) ranging from 0 to 335 if within the coverage of gNB/GNSS or ranging from 336 to 671 if outside the coverage of gNB/GNSS. Additionally, it transmits a coverage indicator (inCoverage flag, IIC).

The SLSS ID and IIC serve to differentiate whether a SyncRef UE is directly or indirectly synchronized to GNSS or a gNB/eNB. In cases of indirect synchronization, the SLSS ID and IIC also indicate whether the SyncRef UE is one hop or multiple hops away from a gNB/eNB or GNSS. Each hop corresponds to another SyncRef UE.

V2X UEs have four primary sources for synchronization: GNSS, gNB/eNB, another UE serving as a synchronization source, and an internal UE clock. Generally, GNSS and gNB/eNB are considered the most reliable sources. Depending on system configuration, either GNSS or gNB/eNB can be designated as the highest priority synchronization source, establishing the respective hierarchy of synchronization priority.

3.3 FUTURE DIRECTIONS

This chapter will examine the new protocols and future directions being pursued to enhance vehicular communication. The DSRC and C-V2X protocols are currently undergoing significant enhancements to meet the demanding requirements of advanced vehicular applications, including high reliability, low latency, and high throughput. These improvements are essential for augmenting existing vehicular sensors to facilitate autonomous driving. In this paper, the recent advancements in the standardization processes of 802.11bd, as well as Releases 18 and 19 of NR V2X by 3GPP will be analysed.

3.3.1 IEEE 802.11bd

In March 2018, the IEEE established a Study Group called Next Generation V2X (NGV) to develop an amendment to the IEEE standard focusing on enhanced V2X communication technologies. Subsequently, in December 2018, the IEEE-SA sanctioned this project, forming a Task Group tasked with producing IEEE 802.11bd, officially released in March 2023.

The IEEE 802.11bd [16],[17] standard is designed to complement and improve upon the existing IEEE 802.11p standard rather than replacing it. While many specifications in 802.11bd are based on 802.11p for communication between NGV STAs, the new standard emphasizes interoperability, coexistence, compatibility, and fairness. A substantial portion of 802.11bd focuses on realizing these attributes, with the remainder addressing technical enhancements and other relevant aspects.

Significantly differing from 802.11p, 802.11bd doubles the available bandwidth and introduces changes in frame format, leading to modifications in processing procedures and various parameters. The increase in data transfer rate primarily results from the wider bandwidth allocation, including channels in the 60 GHz frequency band with a bandwidth of 20 MHz, double that of 802.11p. With the same modulation and coding scheme (MCS) and coding rate (R), 802.11bd achieves an approximate 125% increase in data transfer rate.

As for the length of a single data packet, it is limited by the maximum allowable duration imposed by regulators, typically set at 10,968 μs or less. Higher data transfer rates in 802.11bd result in an increased maximum packet length, nearly doubling that of 802.11p. Specifically, while a single data packet transmitted by 802.11p cannot exceed 4095 bytes, 802.11bd extends this to nearly 7991 bytes. These advancements in packet length can yield significant performance improvements, especially in applications requiring small-security-class packets.

Ensuring data transmission reliability has emerged as a crucial performance metric in evaluating V2X standards. To tackle these challenges, the IEEE 802.11bd standard incorporates several new technologies and services aimed at enhancing Signal-to-Noise Ratio (SNR), including LDPC coding, midambles, and improved beamforming technology.

IEEE 802.11bd adopts LDPC coding to replace the Binary Convolutional Coding (BCC) utilized in 802.11p. LDPC coding offers several advantages over BCC, including lower encoding and decoding complexity, as well as a more evenly distributed check code pattern. These qualities empower LDPC coding to better withstand various channel interferences and deliver superior error correction, especially in high-SNR scenarios.

Concerning the data frame format in *Figure 7*, 802.11bd retains the 40 μs preamble from 802.11p while introducing a NGV preamble lasting at least 36.8 μs. Notably, the NGV preamble incorporates a flexible NGV-LTF field meticulously crafted to bolster data transmission reliability in real-world communication scenarios.

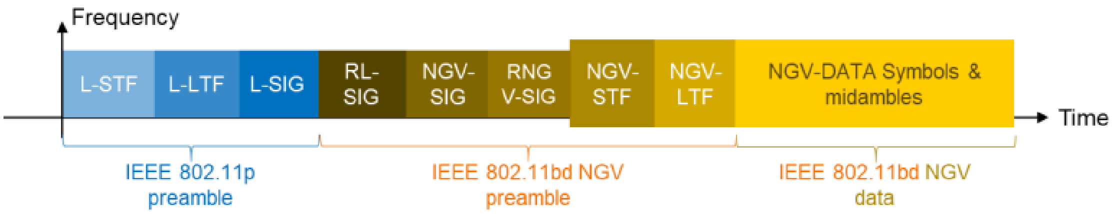

Figure 7 - 802.11bd data frame format.

In addition, the IEEE 802.11bd standard showcases significant advancements in communication performance, notably in data transfer rate, where modulation and coding scheme (MCS) and coding rate (R) play pivotal roles. While Veins has incorporated the MCS and R combinations utilized in 802.11p, this study introduces three new modes from 802.11bd: 64-QAM 5/6, 256-QAM 3/4, and 256-QAM 5/6.

3.3.2 3GPP RELEASE 18 & 19

Release 18

Release 18 marks the initiation of efforts towards 5G Advanced. The scope of Release 18 was deliberated extensively during the 3GPP radio access network (RAN) Release-18 workshop, which garnered over 500 proposals. As it possible to see in *Figure 8,* after six months of intense discussions, 3GPP endorsed its Release-18 package at the December 2021 RAN plenary meeting [S6]. This package encompasses 27 diverse study or work items aimed at enhancing network performance and catering to emerging use cases.

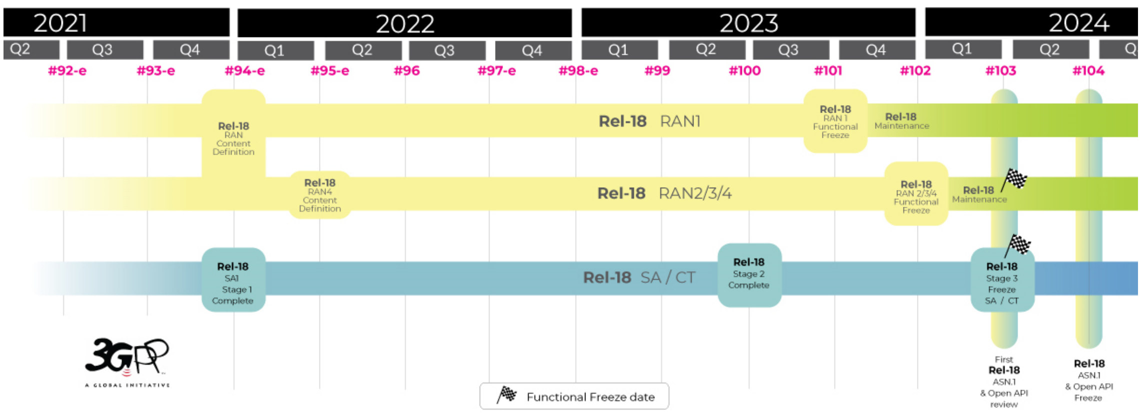

Figure 8 - Timeline of Release 18 evolution by 3GPP.

Among the first aspects analysed in this release, network energy savings are becoming increasingly important due to rising energy costs and the goal of reducing environmental impact.

Mobility will be addressed through the implementation of new lower-level mechanisms to reduce latency associated with mobility management. MIMO evolution aims to improve performance for users with medium or high mobility and increase the number of DMRS ports to support multi-user transmission. Multicast and broadcast transmission will be extended to inactive users, and new solutions will be studied to optimize resource usage. Finally, positioning will continue to improve, with a focus on accuracy, integrity, and energy efficiency in location services [18].

Spectrum is a valuable resource that must be utilized efficiently to maximize societal benefits. In 3GPP Release 18, enhancements will be introduced to enable more flexible and efficient spectrum utilization for 5G deployments across various scenarios with different spectrum allocations.

One key aspect addressed in Release 18 is the support for deploying 5G New Radio (NR) in spectrum allocations with less than 5 MHz bandwidth. This is particularly relevant for scenarios such as railway communication in Europe.

Dynamic Spectrum Sharing (DSS) is another important feature in 5G, enabling a base station (BS) to use shared spectrum to provide connectivity to both LTE and NR user equipment (UE). However, the capacity of the physical downlink control channel (PDCCH) in DSS scenarios can be limited, as NR PDCCH cannot use symbols overlapping with LTE cell-specific reference signals (CRS). In Release 18, 3GPP will study methods to increase PDCCH capacity for DSS by allowing NR PDCCH to be transmitted in symbols overlapping with LTE CRS [18].

Additionally, with the expectation that spectrum bands used by previous generations of mobile networks will be repurposed for 5G Advanced, Release 18 will introduce functionality to efficiently utilize fragmented spectrum blocks across different frequencies. This includes allowing a single downlink control information (DCI) to schedule multiple physical downlink shared channels (PDSCHs) or physical uplink shared channels (PUSCHs) across carriers.

Furthermore, Release 18 will explore the feasibility of enabling simultaneous downlink and uplink transmission within conventional Time Division Duplex (TDD) bands [18]. This will involve dividing TDD bands into non-overlapping sub-bands, with some designated for uplink and others for downlink. Cross-link interference (CLI) handling will also be studied to better support dynamic TDD in commercial deployments.

In addition, 3GPP Release 18 will persist in examining and implementing customized features to enrich and broaden 5G functionality, catering not only to smartphones but also to a diverse range of 5G devices. These include extended reality (XR) and cloud gaming devices, low-complexity UEs, vehicular devices, and unmanned aerial vehicles.

Coverage is paramount for operators aiming to deliver competitive mobile services to customers. While deploying full-stack base stations (BSs) in coverage gaps is a common approach to enhance network coverage, it's not always feasible due to factors like backhaul limitations or cost constraints. Hence, there's a need for a flexible network topology that boosts the resilience of the split Next-Generation

Radio Access Network (NG-RAN) architecture, leveraging diverse nodes such as Integrated Access and Backhaul (IAB) nodes, radio frequency (RF) repeaters, and integration with Non-Terrestrial Networks (NTNs).

As 5G networks evolve in complexity and generate substantial data volumes, harnessing AI techniques becomes increasingly vital for efficient network management. In 3GPP Release 18, there will be a dual focus: enhancing current data collection capabilities and exploring the integration of AI to optimize air interface functions. The release will particularly enhance NR data collection within the self-organizing network/minimization of drive testing framework (SON/MDT), aiming to automate Radio Access Network (RAN) planning, configuration, and optimization while minimizing human intervention. Moreover, it will investigate AI/ML applications to enhance air interface performance and streamline operations. This effort involves establishing a standardized AI/ML framework, identifying areas for enhancement, evaluating techniques, and assessing their potential standardization impact. Notable use cases include CSI feedback, beam management, and positioning, setting the stage for broader AI/ML integration across the air interface.

Release 18 is still in progress, with Stage 3 being moved from December 2023 to March 2024, and the final deadline from March to June 2024.

Release 19

The significant advancements and explorations undertaken in Release 18 serve as a robust groundwork for the progression of 5G Advanced in Release 19. Discussions regarding the scope of Release 19 commenced at the 3GPP RAN Release-19 workshop, attracting approximately 480 submissions from over 80 companies/organizations [S7]. It was agreed at the workshop that 5G-Advanced in Release 19 would primarily cater to commercial deployment requirements while also serving as a transition towards the sixth generation (6G) of mobile communication.

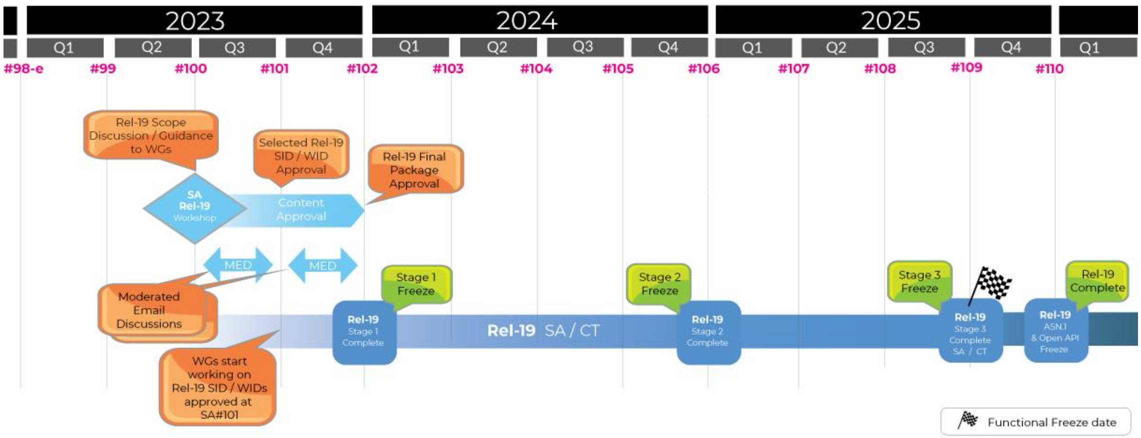

Figure 9 - Timeline of Release 19 by 3GPP.

Following six months of intensive deliberations (*Figure 9)*, 3GPP endorsed the Release-19 package at the December 2023 RAN plenary meeting [S7]. This package encompasses a selective set of study and work items aimed at enhancing network performance, addressing new use cases, and paving the way for 6G.

In this version, 3GPP aims to further boost 5G performance by introducing new functionalities across various domains, including MIMO evolution, mobility management, duplex operation, and XR support [S7].

MIMO stands as an intrinsic and pivotal technological element within the realm of 5G.n the current framework for beam management, a 5G node B (gNB) has the capability to request that a user equipment (UE) conduct beam reporting. This aids the gNB in identifying an optimal beam for communication with the UE. However, striking a balance between reporting overhead and latency/accuracy poses a challenge. To address this issue, 3GPP will delineate enhancements for UE-initiated beam reporting. This will empower the UE to trigger beam reporting autonomously, without waiting for initiation from the gNB.

In addition, 3GPP will increase CSI reporting ports from 32 to 128 to accommodate larger antenna arrays. Enhancements in coherent joint transmission (CJT) will address non-ideal synchronization scenarios. UE measurements for inter transmit-receive point (TRP) alignment and frequency/phase offset will be introduced. Non-coherent uplink codebook enhancements will support three transmit antennas in UEs. Furthermore, improvements will be made to facilitate heterogeneous network support, enabling simultaneous downlink reception from a macro gNB and uplink transmission to micro TRPs, enhancing uplink throughput.

Mobility support will see significant improvements, with the extension of mobility procedures to support inter-central unit (CU) mobility and event-triggered measurement reporting. Conditional handover will be introduced to enhance mobility robustness, reducing interruption time during handover. Time division duplex (TDD) introduced by Release 18 will be enhanced indicating the time and frequency locations of SBFD sub-bands to UEs, defining transmission, reception, and measurement procedures, and supporting random access in SBFD symbols. Cross-link interference (CLI) handling will also be addressed to mitigate interference issues in dynamic TDD scenarios with considerations for both gNB-to-gNB and UE-to-UE co-channel CLI handling schemes.

In Release 19, there will be a continued focus on studying and advancing 5G topology by leveraging various technologies such as 5G femtocells (wireless access point used to enhance indoor cellular connectivity), wireless access backhauls (WAB), multi-hop sidelink relay, and NTNs.

UE energy efficiency is also a priority, with emphasis on improving power-saving mechanisms like Discontinuous Reception (DRX). To address power consumption during idle periods, 3GPP is developing Low-Power Wake-Up Receiver (LP-WUR) technology, enabling devices to wake up only when necessary. LP-WUR will trigger the main radio for tasks like paging and monitoring control channels.

