



Politecnico  
di Torino

ScuDo

Scuola di Dottorato - Doctoral School  
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Computer and Control Engineering (37<sup>th</sup> cycle)

# **Vehicular Micro-Clouds in 5G: Architectures and applications**

By

**Carlos Mateo Risma Carletti**

\*\*\*\*\*

**Supervisor(s):**

Prof. Claudio Ettore Casetti, Supervisor

Prof. Jérôme Härrı, Co-Supervisor

Prof. Fulvio Risso, Co-Supervisor

**Doctoral Examination Committee:**

Prof. Maria Luisa Merani, Università degli Studi di Modena e Reggio Emilia

Ion Turcanu PhD, Luxembourg Institute of Science and Technology

Prof. Roberto Riggio, Università Politecnica delle Marche

Prof. Paolo Giaccone, Politecnico di Torino

Prof. Guido Marchetto, Politecnico di Torino

Politecnico di Torino

2025

## **Declaration**

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Carlos Mateo Risma Carletti  
2025

\* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

*I would like to thank my advisors, Prof. Claudio Casetti, Prof. Jérôme Härrri, and Prof. Fulvio Rizzo, for their encouragement, support and guidance during my research endeavors in my doctoral program.*

*To my mother, the one who from day one has always believed in me and has always given me her unconditional support, without her I could not have achieved any of this and to her I owe everything.*

*I also want to express my gratitude to my family and friends, whose constant support and presence have been a source of comfort and joy during both challenging and rewarding moments of this journey.*

*Finally, I extend my heartfelt thanks to my colleagues at DAUIN and EURECOM. It has been a privilege to collaborate with them, and their dedication and teamwork have been instrumental in achieving many research milestones.*

## **Abstract**

From the invention of the wheel to the dominance of internal combustion engines and now the advent of electric vehicles (EVs), technology has consistently shaped transportation. In recent decades, mobility innovation has expanded its focus beyond individual vehicles toward system-level intelligence powered by advancements in communication technologies, onboard sensors, and edge computing. At the core of this evolution lies the concept of Cooperative Intelligent Transport Systems (C-ITS), which enables seamless data exchange between vehicles and infrastructure to enhance cooperation and safety. Building on this foundation, the paradigm of Cooperative, Connected, and Automated Mobility (CCAM) has emerged to integrate connectivity, automation, and collaboration into modern transportation systems, improving safety, optimizing traffic flow, and reducing environmental impact.

However, this vision brings its own challenges, particularly in ensuring real-time communication and computation for safety-critical applications like collision avoidance and coordinated maneuvers. When hosting such vehicular application services on traditional cloud computing architectures, challenges arise due to latency, bandwidth constraints, and operation costs. Recognizing these limitations, the Multi-access Edge Computing (MEC) paradigm has emerged as a pivotal enabler, relocating computational resources closer to vehicles at the edge of the network. Nevertheless, the growth of data generated by modern perception systems and the highly dynamic nature of vehicles may still pose challenges for MEC-based services needing to serve vehicles roaming under changing network operators or travelling under limited network coverage.

This thesis explores vehicular micro-clouds as a transformative approach to address these challenges. By leveraging vehicle-to-vehicle (V2V) communication, vehicles may form dynamic, localized clusters capable of aggregating, caching, and processing data at the edge. Acting as virtual edge servers, these clusters can

reduce dependency on centralized resources while enhancing system-wide scalability, responsiveness, and cost efficiency. The work presented in this dissertation explores the architectures, protocols, and applications enabled by vehicular micro-clouds, with a particular emphasis on their potential to support CCAM use cases.

The base methodology of this research lies, first, in the development of innovative simulation tools like the `ms-van3t` framework and its integration with the CARLA simulator. This co-simulation framework allows for high-fidelity modeling of vehicular scenarios, including the transmission and processing of Cooperative Perception Messages (CPMs) and the validation of architectures and applications designed for vehicular micro-clouds.

Two key contributions then anchor this research. The first is the design and validation of the Server Local Dynamic Map (S-LDM), a MEC-based middleware service that integrates real-time data from vehicles and infrastructure to maintain a centralized yet localized map of the road environment. This service, evaluated for its scalability and cross-border performance, demonstrates how MEC can bridge gaps in latency-sensitive applications. The second contribution is the development of the Platoon-Local Dynamic Map (P-LDM), a vehicular micro-cloud service tailored for cooperative perception among platooning vehicles. By synchronizing data from multiple platoon members, P-LDM enhances situational awareness and ensures robust performance in dynamic environments. Through extensive simulations, this work demonstrates the benefits of vehicular micro-clouds in reducing latency, optimizing resource utilization, and improving safety-critical applications like automated platooning maneuvers.

Finally, this thesis aims to contribute to a scalable, efficient, and reliable framework for future mobility systems. Vehicular micro-clouds, as envisioned here, represent a significant step toward realizing C-ITS providing an additional underlying layer of virtualized resources for the reliable and efficient execution of CCAM applications.

# Contents

<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Publications . . . . .	3
1.2 Main Contributions . . . . .	4
1.3 Outline . . . . .	7
<b>2 Communication technologies for Cooperative, Connected and Automated Mobility (CCAM)</b>	<b>9</b>
2.1 DSRC-based Protocols . . . . .	15
2.1.1 WAVE . . . . .	15
2.1.1.1 Transport and Network Layers . . . . .	16
2.1.1.2 Data-Link layers . . . . .	17
2.1.1.3 802.11p . . . . .	17
2.1.2 ETSI C-ITS . . . . .	18
2.1.2.1 Application and Facilites layers . . . . .	20
2.1.2.2 Networking and Transport layers . . . . .	26
2.1.2.3 Access layers . . . . .	27
2.1.3 IEEE 802.11bd . . . . .	27

---

2.2	Cellular-based Protocols . . . . .	29
2.2.1	LTE-V2X . . . . .	31
2.2.2	NR-V2X . . . . .	33
2.2.2.1	Mode 1 (Centralized Allocation) . . . . .	35
2.2.2.2	Mode 2 (Distributed Allocation) . . . . .	35
2.3	ETSI MEC . . . . .	37
2.4	AMQP 1.0 for IP-based V2X communications . . . . .	40
2.5	Vehicle Micro Clouds . . . . .	43
<b>3</b>	<b>Development of performance evaluation tools for CCAM</b>	<b>48</b>
3.1	The ms-van3t simulation framework . . . . .	51
3.2	ms-van3t features and architecture . . . . .	53
3.2.1	Multi-stack . . . . .	55
3.2.2	Large scale simulations . . . . .	56
3.2.3	Native integration with SUMO . . . . .	57
3.2.4	Pre-recorded GNSS traces . . . . .	58
3.2.5	Emulation Mode . . . . .	59
3.2.6	ETSI C-ITS stack . . . . .	60
3.2.6.1	Facilities Layer . . . . .	60
3.2.6.2	Basic Transport Protocol . . . . .	62
3.2.6.3	GeoNetworking . . . . .	62
3.3	Evaluation and results . . . . .	63
3.3.1	Performance comparison between V2X access technologies	63
3.3.2	Performance evaluation of emulation capabilities . . . . .	69
3.4	Extending ms-van3t for realistic perception and automation modelling with CARLA simulator . . . . .	74
3.5	Architecture of ms-van3t's CARLA integration . . . . .	75

3.5.1	OpenCDA Modules . . . . .	76
3.5.1.1	LDM . . . . .	76
3.5.1.2	OpenCDA Control Interface . . . . .	78
3.5.2	ms-van3t modules . . . . .	79
3.5.2.1	OpenCDA Client . . . . .	79
3.5.2.2	OpenCDA VDP . . . . .	79
3.5.2.3	OpenCDA Sensor . . . . .	80
3.6	Simulation Setup . . . . .	81
<b>4</b>	<b>Leveraging Connected Vehicle Resources through MEC-Based Services</b>	<b>83</b>
4.1	Review on LDM concept and centralized implementations . . . . .	84
4.2	Server-Local Dynamic Map for enhanced collective perception . . . . .	85
4.3	Service description and deployment architecture . . . . .	87
4.3.1	S-LDM description . . . . .	87
4.3.2	Deployment and scalability options . . . . .	91
4.4	On-road evaluation . . . . .	94
4.5	Scalability analysis . . . . .	98
4.5.1	Real-time data storage scalability . . . . .	100
4.5.2	Data provision to other services scalability . . . . .	104
<b>5</b>	<b>Leveraging Connected Vehicle Resources through Micro Cloud Services</b>	<b>107</b>
5.1	Platoon-Local Dynamic Map: micro-cloud service for cooperative perception . . . . .	109
5.2	System Model . . . . .	110
5.2.1	System overview . . . . .	110
5.2.2	Message Protocols . . . . .	113
5.2.3	Object data processing . . . . .	115
5.2.4	Quantifying object perception . . . . .	116

---

5.3	Preliminary service evaluation . . . . .	118
5.3.1	Simulation setup . . . . .	120
5.3.2	Preliminary results . . . . .	121
5.4	Assignment Problem Formulation . . . . .	124
5.5	Proposed Heuristic Algorithms . . . . .	127
5.5.1	Performance Evaluation . . . . .	131
5.5.2	Scalability . . . . .	133
5.6	Service Validation . . . . .	136
5.6.1	Simulation setup . . . . .	137
5.6.2	Perception computational load analysis . . . . .	140
5.6.3	Perception robustness . . . . .	144
5.6.4	Platooning awareness . . . . .	145
<b>6</b>	<b>Conclusions</b>	<b>149</b>
	<b>References</b>	<b>153</b>

# List of Figures

2.1	FCC spectrum allocation for the 5.9GHz band. . . . .	12
2.2	EU spectrum allocation for the 5.9 GHz band. . . . .	12
2.3	WAVE protocol stack [17]. . . . .	16
2.4	ITS station protocol stack. . . . .	19
2.5	CPM structure defined in [77]. . . . .	24
2.6	Time-frequency allocation in LTE-V2X channels, featuring the neighboring Physical Sidelink Control Channel and Physical Sidelink Shared Channel. . . . .	31
2.7	Sensing-based Semi Persistent Scheduling Scheme in NR-V2X. . .	36
2.8	MEC architecture according to [82][83]. . . . .	39
2.9	Architecture for the exchange of C-ITS messages over AMQP for V2X services in a multi-operator scenario. . . . .	41
2.10	C-ITS message encapsulation scheme as a payload of AMQP 1.0 packets. . . . .	42
2.11	Vehicular Micro Cloud (VMC) architecture. The Cluster Head (CH) and Cluster Members (CMs) aggregate and virtualize their resources through the VMC platform for the deployment of VMC applications. External Vehicles may subscribe to VMC applications without the need to connect to a centralized server (e.g., a MEC server). . . . .	45
3.1	Comparison of ms-van3t features with existing vehicular network simulation and emulation solutions. . . . .	53

---

3.2	Architecture of MS-VAN3T, featuring simulation and emulation capabilities alongside four state-of-the-art access layer models. . . .	54
3.3	One-way latency as a function of the vehicle spatial density. IEEE 802.11p, LTE-V2X, and NR-V2X show a stable and low latency. Vehicles communicating with LTE, instead, experience a higher latency, reaching 390 ms under very high traffic congestion. . . . .	66
3.4	PRR as a function of the vehicle density, for a baseline distance of 150 m and transmission power level of 26 dBm. . . . .	67
3.5	PRR as a function of the baseline distance from the sender, with at the top the zoom in of the region between 0.97 and 1, where the advantages of using cellular-based technologies are evident. Vehicle density: 20 veh/km. . . . .	68
3.6	Laboratory setup for the interoperability tests, performed to validate the emulation feature available in ms-van3t. . . . .	70
3.7	Communication between ms-van3t (configured to emulate one vehicle with stationID equal to 1) and a commercial ETSI C-ITS stack (configured to emulate a vehicle with stationID 699702254). . . . .	70
3.8	Difference between the achieved PPS on the Cohda stack and the expected PPS, for different CPUs and as a function of the total number of vehicles emulated by ms-van3t. . . . .	72
3.9	Architecture of ms-van3t-CARLA. . . . .	75
3.10	LDM representation over the LiDAR view offered in OpenCDA, showcasing the different types of perceptions according to CAM, CPM or local detection. . . . .	81
4.1	S-LDM service internal architecture. . . . .	86
4.2	The S-LDM as part of the 5G-CARMEN architecture, in turn based on C-Roads standards. . . . .	91
4.3	Multiple S-LDM architecture for cross-border and in-country scalability. . . . .	92

4.4	The path followed by one of the two Stellantis vehicles during the cross-border road tests from Austria to Italy is shown. Messages received by the S-LDM from the TIM network via the AMQP broker are represented in blue, whereas those received from the MTA network via its corresponding broker are represented in red. . . . .	95
4.5	Normalized message delay (i.e., the delay between the transmission of messages from vehicles and their reception by the S-LDM, adjusted by subtracting the minimum delay observed during the entire test) over the duration of a single test run for a single vehicle. The S-LDM instance deployed on the TIM MEC platform has been used as the reference point. . . . .	96
4.6	Cumulative Distribution Function of the time between consecutive database updates for each of the Maserati vehicles during the most significant Austria-to-Italy cross-border test. . . . .	97
4.7	Cumulative Distribution Function of the message processing times for each of the Maserati vehicles during the most significant Austria-to-Italy cross-border test. . . . .	98
4.8	Laboratory testbed setup for S-LDM scalability analysis. . . . .	100
4.9	(Top): CPU consumption of S-LDM and AMQP broker pods in milliunits as a function of the number of vehicles being emulated. (Bottom): <i>Instantaneous Update Rate</i> of a reference vehicle (i.e., reference stationID) as the total number of vehicles managed by a single S-LDM instance increases. . . . .	102
4.10	(a) CPU consumption of S-LDM pods, measured in milliunits, with respect to the number of vehicles requesting context information and the number of vehicles located within the context range of each requesting vehicle. (b) <i>Instantaneous update periodicity</i> of a reference vehicle (i.e., reference stationID), with respect to the number of requesting vehicles managed by a single S-LDM instance and different numbers of vehicles in the triggered context. . . . .	105

---

5.1	Scheme showcasing multiple platooning vehicles detecting and broadcasting redundant sensing information about the same non-connected vehicle. . . . .	110
5.2	P-LDM architecture and perception aggregation scheme. . . . .	112
5.3	P-LDM protocol flowchart. . . . .	113
5.4	P-LDM object matching and ID synchronization scheme. . . . .	116
5.5	Cumulative Distribution Function of the Number of Perceived Objects stored in the LDM at a given point in time. . . . .	122
5.6	Cumulative Distribution Function of the Average Number of Perceived Objects handled by each PM, in the LDM at a given point in time. . . . .	122
5.7	Cumulative Distribution Function of the Average Platoon Perception Duration of Objects and the Average Age of Perceived Objects in the LDM. . . . .	123
5.8	Optimization cost achieved by each algorithm for each of the considered costs (a) CPU cost, (b) robustness cost, (c) fairness cost and (d) weighted average of the three costs in Equation (5.18) with equal weights. . . . .	132
5.9	Execution time of each algorithm for a scenario with 10 Platoon Members and 25 Platoon Perceived Objects. . . . .	133
5.10	Average execution time of each algorithm for 10 Platoon Members and an increasing number of Platoon Perceived Objects. . . . .	134
5.11	Weighted average of PPO assignment cost of each algorithm for 10 Platoon Members and an increasing number of Platoon Perceived Objects. . . . .	134
5.12	Weighted average of PPO assignment cost of each algorithm for an increasing number of Platoon Members, where the number of Platoon Perceived Objects is twice the number of Platoon Members. . . . .	135
5.13	Average execution time of each algorithm for an increasing number of Platoon Members, where the number of Platoon Perceived Objects is twice the number of Platoon Members. . . . .	136

5.14	Architecture of the ms-van3t-CARLA framework leveraged in our validation simulations. . . . .	137
5.15	Scenario timeline showing the different phases of the considered simulation scenario: 1. Normal Platooning state (PMs colored in blue and background vehicles colored in green). 2. Intruder vehicle (colored in red) approaches the platoon. 3. Intruder vehicle starts the cut-in maneuver between two PMs. 4. PMs react to the intruder interference by slowing down to create a safe gap. 5. The platoon recovers its normal state, adjusted to the presence of the intruder. . .	140
5.16	Average processing time of each received message (CPMs) for each number of Platoon Perceived Objects included in each message. . .	141
5.17	Cumulative Distribution Function of the number of Platoon Perceived Objects being handled, i.e., being processed for each CPM reception. . . . .	142
5.18	Cumulative Distribution Function of the total number of Platoon Perceived Objects stored in the database at a given point in time. . .	142
5.19	Cumulative Distribution Function of the average Intersection Over Union (IoU) of all objects stored in the database at a given point in time. . . . .	143
5.20	Instantaneous speed of each Platoon Member over time in m/s. The green dashed line at 20 m/s represents the desired platoon speed. The grey background signals the presence of the intruder within the platoon. . . . .	146
5.21	Average member speed difference with platoon leader at intruder cut-in, average member inter-distance at intruder cut-in and average Time To Collision between members at intruder cut-in. . . . .	148

# List of Tables

3.1	Main simulation parameters for the evaluation of the network KPIs.	64
-----	--	----

# Chapter 1

## Introduction

The rapid advancements in onboard perception systems in modern vehicles have unlocked an unprecedented wealth of contextual data to be exploited. This data is immensely valuable, particularly for traffic safety-related applications that represent the primary goal for Cooperative Intelligent Transportation Systems (C-ITS) development and the broader vision of Cooperative, Connected, and Automated Mobility (CCAM). CCAM represents a revolutionizing approach to transportation, where vehicles, infrastructure, and road users work collaboratively through seamless communication and automation, aiming to improve safety and optimize traffic flow. These applications rely heavily on up-to-date information to ensure the correct functioning of their algorithms, as even slight delays in accessing this data can compromise their effectiveness. While the growth of the number of communication-enabled vehicles offers the opportunity to share this data with nearby vehicles and infrastructure, the latency and costs associated with transferring large volumes of data to and from the cloud render the exclusive reliance on centralized cloud-based systems impractical for safety-critical applications.

To address these challenges, the Multi-access Edge Computing (MEC) paradigm has emerged as a pivotal enabler of low-latency and high-speed connectivity. By bringing storage and computational resources closer to vehicles at the edge of the Radio Access Network, MEC facilitates the deployment of latency-sensitive applications under the specifications provided by the European Telecommunications Standards Institute (ETSI). This paradigm empowers vehicles to offload computationally intensive tasks to edge servers, reducing latency and enhancing the responsiveness

of various ITS applications. However, the introduction of MEC is not without challenges. The vast amounts of data generated by onboard perception systems impose a significant load on radio access resources, leading to potential performance bottlenecks. Furthermore, much of the data captured by vehicles not only has a short validity lifetime but is also contextually relevant only within their immediate vicinity, making the channel load imposed by exchanging these data with a MEC server of limited benefit.

Amid these challenges, the concept of vehicular micro-clouds has emerged as a promising solution. These micro-clouds, formed by clusters of vehicles leveraging Vehicle-to-Vehicle (V2V) communications, enable the aggregation, caching, and processing of data locally, reducing the burden on centralized servers. By acting as virtual edge servers, vehicular micro-clouds can provide services not only to their member vehicles but also to external edge servers that might be overloaded with tasks. This decentralized approach introduces a range of benefits, including reduced communication costs, enhanced scalability, and increased computational efficiency.

The evaluation of vehicular micro-cloud solutions, however, requires robust and realistic simulation tools capable of modeling the complexities of vehicular environments. Simulation tools are indispensable for analyzing the performance of communication protocols and task distribution algorithms under diverse scenarios. They provide a controlled environment to test and refine solutions, enabling researchers to measure critical metrics such as latency, reliability, and resource utilization benefits before real-world deployment.

Nonetheless, the deployment of vehicular micro-clouds comes with its own set of challenges. The inherent mobility of vehicles creates unstable wireless links and unpredictable residency times within clusters, posing difficulties in maintaining the reliability and continuity of services. Effective clustering strategies are essential to optimize parameters such as cluster lifetime, available computational resources, and the number of participating vehicles. One particularly promising application CCAM that could benefit from vehicular micro-clouds capabilities is platooning, where a fleet of vehicles travels in close coordination to enhance fuel efficiency, reduce traffic congestion, and improve road safety. Platooning exemplifies the potential of vehicular micro-clouds as it inherently involves stable inter-vehicle distances and operates predominantly on motorways, which are ideal environments for such architectures. Furthermore, the high speeds and frequent handovers required in

cellular network-based communication during platooning underscore the need for localized and efficient data management solutions.