Furthermore, 3GPP is exploring solutions for ultra-low-power IoT devices, such as those operating without batteries or with minimal energy storage. This includes studying waveform characteristics and developing lightweight protocols to support ambient IoT scenarios like asset identification.

This new version will continue to embrace AI/ML to make 5G-Advanced more intelligent. It will introduce support for AI/ML models with a general framework, focusing on one-sided models applicable either at the UE or network level. This entails specifying signalling and protocol aspects related to life cycle management (LCM). Evaluation findings have underscored the advantages of AI/ML-based beam management and positioning. As such, 3GPP will define the necessary signalling and mechanisms to enable AI/ML-based beam management and positioning for both UE-sided and network-sided models.

Furthermore, the Release 19 work item on AI/ML for NR air interface will address outstanding issues identified during the Release-18 study through various study objectives like the examination of CSI compression to tackle the hurdles of inter-vendor training collaboration and enhance the balance between performance and complexity/overhead.

In summary, Release 19 will play a crucial role in bridging the gap between 5G-Advanced and 6G technologies. This will involve the development of channel models that integrate sensing and communication, particularly focusing on the 7-24 GHz spectrum range, which is deemed essential for 6G advancement.

4. SIMULATION SOFTWARE

In this chapter, the methodology used to develop the adopted simulation scenario will be analysed. A detailed description of the various existing connectivity software will be conducted, specifying which is the best for the case I simulated, namely the emergency vehicles preemption. The V-model was adopted for the design of the entire simulation.

The V-Model, which stands for verification and validation, divides the development process into two main sections. The left side of the V encompasses requirement analysis, functional design, and software development. In contrast, the right side focuses on verification and validation activities, culminating in the release.

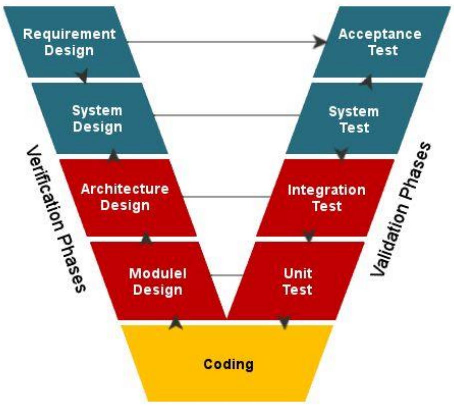

Figure 10 - V-model representation

As an extension of the waterfall methodology, the V-Model places a strong emphasis on testing, particularly highlighting the importance of planning tests early in the development cycle. Therefore, making a careful selection of the software to be used is essential. In the automotive field, one of the most important traffic simulation software is Simulation of Urban Mobility (SUMO).

SUMO, also known as Eclipse SUMO, is a versatile, open-source traffic simulation package [S8] designed for managing extensive networks. Initially developed by the German Aerospace Center and adopted as an Eclipse Foundation project in 2017, SUMO enables detailed, continuous simulation of multi-modal traffic scenarios.

This software facilitates the simulation and analysis of road traffic and traffic management systems, offering tools to implement and evaluate new traffic strategies in a controlled environment before deployment. SUMO plays a critical role in the development and validation of automated driving functions, supporting various X-in-the-Loop and digital twin approaches.

Widely utilized in research, SUMO aids in traffic forecasting, traffic light evaluation, route selection, and vehicular communication systems. Its open-source nature allows for modification of the source code, encouraging experimentation and innovation.

Additionally, SUMO integrates seamlessly with other connectivity software for analyzing Vehicle-to-Everything (V2X) communication. This integration enhances simulation realism by incorporating diverse communication scenarios, thereby improving the relevance of research outcomes.

4.1 OVERVIEW OF CONNECTIVITY SIMULATION SOFTWARE

This chapter explores various software tools used for simulating and evaluating vehicular communication scenarios. These tools are critical for advancing communication protocols and network architectures, which are essential for the deployment of connected and autonomous vehicles. This chapter will be particularly useful in identifying the most appropriate software for the specific objective of this project.

Network simulation software

Network simulation software plays a pivotal role in modelling and testing vehicular communication systems. Key tools in this category include ns-2 and ns-3.

ns-2 and ns-3 are both open-source network simulators extensively used for research and development in communication networks. While they aim to model and analyse network behaviour, they possess distinct characteristics that suit different simulation scenarios. ns-2, established in 1996, uses the OTcl scripting language and is known for its large user community, extensive resources, and support for a broad range of network protocols and technologies. However, it has a steeper learning curve, lower performance for large-scale simulations, and lacks support for emerging technologies like 5G and SDN. On the other hand, ns-3 [S9], which began in 2006, uses C++ with an optional Python interface. It boasts a modern and modular architecture, enhanced performance for large-scale simulations, and support for recent technologies like 5G and SDN. Despite these strengths, ns-3 has a smaller user community, limited resources and documentation, and is less stable than ns-2. If you need a mature, stable simulator with ample documentation and support, and you are not familiar with C++, ns-2 is a good choice. If you need to simulate large-scale networks with modern technologies and are proficient in C++, ns-3 is preferable. Note that ns-3 is not backward compatible with ns-2, and other network simulators like OPNET, OMNeT++, and QualNet are available alternatives. Carefully evaluate your needs and research different simulators before deciding.

V2X-Sim

V2X-Sim, an acronym for Vehicle-to-Everything Simulation, is a synthetic dataset for collaborative perception in autonomous driving developed by AI4CE Lab at NYU and MediaBrain Group at SJTU [19]. It is a valuable resource for researchers working in this field, as it provides realistic simulation scenarios, multimodal sensor data, and accurate ground truth.

V2X-Sim stands out for several features [19]:

- Realistic scenarios: The dataset simulates complex autonomous driving scenarios with different types of vehicles, pedestrians, and cyclists, replicating real driving situations.
- Multimodal data: V2X-Sim includes data from multiple sensors, including LiDAR, cameras, and radar, allowing for the analysis of diverse and complementary sensory information.
- Accurate ground truth: For each object present in the scenario, V2X-Sim provides precise and detailed information, facilitating the evaluation of perception algorithm performance.
- Scalability: The simulation platform on which V2X-Sim is based allows for the simulation of large-scale scenarios with numerous agents, making it suitable for testing algorithms on a large scale.

The use of V2X-Sim has led to the development of new algorithms for object detection, tracking, and segmentation in autonomous driving. Additionally, the dataset has been employed to evaluate the performance of various collaborative perception systems.

V2X-Sim represents a benchmark for research on collaborative perception in autonomous driving, offering researchers a versatile and reliable tool for developing and testing new algorithms and systems. The use of this dataset will continue to promote the advancement of the field and contribute to the realization of safer and more reliable autonomous driving.

In theory, the V2X-Sim software could be adapted to simulate the preemption of the emergency vehicles, but currently, there are no specific functionalities dedicated to this scenario.

ms-van3t

ms-van3t is a sophisticated simulation tool that leverages the ns-3 and SUMO simulators to manage both vehicle mobility and connectivity efficiently [20]. One of the standout features is its ability to integrate multiple open-source models for V2X communications, allowing simulated connected vehicles to flexibly utilize and switch between these models based on the scenario requirements. This flexibility is crucial for testing various communication protocols under different conditions.

It includes a complete ETSI C-ITS stack, enabling the simulation of Cooperative Awareness Messages (CAM), event-triggered Decentralized Environmental Notification Messages (DENM), and Infrastructure to Vehicle Information Messages (IVIMs). This comprehensive support for ETSI C-ITS standards ensures that researchers can accurately model and evaluate cooperative vehicular communication scenarios.

Another significant capability of ms-van3t is its support for Hardware-in-the-Loop (HIL) testing. This feature allows the simulation environment to emulate the presence of vehicles and facilitates communication with real vehicles through physical interfaces. HIL testing is essential for validating the interaction between simulated and real-world components, enhancing the realism and applicability of the simulation results.

Ms-van3t also supports the use of real GNSS traces, which adds a layer of realism by incorporating actual vehicle movement data. Additionally, it offers easy visualization of mobile nodes on real maps, providing a clear and intuitive understanding of vehicle dynamics and communication interactions within the simulated environment.

The tool's capability for efficient comparison of V2X technologies makes it an asset for researchers and developers.

VSimRTI

VSimRTI is a software framework designed to facilitate the assessment of novel C-ITS solutions. Its core strength lies in its ability to couple various simulators, including traffic simulators like SUMO and VISSIM, with network simulators and application simulators [21]. This integrated approach allows for the comprehensive simulation of C-ITS functionalities, encompassing not only vehicle movement but also communication networks and the behavior of C-ITS applications themselves.

VSimRTI leverages an innovative "ambassador" concept inspired by the High-Level Architecture (HLA). Ambassadors act as intermediaries, enabling communication between different simulators even if they adhere to different communication protocols. This design principle fosters flexibility and scalability, allowing for the seamless integration of new simulators into the VSimRTI framework.

VSimRTI offers a multitude of advantages for researchers in the C-ITS domain [21]. Here are some key aspects:

- *Scalability:* VSimRTI can handle large-scale simulations, enabling researchers to evaluate C-ITS performance in complex scenarios with numerous vehicles and infrastructure elements.
- *Diversity of Applications:* VSimRTI supports the simulation of a wide range of C-ITS applications, such as cooperative collision avoidance, lane changing assistance, and traffic signal optimization.
- *Realistic Traffic Representation:* By coupling with established traffic simulators, VSimRTI allows for the creation of realistic traffic environments, ensuring the generalizability of simulation results.
- *Flexibility:* The ambassador concept facilitates the integration of new simulators, catering to the specific needs of a research project.

VSimRTI can be used to simulate the right of way for emergency vehicles. Its ability to couple different simulators makes it suitable for recreating realistic scenarios involving emergency vehicles, road infrastructure, and other road users. However, since my priorities are ease of use, computational efficiency, and a focus on V2I communication, this software might not be the most suitable solution.

SimuLTE

SimuLTE [S10] stands out as an innovative and versatile simulation software for Long Term Evolution (LTE) networks, providing researchers and engineers with a comprehensive environment to evaluate the performance and behavior of these complex networks. Its modular and scalable architecture enables the simulation of large-scale realistic scenarios, incorporating numerous LTE nodes, user devices, and base stations. SimuLTE distinguishes itself through several key features that make it a valuable tool for LTE research and development:

- *Accuracy and Realism:* SimuLTE employs realistic physical models to simulate signal propagation, interference, and other physical phenomena that influence LTE network performance. This ensures reliable and accurate simulation results that reflect real-world behavior.
- *Scalability and Performance:* SimuLTE's modular structure allows for simulating large-scale scenarios with numerous LTE nodes, user devices, and base stations. Its optimization for parallel execution on high-performance computing clusters makes it suitable for complex and computationally demanding simulations.
- *Flexibility and Customization*: SimuLTE offers a wide range of configuration and customization options, enabling users to tailor simulations to their specific needs. Network parameters, traffic models, user device behaviors, and other aspects can be modified to replicate realistic scenarios and evaluate different network configurations.
- *Integration with Other Tools: SimuLTE* can be integrated with other simulation and analysis tools, allowing users to combine its capabilities with other models and methodologies to gain a more comprehensive view of network performance.

In conclusion, SimuLTE establishes itself as a powerful tool for LTE network simulation, providing researchers and engineers with a comprehensive environment to evaluate performance, analyse complex scenarios, and develop new solutions for next-generation LTE networks. Its accuracy, scalability, flexibility, and integration capabilities make it a valuable tool for LTE research and development, contributing to the advancement of these high-performance mobile networks.

SimuLTE isn't ideal for directly simulating V2I communication for emergency vehicle preemption with traffic light control, it can be part of a larger framework with other simulators to achieve your goal.

Veins

Veins stands out as a powerful and versatile open-source software framework for simulating vehicular networks [S11], providing researchers and engineers with a comprehensive environment to model, analyse, and evaluate the performance of communication protocols and applications in these complex networks. Its modular architecture, extensive functionality, and active community make it a valuable tool for a wide range of vehicular network research endeavours.

Veins distinguishes itself through several key features [S11] that contribute to its widespread adoption:

- *Comprehensive Protocol Modelling:* Veins offers a rich set of modules for modelling various communication protocols commonly used in vehicular networks, including IEEE 802.11p (Wi-Fi), LTE-A (Long Term Evolution-Advanced), and C-V2X (Cellular Vehicle-to-Everything). This allows researchers to investigate the behaviour of these protocols under different network conditions and traffic scenarios.
- *Application Development and Testing:* Veins provides a robust platform for developing and testing vehicular network applications. Its event-driven architecture and flexible programming interfaces enable researchers to create realistic application scenarios and evaluate their performance in terms of latency, throughput, and reliability.
- *Realistic Traffic Simulation:* Veins integrates with traffic simulators like SUMO (Simulation of Urban Mobility) and OMNeT++ to accurately model the dynamics of vehicular traffic. This allows researchers to study the impact of traffic patterns and congestion on the performance of communication protocols and applications.
- *Scalability and Performance:* Veins' modular design and support for parallel processing make it scalable to large-scale simulations involving numerous vehicles, infrastructure elements, and communication protocols. This enables researchers to model realistic and complex vehicular network scenarios.

In conclusion, Veins establishes itself as an indispensable tool for vehicular network research, offering researchers and engineers a comprehensive environment to model, analyze, and evaluate the performance of communication protocols and applications. Its versatility, scalability, and active community make it an asset for advancing the development and deployment of efficient, reliable, and secure vehicular networks.

Veins is a powerful tool for simulating V2I communication and emergency vehicle preemption.

Artery

Artery is a simulation framework designed for modelling and simulating ETSI ITS-G5 protocols, serves as a comprehensive platform for integrating multiple ITS-G5 services within individual vehicles [S12]. Its middleware provides essential facilities for these services, including Cooperative Awareness (CAMs) and Decentralized Notification (DENMs). Originally developed as an extension of Veins, Artery initially concentrated on WAVE, the U.S. counterpart of ITS-G5, when development commenced in 2014. Its fundamental purpose lies in facilitating the analysis of next-generation mobile communication technologies, particularly their impact on vehicular connectivity. By offering a means to evaluate and optimize communication protocols and network architectures, Artery plays a crucial role in addressing the evolving demands of connected vehicles and their diverse use cases.

Artery doesn't natively handle traffic light control; its strength lies in large-scale communication network simulation. By combining it with a separate traffic light simulator like SUMO, you can achieve your desired V2I communication and preemption scenario. However, this approach might require more complex integration compared to using Veins, which can directly integrate with SUMO for traffic light control.

But why did I choose Veins for my thesis project?

Veins is suitable for simulating a V2I communication scenario involving emergency vehicle preemption with traffic light control. In fact, it's one of its core strengths.

Veins offers modules for simulating V2I communication protocols like IEEE 802.11p (used for DSRC), allowing you to define messages carrying information about the emergency vehicle's location, speed, and direction.

Veins integrates with traffic simulators like SUMO, enabling you to model traffic lights and their control logic. A custom SUMO script can be developed to listen for incoming V2I messages from emergency vehicles and implement the preemption logic. Upon receiving a V2I message, the script can dynamically modify the traffic light timing to prioritize the emergency vehicle, such as extending green phases or shortening red phases.

Benefits of using Veins include its open-source and customizable nature, a large and active community for support, and seamless integration with other simulators like SUMO.

4.2 PREEMPTION OF EMERGENCY VEHICLES

As specified at the beginning of the chapter, this thesis will analyse a particular case of vehicular communication: emergency vehicles preemption. But why is it important to study this scenario?

This simulation is a crucial aspect of contemporary automotive research, playing a fundamental role in road safety, operational efficiency, and the integration of emerging technologies. The ability to predict and model the behaviour of various types of vehicles in emergency situations not only reduces the risk of accidents but also significantly improves the response times of emergency services.