This thesis is driven by the motivation to explore and address the challenges and opportunities associated with vehicular micro-clouds. By developing novel architectures and applications tailored to this paradigm, the work aims to contribute to the realization of efficient, scalable, and reliable vehicular micro-cloud systems that can enhance the safety, efficiency, and intelligence of future transportation systems.

## 1.1 Publications

The following is a list of publications resulting from this research work:

1. F. Raviglione, C.M.R. Carletti, C. Casetti, F. Stoffella, G.M. Yilma, F. Visintainer, “S-LDM: Server Local Dynamic Map for Vehicular Enhanced Collective Perception,” *2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring)*, pp. 1-5, 2022.  
Available at: <https://doi.org/10.1109/VTC2022-Spring54318.2022.9860701>.
2. G. La Bruna, C.R. Carletti, R. Rusca, C. Casetti, C.F. Chiasserini, et al., “Edge-Assisted Federated Learning in Vehicular Networks,” *2022 18th International Conference on Mobility, Sensing, and Networking (MSN)*, 2022.  
Available at: <https://doi.org/10.1109/MSN57253.2022.00038>.
3. C.M.R. Carletti, C. Casetti, J. Härri, F. Risso, “Platoon-Local Dynamic Map: Micro Cloud Support for Platooning Cooperative Perception,” *2023 19th International Conference on Wireless and Mobile Computing (WiMob)*, 2023.  
Available at: <https://doi.org/10.1109/WiMob58348.2023.10187883>.
4. F. Raviglione, C.M.R. Carletti, M. Malinverno, C. Casetti, C.F. Chiasserini, “ms-van3t: An Integrated Multi-Stack Framework for Virtual Validation of V2X Communication and Services,” *Computer Communications*, vol. 217, pp. 70-86, 2024.  
Available at: <https://doi.org/10.1016/j.comcom.2024.01.022>.

5. C.M.R. Carletti, C. Casetti, J. Härri, F. Risso, “ms-van3t-CARLA: An Open-Source Co-Simulation Framework for Cooperative Perception Evaluation,” *2024 19th Wireless On-Demand Network Systems and Services Conference (WONS)*, 2024.

Available at: <https://doi.org/10.23919/WONS60642.2024.10449533>.

6. C.M.R. Carletti, F. Raviglione, C. Casetti, F. Stoffella, G.M. Yilma, F. Visintainer, “S-LDM: Server Local Dynamic Map for 5G-Based Centralized Enhanced Collective Perception,” *Vehicular Communications*, vol. 49, article no. 100819, 2024.

Available at: <https://doi.org/10.1016/j.vehcom.2024.100819>.

## 1.2 Main Contributions

This Section summarizes the main contributions of this thesis:

- **Chapter 2.** Introduction to existing protocols for Cooperative, Connected, and Automated Mobility (CCAM), including an overview of the access technologies based on both DSRC (IEEE 802.11p and ETSI ITS-G5) and Cellular (LTE-V2X and NR-V2X) communications. In addition to the available access technologies, an overview of the communication stacks, namely WAVE and C-ITS, is provided, detailing the layers and protocols specifically developed to build on these technologies and tailored to meet the demands of V2X communication. Focusing on the European landscape, the discussion extends beyond the V2V-centered C-ITS architecture to include architectures designed to enhance V2I and V2N communications, with detailed coverage of the ETSI MEC framework and the IP-based protocols that leverage it.
- **Chapter 3**
  1. Design, development, and validation of ms-van3t, an open-source simulation and emulation framework. This innovative tool introduces advanced features that set it apart from existing solutions, such as the ability to utilize pre-recorded GNSS traces in place of traditional mobility models, and supports the simulation of large-scale V2X scenarios adhering to

ETSI C-ITS standards. The framework is utilized for the performance evaluation of V2X access technologies available, specifically testing models such as LTE, IEEE 802.11p, LTE-V2X, and NR-V2X, with a focus on latency and Packet Reception Ratio (PRR) metrics. Additionally, the emulation feature is validated in terms of its interoperability with commercial protocol stacks and its scalability capabilities for Hardware-In-the-Loop (HIL) applications.

2. Design, development, and validation of an extension of the ms-van3t simulation framework with the CARLA simulator, enabling high-fidelity modeling of vehicular scenarios. This integration provides an implementation of key C-ITS services for cooperative perception, the Local Dynamic Map, and the Collective Perception Service. Notably, ms-van3t is currently the only simulation framework to integrate the ns-3 network simulator [4] with the CARLA simulator [63]. Furthermore, ms-van3t is the only simulation framework that integrates sensor data generated within the CARLA simulator to create and exchange simulated Collective Perception Messages (CPMs) within a network simulation environment.

- **Chapter 4.**

1. Design and development of a 5G-enabled MEC service, called the Server Local Dynamic Map (S-LDM). Acting as a middleware entity, the S-LDM integrates real-time and historical data from all detected vehicles and non-connected objects in a defined area, maintaining a centralized, dynamic map of the road. Automated maneuver management services utilizing 5G MEC infrastructure typically require only a subset of pre-processed, context-specific information from the broader pool of data shared through C-ITS messages. This localized context encompasses only data about vehicles and non-connected objects involved in the considered maneuvers. For such services, it is crucial to ensure that the context knowledge is consistently updated, with low latency, and that computational demands of processing raw vehicle data at the MEC level are minimized. The S-LDM efficiently filters and processes data to provide relevant information to MEC applications only under specific road-triggering conditions, such as when a vehicle initiates a centralized

lane merge. By focusing on pre-processed and filtered data, S-LDM reduces latency and computational overhead.

2. Validation and evaluation of the S-LDM scalability and cross-border performance, showcasing its ability to operate effectively under high-density traffic and across multiple operator domains. These findings emphasize the importance of scalable and interoperable designs in real-world CCAM deployments.

- **Chapter 5.**

1. Design and development of the Platoon-Local Dynamic Map (P-LDM), a vehicular micro-cloud service designed to facilitate cooperative perception and enhance data-sharing efficiency among platooning vehicles. The P-LDM aggregates and synchronizes perception data from platoon members, ensuring a unified and up-to-date view of the surrounding environment. This includes the detection and tracking of both V2X-enabled and non-V2X objects, leveraging onboard sensors to extend situational awareness beyond individual vehicles.
2. Formulation of the platoon perceived object assignment optimization problem within the P-LDM service. This problem is modeled to assign perceived objects to platoon members efficiently, minimizing computational overhead while maximizing perception accuracy and fairness. Two heuristic algorithms are proposed to address the online computation of Platoon Perceived Object assignments. Their effectiveness is demonstrated against an adaptation of the NSGA-II [62] genetic algorithm used as a baseline for evaluating solution optimality. The NSGA-II algorithm provides a more computationally intensive but robust approach, serving as a reference to assess the trade-offs in the greedy algorithm's performance. Their effectiveness is evaluated through extensive simulations, which highlight their capability to maintain low latency and high reliability, as well as the ability to manage large numbers of vehicles and perceived objects.
3. Implementation and validation of the P-LDM leveraging the ms-van3t simulation framework, harnessing the powerful capabilities of the integration with the CARLA simulator. This implementation allows for a

highly realistic validation of the service, providing insights into its functionality under realistic simulated conditions. The evaluation focuses on two critical aspects: the accuracy and reliability of the data gathered by the P-LDM and the efficiency of computational load distribution among platoon members. A comprehensive evaluation compares the P-LDM with the conventional fully distributed scheme for cooperative perception devised by ETSI.

4. Lastly, the effects of cooperative perception on platooning maneuver execution are thoroughly evaluated. By utilizing the realistic control dynamics provided by CARLA, the field-tested ENSEMBLE platooning protocol [49] has been implemented within the ms-van3t framework. This setup enables an in-depth analysis of how enhanced situational awareness influences the execution of complex platooning maneuvers, particularly in maintaining stability and safety during dynamic interactions with non-platooning vehicles.

## 1.3 Outline

The remainder of this thesis is organized as follows:

- Chapter 2 presents an overview of key CCAM protocols and technologies, including DSRC (IEEE 802.11p and ETSI ITS-G5) and cellular-based solutions (LTE-V2X and NR-V2X). Communication stacks, such as WAVE and C-ITS, are detailed, emphasizing their relevance to V2X applications. The chapter also explores European-centric architectures for V2I and V2N communications, including the ETSI MEC framework.
- Chapter 3 discusses the design and validation of ms-van3t, an open-source simulation and emulation framework providing ETSI C-ITS standard compliance on top of a comprehensive set of access technologies for vehicular communications. The extension of ms-van3t with CARLA is highlighted, enabling high-fidelity cooperative perception simulations and the integration of Collective Perception Messages (CPMs).
- Chapter 4 introduces the Server Local Dynamic Map (S-LDM), a 5G-enabled MEC service that maintains a centralized dynamic map of the road envi-

ronment. By pre-processing and filtering context-specific data, the S-LDM supports low-latency MEC applications. Its scalability and cross-border performance are evaluated, demonstrating its effectiveness in high-density traffic and multi-operator scenarios.

- Chapter 5 focuses on the design and validation of the Platoon-Local Dynamic Map (P-LDM), a micro-cloud service for cooperative perception in platooning. It introduces optimization algorithms for object assignment and evaluates the P-LDM's implementation using ms-van3t with CARLA. The chapter also examines the effects of cooperative perception on platooning maneuvers, leveraging the ENSEMBLE protocol to assess stability and safety during dynamic interactions.
- Chapter 6 presents the conclusions and main take-away messages.

## **Chapter 2**

# **Communication technologies for Cooperative, Connected and Automated Mobility (CCAM)**

Since the beginning of the automotive industry, car makers have allocated a significant portion of their research and development efforts to improving the safety of vehicles. From the introduction of built-in seat belts, followed by the widespread deployment of airbags, to the integration of radar sensors for automated emergency braking, emerging technologies have subsequently been leveraged to provide safety features. Before such features are released and become standard for new vehicles, a considerable amount of time needs to be invested in investigation and testing.

Over the past twenty years, research and development efforts have been made towards the realization of Cooperative, Connected, and Automated Mobility (CCAM), paving the way for an extensive set of communications technologies and protocols. This paradigm empowers road safety and traffic efficiency services for Cooperative Intelligent Transport Systems (C-ITS). Indeed, traffic efficiency implicitly benefits road safety, which is why in Europe, automotive industries (through the CAR2CAR Communication Consortium [1]), road operators (through the C-Roads platform [2]), and academic researchers have been working together for the standardization of all C-ITS aspects.

As described in [58], in order to achieve an acceptable level of safety for autonomous vehicles, the information from onboard sensors is not enough, and thus,

an exchange of information is needed between vehicles. For such exchange of information, Vehicle-to-Everything (commonly referred to as V2X) communications constitute the main enabler, providing fast and direct exchange of messages between vehicles. Indeed, the wireless communication for vehicles to share information is not only to other vehicles (Vehicle-to-Vehicle) but also to other road users or road operators and traffic managers. As defined by [76], the main types of communications encompassing V2X are the following:

- **Vehicle-to-Vehicle (V2V):** represents the direct exchange of messages between vehicles over a common wireless channel. Providing the lowest latency, this type of communication is often used for safety-critical use-cases such as collision avoidance applications.
- **Vehicle-to-Infrastructure (V2I):** represents the direct exchange of messages between vehicles and a Road-Side Unit (RSU) or a Base Station (eNB/gNB). V2I generally implies the exchange of messages for the execution of a service running locally at the RSU/eNB/gNB, managed by a road operator. The main types of applications relying on this type of communication are related to mandatory and advisory road signage, road hazard warnings, and traffic light maneuvers.
- **Vehicle-to-Network (V2N):** represents the exchange of messages between vehicles and a remote server. This type of communication is mostly used for traffic surveillance and computation offloading usecases.
- **Vehicle-to-Pedestrian (V2P):** represents the exchange of messages between vehicles and pedestrians over a common wireless channel. This sub-type considers the communication not only with pedestrians but it extends to Vulnerable Road Users (e.g. cyclists). Providing the same latency performance as V2V communications, most application relying on V2P communications are related to collision avoidance as well.

There are two main families of technologies for the execution of V2X communications at the time of writing:

- Dedicated Short Range Communications (DSRC), with standards mainly developed by IEEE, including the WAVE set of standards for the US, and

---

ETSI, including the ITS-G5 set of standards for the European market. Both sets of standards leverage the IEEE 802.11p standard for their Physical and MAC layers.

- Cellular-based standards, often referred to as Cellular-V2X (C-V2X), developed by 3GPP from the introduction of the PC5 interface for V2V communications in Release 14 [25]. Several standards have been released by ETSI for the interoperability between C-V2X and ITS-G5 as access layers for the C-ITS protocol stack.

For the reliable execution of V2X communications, organizations responsible for managing the radio spectrum have recognized the importance of allocating exclusive bandwidth to support the development of Intelligent Transportation Systems (ITS). In 1999, the U.S. Federal Communications Commission (FCC) allocated 75 MHz within the 5.9 GHz band for Dedicated Short-Range Communication (DSRC). The allocated spectrum spanned from 5850 MHz to 5925 MHz and was divided into seven 10 MHz channels (from channel 172 up to channel 184), recommended for ITS use, with a 5 MHz guard band at the lower end, as depicted in Figure 2.1. Over the years, follow-up amendments were issued to refine the regulations governing this allocation. Specifically, channel 178 was designated as Control Channel, while channels 174, 176, 180, and 182 were assigned as Service Channel for non-safety applications. Lastly, channels 172 and 184 were reserved as Service Channels for safety applications. In November 2020, the FCC unanimously approved a First Report and Order that reallocated a significant portion of the spectrum. Specifically, the lower 45 MHz of the band (5850–5895 MHz) was repurposed for unlicensed commercial uses, such as Wi-Fi. As a result, only the upper 30 MHz of the spectrum (5895–5925 MHz) is still designated for Intelligent Transportation Systems (ITS) applications [3].

In 2010, the European framework for ITS was established following decisions by the Electronic Communications Committee (ECC), outlined in [117]. Based on these decisions, ETSI defined technical specifications in [67] for allocating the 5.9 GHz frequency band for ITS applications. As depicted in Figure 2.2, the 5855–5875 MHz range was designated for ITS non-safety applications, the 5875–5905 MHz range for ITS safety-critical applications, and the 5905–5925 MHz range was reserved for future ITS developments.

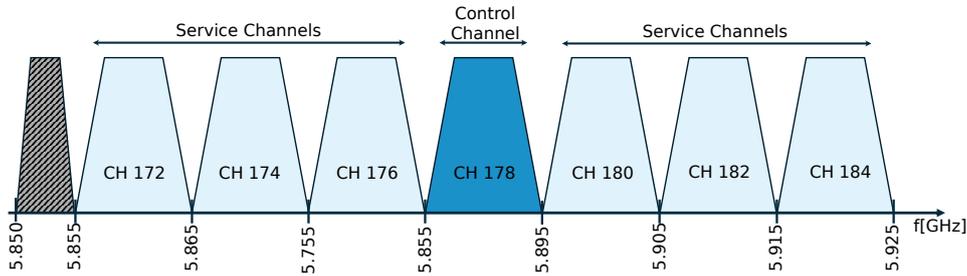


Fig. 2.1 FCC spectrum allocation for the 5.9GHz band.

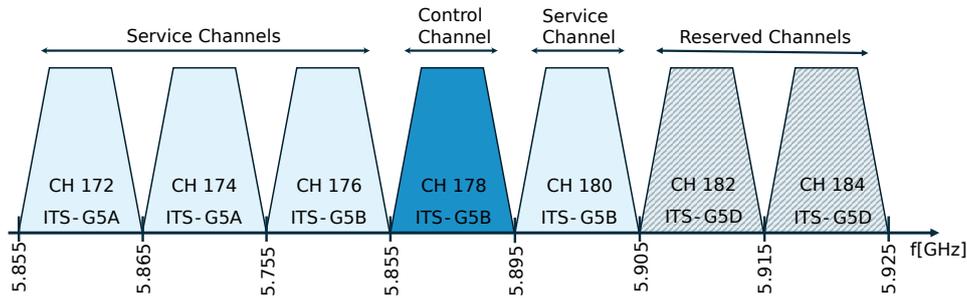


Fig. 2.2 EU spectrum allocation for the 5.9 GHz band.

It is worth mentioning that these frequency bands are shared spectrums, which multiple technologies are allowed to access. Indeed, both WiFi-based and Cellular-based technologies leverage the same frequency band for which a considerable amount of effort has been made by ETSI and the CAR2CAR Communication Consortium toward pre-standardization analysis of various channel co-existence methods in [74][55]. The channel co-existence remains an open research issue of great importance for the future of C-ITS due to the incompatibility between both families of access technologies, resulting in reduced performance when sharing the spectrum in an unstructured manner [146]. As the landscape of access technologies for V2X communications continues to be polarized between DSRC and C-V2X, mainly due to even advantages and disadvantages between the two, the European standardization entities are committed to achieving a fair, non-discriminatory, and future-proof channel co-existence.

The success of V2X (Vehicle-to-Everything) communication technologies depends on vehicles' ability to process and respond to large amounts of real-time data. While cloud computing solutions offer the necessary computational capacity for specific applications, relying exclusively on remote servers can introduce latency and bandwidth limitations that are unsuitable for safety-critical applications, such

---

as collision avoidance or maneuver coordination. The primary focus of C-ITS development is to ensure that contextual information is as up-to-date as possible for the proper functioning of application algorithms. Additionally, most potential users of the data generated by a given vehicle are typically those in close proximity. This proximity means that the latency and costs associated with transferring data to and from the cloud make deploying applications on remote servers impractical. To address the challenge of reducing end-to-end communication latency, the Multi-access Edge Computing (MEC) paradigm has emerged with its architecture standardized by ETSI in [78]. ETSI MEC offers an open framework enabling vehicles to take advantage of edge servers' computational resources, reducing latency for a wide range of applications and services, for which a comprehensive set of API specifications have been defined. Furthermore, MEC enables the implementation of real-time applications that demand high bandwidth requirements and low latency, such as intersection movement assist, which alerts drivers of potential collision risks, and real-time situational awareness for hazard detection and high-definition mapping. MEC also supports see-through applications, streaming real-time video from a remote vehicle to enhance visibility, and Cooperative Lane Change (CLC), enabling coordinated lane-switching signals between vehicles. Additionally, MEC facilitates the delivery of periodic, non-latency-sensitive tasks like software updates and critical safety features like vulnerable road user (VRU) discovery, which detects pedestrians and cyclists to warn nearby drivers. These applications showcase MEC's pivotal role in advancing safety, efficiency, and cooperation in vehicular systems [89].

One of the key strengths of MEC lies in its technology-agnostic nature, which ensures seamless interoperability regardless of the underlying radio access technology used by vehicles. Whether vehicles are equipped with DSRC or C-V2X devices for communication, MEC can serve as a unifying layer that bridges these technologies by leveraging its ability to process and relay data through edge servers. By leveraging Vehicle-to-Infrastructure-to-Vehicle (V2I2V) communication, Multi-access Edge Computing (MEC) can effectively bridge the gap between heterogeneous radio technologies—such as DSRC and C-V2X—so that vehicles equipped with different standards can still exchange time-sensitive information. As showcased by the authors in [53], MEC nodes positioned at the edge are capable of receiving and *transcoding* protocols from one radio technology to another (e.g., DSRC to C-V2X and vice versa). This allows vehicles to remain interoperable and share critical data through

localized MEC-based infrastructure, even though they cannot communicate directly over incompatible radio interfaces.

Communication between MEC nodes and vehicles, as well as with other network components, requires a robust and flexible messaging protocol to ensure the efficient exchange of information in diverse and dynamic environments. The Advanced Message Queuing Protocol (AMQP) emerges as a fitting solution for this requirement. As an open-standard messaging protocol, AMQP enables reliable and interoperable communication by supporting features like message queuing, topic-based subscriptions, and guaranteed delivery. These capabilities make it particularly well-suited for V2X scenarios where a wide range of entities—vehicles, infrastructure, and cloud systems—need to exchange data with minimal latency and maximum reliability.