Recent data from 2014 to 2019 underscores the need to enhance ambulance safety across different countries [S13]:

- Germany: 597 ambulance accidents, 1,170 injuries, 31 fatalities.
- Austria: 62 ambulance accidents, 115 injuries, 7 fatalities.
- Switzerland: 25 ambulance accidents, 18 injuries.

In London, the number of ambulance accidents has been steadily increasing, with 2,265 accidents recorded in 2017 alone [S14].

The situation is equally concerning for other types of emergency vehicles. According to the National Institute of Health (NIH), from 2001 to 2010, traffic accidents were the second leading cause of on-duty fatalities for firefighters, with nearly 30,000 fire truck-related accidents per year [S15]. The United States Fire Administration reported almost 100 firefighter deaths annually, with 20-25% of these deaths attributable to road accidents [S15]. For police officers, transportation-related incidents are the leading cause of death, with nearly 36% of on-duty fatalities resulting from traffic accidents and an additional 10% due to being struck by a vehicle. Between 2004 and 2013, the average number of police officer deaths due to traffic accidents was nearly 68 per year [S16].

These statistics underscore the necessity of developing and testing systems that can enhance the safety and operational efficiency of ambulances. This study aims to evaluate the benefits of simulation, with a particular focus on the interaction between emergency vehicles, road-side unit, and traffic lights and to explore the potential offered by the integration of advanced driver assistance systems and intelligent infrastructure.

Three software tools were employed in the development of this project: SUMO for traffic simulation, and OMNeT++ and Veins for connectivity.

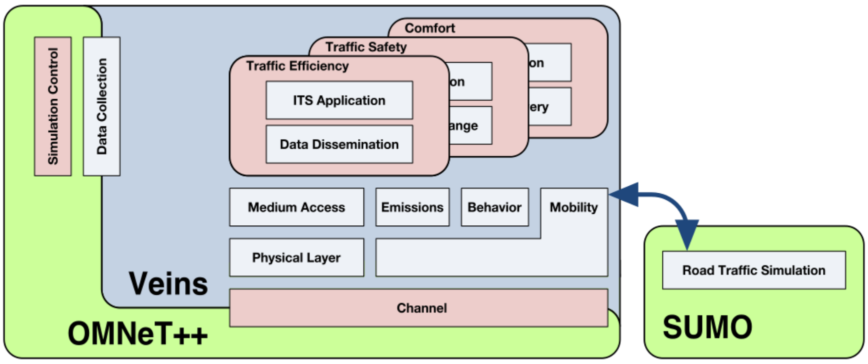

Figure 11 - Modular structure of Veins

Veins is the ideal choice because it is specifically designed for vehicular connectivity. Unlike other tools which are extensions to evaluate connectivity in specific scenarios, Veins tightly integrates the SUMO traffic simulator with V2X communications, allowing for realistic and detailed simulation of traffic and communication between vehicles and infrastructure. Additionally, Veins offers modular flexibility, an active support community, and a powerful ecosystem of tools, making it a robust and widely adopted solution in academic and research settings.

4.2.1 SUMO

The scenario was created using the Netedit tool of SUMO and represents a 4-way intersection controlled by traffic lights as we can see in *Figure 12*.

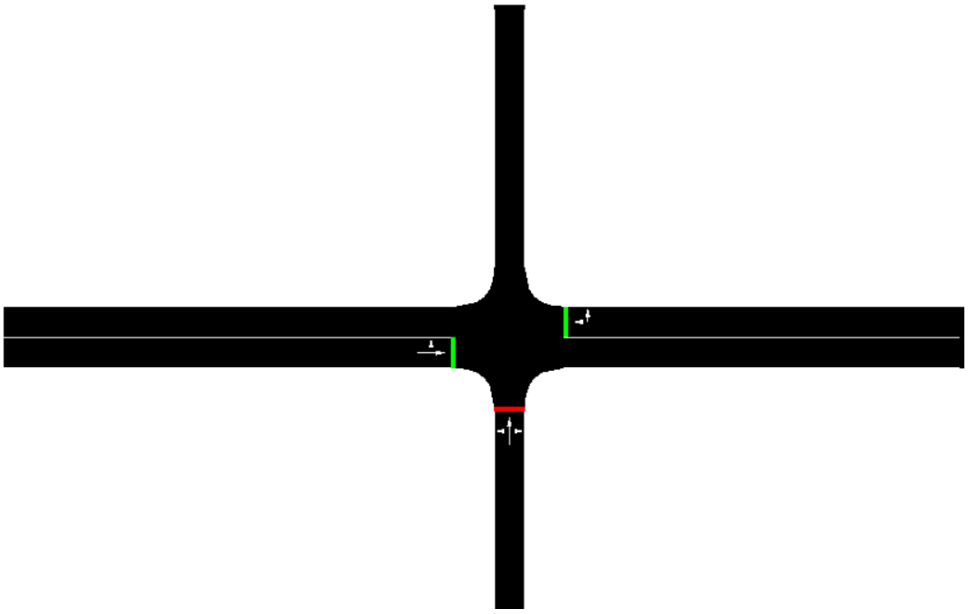

Figure 12 - SUMO scenario

In this case, the main files enabling the simulation in SUMO are three: EVP.net, EVP.rou, and EVP.sumocfg.

EVP.net is automatically generated and describes the traffic-related aspects of a map, including roads and intersections through which simulated vehicles move. At a high level, a SUMO network is a directed graph where nodes, commonly referred to as "junctions" in SUMO, represent intersections and "edges" represent streets or roads.

Specifically, EVP.net contains detailed information about each road, including its lanes with position, shape, and speed limit. It also includes the traffic light logic governing intersections, precise definitions of intersections with their priority rules, and detailed lane connections at intersections using nodes to facilitate vehicle movement along the road network.

In *Figure 13* It is possible to see the code that governs the normal operation of the traffic light.

```
<tlLogic id="2" type="static" programID="0" offset="0">
    <phase duration="28" state="GGrrrGG"/>
    <phase duration="6"  state="yyrrryy"/>
    <phase duration="28" state="rrGGGrr"/>
    <phase duration="6"  state="rryyyrr"/>

</tlLogic>
```

Figure 13 - Basic Traffic Light Program in the file EVP.net

The logic is defined through a series of phases that dictate the behaviour of traffic lights at an intersection. The `<tlLogic>` tag specifies a static traffic light program with a unique identifier and a zero offset, indicating that the cycle starts immediately without delay. Each phase within this logic includes a duration and a state, where the state string represents the traffic light status (green, yellow, red) for each lane. The cycle typically involves two main phases of 21 seconds each, where different sets of lanes are green, allowing specific traffic flows. These are interspersed with 6-second yellow transition phases, ensuring that vehicles have time to prepare to stop before the next set of lanes turns green. This structured alternation of traffic light states ensures an organized and efficient flow of vehicles through the intersection, minimizing the risk of collisions and optimizing traffic movement.

The file EVP.rou in *Figure 14* defines the vehicle types, routes, and individual vehicles within the simulation.

The file specifies two types of vehicles: "passenger" with a maximum acceleration of 2.5 m/s^2, deceleration of 4.5 m/s^2, length of 5 meters, maximum speed of 20 m/s, and yellow colour; and "emergency" with an acceleration of 2.6 m/s^2, deceleration of 4.5 m/s^2, length of 6.5 meters, maximum speed of 20 m/s, red colour, and a special shape for emergency vehicles.

Seven routes are defined, each identified by a unique ID and a sequence of edges, such as "route0" with edges "2to5" and "2to1". Vehicles are specified with a unique ID, vehicle type, departure time, and the route they will follow, such as "car1", which is a "passenger" vehicle departing at 10.00 seconds and following "route1". This structure allows for a detailed and accurate traffic simulation, organizing vehicle movements through the road network.

```
<routes>
    <vType id="passenger" accel="2.5" decel="4.5" sigma="0.1"
length="5" minGap="2.5" maxSpeed="20.0" color="1,1,0"/>
      <vType id="emergency" accel="2.6" decel="4.5" sigma="0"
length="6.5" minGap="2.5" maxSpeed="20.0" color="1,0,0"
guiShape="emergency"/>


    <route id="route0" edges="2to5 2to1"/>
      <route id="route1" edges="1to2 2to3"/>
      <route id="route2" edges="2to5 4to2"/>
      <route id="route3" edges="3to2 4to2"/>
      <route id="route4" edges="3to2 2to1"/>
      <route id="route5" edges="2to5 2to3"/>
      <route id="route6" edges="1to2 4to2"/>

    <vehicle id="car1" type="passenger" depart="10.00"
route="route1"/>
    <vehicle id="emergency1" type="emergency" route="route0"
depart="15.00"/>
      <vehicle id="car4" type="passenger" depart="25.00"
route="route6"/>
      <vehicle id="car3" type="passenger" depart="30.00"
route="route2"/>
      <vehicle id="emergency2" type="emergency" route="route3"
depart="35.00"/>
      <vehicle id="car2" type="passenger" depart="40.00"
route="route1"/>
    <vehicle id="emergency3" type="emergency" route="route6"
depart="48.00"/>
      <vehicle id="car5" type="passenger" depart="48.00"
route="route3"/>
</routes>
```

Figure 14 - File EVP.rou

The last file ‘**EVP.sumocfg**’ in *Figure 15* defines a configuration file which contains all the parameters needed to start the simulation.

```
<configuration>
 <input>
        <net-file value="EVP.net.xml"/>
        <route-files value="EVP.rou.xml"/>
    </input>
    <time>
        <begin value="0"/>
        <end value="100"/>
    </time>
</configuration>
```

Figure 15 - File EVP.sumocgf

Once the scenario was defined on SUMO, I moved to OMNeT++ to configure the necessary files for simulating the preemption of emergency vehicles.

4.2.2 OMNeT++ AND Veins

OMNeT++ (Objective Modular Network Testbed in C++) is a modular, component-based C++ simulation library and framework utilized by Veins for constructing network simulators [S17].

An OMNeT++ model is composed of several key components:

- *NED Language Topology Descriptions* (.ned files): These files define the module structure, including parameters, gates, and other elements.
- *Message Definitions* (.msg files): These files allow for the definition of message types and the addition of data fields to them. OMNeT++ translates these message definitions into fully functional C++ classes.
- *Simple Module Sources*: These are C++ source files with .h and .cc extensions. They implement the behaviour and functionality of the simple modules defined in the NED files.

The initial step in my OMNeT++ workflow was to create a directory named '**EmergencyVehiclePreemption**' and upload the files generated in SUMO. This preliminary action facilitated the testing of the simulation using the example files within Veins, thereby verifying the successful installation and integration of the simulation environment.

I began to conceptualize the structure of the simulation and define my objectives. The primary goal was to implement Vehicle-to-Infrastructure (V2I) communication between the emergency vehicle and a Roadside Unit (RSU) situated at the edge of the road. The RSU, upon detecting the presence of the emergency vehicle, would communicate with the traffic signal to adjust its timing program appropriately. This adjustment would also consider the presence of "passenger" vehicles in the other lanes to ensure an optimal flow of traffic.

Following the outlined structure, I created the '**EVPNetwork.ned**' file to define the topology of my simulation. This file specifies the structure of the modules, including the morphology of the RSU.

```
import org.car2x.veins.nodes.RSU;
import org.car2x.veins.nodes.Scenario;

network EVPNetwork extends Scenario {
    submodules:
        rsu[1]: RSU {
            @display("p=150,140;i=veins/sign/yellowdiamond;is=vs");

        }
        connections allowunconnected:
}
```

Figure 16 - Network file definition

This file creates a new network by extending the file '**Scenario.ned**' from Veins and adding an RSU module. The Roadside module facilitates communication with vehicles within the simulation area. The new network inherits all parameters and modules defined in the Scenario, enhancing the simulation with the added functionality of the RSU.

I utilize the '**TraCIDemo11pMessage.ms**' file as a message type for facilitating DSRC within the simulation of vehicular networks. This message type is specifically designed to transmit structured data between network nodes, incorporating essential details such as `*demoData*`, `*senderAddress*`, and a serial number (`*serial*`). Importantly, it extends the `*BaseFrame1609_4*` file, aligning with the standards and protocols of DSRC. Its implementation supports the development and testing of communication algorithms, routing protocols, and traffic management strategies within the simulated environment, thereby enabling realistic simulation and comprehensive analysis of ad hoc vehicular networks (VANETs).

The communication process was divided into two distinct phases. The first phase involves the interaction between the emergency vehicle and the RSU, while the second phase encompasses the communication between the RSU and the traffic lights. Consequently, it was necessary to create four separate files to manage these message exchanges effectively.

The first phase is realized through two files which are called '**EmergencyVehicle.h**' and '**EmergencyVehicle.cc**'.

The first file defines the header of a class *EmergencyVehicle* that extends the functionalities of the base class *DemoBaseApplLayer* to manage an emergency vehicle in a Veins simulation.

```
#ifndef EMERGENCYVEHICLE_H
#define EMERGENCYVEHICLE_H

#include
"veins/modules/application/ieee80211p/DemoBaseApplLayer.h"
namespace veins {
class EmergencyVehicle : public veins::DemoBaseApplLayer {
protected:
    virtual void initialize(int stage) override;
    virtual void handleSelfMsg(cMessage* msg) override;
    virtual void handlePositionUpdate(cObject* obj) override;

private:
    void sendEmergencyMessage();
    bool isEmergencyVehicle = false; // to identify if the
vehicle is an emergency one
};}

#endif // EMERGENCYVEHICLE_H
```

Figure 17 - File EmergencyVehicle.h

The following section declares three methods that override the virtual functions in the base class:

- initialize(int stage): Method called during the module initialization. stage indicates the initialization phase.
- handleSelfMsg(cMessage* msg): Method called to handle internal messages (self-messages).
- handlePositionUpdate(cObject* obj): Method called when there is a position update for the vehicle.

```
protected:
    virtual void initialize(int stage) override;
    virtual void handleSelfMsg(cMessage* msg) override;
    virtual void handlePositionUpdate(cObject* obj) override;
```

This second section declares two private members:

- void SendEmergencyMessage(): method to send an emergency message.
- bool isEmergencyVehicle=false: boolean variable to identify if the vehicle is an emergency one

```
private:
    void sendEmergencyMessage();
    bool isEmergencyVehicle = false;
```

The second file '**EmergencyVehicle.cc**' contains the implementation of the code, detailing how the class works. It will be described step by step.

This part includes necessary headers for the class, message handling, and mobility. It uses the veins namespace to simplify code references.

```
#include "EmergencyVehicle.h"
#include "veins/modules/application/traci/TraCIDemo11pMessage_m.h"
#include "veins/modules/mobility/traci/TraCIMobility.h"

using namespace veins;
```

The *initialize* method is invoked at the beginning of the simulation. During initialization, it checks if the simulation is in stage 0. If so, it accesses the mobility module to retrieve the vehicle's ID.

If the vehicle ID matches one of the predefined emergency vehicle IDs, it flags *isEmergencyVehicle* as true and schedules an emergency message to be dispatched after a 15-second delay.

If the mobility module is inaccessible, an error message is logged to indicate the issue.

This setup ensures that emergency vehicles are correctly identified and prepared to send timely emergency notifications within the simulation

framework.

```
void EmergencyVehicle::initialize(int stage) {
    DemoBaseApplLayer::initialize(stage);

    if (stage == 0) {
        veins::TraCIMobility* mobility =
TraCIMobilityAccess().get(getParentModule());


        if (mobility) {
            std::string vehicleId = mobility->getExternalId();
            if (vehicleId == "emergency1" || vehicleId ==
"emergency2" || vehicleId == "emergency3") { // Check if the incoming
ID is one of the emergency vehicle
                isEmergencyVehicle = true;
                // Auto-trigger to send an emergency message
                scheduleAt(simTime() + 15, new
cMessage("sendEmergencyMessage"));
            }
        } else {
            EV << "Error: TraCIMobility not available." << endl;
    }
}
}
```

The *handleSelfMsg* method manages internal messages (self-messages). It checks the name of the incoming message using `*strcmp*`.