To bridge the limitations of centralized cloud computing and edge solutions, Vehicular Micro-Clouds (VMCs) offer a complementary approach that builds upon the Multi-access Edge Computing (MEC) paradigm. While MEC significantly reduces latency by bringing computational resources closer to vehicles at the network edge, its reliance on fixed infrastructure introduces challenges in environments where connectivity to edge servers is intermittent or where computational demands exceed server capacity. Vehicular Micro-Clouds address these gaps by forming dynamic clusters of vehicles using V2V communication, effectively creating localized processing units that act as virtual edge servers.

By aggregating, caching, and processing data within these clusters, VMCs reduce the communication load on MEC servers while ensuring that critical, contextually relevant information is processed in real-time. This decentralized hierarchical model not only enhances the scalability and flexibility of MEC-enabled systems but also ensures robust performance in scenarios with high mobility or sparse infrastructure coverage. VMCs, therefore, provide a seamless extension to MEC, particularly for CCAM applications like platooning, cooperative perception, and real-time hazard detection, where immediate, localized processing is paramount.

Vehicular Micro-Clouds (VMCs) not only address the limitations of centralized solutions, such as high latency and dependency on fixed infrastructure, but also overcome the inefficiencies of standard decentralized systems. Unlike traditional decentralized approaches, which often suffer from inefficiencies in resource utilization, lack of coordination, and reliability issues due to the inherent dynamism of vehicular networks, VMCs form coordinated clusters of vehicles that collectively process and

share data. This architecture improves resource utilization, minimizes redundant transmissions, and ensures reliable service continuity. By bridging the strengths of centralized and decentralized paradigms, VMCs provide a scalable and robust platform tailored for CCAM applications like platooning and cooperative perception.

The following sections offer a detailed description of the various DSRC and C-V2X protocols, with some emphasis on C-V2X, which is the focus of this thesis thanks to its integration with MEC use cases. Additionally, the ETSI MEC architecture is illustrated, and the AMQP 1.0 protocol is further described, showcasing its importance within the MEC scheme. Lastly, an introduction to Vehicular Micro Clouds is provided, detailing its architecture, operations, and main challenges.

## 2.1 DSRC-based Protocols

### 2.1.1 WAVE

The IEEE WAVE (Wireless Access in Vehicular Environments) protocol suite encompasses the IEEE 1609 family of standards [17][20][21][22], along with IEEE 802.2 at the Logical Link Control (LLC) layer and IEEE 802.11p at the MAC and Physical layers. Collectively, these standards establish a communication stack specifically designed to facilitate Intelligent Transportation Systems (ITS), enabling Vehicle-to-Vehicle (V2V) communication between On-Board Units (OBUs) and Vehicle-to-Infrastructure (V2I) communication between OBUs and Roadside Units (RSUs).

These standards establish the functional framework for applications operating within the WAVE environment. This functionality is guided by the management activities specified in IEEE P1609.1, the security protocols outlined in IEEE P1609.2, and the network-layer protocol defined in IEEE P1609.3. Positioned above IEEE 802.11p, IEEE 1609.4 facilitates the seamless operation of higher layers by abstracting physical channel access parameters, enabling multi-channel operation, and coordinating channel usage among WAVE devices [150]. It is important to note that the higher OSI layers—namely the Application, Presentation, and Session layers—remain unspecified within the WAVE standard framework.

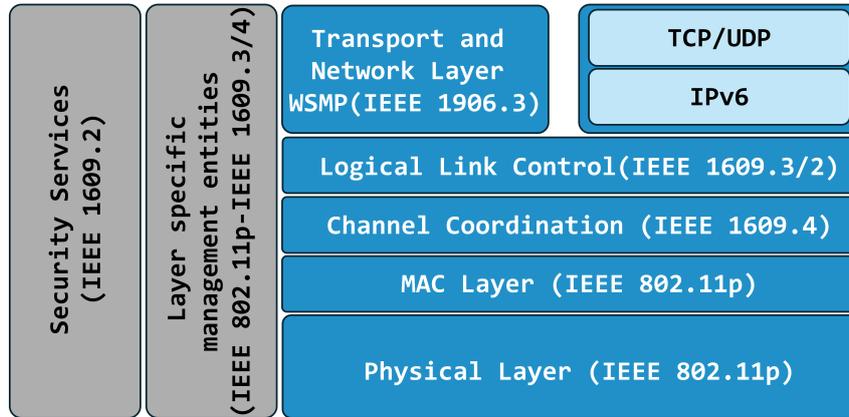


Fig. 2.3 WAVE protocol stack [17].

The components of the WAVE protocol stack are depicted in Figure 2.3. The stack is structured with a data plane, dedicated to protocols that handle the transfer of higher-layer information, and a management plane, which encompasses functions that indirectly facilitate information exchange between the layers by overseeing their operations.

### 2.1.1.1 Transport and Network Layers

The IEEE 1609.3 standard defines two parallel data plane protocol stacks: the IPv6 stack and the Wave Short Message Protocol (WSMP) stack, the latter optimized for wireless vehicular environments. WSMP allows applications to directly control transmission characteristics using a unique Provider Service Identifier (PSID) to ensure messages reach their intended recipients. Unmatched PSIDs result in the message being disregarded. While IPv6 is supported for scalability, the standard remains flexible regarding transport and higher-layer protocols, supporting UDP, TCP, and other IP-based protocols. In the Management Plane, the WAVE Management Entity (WME) coordinates system operations by managing time synchronization, processing service requests, and generating WAVE Service Advertisements (WSAs), which are transported via WSMP to fulfill higher-layer requirements [17].

### 2.1.1.2 Data-Link layers

At the Data-Link Layer, the Logical Link Control (LLC) sublayer distinguishes between the two upper-layer stacks (IPv6 and WSMP) using the EtherType field. This 2-octet field, located in the LLC header, identifies the specific networking protocol to be employed above the LLC sublayer. While EtherType is defined by IEEE 802.2, IEEE 1609.3 specifies its usage in WAVE devices. The hexadecimal values for IPv6 and WSMP are 0x86DD and 0x88DC, respectively [17].

WAVE employs the 802.11p standard for both the MAC and Physical Layers. To extend its capabilities, IEEE 1609.4 specifies MAC sublayer extensions that enhance channel coordination for multi-channel operations [22]. These extensions are essential for enabling the multi-channel functionalities required in vehicular environments.

### 2.1.1.3 802.11p

As previously mentioned, IEEE 802.11p is the standard utilized for the WAVE MAC and Physical Layers. This standard was specifically developed to provide the necessary specifications for ensuring interoperability between wireless devices operating in dynamic and rapidly changing communication environments, as well as scenarios requiring the completion of transactions within extremely short timeframes.

IEEE 802.11p is essentially a modified version of IEEE 802.11a. At the Physical Layer, 802.11p reduces the channel bandwidth to 10 MHz, compared to the 20 MHz bandwidth of the 802.11a amendment. This reduced bandwidth provides a wider guard band between adjacent channels, mitigating issues such as multi-path fading and Doppler effect phenomena, which can otherwise lead to Inter-Symbol Interference (ISI) [97].

Like other Wi-Fi-based technologies, IEEE 802.11p employs Orthogonal Frequency Division Multiplexing (OFDM), utilizing 64 orthogonal subcarriers. It supports data rates ranging from 3 Mb/s to 27 Mb/s, depending on the selected Modulation and Coding Scheme (MCS). Supported modulation techniques include BPSK, QPSK, 16QAM, and 64QAM, enabling flexibility and adaptability to varying communication requirements.

At the MAC Layer, IEEE 802.11p introduces a new mode of operation alongside the traditional ad hoc and infrastructure modes, known as Outside the Context of a BSS (OCB). This mode enables vehicular stations (STAs) to transmit and receive messages without the need to associate with a Basic Service Set (BSS) in the conventional manner, facilitating rapid and flexible communication in dynamic vehicular environments.

Additionally, 802.11p incorporates priority schemes based on the IEEE 802.11e standard, specifically tailored for DSRC (Dedicated Short Range Communications) using the Enhanced Distributed Channel Access (EDCA) mechanism. EDCA defines four distinct Access Categories (ACs) to prioritize data transmission, enabling differentiation between various types of messages. These ACs, ranked from lowest to highest priority, are as follows [23]:

- AC\_BK: Background.
- AC\_BE: Best Effort.
- AC\_VI: Video.
- AC\_VO: Voice.

The primary goal of this prioritization is to enable service differentiation, ensuring that safety-critical messages are transmitted with higher priority over non-safety messages, in this way enhancing the reliability and robustness of vehicular applications relying on V2X communications.

### **2.1.2 ETSI C-ITS**

Focusing on the European landscape of vehicular communications, the European Telecommunications Standards Institute (ETSI) has developed the C-ITS set of protocols, drawing inspiration from the WAVE architecture. For the C-ITS stack of an ITS station, depicted in Figure 2.4, ETSI has developed the ITS-G5 technology to be used as the Access Layer. ITS-G5 has been designed to leverage IEEE 802.11p as the underlying standard, as detailed in [8] and [11]. In recent years, ETSI has expanded its efforts to incorporate emerging access technologies, such as LTE-V2X and NR-V2X, as Access layer for the C-ITS stack. These efforts include providing

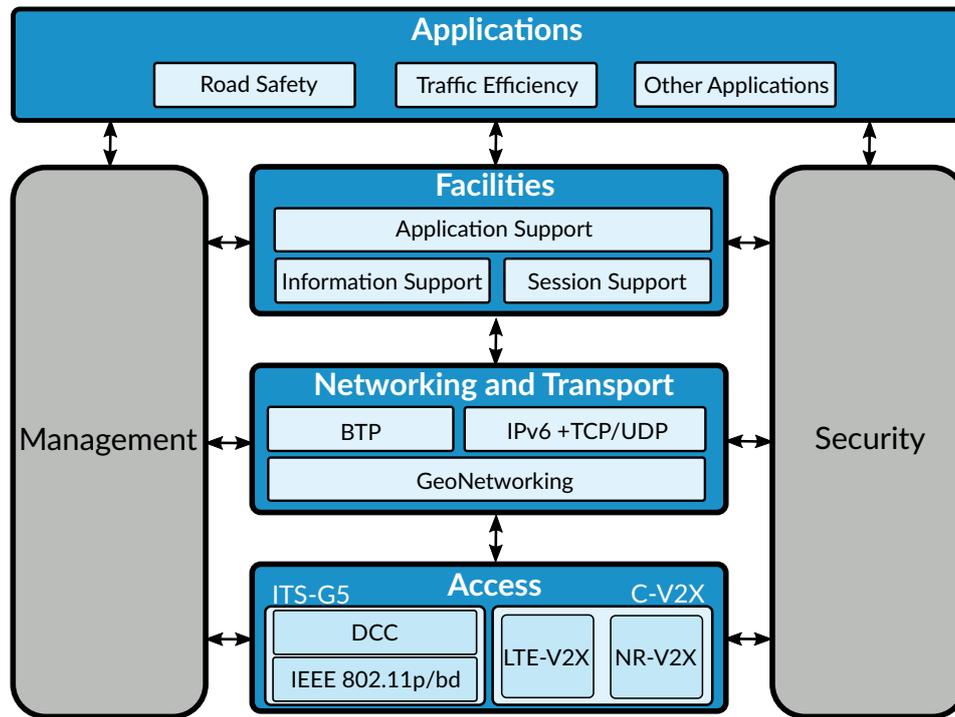


Fig. 2.4 ITS station protocol stack.

guidelines for the integration and interconnection of these technologies with higher layers of the protocol stack, as specified in [31].

The C-ITS stack of an ITS station, as specified in [7], aligns with the ISO/OSI reference model defined in ISO/IEC 7498-1. As depicted in Figure 2.4, the stack is organized into four data plane layers, each serving specific roles:

- **ITS Applications Layer:** Defined in [13], [26], and [14], this layer specifies the technical requirements for applications such as Road Hazard Signaling (RHS), Intersection Collision Risk Warning (ICRW), and Longitudinal Collision Risk Warning (LCRW).
- **ITS Facilities Layer:** Defined in [10], it is designed to cover the three highest layers of the ISO/OSI model. ITS applications rely on it, ensuring consistent and interoperable message exchange across different systems.
- **ITS Access Layer:** Defined in [8], this layer defines the physical and MAC sub-layers, along with mechanisms for Decentralized Congestion Control (DCC) as detailed in [27] and [18].

In addition, 2 vertical layers are defined :

- **ITS management entity:** responsible for coordinating various management functionalities for the operation of ITS stations. These functionalities include tasks such as station management, congestion control, and service advertisement. To support these operations, the entity leverages a Management Information Base (MIB) and several interfaces defined in [7], connecting it to all layers for the integration and management across the ITS station architecture.
- **ITS security entity:** designed to ensure security across the C-ITS protocol stack. Its functionalities include managing firewalls, intrusion detection, authentication, authorization, cryptographic keys, and certificates following the specifications in [80] [81] [79]. It also incorporates a Security Information Base (SIB) and Hardware Security Modules (HSM). To support secure operations, a set of interfaces is defined to connect with the rest of the layers of the C-ITS architecture providing specific security management services for each one.

To provide context for the primary focus of this work, which centers on the development of novel Facilities Layer applications under Vehicular Micro Cloud architectures, the following sections present an overview of the layers within the C-ITS stack. This overview aims to highlight the foundational framework upon which the contributions of this thesis are structured.

### 2.1.2.1 Application and Facilities layers

The C-ITS communication stack broadens the scope of vehicular communication beyond the WAVE counterpart by standardizing layers 5, 6, and 7 of the ISO/OSI model, corresponding to the Session (e.g., inter-host communication), Presentation (e.g., ASN.1 encoding, decoding, and encryption), and Application layers. Within the Application layer, as previously discussed, ETSI has initially identified three major applications composing the so-called ITS Basic Set of Applications (ITS BSA), utilizing Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs) for safety-critical use cases. Specifically, the Road Hazard Signaling (RHS) application, defined in [13], is introduced to enhance situational awareness among ITS stations (ITS-S) and between ITS-S and vehicle drivers. This is achieved through two operational modes: the originating mode, where an

ITS-S detects and transmits warnings about road hazards to nearby ITS-S, and the receiving mode, which notifies drivers of these hazards when relevant. Furthermore, [26] defines the Intersection Collision Risk Warning (ICRW) application, which predicts potential collision risks involving vehicles or obstacles in intersections. The application accounts for several risk types, including crossing collisions, violations of traffic signals, collisions with Vulnerable Road Users (VRUs), and rear-end collisions. Lastly, the Longitudinal Collision Risk Warning (LCRW) application is outlined in [14], addressing other collision types such as frontal and side-forward collisions.

Descending one layer in the ITS-G5 stack, the Facilities layer provides essential support for ITS applications by offering shared functionalities and data provision depending on their operational and functional requirements [7]. This layer thus acts as an intermediary ensuring the exchange of information across diverse vehicular applications, aligning with the functional goals of C-ITS to enhance road safety and communication. As specified in [10], this layer encompasses the upper three layers of the OSI model and organizes its services into sub-layers categorized into the following classification:

- **Application Support Facilities:** These facilities are responsible for managing functionalities that support ITS Basic Set of Applications (ITS BSA). They include components for managing the Cooperative Awareness Messages (CAM) and Decentralized Environmental Notifications Messages (DENM). CAM enables vehicles to share real-time information about their position, speed, and status, with the aim of improving situational awareness among nearby entities. DENM, on the other hand, communicates notifications about specific events, such as hazards or road conditions, enhancing safety and responsiveness in dynamic environments.
- **Information support facilities:** These facilities provide essential data and database management functionalities for the ITS BSA. The main one is the Local Dynamic Map (LDM), which integrates both dynamic and static data from surrounding sources. This map serves as a comprehensive representation of the road environment, supporting applications such as collision avoidance, route optimization, and urban traffic management. LDM enables vehicles and infrastructure to access and utilize data effectively, ensuring informed decision-making in real-time.

- **Communication support facilities:** These facilities are tasked with managing communication and session requirements for ITS applications. They include services like addressing modes for V2X message transmissions and support for session management. Such functionalities provide message delivery reliability in compliance with the stringent latency requirements of safety-critical vehicular applications. These facilities articulate the coordination among vehicles, infrastructure, and other different types of entities of ITS.

Regarding Application Support Facilities, in addition to the CA and DEN services, which represent the main pillars of what the CAR2CAR Communication Consortium calls Day 1 applications [58], ETSI has standardized a comprehensive spectrum of facilities layer services to support the next generation of applications, namely Day 2 and Day 3. While Day 1 focused on information dissemination about the vehicle itself (e.g., through CAMs), Day 2 broadens this by integrating data collected about the vehicle's surroundings using on-board sensors. Indeed, Day 2 emphasizes the detection of objects and conditions beyond the vehicle's line of sight for use cases that include semi-automated vehicle maneuvers such as emergency braking and cooperative adaptive cruise control. One of the main Application Support Facilities for Day 2 applications standardized by ETSI in recent years is the Collective Perception Service (CPS) [77]. The primary objective of the CPS is to enhance situational awareness by enabling ITS Stations (ITS-S) to exchange information about their local environment, particularly regarding non-V2X-equipped entities such as legacy vehicles, pedestrians, animals, and static objects, as well as the status of perception regions. To facilitate the transmission and reception of Cooperative Perception Messages (CPMs), the CPS supports the following sub-functions:

- **Encode CPM:** Constructs the CPM according to the specified ASN.1 format. The encoding process includes the most recent object information, sensor information, and perception region information data.
- **Decode CPM:** Decodes the received CPM, extracting the necessary information for further use.
- **CPM Transmission Management:** Implements the protocol involving the management of the activation and termination of CPM transmission operations, determining the frequency of CPM generation events, and triggering CPM generation.

- **CPM Reception Management:** Implements the protocol decoding incoming CPMs and ensuring that the decoded information is made available to the Local Dynamic Map (LDM) and/or ITS applications. Additionally, it may optionally validate the accuracy of the information contained in the received CPMs.

The information within CPMs is encapsulated as depicted in Figure 2.5, which includes the ITS PDU header (included in all Facilities layer messages) and a set of containers comprising the CPM payload. In addition to the mandatory management container, containing the basic information necessary to interpret the rest of the CPM's information, a total of 4 different optional containers are defined to complete the CPM payload:

- **Originating ITS-S Container:** provides details about the ITS station (ITS-S) generating the message. In the case of vehicles, this includes orientation information that helps in interpreting velocity and other kinematic data. The orientation may also include pitch and roll angles if available, while trailer configurations are described when applicable. The container specifies attributes such as hitch points and angular offsets for trailers, along with reference points to clearly define the relationship between the towing vehicle and its trailers.
- **Sensor Information Container:** Specifies the capabilities and attributes of the sensors involved in object detection. Each sensor is uniquely identified and categorized by type, such as radar, LiDAR, or camera. Additionally, the container can specify that the provided information results from sensor fusion, where data from multiple sensors is aggregated to produce a more accurate and comprehensive environmental representation. The container defines the sensor's perception region, specifying its coverage area geometrically. For vehicle-mounted sensors, this is expressed in a body-fixed coordinate system, while infrastructure sensors use an East-North-Up system. The container also includes detection confidence levels for these regions and indicates whether shadowing effects (caused by occlusions) impact detection accuracy.
- **Perception Region Container:** Describes the spatial regions covered by the sensors, describing specific regions where objects may be detected. It captures changes in detection capabilities that may differ from the static

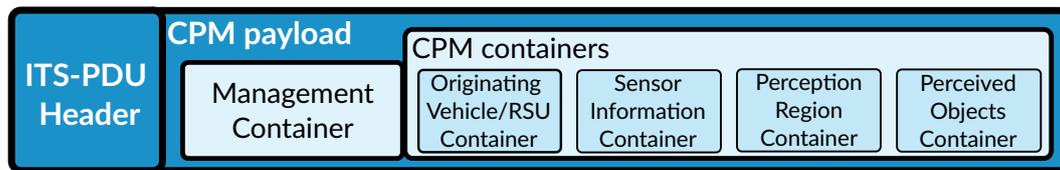


Fig. 2.5 CPM structure defined in [77].

specifications in the Sensor Information Container. The container specifies confidence levels for different areas within the perception region and accounts for shadowed or occluded zones. It also lists the objects detected within each region and identifies the sensors responsible for these detections, enabling detailed tracking of the environment.