If the message name matches "*sendEmergencyMessage*" and the vehicle is identified as an emergency vehicle (isEmergencyVehicle is true), it proceeds to invoke the *sendEmergencyMessage* method to dispatch an emergency notification.

Subsequently, it schedules the next self-message to trigger again after 15 seconds, ensuring periodic transmission of emergency alerts.

If the vehicle is not an emergency vehicle, the method simply deletes the incoming message to avoid unnecessary processing.

For any other type of self-message, the method delegates the handling to the base class `*DemoBaseApplLayer*` for appropriate processing, maintaining the integrity of message handling across different simulation scenarios.

This approach ensures efficient and timely communication of emergency situations within the simulated environment, adhering to specified protocols and operational requirements.

```
void EmergencyVehicle::handleSelfMsg(cMessage* msg) {
    if (strcmp(msg->getName(), "sendEmergencyMessage") == 0) {
        if (isEmergencyVehicle) {
            sendEmergencyMessage();
            // organize the next message
            scheduleAt(simTime() + 15, msg);
        } else {
            delete msg; //delete the message if it isn't an emergency
vehicle
        }
    } else {
        DemoBaseApplLayer::handleSelfMsg(msg);
    }
}
void EmergencyVehicle::handlePositionUpdate(cObject* obj) {
    DemoBaseApplLayer::handlePositionUpdate(obj);

    if (isEmergencyVehicle) {
        sendEmergencyMessage();
    }
}
```

The *handlePositionUpdate* method manages updates to the vehicle's position within the simulation. It extends the functionality provided by the base class `*DemoBaseApplLayer*` to incorporate specific actions for emergency vehicles. Upon receiving a position update notification, the method first delegates to the base class to handle any generic position-related tasks. Subsequently, it checks if the vehicle is designated as an emergency vehicle (*isEmergencyVehicle* flag is true). If confirmed, the method triggers the “*sendEmergencyMessage*” function to initiate the transmission of an emergency notification. This streamlined approach ensures that emergency vehicles promptly broadcast their status and location updates to relevant entities within the simulation environment. By leveraging this method, the simulation accurately reflects real-world scenarios where timely communication of emergency situations is critical for effective response and coordination among vehicles and infrastructure.

```
void EmergencyVehicle::handlePositionUpdate(cObject* obj) {
    DemoBaseApplLayer::handlePositionUpdate(obj);

    if (isEmergencyVehicle) {
        sendEmergencyMessage();
    }
}
```

The *sendEmergencyMessage* method facilitates the transmission of emergency notifications in the Veins simulation environment. When invoked, the method first creates a new instance of TraCIDemo11pMessage.

It sets the message content to indicate an emergency situation using '*setDemoData("EmergencyVehicle")*'. Next, the method accesses the vehicle's ID from the mobility module through '*TraCIMobilityAccess().get(getParentModule())*', ensuring the message includes accurate identification of the sending vehicle. Subsequently, the vehicle ID is assigned to the message using '*setVehicleId(vehicleId.c_str())*', preparing it for transmission.

Finally, the method calls '*sendDown(wsm)*' to dispatch the prepared message downwards through the communication stack.

Upon successful transmission of the emergency message, a log message is generated to notify the user. This log message confirms that the emergency message has been successfully sent, providing visibility and acknowledgment within the simulation environment.

This process enables emergency vehicles to effectively broadcast their status and critical information, such as location and emergency nature, within the simulation. It ensures that emergency responses and coordination mechanisms can be realistically simulated, enhancing the fidelity and effectiveness of emergency management scenarios in the simulated environment.

```
void EmergencyVehicle::sendEmergencyMessage() {
    TraCIDemo11pMessage* wsm = new TraCIDemo11pMessage();
    wsm->setDemoData("EmergencyVehicle");

    // obtain the vehicle ID from the mobility module
    veins::TraCIMobility* mobility =
TraCIMobilityAccess().get(getParentModule());
    std::string vehicleId = mobility->getExternalId();
    wsm->setVehicleId(vehicleId.c_str()); //set the vehicle ID in the
message

    sendDown(wsm);
    EV << "Emergency message sent by vehicle: " << vehicleId << endl;
}
```

The second phase of the project involves managing traffic light control through communication between the RSU and the traffic lights. Before addressing the management of the traffic light program, I analysed the configuration of the road network. The EVP.net file defines the traffic lights based on the junctions. For example, at junction ID=2, there are three traffic lights managing that specific intersection. Since it is not possible to directly control each individual traffic light (as they do not have a specific ID), I decided to work with lanes and intersection priorities.

I evaluated the network file containing the predefined traffic light program and introduced two emergency programs: one for the case when the emergency vehicle is on the main road and one for when it is on the secondary road. In both cases, I also outlined the duration of the traffic light phases. In *Figure 18* it is possible to see the two emergency traffic light programs that have been added as <*additional*> in the EVP.net file.

```
<additional>
   <!-- Emergency Program for main lane -->
    <tlLogic id="2" type="static" programID="2_emergency_main"
offset="0">
        <phase duration="40" state="GGrrrGG"/>
        <phase duration="6"  state="yyrrryy"/>
        <phase duration="28"  state="rrGGGrr"/>
        <phase duration="6"  state="rryyyrr"/>

    </tlLogic>

    <!-- Emergency Program for side lane -->
    <tlLogic id="2" type="static" programID="2_emergency_side"
offset="0">
        <phase duration="29" state="rrGGGrr"/>
        <phase duration="6"  state="ggyyygg"/>
        <phase duration="28" state="yyrrryy"/>
        <phase duration="6"  state="rrGGGrr"/>

    </tlLogic>
</additional>
```

Figure 18 - Traffic lights emergency programs

After introducing the two emergency programs, I worked on the code needed to activate them. To this end, I created the files '**EmergencyRSU.h**' and '**EmergencyRSU.cc**'.

The `` `**EmergencyRSU.h**` `` file in *Figure 19* defines the `` `*EmergencyRSU*` `` class, which inherits from `` `*DemoBaseApplLayer*` `` within the veins namespace. This class is designed to manage traffic light programs in response to emergency scenarios. It includes several essential header files for functionality related to IEEE 802.11p communication and TraCI (Traffic Control Interface). The class provides methods for initialization, handling Wave Short Messages (WSM), changing traffic light programs based on lane IDs, and managing incoming messages. Additionally, it declares private member variables for interfacing with the TraCI simulation manager and command interface, facilitating communication and control within the traffic simulation environment. This setup enables the `*EmergencyRSU*` to dynamically alter traffic light sequences to prioritize emergency vehicles, enhancing the response efficiency in critical situations.

```
#ifndef EMERGENCYRSU_H
#define EMERGENCYRSU_H

#include
"veins/modules/application/ieee80211p/DemoBaseApplLayer.h"
#include
"veins/modules/application/traci/TraCIDemo11pMessage_m.h"
#include
"veins/modules/mobility/traci/TraCICommandInterface.h"
#include "veins/modules/mobility/traci/TraCIScenarioManager.h"

namespace veins {

class EmergencyRSU : public DemoBaseApplLayer {
public:
    virtual void initialize(int stage) override;

protected:
    virtual void onWSM(BaseFrame1609_4* wsm) override;
    void changeTrafficLightProgram(const std::string& laneId);
    virtual void handleMessage(cMessage *msg) override;

private:
    void initializeAfterDelay();

    veins::TraCIScenarioManager* manager;
    veins::TraCICommandInterface* traci;

};

}

#endif // EMERGENCYRSU_H
```
Figure 19 - EmergencyRSU.h file

The next step was to define the '**EmergencyRSU.cc**' file, which enables the actual traffic management. This file will be analysed step by step.

The file begins by including the necessary headers, such as '**EmergencyRSU.h**', to access the EmergencyRSU class definition. It utilizes the veins namespace to organize and avoid naming conflicts within the codebase. Additionally, the Define_Module(EmergencyRSU) statement registers the EmergencyRSU class with the OMNeT++ simulation framework, allowing it to be instantiated and interact with the simulation environment.

```
#include "EmergencyRSU.h"

using namespace veins;

Define Module(EmergencyRSU);
```

The *initialize* function serves as a pivotal method for initializing the `*EmergencyRSU*` class during different stages of the simulation lifecycle. Inherited from the `*DemoBaseApplLayer*` base class, this function is overridden to accommodate specific initialization routines.

During the *stage 0*, it performs standard initialization tasks typical for the application layer.

In *stage 1*, it delays further initialization until a simulated time of 2 seconds has elapsed, ensuring synchronization with other simulation processes. This is achieved by scheduling a self-message named *InitializeAfterDelay* using `*scheduleAt(simTime() + 2.0, new cMessage("InitializeAfterDelay"))*`. Once the delay is met, the function invokes `*initializeAfterDelay()*` to initialize crucial components required for traffic management.

This staged approach ensures that the `*EmergencyRSU*` class is ready to handle emergency scenarios effectively within the simulated traffic environment.

```
void EmergencyRSU::initialize(int stage) {
    DemoBaseApplLayer::initialize(stage);

    if (stage == 0) {
        // base initialization for stage 0
    } else if (stage == 1) {
        // wait for a simulated time
        if (simTime() < 2.0) {  // move the initialization
scheduleAt(simTime() + 2.0, new
cMessage("InitializeAfterDelay"));
        } else {
            initializeAfterDelay();
        }
    }
}
```

The *handleMessage* function plays a critical role in managing incoming messages within the `*EmergencyRSU*` class.

Overriding the base class method, it first checks if the received message is a self-message named '*InitializeAfterDelay*'. Upon receiving this self-message, it initiates the delayed initialization process by invoking `*initializeAfterDelay*()` and then deletes the message to maintain memory efficiency.

For other types of messages, the function delegates handling responsibilities to the base class method *(`DemoBaseApplLayer::handleMessage(msg)`),* ensuring comprehensive message processing across various scenarios within the simulation environment.

This structured approach enables the `*EmergencyRSU*` class to effectively manage message reception and initialization tasks, facilitating responsive and efficient traffic management operations during emergency situations.

```
void EmergencyRSU::handleMessage(cMessage *msg) {
    if (msg->isSelfMessage() && strcmp(msg->getName(),
"InitializeAfterDelay") == 0) {
        EV << "Received self message 'InitializeAfterDelay'.
Initializing after delay." << endl;
        initializeAfterDelay();
        delete msg;
   } else {
        DemoBaseApplLayer::handleMessage(msg);
    }
}
```

The *initializeAfterDelay()* function pivotal for setting up the `*EmergencyRSU*` class after a specified delay in the simulation. This function begins by logging its initiation, signaling the start of setup operations. It first accesses the TraCI scenario manager through `*TraCIScenarioManagerAccess().get()*`, aiming to establish communication with the simulation environment.

Upon successful acquisition of the scenario manager, it retrieves the TraCI command interface using `*manager->getCommandInterface()*`.

If both components are successfully obtained, the function proceeds to configure the traffic light program for a specific ID=2 to the base program=0 using `*traci->trafficlight("2").setProgram("0")*`.

Each step is accompanied by informative log messages confirming the successful initialization of TraCI components and the setting of the traffic light program.

In cases where either the scenario manager or command interface is unavailable, corresponding error messages are logged, ensuring robust error handling throughout the initialization process.

This meticulous setup ensures that the `*EmergencyRSU*` class is fully prepared to manage traffic scenarios effectively within the simulated environment, enhancing its responsiveness to dynamic simulation events and scenarios.

```
void EmergencyRSU::initializeAfterDelay() {
    EV << "Initializing EmergencyRSU after delay." << endl;

    manager = TraCIScenarioManagerAccess().get();
    if (manager) {
        EV << "TraCIScenarioManager obtained successfully."
<< endl;
        traci = manager->getCommandInterface();
        if (traci) {
            EV << "TraCICommandInterface obtained successfully."
<< endl;
            // Imposta il programma di base per il semaforo
            traci->trafficlight("2").setProgram("0");
            EV << "Traffic light program set to base program."
<< endl;
        } else {
            EV << "Error: TraCICommandInterface not available."
<< endl;
        }
    } else {
        EV << "Error: TraCIScenarioManager not available."
<< endl;
        traci = nullptr;
    }
}
```

The *onWSM* function handles incoming Wave Short Messages (WSM) within the `*EmergencyRSU*` class, which is crucial for managing emergency vehicle scenarios in the simulation.

When invoked, the function first logs the receipt of a message, indicating the start of message processing. It then casts the received `*BaseFrame1609_4*` pointer `*wsm*` to a `*TraCIDemo11pMessage*` pointer `*wsmMsg*` using `*check_and_cast*`, allowing access to specific message data such as the type of message (`*wsmMsg->getDemoData()*`*)* and the ID of the vehicle sending the message (`*wsmMsg->getVehicleId()*`).

The function checks if the message indicates an "*EmergencyVehicle*" and ensures that the TraCI command interface is available. If both conditions are met, it retrieves the vehicle ID from the message and uses TraCI to obtain the vehicle object.

Subsequently, it retrieves the lane ID on which the vehicle is located using `*vehicle.getLaneId()*`.

Finally, it calls `*changeTrafficLightProgram(laneId)*` to adjust the traffic light program based on the detected lane ID, facilitating dynamic traffic management to prioritize emergency vehicles.

This function encapsulates the core logic for responding to emergency vehicle messages, leveraging simulation tools and data to enhance traffic control and emergency response mechanisms within the simulated environment.

```
void EmergencyRSU::onWSM(BaseFrame1609_4* wsm) {
    EV << "onWSM called. Message received." << endl;
    TraCIDemo11pMessage* wsmMsg =
check_and_cast<TraCIDemo11pMessage*>(wsm);
    EV << "Received WSM: " << wsmMsg->getDemoData() << " from
vehicle ID: " << wsmMsg->getVehicleId() << endl;

    if (std::string(wsmMsg->getDemoData()) == "EmergencyVehicle"
&& traci) {
        // Use vehicle ID from the message
        std::string vehicleId = wsmMsg->getVehicleId();
        EV << "Vehicle ID from message: " << vehicleId << endl;

        // obtain vehicle object with TraCI
        TraCICommandInterface::Vehicle vehicle = traci->
vehicle(vehicleId);

        // obtain lane ID
        std::string laneId = vehicle.getLaneId();
        EV << "Vehicle ID: " << vehicleId << " is on lane: "
<< laneId << endl;

        // Change program based on lane ID
        changeTrafficLightProgram(laneId);
    }
}
```

The *changeTrafficLightProgram* function is crucial for dynamically adjusting traffic light programs based on the lane ID provided as `*laneId*`.

At the outset, the function checks if the TraCI command interface is available. If TraCI is not available, an error message is logged, and the function exits early.

Assuming TraCI is accessible, the function evaluates the `*laneId*` parameter to determine which traffic light program to set for a specific traffic light ID.

If `*laneId*` corresponds to "3to2_0" or "1to2_0", indicating vehicles moving towards a junction, the function sets the traffic light program to "2_emergency_main", prioritizing emergency vehicles approaching the main route. This action is logged with a message indicating the program switch to "EMERGENCY_MAIN" for the identified lane.

Conversely, if `*laneId*` corresponds to "2to5_0", indicating vehicles moving towards a side route, the function sets the traffic light program to "2_emergency_side", prioritizing emergency vehicles on the side route. Again, this action is logged with a message indicating the program switch to "EMERGENCY_SIDE".

If `*laneId*` does not match any of these specific values, the function defaults the traffic light program to "0", representing the standard program for other lanes.

Each program change is accompanied by an appropriate log message detailing the lane for which the traffic light program has been set. This function is essential for enhancing traffic management in emergency scenarios within the simulation environment, ensuring efficient vehicle prioritization and traffic flow control based on real-time data from the simulation.

```
void EmergencyRSU::changeTrafficLightProgram(const std::string&
laneId) {
    if (!traci) {
        EV << "TraCI is not available." << endl;
        return;
    }
    if (laneId == "3to2_0" || laneId == "1to2_0") {
    traci->trafficlight("2").setProgram("2_emergency_main");
    EV << "Traffic light program set to EMERGENCY_MAIN for lane:
" << laneId << endl;
    } else if (laneId == "2to5_0") {
    traci->trafficlight("2").setProgram("2_emergency_side");
    EV << "Traffic light program set to EMERGENCY_SIDE for lane:
" << laneId << endl;
    } else {
        traci->trafficlight("2").setProgram("0");
        EV << "Traffic light program set to STANDARD for lane: "
<< laneId << endl;
    }
  }
```

The final step was to create the "omnetpp.ini" configuration file required to run the simulation. To do this, I based my work on the structure of the existing file from the example Veins scenario. I made several adjustments to tailor it to my specific simulation needs.

First, I set various general parameters such as enabling command-line environment express mode and auto-flushing, and configured the logging level to 'info.' The simulation network was specified as "EVPNetwork," and I defined several key simulation parameters including a simulation time limit of 100 seconds, enabling scalar and vector recording, and setting the playground size to 2000m x 2000m x 50m.

```
[General]
cmdenv-express-mode = true
cmdenv-autoflush = true
cmdenv-status-frequency = 1s
**.cmdenv-log-level = info

image-path = ../../images

network = EVPNetwork

##########################################################
#            Simulation parameters                       #
##########################################################
debug-on-errors = true
print-undisposed = true

sim-time-limit = 100s

**.scalar-recording = true
**.vector-recording = true

*.playgroundSizeX = 2000m
*.playgroundSizeY = 2000m
*.playgroundSizeZ = 50m
```

For the annotations, I enabled drawing to visualize the simulation better. Although I included a section for obstacle parameters, it is currently commented out, indicating no obstacles are defined in this scenario at this time.

```
##########################################################
# Annotation parameters                                  #
##########################################################
*.annotations.draw = true

##########################################################
# Obstacle parameters                                    #
##########################################################
#*.obstacles.obstacles = xmldoc("config.xml",
"//AnalogueModel[@type='SimpleObstacleShadowing']/obstacles")
```

Next, I configured the '*TraCIScenarioManager*' parameters, specifying the update interval, host, and port settings, and enabled automatic shutdown. I also referenced a launch configuration file for the manager.

```
##########################################################
#            TraCIScenarioManager parameters             #
##########################################################
*.manager.updateInterval = 1s
*.manager.host = "localhost"
*.manager.port = 9999
*.manager.autoShutdown = true
*.manager.launchConfig = xmldoc("EVP.launchd.xml")
```

In the application layer section, I set the application type for the nodes to "*EmergencyVehicle*" and defined specific settings for the RSUs (Roadside Units), including their coordinates and application types called "*EmergencyRSU*".

```
############################################################
#                       App Layer                          #
############################################################
*.node[*].applType = "EmergencyVehicle"

############################################################
#                        RSU SETTINGS                      #
#                                                          #
#                                                          #
############################################################
*.rsu[0].mobility.x = 100
*.rsu[0].mobility.y = 80
*.rsu[0].mobility.z = 3
*.rsu[*].applType = "EmergencyRSU"
```

For the 11p specific parameters, I enabled direct sending, set the maximum interference distance, and configured various MAC layer settings including the use of the service channel, transmission power, and bitrate. PHY layer configurations were also detailed, specifying the noise floor, propagation delay options, and referencing an antenna model from an XML file. Additionally, I set the antenna offset for nodes.

```
############################################################
#             11p specific parameters                      #
#                                                          #
#                    NIC-Settings                          #
############################################################
*.connectionManager.sendDirect = true
*.connectionManager.maxInterfDist = 2600m
*.connectionManager.drawMaxIntfDist = false

*.**.nic.mac1609_4.useServiceChannel = true

*.**.nic.mac1609_4.txPower = 50mW
*.**.nic.mac1609_4.bitrate = 6Mbps
*.**.nic.phy80211p.minPowerLevel = -110dBm

*.**.nic.phy80211p.useNoiseFloor = true
*.**.nic.phy80211p.noiseFloor = -98dBm

*.**.nic.phy80211p.decider = xmldoc("config.xml")
*.**.nic.phy80211p.analogueModels = xmldoc("config.xml")
*.**.nic.phy80211p.usePropagationDelay = true

*.**.nic.phy80211p.antenna = xmldoc("antenna.xml",
"/root/Antenna[@id='monopole']")
*.node[*].nic.phy80211p.antennaOffsetY = 0 m
*.node[*].nic.phy80211p.antennaOffsetZ = 1.895 m
```

Lastly, I defined mobility parameters for the nodes, initializing their positions to zero and indicating they remain stationary during the simulation.

```
############################################################
#                        Mobility                          #
############################################################
*.node[*].veinsmobility.x = 0
*.node[*].veinsmobility.y = 0
*.node[*].veinsmobility.z = 0
*.node[*].veinsmobility.setHostSpeed = false
```

Overall, this configuration sets up a detailed scenario to study the communication and interaction of emergency vehicles within a vehicular network.

In this chapter, we detailed the implementation and results of our simulation using OMNeT++ and Veins. We constructed a comprehensive simulation environment to test Vehicle-to-Infrastructure (V2I) communication for emergency vehicle preemption scenarios. Through a systematic approach, we outlined the creation and configuration of necessary modules, message definitions, and traffic management protocols.

Key components such as the *EmergencyVehicle* and *EmergencyRSU* classes were meticulously developed to handle specific aspects of the simulation, from sending emergency notifications to dynamically adjusting traffic light programs. The successful integration of these elements was demonstrated through detailed code explanations and corresponding log messages, showcasing the robustness of the simulation framework.

The results obtained provide valuable insights into the effectiveness of V2I communication in enhancing traffic management during emergency situations. By prioritizing emergency vehicles and optimizing traffic signal timing, our simulation underscores the potential for improved response times and overall traffic flow in real-world applications. These findings pave the way for future research and development in intelligent transportation systems and vehicular ad hoc networks (VANETs).

4.2.3 SIMULATION RESULTS

In this chapter, I present a thorough analysis of the simulations conducted using the codes and methodologies detailed in the previous chapter. My primary focus is to evaluate the effectiveness and impact of the implemented V2I communication protocols in emergency vehicle preemption scenarios.

I begin by examining the performance and behaviour of the simulation environment equipped with the custom modules and message definitions introduced earlier. This involves a detailed look at how the emergency vehicles interact with RSU and traffic signals to ensure optimal traffic flow during emergency situations. Specific metrics such as message transmission success messages, response times, and traffic signal adjustments are analysed to provide a comprehensive understanding of the system's functionality.

Following this analysis, I compare the performance of the simulation environment with and without the implemented V2I communication code. By contrasting scenarios where emergency vehicle preemption is active against those where it is not, I aim to highlight the improvements and benefits brought about by our proposed system.

In *Figure 20*, we observe the entry of a passenger vehicle into the simulation. The traffic light is set to operate with the base program, showing green on the main lanes and red on the side lane. Since the green phase of the base program lasts 28 seconds, we expect this vehicle to cross the intersection without any issues.

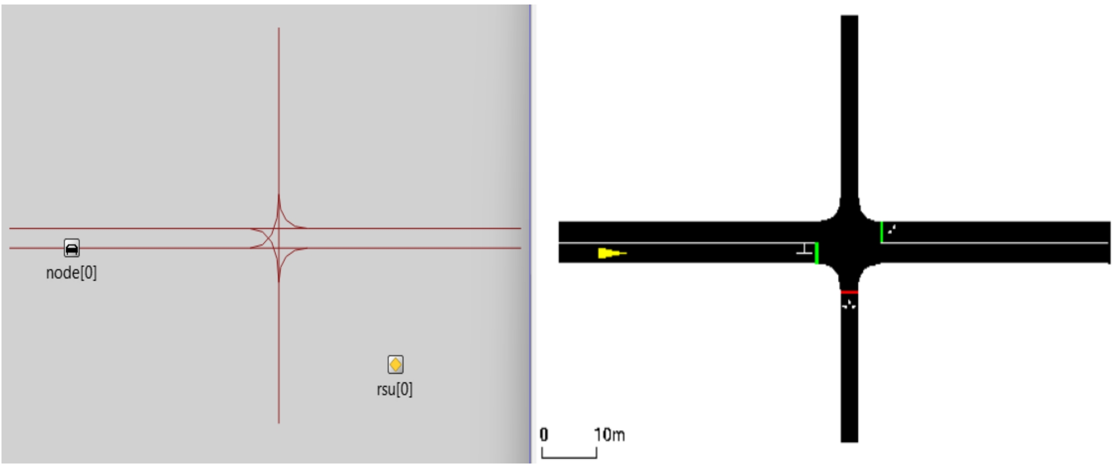

Figure 20 - Simulation at t=14s

According to the program defined in the '**EVP.rou**' file, the first emergency vehicle enters the simulation at t=15s.

In *Figure 21*, we can see that the message sent by the vehicle with ID "emergency1" has been correctly received by the RSU. Based on the lane in which the vehicle is located, the traffic light program "2_emergency_side" has been correctly set, as the vehicle is in the side lane.

```
** Event #848  t=17.000062041432  EVPNetwork.rsu[0].appl (EmergencyRSU, id=9)  on  (veins::TraCIDemo11pMessage, id=382)
INFO: onWSM called. Message received.
INFO: Received WSM: EmergencyVehicle from vehicle ID: emergency1
INFO: Vehicle ID from message: emergency1
INFO: Vehicle ID: emergency1 is on lane: 2to5_0
INFO: Traffic light program set to EMERGENCY_SIDE for lane: 2to5_0
```

Figure 21 - Message Confirmation and Traffic Lights Program Changing

Indeed, in *Figure 22*, it is possible to observe that the traffic light program has been correctly modified to allow the safe passage of the emergency vehicle.

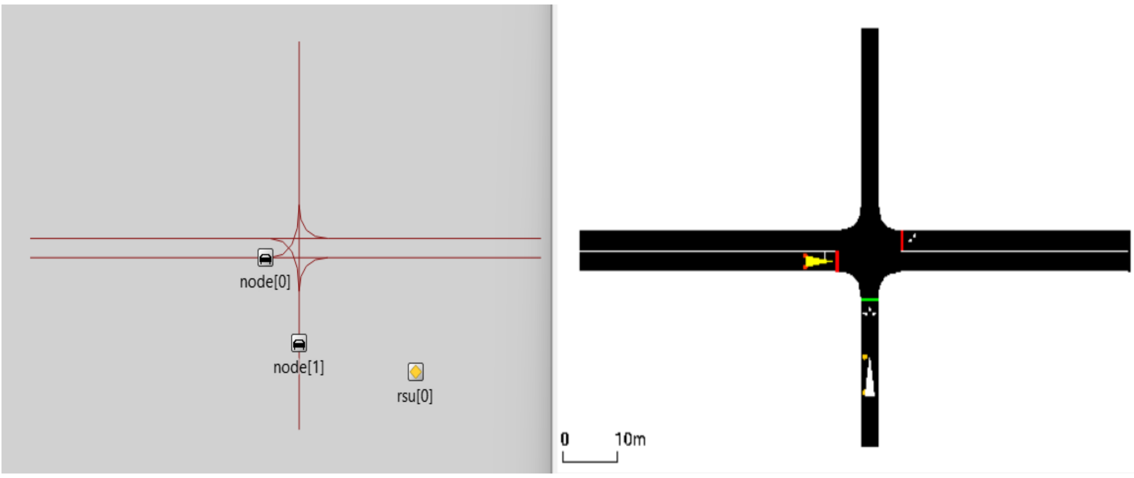

Figure 22 - Simulation at t=17s

In *Figure 23*, both simulations are captured at t=17s.

The left side shows the simulation conducted using Veins, including the custom codes I generated. The right side shows the simulation performed using only SUMO, without implementing preemption for emergency vehicles.

On the left, the emergency vehicle (indicated by a special symbol or color) is given priority at the intersection, as reflected by the green light in the side lane. This demonstrates the effective application of the emergency vehicle preemption system, allowing the emergency vehicle to pass through the intersection safely and without delay.

On the right, the simulation without Veins shows the emergency vehicle waiting at a red light, just like the other vehicles. This scenario highlights the lack of priority for emergency vehicles, resulting in potential delays and a less efficient response in real-world situations.

This comparison clearly illustrates the advantage of integrating Veins with SUMO to manage emergency vehicle preemption at traffic lights, ensuring quicker and safer passage for emergency services.

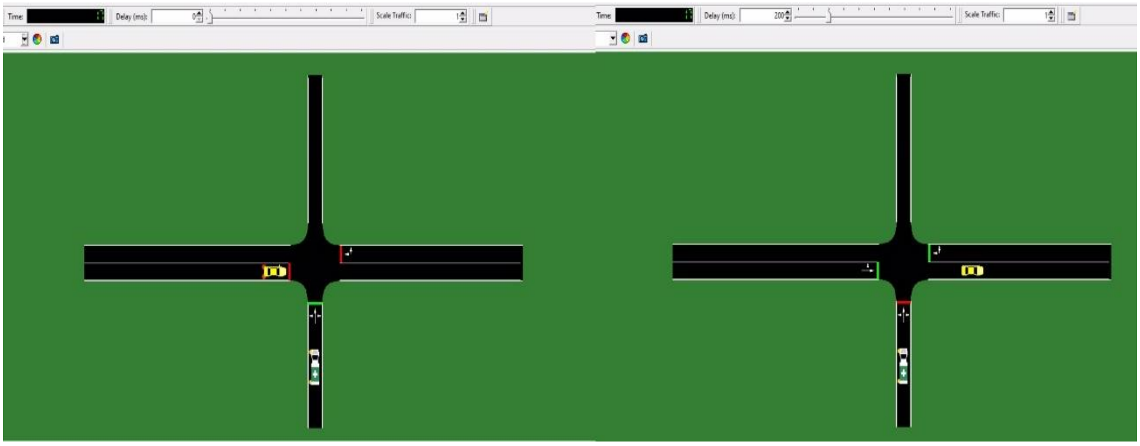

Figure 23 - Simulations Comparison at t=17s

The second emergency vehicle enters the simulation at 35 seconds.

Figure 24 is introduced to highlight the situation at t=34 seconds, just before the emergency vehicle arrives on the scene, demonstrating how the emergency program is correctly set in the following seconds.

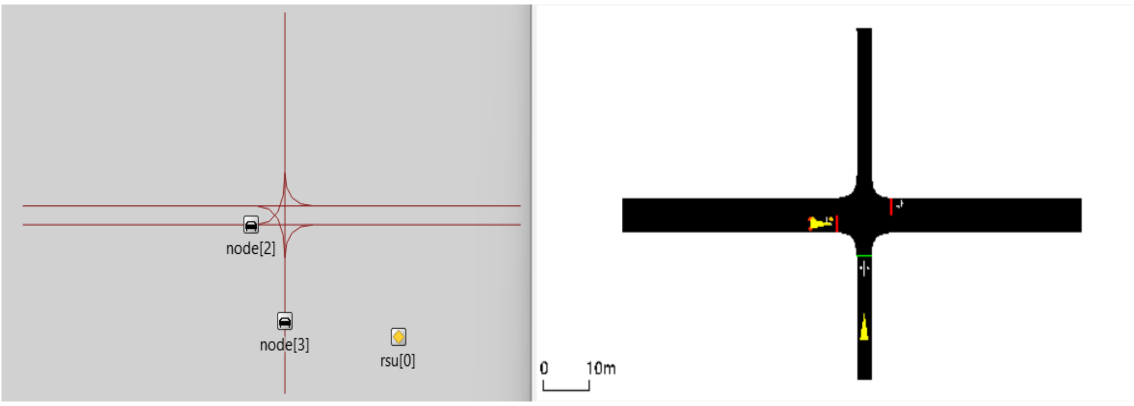

Figure 24 - Simulation at t=34s

In the following sections, it will be analysed what happens at 39 seconds, evaluating whether everything functions correctly and examining the differences between the two simulations.