- **Perceived Object Container:** contains detailed information about detected objects, specifying their kinematic state, classification, and associated meta-data. Each object is assigned a unique identifier and is described in terms of its position, velocity, and orientation. These attributes are defined relative to the vehicle reference point, which is specified in the Management Container, representing the ground position of the center of the front of the vehicle’s bounding box. It is worth noting that all objects listed in this container share the same reference point for positional and kinematic data, independent of the position or orientation of the sensor that detected each object. Information about the sensors themselves is managed separately in the Sensor Information Container. The container also includes object classifications, such as vehicle or pedestrian, along with confidence levels that indicate the reliability of the classification. Additional attributes, such as object dimensions and quality metrics, further quantify detection accuracy. While the vehicle reference point provides a unified spatial origin for all kinematic data, the specific contributions of individual sensors are accounted for through offsets and orientations defined in the Sensor Information Container.

The dissemination of CPMs operates on an event-driven system scheme. The generation is not on a rigid periodic schedule but is instead triggered by specific events, such as significant changes in the environment. For instance, updates occur when a perception region is modified or when a detected object experiences a significant change in its kinematic state, requiring its inclusion in the shared information. Regarding the inclusion of data in CPMs, a dynamic set of rules based on the ones

used for CAMs ensures the relevance of the exchanged information. Newly detected objects, or those whose characteristics have changed significantly, are considered for inclusion. Conversely, static objects or those that change state infrequently are updated less often. In this manner, a trade-off between data relevancy and channel congestion is achieved. Finally, the assembly and transmission of CPMs are carefully managed to align with size constraints, such as Maximum Transmission Unit (MTU) limits. Multi-Channel Operation (MCO) is supported to optimize delivery, leveraging multiple communication channels to distribute data. Within this paradigm, high-priority messages are assigned to the appropriate channel, minimizing delays and ensuring that critical information reaches its intended recipients without interruption.

The Information Support Facilities most closely related to the CPS is the Local Dynamic Map (LDM) [68], which serves as a repository for road traffic information. It encompasses data on both moving objects, such as nearby vehicles, and stationary objects, such as traffic signs. To achieve this, the LDM integrates information from various sources, including vehicles, infrastructure, road operators, and sensors. For instance, vehicle data is supplied through Cooperative Awareness services and represented as objects derived from CAMs, while sensor data can be provided through the CPS, based on object information extracted from CPMs. These diverse data inputs enable the LDM to support its core functionalities, which include registering facilities and applications as LDM data providers or consumers, managing subscriptions for notifications, retaining information, and prioritizing requests. Regarding the CPS, it acts as both a data provider and as a data consumer by providing object information from received CPMs and requesting information about objects detected by onboard sensors to be considered for inclusion in a new CPM. Additionally, as detailed in [68], the data management functionalities to be performed by the LDM include the maintenance and validation of all LDM Data Objects received from registered and authorized LDM Data Providers. The LDM ensures that each Data Object remains valid within a defined time validity and within the geographical boundaries of the LDM Area of Maintenance. A Data Object is considered valid from the timestamp provided by the Data Provider until the end of its specified validity period. The default time validity, established during the Data Provider's registration, can be overridden by specific time validity values assigned when adding or updating individual Data Objects. Furthermore, the LDM validates the spatial relevance of each Data Object by ensuring its location intersects with the LDM Area of

Maintenance, a geographical region defined relative to the current location of the hosting ITS-S.

### 2.1.2.2 Networking and Transport layers

GeoNetworking (GeoNet) is a network-layer protocol developed to address the challenges of mobile ad hoc communication in vehicular networks (VANETs) [15]. This protocol uses geographical positions to disseminate information and route data packets efficiently, enabling the execution of V2X communications. The main goal of GeoNet is to support a wide array of applications within ITS, prioritizing functionalities like high-rate periodic status messages for road safety and multi-hop packet dissemination for emergency warnings in geographically defined areas.

GeoNet operates in a decentralized, *connectionless* manner, specifically designed to adapt to the dynamic mobility and topology of VANETs. It enables the addressing of packets to individual nodes or groups of nodes based on their geographic location, relying on nodes to maintain a partial network topology view. Each node, upon receiving a packet, processes its geographical address and forwards it accordingly based on its current knowledge of the topology. GeoNet provides geographical addressing and forwarding capabilities, enabling various communication scenarios tailored for ITS applications, as described in [12]. These scenarios include:

- **Point-to-Point (GeoUnicast):** Communication occurs between two specific ITS-S. When performing GeoUnicast, packets are forwarded hop-by-hop based on the destination's geographical location until they reach the target node.
- **Point-to-Multipoint (Topologically-Scoped Broadcast):** A single ITS-S communicates with multiple stations using Topologically-scoped broadcast. Here, packets are broadcast to neighbors, which re-broadcast them further. For the special case of Single-hop broadcast, commonly used for CAMs and CPMs, messages are not re-forwarded by receivers, restricting the broadcast to one-hop neighbors.
- **GeoAnycast:** In this scenario, the communication targets any ITS-S within a specified geographical area. Within this scenario, packets are forwarded hop-by-hop until they reach the area of interest. Once inside the area, the first node to receive the packet terminates the forwarding.

- **GeoBroadcast:** A single ITS station sends packets to all stations in a geographical area. Packets are forwarded similarly to GeoAnycast, with the difference being that nodes inside the area of interest rebroadcast the packet to ensure complete coverage within said area. This scheme is generally applied for the dissemination of DENMs.

### 2.1.2.3 Access layers

As introduced earlier in this section, ETSI has developed the ITS-G5 standard, which leverages the IEEE 802.11p (outlined in 2.1.1.3) for the MAC and Physical layers, as the primary standard for the ITS Access layer. It is worth noting that due to increasing development from 3GPP, the specifications of this layer were later extended to support C-V2X in [31], which will be described more in detail in Section 2.2. The main feature of ITS-G5 is the addition of Decentralized Congestion Control (DCC) methods specified in [27] for the Logical Link Control sub-layer. DCC aims to mitigate channel overload in congested traffic scenarios in which an elevated number of vehicles are in close proximity and find themselves in radio range. The way in which this issue is mitigated by DCC is through the regulation of data traffic of each ITS-S to avoid surpassing the communication channel capacity [27]. As the information available at the MAC layer is not sufficient, ETSI details cross-layer operations of DCC within ITS-S in [18]. The DCC mechanism relies on higher-layer services to prioritize data traffic sources when channel congestion demands a restricted transmission. In addition to controlling the overall channel usage, DCC offers mechanisms to regulate channel allocation fairness among ITS-S, avoiding an unbalanced distribution of channel resources.

### 2.1.3 IEEE 802.11bd

Even though IEEE 802.11p has demonstrated its ability to meet the requirements of most Day 1 safety and traffic efficiency applications under moderate network traffic conditions [136], IEEE 802.11p networks are not expected to fulfill the stricter requirements of Day2 and Day3 applications, some of them including maximum latencies of 3 ms and Packet Reception Ratio (PRR) as high as 99.99%.

To address these emerging demands, IEEE has put significant efforts towards developing the IEEE 802.11bd standard. The IEEE 802.11bd standard aims to en-

hance transmission reliability, increase throughput, and extend transmission range compared to the legacy IEEE 802.11p standard. Additionally, it ensures interoperability, coexistence, backward compatibility, and fairness with existing IEEE 802.11p devices. Indeed, IEEE 802.11bd devices (referred to as Next Generation V2X Station (NGV STA) in [37]) can receive transmissions from any IEEE 802.11p-based devices (referred to as non-NGV STA) such as ITS-G5 devices and vice-versa. In order to secure a seamless interoperability, stations using IEEE 802.11bd-based ETSI ITS-G5 prioritize compatibility with existing IEEE 802.11p-based stations for Day 1 messages (e.g., CAM and DENMs). When IEEE 802.11p stations are detected nearby, these messages are transmitted via the IEEE 802.11p access layer to ensure interoperability, whereas in the absence of such stations, IEEE 802.11bd is utilized. Regardless of the format, packets transmitted with IEEE 802.11bd remain detectable by IEEE 802.11p stations due to a shared preamble. This allows existing stations to integrate these packets into their CSMA/CA channel access mechanism, even if they cannot decode the IEEE 802.11bd data fields [56]. The IEEE 802.11bd standard introduces several technical advancements to address the requirements of NGV communications. In particular, it incorporates advanced modulation schemes, such as BPSK-DCM and 256-QAM, allowing for increased data rates and improved spectral efficiency. With the addition of Low-Density Parity-Check (LDPC) error correction, IEEE 802.11bd for a more more reliable communication under challenging conditions. Flexible channel bandwidths are supported with interoperability between 10 MHz and 20 MHz channels. Although the same subcarrier spacing of IEEE 802.11p is utilized, a more significant portion of the channel is used in IEEE 802.11bd, supporting 56 subcarriers instead of 52 of IEEE 802.11p. IEEE 802.11bd also incorporates 2x2 MIMO technology, meant for unicast transmissions, and it extends the operating frequency spectrum to include the 60 GHz band alongside the traditional 5.9 GHz band, facilitating support for additional use cases and increased channel capacity. To improve communication reliability, IEEE 802.11bd employs adaptive repetition mechanisms (repetitions based on the channel busy ratio), as well as the addition of DCM (Dual-Carrier Modulation), improving performance by offering a 3 dB gain in sensitivity, contributing to more reliable transmissions over longer distances. Lastly, the standard includes advanced localization capabilities, supporting positioning functionalities through fine timing measurements, which can be leveraged for precise vehicle positioning in cooperative scenarios. Midamble-

based channel tracking is also introduced, simplifying receiver complexity while maintaining high performance.

IEEE 802.11bd has been specifically designed to enhance the ETSI ITS-G5 protocol. Its features support Day 2 applications such as the Cooperative Perception Service (CPS) and Cooperative Maneuvering Service (MCS), leveraging the improved capabilities of its access layer. By supporting multi-channel operations (MCO) with two adjacent 10 MHz channels, the mechanisms outlined in [77] for CPS are natively integrated, enabling more robust and efficient V2X communication.