In *Figure 25,* it is possible to observe how, within just 4 seconds, the message is correctly received by the RSU, emphasizing the lane and the traffic light program tailored to that lane.

```
** Event #3666  t=39.000052173342  EVPNetwork.rsu[0].appl (EmergencyRSU, id=9)  on  (veins::TraCIDemo11pMessage, id=1831)
INFO: onWSM called. Message received.
INFO: Received WSM: EmergencyVehicle from vehicle ID: emergency2
INFO: Vehicle ID from message: emergency2
INFO: Vehicle ID: emergency2 is on lane: 3to2_0
INFO: Traffic light program set to EMERGENCY_MAIN for lane: 3to2_0
```

Figure 25 - Message Confirmation and Traffic Lights Program Changing at t=39s

Consequently, as noted in the image below, the scenario changes accordingly.

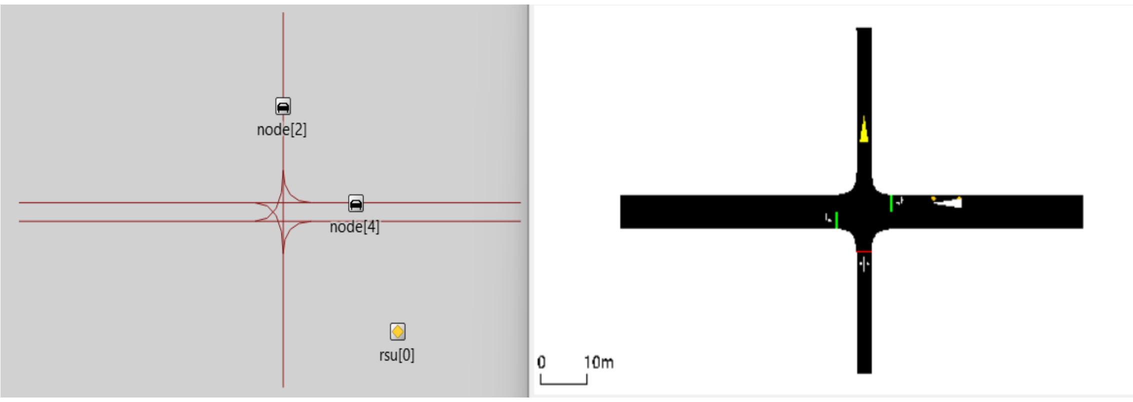

Figure 26 - Simulation at t=39s

Both simulations in *Figure 27* capture the scene at the 39th second. The contrasting scenarios highlight the significant impact of preemption. In the Veins simulation, the emergency vehicle efficiently navigates the intersection due to the prioritized traffic signal. Conversely, the SUMO simulation forces the emergency vehicle to wait alongside regular traffic, leading to congestion. This imagery underscores the importance of incorporating emergency vehicle behaviour into traffic simulation models.

A closer look at the image reveals another interesting detail: in the SUMO environment, the first emergency vehicle, is still present at the intersection. Despite the lack of preemption, "*emergency 1*" manages to cross the intersection after approximately 20 seconds from its arrival. In contrast, in the Veins simulation, the use of the preemption system allows the single emergency vehicle to clear the intersection in just 7 seconds. This comparison further highlights the effectiveness of preemption in reducing waiting times and facilitating the rapid passage of emergency vehicles avoiding traffic congestion.

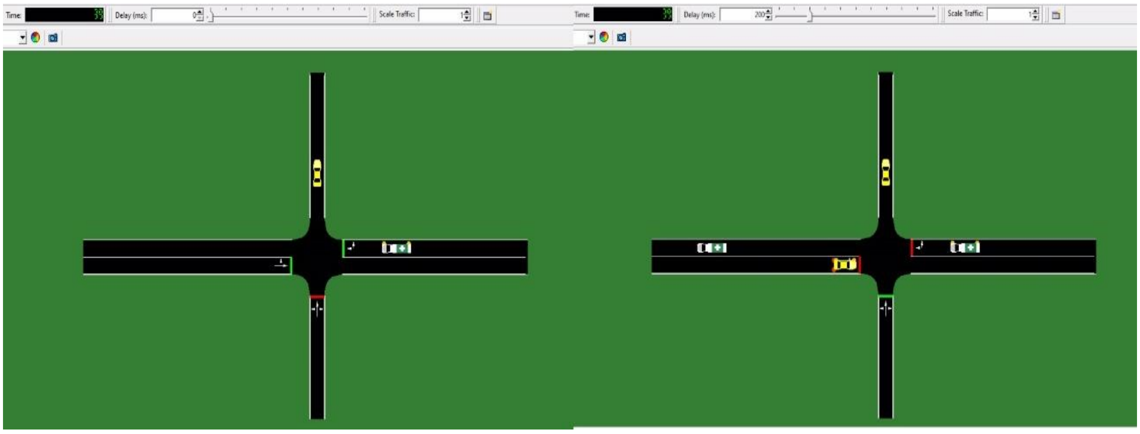

Figure 27 - Simulations Comparison at t=39s

The last scenario occurs around second 48, when the last emergency vehicle enters this simulation. As observed in the *Figure 28*, at second 48, just as the "*emergency*

3" vehicle enters the simulation, the traffic light is still set to the default program: red for the main lane and green for the secondary lane.

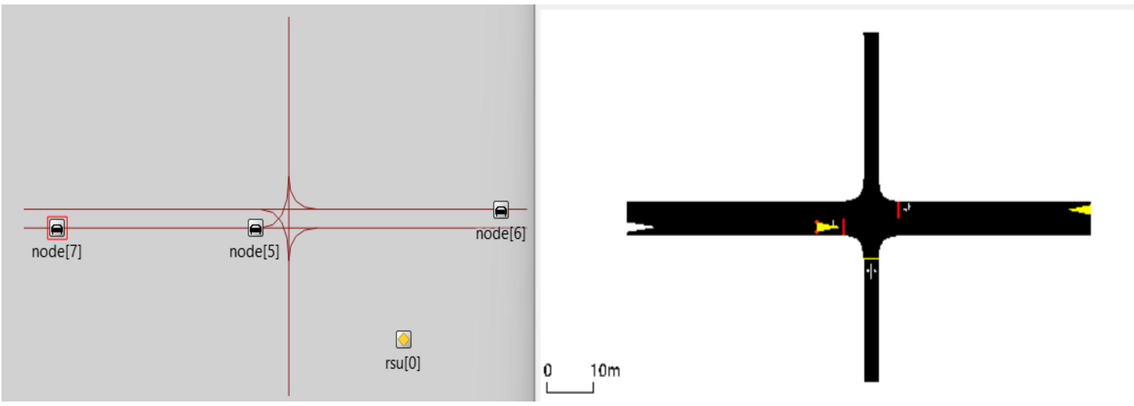

Figure 28 - Simulation at t=48s

At second 50, the emergency vehicle sends a message to the RSU to declare its presence, specifying its position, so that the traffic light program is changed to "*2_emergency_main*".

```
** Event #5104  t=50.00004978448  EVPNetwork.rsu[0].appl (EmergencyRSU, id=9)  on  (veins::TraCIDemo11pMessage, id=2601)
INFO: onWSM called. Message received.
INFO: Received WSM: EmergencyVehicle from vehicle ID: emergency3
INFO: Vehicle ID from message: emergency3
INFO: Vehicle ID: emergency3 is on lane: 1to2_0
INFO: Traffic light program set to EMERGENCY_MAIN for lane: 1to2_0
```

Figure 29 - Message Confirmation ad Traffic Lights Program Changing at t=50s

From *Figure 30*, it is evident how the traffic light immediately turns green for the main lane and red for the secondary lane, ensuring the emergency vehicle can cross the intersection safely without concerns about other vehicles in the secondary lanes.

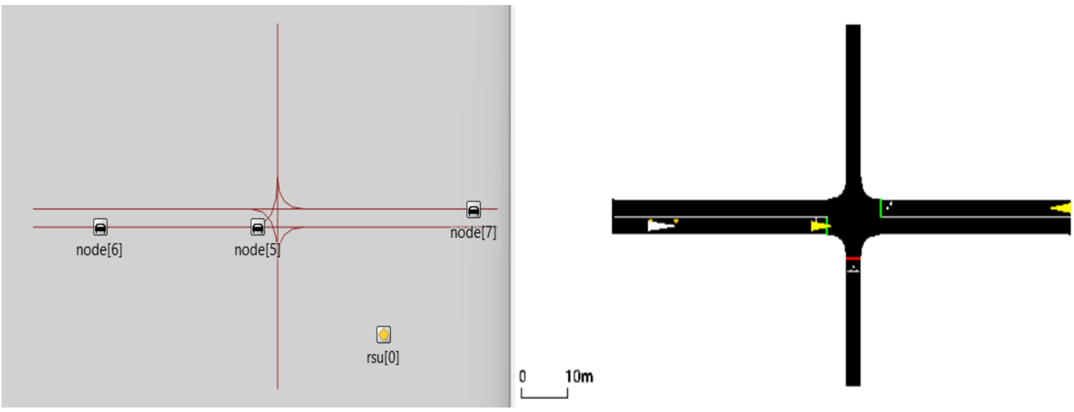

Figure 30 - Simulation at t=50s

Both simulations in *Figure 31* capture the scene at the 50th second. The left frame, generated by VEINS, showcases the power of preemption.

Here, the emergency vehicle seamlessly navigates the intersection due to the prioritized traffic signal granted by preemption system.

In stark contrast, the right frame, captured by SUMO which lacks preemption functionality, reveals a different story. It is important to underline the presence of another emergency vehicle, labelled "Emergency 2" in the SUMO simulation, it's forced to wait alongside regular traffic due to the absence of preemption. This congestion significantly hinders its progress, highlighting the importance of traffic light control for facilitating emergency response.

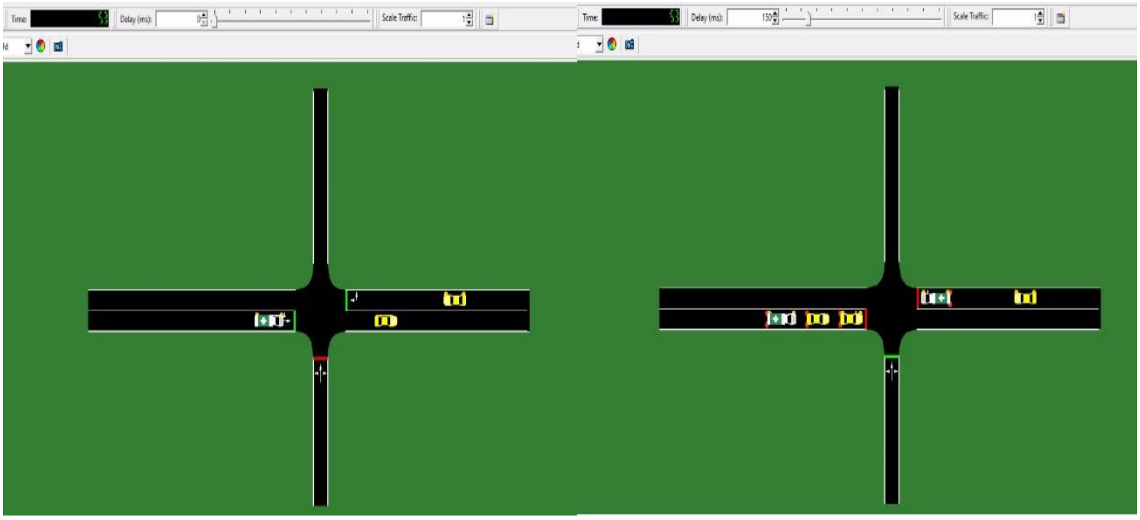

Figure 31 - Simulations Comparison at t=50s

In conclusion, the simulation results convincingly demonstrate the effectiveness of the implemented V2I communication protocols for emergency vehicle preemption. The analysis of *Figures 20* through *31* showcases the successful interaction between emergency vehicles, RSU, and traffic signals facilitated by the custom Veins modules.

Specific metrics such as message transmission success and response times confirm the system's functionality. Furthermore, the contrasting scenarios with and without preemption highlight the clear advantages of our proposed system.

Veins with preemption functionality allows emergency vehicles to swiftly navigate intersections, as evidenced by the prioritized traffic signals *in Figures 22, 26,* and *30*. This significantly reduces emergency response times compared to scenarios without preemption, where emergency vehicles are forced to wait at red lights alongside regular traffic (Figures 23,27 and 31). The analysis also revealed the positive impact of Veins on overall traffic flow. The dynamic adjustment of traffic light timings based on emergency vehicle presence leads to smoother traffic flow with lower vehicle density compared to SUMO simulations.

These findings strongly support the value of integrating V2I communication protocols for emergency vehicle preemption into traffic simulation models. By optimizing traffic flow and facilitating quicker emergency response times, this technology can significantly contribute to improved public safety and efficiency on the roadways.

4.2.4 CRITICAL ISSUES AND IMPROVEMENT PROPOSALS

The development and simulation of the emergency vehicle preemption system have unearthed critical issues that hinder its widespread implementation and effectiveness. These limitations primarily stem from the simulation environment's inability to fully encapsulate real-world traffic complexities and the system's design constraints. Addressing these shortcomings is crucial for developing a robust and reliable emergency vehicle preemption system.

Limited Realism of Traffic Patterns

The current simulation environment, while robust, does not fully capture the complexity and variability of real-world traffic. Real traffic conditions are influenced by unpredictable human behaviours, a diverse array of vehicle types, and varying road conditions. These factors are often oversimplified in simulation models, which can lead to discrepancies between simulated and actual traffic dynamics.

To bridge this gap, it is essential to incorporate more sophisticated traffic models that account for the diversity and unpredictability of real-world traffic. Enhancements could include integrating stochastic models that simulate a range of driver behaviours and decision-making processes, including different vehicle types such as motorcycles, trucks, and buses, each with distinct operational characteristics. Additionally, varying road conditions such as construction zones, accidents, and other incidents that impact traffic flow should be simulated. By increasing the realism of traffic patterns, the simulation can provide more accurate insights and better predict the system's performance in real-world scenarios.

Scalability Issues

The scalability of the simulation poses a significant challenge. As the number of vehicles and intersections increases, the computational resources required to maintain the simulation's performance and accuracy become substantial. This limitation may hinder the application of the simulation model to larger, more complex urban areas without significant modifications and resource investments.

To address scalability, utilizing cloud computing and distributed processing to handle larger datasets and more complex simulations is necessary. Implementing efficient algorithms to reduce computational load, such as parallel processing techniques and data partitioning, can further enhance performance. Developing a modular framework that allows for incremental scaling will enable the simulation to handle different parts of a city independently while maintaining overall coherence. Improving scalability will allow the system to be applied to larger urban areas, thereby enhancing its utility and effectiveness.

Communication Delays and Errors

The simulation currently assumes an ideal communication environment between vehicles and infrastructure, which does not reflect real-world conditions. In practice, communication delays, packet losses, and other forms of signal degradation can occur, impacting the effectiveness of V2I communication protocols.

To enhance the realism and applicability of the simulation results, it is important to model communication imperfections. Incorporating realistic delay models that account for network congestion and signal propagation times is essential. Simulating packet losses and implementing robust error correction mechanisms will provide a more accurate representation of real-world communication conditions. Additionally, varying signal quality based on environmental factors and infrastructure constraints can be modelled. These enhancements will lead to more reliable system performance.

Hardcoding of Emergency Vehicle IDs

Currently, emergency vehicle identifiers are hardcoded in the system, limiting scalability and maintainability. Each time an emergency vehicle is added or removed, the source code must be manually updated, increasing the risk of errors and reducing operational efficiency.

Adopting a more dynamic approach to manage emergency vehicle identifiers can significantly improve the system. Implementing a centralized database to dynamically manage emergency vehicle identifiers will ensure all vehicles are accurately tracked without manual code modifications. Using generic identifiers based on vehicle type rather than specific IDs will simplify updates and reduce configuration errors. These changes will enhance the system's flexibility, making it easier to manage and maintain.

Limited Traffic Signal Programs

The traffic signal program used in the simulation is tailored specifically for the study scenario, making it rigid and unsuitable for different situations. This configuration does not allow for dynamic adaptation of traffic signals in response to the position and direction of emergency vehicles.

To improve traffic signal management, developing a more flexible traffic signal control system that can dynamically adjust based on the GPS coordinates of emergency vehicles is necessary. This system should consider the travel directions and relative positions of emergency vehicles with respect to Roadside Units (RSUs). Implementing adaptive algorithms that respond in real-time to changes in traffic conditions and emergency vehicle movements will lead to more precise and responsive traffic signal management, enhancing traffic flow and reducing emergency response times.

Impact of Traffic Congestion on Signal Timing

Another critical issue is that the current approach to managing traffic signals based on predefined timings may not be effective in the presence of traffic congestion. If an emergency vehicle encounters a traffic jam, it may not reach the intersection within the pre-set signal change intervals, rendering the signal preemption ineffective.

To mitigate this issue, it is essential to develop a dynamic traffic signal control system that can adapt to real-time traffic conditions. This system should use real-time traffic data to adjust signal timings dynamically, ensuring that emergency vehicles can navigate through congested intersections effectively. Additionally, incorporating predictive algorithms that anticipate traffic congestion and adjust signals proactively can further enhance the system's reliability and effectiveness.

Vehicle-to-Vehicle Communication

The current system focuses on communication between emergency vehicles and infrastructure, neglecting the importance of notifying other vehicles on the road. Effective Vehicle-to-Vehicle (V2V) communication is crucial for road safety and efficient emergency vehicle passage.

Integrating V2V communication can significantly enhance road safety. Developing systems that alert other vehicles of the presence of emergency vehicles will allow them to clear lanes and reduce speed. Encouraging collaborative driving behaviours through V2V communication will create a safer and more cooperative traffic environment. This integration will reduce the risk of accidents and facilitate the passage of emergency vehicles, contributing to a safer traffic ecosystem.

Lack of Environmental Factors

The simulation does not account for environmental factors such as weather conditions, visibility, and road surface conditions. These factors can significantly impact vehicle performance and safety, especially during emergency situations.

Incorporating environmental factors into the simulation can provide a more comprehensive understanding of system performance. Integrating weather conditions such as rain, fog, and snow, which affect visibility and road friction, is necessary. Simulating varying road surface conditions, including wet, icy, or damaged roads, will provide a more realistic assessment. Adjusting vehicle behaviour models to account for changes in environmental conditions will offer a more accurate understanding of system performance under different scenarios.

5. SMARTPHONES IN VEHICULAR COMMUNICATION

The evolution of vehicular communication technologies, such as Vehicle-to-Everything (V2X) and Cellular Vehicle-to-Everything (C-V2X), has opened new opportunities to enhance road safety, traffic efficiency, and the overall driving experience. Traditionally, these technologies have relied on dedicated infrastructure and specialized devices installed in vehicles and road infrastructures. However, the emergence of smartphones as ubiquitous devices equipped with advanced sensors and connectivity capabilities has introduced an innovative paradigm in vehicular communications: utilizing smartphones themselves as active nodes in V2X and C-V2X networks.

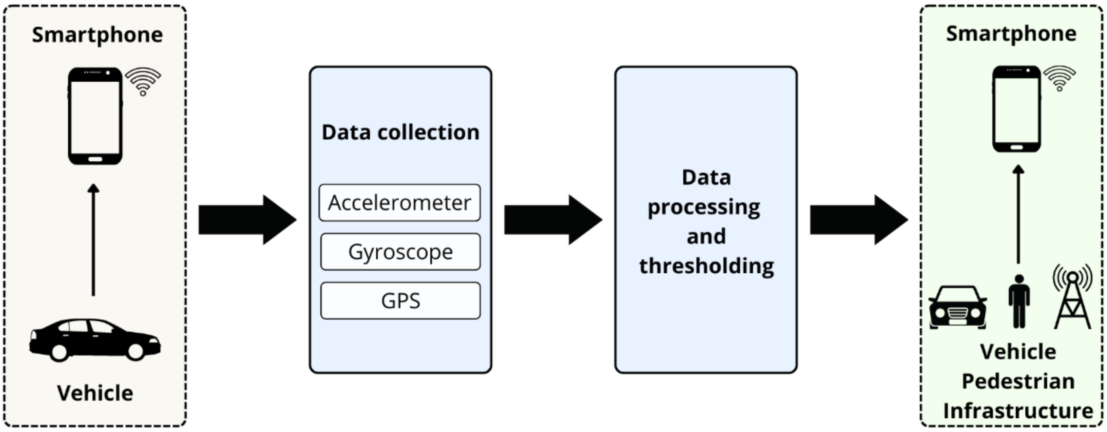

Figure 32 - Smartphones in vehicular communication

Modern smartphones are equipped with a wide range of sensors [22],[23], including GPS, accelerometers, gyroscopes, magnetometers, and cameras, making them powerful tools for detection and data collection. Furthermore, their Wi-Fi, Bluetooth, and cellular connectivity enables effective communication with other devices and infrastructures. This combination of sensors and connectivity makes smartphones ideal for integration into V2X and C-V2X networks, performing roles such as:

- *Traffic Monitoring*: Smartphones can collect real-time data on traffic flows, detect congestion and accidents, and transmit this information to traffic control centers and nearby vehicles. Using GPS, smartphones can provide precise location data, while motion sensors can identify sudden accelerations or braking, which may indicate hazardous road conditions.
- *Vehicle-to-Vehicle Communication (V2V):* Smartphones can facilitate direct communication between vehicles using technologies like Wi-Fi Direct or Bluetooth [22]. This type of communication can be used to coordinate

overtaking maneuvers, warn drivers of potential collisions, and share real-time information about road conditions.

- *Interaction with Infrastructure (V2I):* Smartphones can communicate with smart road infrastructure, such as traffic lights and road signs, to receive traffic status updates and optimize driving routes [24]. This capability to communicate with road infrastructure can significantly improve traffic flow and reduce wait times at traffic signals.
- Protection of Vulnerable Road Users (VRU): Smartphones can be used to improve safety for VRUs, such as pedestrians and cyclists, by sending alerts to nearby vehicles and infrastructure. This enables early warning systems that can prevent accidents by notifying drivers of VRU presence, particularly at intersections and crosswalks.
- *Participation in C-V2X Networks*: Beyond Wi-Fi-based communication, smartphones can also leverage 4G and 5G cellular networks to participate in C-V2X networks, providing greater coverage and reliability in communication. C-V2X networks offer reduced latency and higher data transmission capacity, allowing smartphones to handle complex vehicular communication scenarios.

Integrating smartphones into V2X and C-V2X networks offers numerous advantages, making V2X technology more accessible and scalable. The use of existing devices, such as smartphones, significantly reduces the need for investment in dedicated infrastructure, enhancing cost-effectiveness. Smartphones also offer flexibility and adaptability [22],[23], as they can be easily updated with new features and applications, allowing for rapid adaptation to technological innovations and new market demands. With billions of smartphones in use worldwide, their integration into vehicular networks can accelerate the adoption and spread of V2X technologies, potentially improving traffic safety and efficiency globally. However, using smartphones as V2X and C-V2X nodes also presents several challenges. The collection and transmission of sensitive data necessitate adequate measures to protect user privacy and prevent unauthorized access, addressing security and privacy concerns. Connection reliability is another issue, as dependence on cellular and Wi-Fi networks for V2X communication may be affected by coverage issues and network congestion, potentially compromising real-time communications. Furthermore, the intensive use of sensors and communication features on smartphones can lead to high energy consumption, reducing battery life and limiting the device's usability for other applications.

Modern smartphones come equipped with numerous interfaces that enable connectivity to various networks and devices, which are crucial for vehicular communication in a V2X environment. Wi-Fi, including Wi-Fi Direct, allows direct communication between devices for peer-to-peer connections without a central access point, while Wi-Fi Ad Hoc facilitates temporary communication networks between vehicles, allowing for information exchange without a fixed infrastructure. Wi-Fi Hotspot enables devices such as smartphones, tablets, or computers to connect to the Internet or a local wireless network. Bluetooth offers both Bluetooth Low Energy (BLE) for low-power data transfers with nearby smart devices and Bluetooth Classic for less latency-sensitive applications, such as communication with infotainment devices or wearables [S18]. Cellular networks, including 5G and 4G LTE, provide connectivity with 5G offering high data speeds, low latency, and advanced network capabilities ideal for real-time V2X applications, supporting C-V2X communication for V2V, V2I, and V2P interactions, while 4G LTE supports some V2X applications with lower capacity and latency. NFC, though limited to very short distances, can be used for specific applications like device pairing or access management. UWB (Ultra-Wideband) provides precise localization and short-range communication, useful for accurate position detection in parking navigation or object tracking around the vehicle. Finally, mmWave, part of the 5G specification, enables ultra-high-speed communication over short distances, suitable for high-resolution data streaming between vehicles and infrastructure.

SMARTPHONES FOR DSRC

Vehicular communication using DSRC (Dedicated Short-Range Communications) is based on the IEEE 802.11p protocol, part of the WAVE (Wireless Access in Vehicular Environments) standard or the ETSI ITS-G5. This protocol is designed to provide low-latency, high-speed communication between vehicles and infrastructure.

Modern smartphones are not directly compatible with the 802.11p standard used in DSRC, so certain integrations [23] are necessary to enable mobile devices to work with vehicular communication protocols:

- External Modules and Adapters:
  - *External Hardware*: Smartphones can be connected to external DSRC modules via USB or Bluetooth interfaces. These modules are designed to support IEEE 802.11p and handle the transmission and reception of DSRC signals.
  - *Telematics Control Unit (TCU)*: In some vehicles, an integrated TCU can interface with the smartphone to collect data and manage DSRC communications.
- Software Applications:
  - *Data Aggregation:* Smartphones can collect and process data from internal sensors such as GPS and accelerometers and send this data to a DSRC unit for transmission.
  - *User Interface*: Smartphone applications can provide user interfaces to notify and display safety alerts derived from DSRC communications.
- Integration with Infotainment Systems:

- *Vehicle System Connection:* Smartphones can connect to infotainment systems in vehicles that are already equipped with DSRC capabilities, serving as input/output terminals for the information collected and transmitted via DSRC.

- Cellular Network Utilization as Support:
    - *Hybrid Communications:* In some architectures, smartphones can use their 4G/5G capabilities to complement DSRC communications, improving coverage and latency where cellular networks offer advantages.
- Beaconing and Data Management:
    - Smartphones can perform beaconing operations and manage small data packets that are transmitted or received by DSRC devices.

Early studies on the use of smartphones as nodes for DSRC (Dedicated Short-Range Communications) have focused on Vulnerable Road Users (VRUs), which include pedestrians, cyclists, and motorcyclists. These individuals are particularly at risk of road accidents due to their limited physical protection compared to motor vehicles. Implementing smartphones as DSRC communication nodes for VRUs is crucial for several reasons [25].

VRUs are exposed to higher risks in urban and suburban traffic, and the use of smartphones can significantly enhance their safety. Equipped with sensors such as GPS and accelerometers, smartphones can detect the position of pedestrians and cyclists, sending alerts to nearby vehicles and infrastructure. This real-time communication helps signal the presence of VRUs, reducing accident risks and improving driver awareness.

For motorcyclists, smartphones can provide safety alerts about hazardous road conditions or heavy traffic. These notifications assist motorcyclists in reacting more effectively to changing road conditions and in preventing dangerous situations.

Regarding this topic, in the paper [25] *is* proposed a system to protect vulnerable road users. This system predicts vehicle collision probabilities and uses smartphones to warn drivers when GPS and 4G wireless communication are active. The key challenge addressed is minimizing false positives and false negatives, which can lead to driver distraction and reduced trust in the alert system. The proposed system operates in three phases:

- *Activation Phase:* The system activates when a vehicle and a pedestrian are within a designated activation area around the vehicle.
- *Prediction Phase:* It estimates future collision probabilities by analysing factors such as the vehicle's location, speed, acceleration, and direction.

- *Warning Phase:* The system decides if a warning is necessary and issues alerts based on the assessed collision risk, aiming to reduce false alarms and effectively identify dangerous situations.

In 2013, Honda launched a project focused on Vehicle-to-Pedestrian (V2P) communication. The company successfully demonstrated the ability of vehicles equipped with Dedicated Short-Range Communications (DSRC) technology to detect pedestrians using DSRC-enabled smartphones [S19]. This V2P technology facilitates cooperative communication between a pedestrian's smartphone and nearby vehicles, providing both acoustic and visual alerts to pedestrians and drivers to prevent collisions.

The V2P system utilizes the pedestrian's smartphone GPS, dynamic detection capabilities, and DSRC wireless technology operating in the 5.9 GHz band. This setup allows the smartphone and surrounding vehicles to establish a communication channel to monitor and determine if a pedestrian is at risk of being hit. The system is particularly valuable in situations where the pedestrian is not easily visible to the driver, such as when stepping off a curb behind a parked vehicle or another traffic obstruction.

Through a dedicated app, the system can ascertain the pedestrian's position, direction, and speed, as well as the positions of nearby vehicles. If an imminent collision risk is detected, the app sends a high-intensity acoustic alert and a message to the pedestrian's smartphone. Simultaneously, the vehicle receives alerts via an acoustic alarm and visual signals on the head-up display and navigation screen. Additionally, the system can detect if the pedestrian is using their phone for texting, listening to music, or making calls, incorporating this information into the collision risk assessment.

One of the leading mobile and fixed-line telecom companies is developing an innovative project that uses smartphones as nodes for vehicular communication [S20]. The platform, called Vodafone STEP, is designed based on the guidelines of key standardization bodies such as ETSI and 5GAA. This system addresses the issue of data fragmentation in mobility by effectively distributing information across various traffic and transport domains throughout Europe. Vodafone STEP provides a secure and reliable communication channel for managing and validating V2X (Vehicle-to-Everything) messages in real-time, ensuring smooth and coordinated data distribution on mobility.

In conclusion, the integration of smartphones into Dedicated Short-Range Communications (DSRC) for vehicular communication marks a significant advancement in road safety technology. By leveraging the IEEE 802.11p protocol and DSRC capabilities, smartphones can bridge the gap between vulnerable road users (VRUs) and vehicles, enhancing situational awareness and reducing collision risks. The use of external modules, software applications, and integration with infotainment systems allows for effective communication and data management, while hybrid communications with cellular networks further extend coverage and reliability.

Honda's 2013 project on Vehicle-to-Pedestrian (V2P) communication and ongoing innovations, such as Vodafone STEP, illustrate the potential of these technologies to transform traffic safety. With real-time alerts and comprehensive risk assessments, these systems aim to protect pedestrians, cyclists, and motorcyclists by improving detection and communication in complex traffic environments. As these technologies continue to evolve and integrate, they promise to play a crucial role in creating safer and more efficient roadways for all users.

SMARTPHONES FOR C-V2X

C-V2X (Cellular Vehicle-to-Everything) is a technology that leverages cellular networks for vehicular communication, enabling communication between vehicles (V2V), vehicles and infrastructure (V2I), and vehicles and pedestrians (V2P). This technology benefits from the existing infrastructure of 4G LTE and 5G networks to ensure efficient and reliable communication.

C-V2X offers low latency, which is crucial for applications requiring real-time responses, such as traffic control and accident prevention. 5G networks provide high data capacity, essential for handling large volumes of data and numerous connected devices. Additionally, cellular communications ensure extensive coverage and a stable connection, which are critical for high-priority V2X applications.

Modern smartphones, equipped with support for 4G and 5G networks, can play a key role in enabling C-V2X communication in various ways:

- Direct Connections
    - *5G NR (New Radio):* Smartphones can utilize 5G connectivity to communicate directly between vehicles at high speeds, bypassing centralized network infrastructure.
    - *PC5 Mode:* This mode allows direct communication between devices, reducing reliance on cellular networks and enhancing responsiveness in low-latency scenarios.
- Utilizing Existing Infrastructure
    - *Edge Computing*: Smartphones can interact with 5G network edge computing platforms to process data locally, reducing latency and increasing communication efficiency.
    - *Cloud Integration:* By connecting to cloud services, smartphones can collect and analyse real-time data, optimizing traffic management and incident response.
- Advanced Applications

- *Navigation and Infotainment:* Smartphones can provide real-time route information and infotainment services based on data from C-V2X networks.
- *Safety Systems:* Smartphone applications can send alerts and notifications about risk situations, such as imminent collisions or hazardous road conditions.

Also, for C-V2X, VRUs (Vulnerable Road Users) are among the most studied areas for smartphone-based communication. The document [26], presents a new collision warning system specifically designed for VRUs using Vehicle-to-Pedestrian (V2P) communication.

The key contributions of this research are:

- Design of a Phone Case: The study introduces a specialized phone case that enables smartphones to communicate directly with vehicles via Cellular Vehicle-to-Everything (C-V2X) technology. This design aims to significantly enhance the safety of VRUs by reducing the risk of collisions with vehicles.
- Collision Warning Method: A novel collision warning method based on s-t-coordinates is proposed and tested in the field. This approach provides effective alerts to both drivers and VRUs, helping to prevent accidents on both straight and curved roads.
- Performance Evaluation: The paper evaluates the performance of the V2P-based VRU warning system, focusing on aspects such as communication range, end-to-end latency, and localization accuracy, ensuring its reliability and effectiveness.

This work highlights the potential of C-V2X technology to improve the safety of VRUs through advanced warning systems and innovative hardware solutions.

A key element in the C-V2X landscape is *connected cars*. A connected car is essentially a vehicle equipped with internet access, allowing it to share data with devices both inside and outside the vehicle. This internet connection, typically achieved through mobile data networks, enables a wide range of services that can be remotely managed through smartphones or other devices. To ensure the transmission and reception of information, the car must maintain a constant internet connection, a fundamental feature at the heart of connected car functionality. This capability enables a variety of innovative features that significantly transform our interaction with vehicles [S21].

The core technology behind connected cars involves two main approaches:

- *Embedded Systems:* vehicles come equipped with a pre-installed chipset and antenna, making them internet-ready straight from the factory. This system allows the vehicle to download updates, transmit data, and connect to other devices via built-in Wi-Fi.

- *Tethered Systems:* vehicle’s internet connectivity is provided by the user's smartphone. Although not as integrated as embedded systems, tethered connectivity still enhances the vehicle’s functionality.

Smartphones play a crucial role in enhancing the connected car experience. They provide advanced display technology and intuitive interfaces that are ideal for presenting V2X (Vehicle-to-Everything) information, such as Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs). This integration allows drivers to access real-time traffic information, notifications, and alerts in a user-friendly format. Additionally, smartphones can be customized with various applications and widgets to display relevant V2X data, including navigation updates, traffic conditions, and vehicle status, tailored to individual preferences.

The improved communication and connectivity offered by smartphones are essential. By connecting to cellular networks like 4G or 5G, smartphones can continuously receive and transmit V2X data, providing real-time updates. This connectivity facilitates the seamless integration of live V2X information with other data sources, such as weather forecasts or public transportation schedules. For example, if the V2X system detects a roadblock or traffic jam, the smartphone app can automatically calculate an alternative route by consulting public transportation schedules. This integration not only enhances the driving experience but also ensures drivers are well-informed and can make better decisions.

The use of smartphones in the V2X ecosystem also offers economic benefits and flexibility. Smartphones are frequently updated with modern technologies and features, providing a cost-effective solution for incorporating the latest advancements into vehicle systems. This approach helps bridge the technological gap between vehicle-integrated systems and rapidly evolving consumer electronics. By relying on smartphones for advanced features and updates, automakers can avoid costly frequent updates to vehicle-integrated systems.

From a technical perspective, a dual-component system comprising a vehicle-integrated V2X communication unit (OBU) and a personal portable device (PPD), such as a smartphone, requires careful management of message processing. The OBU handles critical, low-latency V2X communication tasks, while the smartphone processes less time-sensitive data. This distribution of workload ensures that time-sensitive information is processed promptly, while less urgent data is managed by the smartphone. Effective system design must balance the low-latency requirements of V2X communications with the higher latency associated with mobile data networks.

Security and privacy are paramount in integrating smartphones into V2X systems. Protecting V2X communication and personal data transmitted via smartphones requires robust encryption protocols and secure authentication methods to prevent vulnerabilities. Additionally, managing user privacy involves handling location data and personal information transparently and obtaining user consent.

Several examples illustrate the potential of smartphone integration in V2X communication. Platforms such as Android Auto [S22] and Apple CarPlay [S23] demonstrate how smartphones can seamlessly connect with vehicle infotainment systems, providing access to navigation, communication, and entertainment features. This integration highlights how smartphones can enhance V2X communication by combining real-time data with advanced user interfaces.

Always considering C-V2X communication, *Applied Information* is at the forefront of utilizing smartphones as key nodes in intelligent traffic management and road safety. Among its main projects, the *Glance Smart City Supervisor System* [S24] allows cities to centrally manage traffic and ITS resources through a user-friendly web-based application. This cloud-based platform provides timely and relevant information, optimizing ITS investments and integrating with the *TravelSafely* app [S25], which combines Smart City solutions with advancements in connected vehicle technology to create a safer, more informed network. *TravelSafely* connects the user's phone to a network of traffic intersections, school crossing signals, motorists, cyclists, and pedestrians, providing acoustic alerts for hazardous road conditions and using vocal notifications to allow drivers to stay focused on the road while using their preferred navigation app. Additionally, *Glance Preemption [S26]* is designed to enhance road safety and traffic efficiency through traffic signal preemption for emergency vehicles. This system leverages smartphones to enable emergency vehicle drivers to communicate in real-time with the central system via the mobile app mentioned above, displaying signal statuses and optimized routes, and sending the vehicle's location and direction to calculate the most effective signal sequence. Smartphones, through cellular connectivity, ensure real-time updates and rapid decision-making, while collected data is analysed to improve traffic flow and identify potential issues. The integration of smartphones in these systems offers numerous benefits, including increased accessibility and ease of use for emergency personnel, flexibility in software update implementation, and reduced costs associated with dedicated hardware infrastructure.

In the evolving landscape of vehicular communication, C-V2X (Cellular Vehicle-to-Everything) technology represents a significant leap forward by leveraging existing cellular networks to enable robust and efficient communication between vehicles, infrastructure, and pedestrians. The integration of C-V2X technology with modern smartphones enhances this capability by providing advanced connectivity, real-time data processing, and intuitive user interfaces.

The role of smartphones in the C-V2X ecosystem is multifaceted. They facilitate direct and indirect connections between vehicles and infrastructure, support edge and cloud computing for improved data handling, and enable advanced applications that enhance navigation and safety. The potential of smartphones extends to improving the safety of vulnerable road users (VRUs) through innovative solutions like specialized phone cases and collision warning systems. Such integrations highlight the versatility and effectiveness of smartphones in enhancing V2X communication and road safety.

Smartphones offer a cost-effective and flexible solution for incorporating the latest technological advancements into vehicle systems, bridging the gap between rapidly evolving consumer electronics and traditional vehicle technologies. Their ability to provide real-time updates, process vast amounts of data, and integrate seamlessly with existing systems underscores their critical role in modern automotive technology.

The advancements and examples of smartphone integration, such as those demonstrated by Applied Information's *Glance Smart City Supervisor System* and the *TravelSafely* app, illustrate the tangible benefits of this technology. These systems enhance traffic management, optimize emergency response, and improve overall road safety by leveraging the real-time connectivity and data processing capabilities of smartphones.

As we continue to embrace and develop C-V2X technology, the synergy between connected cars and smartphones will play a pivotal role in shaping the future of intelligent transportation systems. By ensuring robust, real-time communication and leveraging the capabilities of modern smartphones, we can look forward to a more connected, safer, and efficient road network that benefits all road users.

In conclusion, the integration of C-V2X technology with smartphones represents a transformative advancement in automotive communication, offering enhanced safety, improved traffic management, and a more connected driving experience. The ongoing innovation in this field promises to further revolutionize how we interact with vehicles and infrastructure, paving the way for a smarter, more responsive transportation ecosystem.

BIBLIOGRAPHY

[1] Maede Fotros, Javad Rezazadeh, and Omid Ameri Sianaki, “A Survey on VANETs Routing Protocols for IoT Intelligent Transportation Systems”, 2020.

[2] Tanya Garg, Gurjinder Kaur, “A Systematic Review on Intelligent Transportation Systems”, 2022.

[3] Felipe Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Viana, Raquel A. F. Mini, Antonio A. F. Loureiro, “Data communication in VANETs: Protocols, applications and challenges”, 2014.

[4] Aneesh Paul, Rohan Chauhan, Rituraj Srivastava, and Mriganka Baruah, “Advanced Driver Assistance Systems”, 2016.

[5] Adam Ziebinski, Rafal Cupek, Damian Grzechca, Lukas Chruszczyk, “Review of Advanced Driver Assistance Systems (ADAS)”, 2017.

[6] Casetti C., “DSRC-based Standards” [slide presentation], Networking Technologies for Connected Vehicles, Politecnico di Torino, 2024.

[7] Ruobing Jiang and Yanmin Zhu, “Wireless Access in Vehicular Environment”, 2019.

[8] ETSI, “ETSI EN 302 663 V1.3.1 (2020-01) – Intelligent Transport System (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band”, 2020.

[9] ETSI, “ETSI TS 102 687 V1.2.1 (2018-04) – Intelligent Transport System (ITS); Decentralized Congestion Control Mechanism for Intelligent Transport Systems operating in the 5GHz range; Access layer part”, 2018.

[10] David Eckhoff, Nikoletta Sofra, and Reinhard German, “A Performance Study of Cooperative Awareness in ETSI ITS-G5 and IEEE WAVE”, 2013.

[11] ETSI, “ETSI EN 302 636-4-1 V1.4.1 (2020-01) - Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality”, 2020.

[12] ETSI, “ETSI TS 102 894-2 V1.3.1 (2018-08) – Intelligent Transport System (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary”, 2018.

[13] Shanzhi Chen, Jinlink Hu, Li Zhao, Rui Zhao, Jiayi Fang, Yan Shi and Hui Xu, “Cellular Vehicle-to-Everything (C-V2X)”, 2023.

[14] Casetti C., “Cellular-based Vehicular Networks C-V2X” [slide presentation], Networking Technologies for Connected Vehicles, Politecnico di Torino, 2024.

[15] Mario H. Castañeda Garcia, Alejandro Molina-Galan, Mate Boban, Javier Gozalvez, Baldomero CollPerales, Taylan Şahin and Apostolos Kousaridas,” A Tutorial on 5G NR V2X Communications”, 2021.

[16] Gaurang Naik, Biplav Choudhury, Jung-Min Park, “IEEE 802.11bd & 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications “, 2019.

[17] Shanzheng Xue, Siyuan Gong,Xinyi Li, “A Comparative Study of IEEE 802.11bd and IEEE 802.11p on the Data Dissemination Properties in Dynamic Traffic Scenarios “, 2024.

[18] Xingqin Lin, “An Overview of 5G Advanced Evolution in 3GPP Release 18”, 2022.

[19] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, Chen Feng, “V2X-Sim: Multi-Agent Collaborative Perception Dataset and Benchmark for Autonomous Driving”, 2022.

[20] F. Raviglione, C.M. Risma Carletti, M. Malinverno, C. Casetti, C.F. Chiasserini, “An integrated multi-stack framework for virtual validation of V2X communication and services”, 2024.

[21] Queck Bjoern Schuenemann, Nico Naumann, David Rieck, Stefan Reichel, Robert Protzmann, Franz Kage, Erik Schlei, Jiajun Hu, “Vehicle-2-X Simulation Runtime Infrastructure (VSimRTI) Tobias”, 2010.

[22] Jason H. Gao, Li-Shiuan Peh, “Automotive V2X on Phones: Enabling next-generation mobile ITS apps”, 2016.

[23] Stefan Diewald, Tim Leinmuller, Boris Atanassow, Lutz-Peter Breyer and Matthias Kranz, “Mobile device integration and interaction with V2X communication”, 2012.

[24] Rishu Chhabra, C. Rama Krishna, Seema Verma, “Augmenting Driver’s Situational Awareness using Smartphones in VANETs “, 2021.

[25] Roya Bastani Zadeh, Mehdi Ghatee, Hamid Reza Eftekhari, "Three-Phases Smartphone-Based Warning System to Protect Vulnerable Road Users Under Fuzzy Conditions", 2017.

[26] Changlong Zhang, Jimin Wei, Shibo Qu, Changying Huang, Jingang Dai, Peipei Fu, Zhixiong Wang, and Xinquan Li, "Implementation of a V2P-Based VRU Warning System With C-V2X Technology", 2023.

SITOGRAPHY

[1] https://www.sae.org/blog/sae-j3016-update
[2] https://www.sae.org/standards/content/j3016_202104/
[3] http://www.adasitalia.com/
[4] https://www.everythingrf.com/community/what-is-lte-v2x-pc5
[5] https://www.3gpp.org/technologies/sa5-r17
[6] https://www.3gpp.org/technologies/rel-18
[7] https://www.3gpp.org/specifications-technologies/releases/release-19
[8] https://sumo.dlr.de/docs/index.html
[9] *https://www.nsnam.org/*
[10] *https://simulte.omnetpp.org/*
[11] *https://veins.car2x.org/*
[12] https://www.arterychip.com/en/index.jsp
[13] *https://publichealth.jmir.org/2021/11/e25897/*
[14] *https://www.mylondon.news/news/health/number-accidents-involving-ambulances-london-15615134*
[15] *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3503424/*
[16] *https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812169*
[17] https://omnetpp.org/intro/
[18] *https://www.bluetooth.com/blog/bluetooth-use-cases-for-v2x-automotive-applications/*
[19] https://hondanews.com/en-US/releases/honda-demonstrates-advanced-vehicle-to-pedestrian-and-vehicle-to-motorcycle-safety-technologies
[20] *https://de-he.step.vodafone.com/*
[21] https://www.wejo.com/resources/what-is-a-connected-car#
[22] https://www.android.com/intl/it_it/auto/
[23] https://www.apple.com/it/ios/carplay/
[24] https://appinfoinc.com/solutions/glance-smart-city-system/
[25] https://appinfoinc.com/solutions/glance-travelsafely/
[26] https://appinfoinc.com/solutions/preemption-priority/