## 2.2 Cellular-based Protocols

Despite its early adoption, DSRC faced challenges in scalability, range, and interoperability, particularly as use cases expanded. Recognizing these limitations, 3GPP introduced C-V2X technologies, starting with LTE-V2X in 3GPP Release 14 [25]. LTE-V2X represented an adaptation of cellular communications, including enhancements to the Radio Access Network (RAN) and more importantly, the introduction of the PC5 interface allowing direct sidelink communication between vehicles, other road users, and infrastructure [94]. These adaptations aim to provide reliable, low-latency, and high-throughput communication tailored for vehicular environments, enabling direct communication between vehicles and infrastructure in centralized scenarios, as well as between vehicles in out-of-coverage scenarios. In particular, in addition to the Uu interface for V2N communication to the cellular infrastructure, LTE-V2X introduced two operational modes to use the PC5 interface, namely, Mode 3 for centralized resource management and Mode 4 for autonomous operation. The support for both modes attempts to cover not only urban scenarios with the availability of cellular infrastructure (Mode 3) but also rural or disaster scenarios without network coverage (Mode 4).

Release 14 established LTE V2X's core functionality by enabling the transmission of Day 1 road safety messages such as CAMs, and DENMs while defining the architecture for LTE-based V2X technologies. This foundation was further enhanced in Release 15 [28], which introduced advanced use cases like real-time situational awareness, remote driving, and software updates. Significant technical upgrades included Carrier Aggregation (CA) for mode 4, support for 64-QAM modulation,

and higher message generation rates, alongside initial steps to integrate V2X services into the emerging 5G ecosystem.

Release 16 [30] marked a significant leap, introducing Ultra Reliable Low Latency Communications (URLLC) for 5G V2X, particularly enhancing short-range direct communication. This upgrade supported use cases requiring extreme reliability and minimal latency, such as sensor data sharing for collective perception, vehicle trajectory exchange for collision avoidance, and coordinated platoon driving with very short gaps. Building on the advancements of Release 16 and recognizing the availability of large bandwidths in unlicensed spectrum, Release 17 enabled sidelink operations in the 5 GHz and 6 GHz unlicensed bands (n46 and n96/n102), aligned with the latest regulatory requirements. Release 17 [40] continued to expand the capabilities of NR sidelink and V2X communication with a strong focus on addressing power saving and enhancing resource allocation coordination. Another important initiative focused on single-hop NR sidelink-based relay, studying how to enable efficient relay communication over NR sidelink. In particular, Release 17 introduced partial sensing-based and random resource allocation methods, enabling UEs in Mode 2 to adaptively select resources based on their configuration and operational state. Additionally, Inter-UE coordination (IUC) mechanisms were introduced to allow UEs to share information about preferred or non-preferred resources and to identify overlapping resources based on sidelink control information. The momentum has continued with Release 18 [41], recognizing the potential of unlicensed spectrum to support high data rates and broader connectivity, focusing on expanding sidelink communication into unlicensed frequency bands. Specifically, it targeted FR1 unlicensed bands in the 5 GHz and 6 GHz ranges, aligning with updated regulatory standards. In addition to unlicensed spectrum support, Release 18 has extended the efforts of Release 17 to implement mechanisms to ensure co-channel coexistence between LTE sidelink and NR sidelink. By supporting both semi-static and dynamic resource pool sharing, these mechanisms allow the two technologies to operate efficiently on the same frequencies without interfering. Furthermore, Release 18 introduces an enhanced resource selection feature in sidelink Mode 2, enabling UEs to select resources across consecutive slots for continuous transmission of one or more transport blocks.

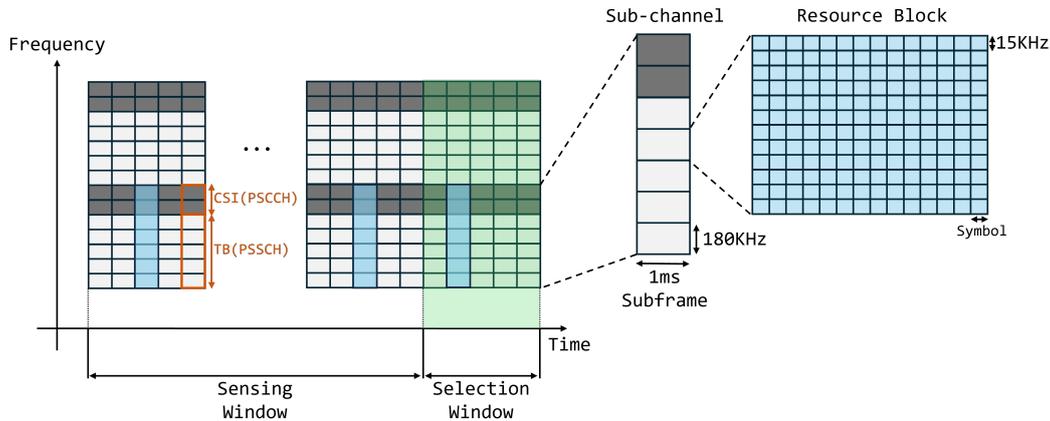


Fig. 2.6 Time-frequency allocation in LTE-V2X channels, featuring the neighboring Physical Sidelink Control Channel and Physical Sidelink Shared Channel.

### 2.2.1 LTE-V2X

The physical layer of LTE-V2X utilizes the Single Carrier Frequency Division Multiple Access (SC-FDMA) modulation scheme, chosen for its resilience to Doppler effects and its ability to maintain low latency that is a critical requirement for safety applications. The operation band of LTE-V2X is the 5.9 GHz ITS band, with channel bandwidths of 10 MHz and 20 MHz. Each channel is divided into Resource Blocks (RBs), with each RB spanning 180 kHz and comprising 12 subcarriers, each 15 kHz wide. A pre-configured number of RB constitutes a sub-channel, representing the minimum unit for data transmission. The time domain is organized into 1 ms subframes, each containing 14 Orthogonal Frequency-Division Multiplexing (OFDM) symbols. In each subframe, nine symbols are dedicated to data transmission, four are used for Demodulation Reference Signals (DMRS) to support channel estimation, and one serves as a guard period for switching between transmission and reception. The Transport Block (TB) represents the smallest data unit transmitted over the Physical Sidelink Shared Channel (PSSCH). Each TB contains a complete message, e.g., a CAM or CPM. Furthermore, an associated Sidelink Control Information (SCI) is transmitted over the Physical Sidelink Control Channel (PSCCH). The SCI contains metadata essential for the proper decoding of the TB, including Modulation and Coding Scheme (MCS), resource allocation, and transmission priority.

LTE-V2X supports QPSK, 16-QAM as well as 64-QAM since Release 15 [84]. LTE-V2X supports two distinct modes of resource allocation to cover different requirements of vehicular networks scenarios:

- **Mode 3 (Centralized Allocation):** Mode 3 represents the central resource allocation mode for urban scenarios and relies on an eNB for centralized scheduling. It requires network coverage and leverages the LTE infrastructure for resource management. Subchannels are allocated dynamically or semi-persistently, with the eNB considering factors such as channel conditions and network congestion. Under a dynamic scheduling scheme, each ITS-S requests resources from the eNB for each TB transmission, whereas on the Semi-Persistent Scheduling (SPS) counterpart, the eNB reserves resources for periodic TB transmissions with reduced signaling.
- **Mode 4 (Distributed Allocation):** Mode 4, designed for scenarios where cellular infrastructure is unavailable, employs a Sensing-Based SPS algorithm, allowing vehicles to select and manage communication resources autonomously. Under this scheme, the reservation details, including the interval for the subsequent transmission, are shared with nearby vehicles through the Resource Reservation Interval (RRI) announced in the SCI. Vehicles choose RRI values from a pre-configured list, which includes intervals ranging from 0 ms to 1000 ms. The chosen RRI influences the duration of the reservation and the Reselection Counter, which is randomly assigned within specific ranges based on the RRI value. For instance, shorter RRIs result in higher counter values, allowing for more frequent re-selection of channels. When the counter reaches zero, or if a new transmission block cannot fit the reserved sub-channels or meet its latency requirements, vehicles select new sub-channels using the sensing-based scheduling process. After identifying potential Candidate Sub-frame Resources (CSR) within a specific time frame known as the Selection Window (SW), the ITS-S evaluates the size and timing of the available sub-channels to ensure they can accommodate the upcoming data block. Next, vehicles refine their options by excluding sub-channels likely to experience collisions. This step relies on historical sensing data, from where the ITS-S evaluates which sub-channels were previously occupied by themselves or nearby vehicles. From the remaining options, vehicles focus on prioritizing sub-channels with the least interference, evaluating the average signal strength of the re-

maining candidates, and selecting those with the lowest levels of interference, creating a shortlist of high-quality sub-channels. Finally, to prevent multiple vehicles from unintentionally selecting the same sub-channel, a random choice is made from the prioritized shortlist. This random selection ensures fairness and reduces the likelihood of collisions, allowing vehicles to secure reliable communication channels for their upcoming transmissions.

### 2.2.2 NR-V2X

To support a wide range of V2X applications with varying quality of service (QoS) requirements, including scenarios with high vehicular density, NR-V2X was introduced in Release 16, leveraging the flexibility of 5G NR to expand LTE-V2X's periodic safety message scope and enabling diverse use cases outlined in the 5G V2X design requirements. The V2X use cases have been categorized by 3GPP into four groups [84], aiming to collectively address the diverse and demanding needs of V2X communication applications, each tailored to specific applications with distinct communication requirements: Vehicle Platooning, Advanced Driving, Extended Sensors and Remote Driving. Platooning focuses on enabling coordinated platoon formation and management, requiring low latency (10-25 ms) and moderate reliability in the 90-99.99% range to support periodic data exchange. Advanced Driving supports automated and semi-automated driving by facilitating the sharing of sensor data and trajectory coordination, with reliability in the 90-99.999% range and a latency of 10-100 ms to ensure safety and traffic efficiency. Extended Sensors aim to enhance situational awareness by allowing vehicles and RSUs to share sensor data, improving perception beyond onboard sensor capabilities, e.g., through CPMs. This group operates with a latency requirement of 3-100 ms. Lastly, Remote Driving enables the teleoperation of vehicles in complex or hazardous scenarios, demanding ultra-low latency (5 ms) and extremely high reliability (99.999%) to guarantee seamless control. To support this wide set of application requirements, substantial enhancements have been introduced for NR-V2X for sidelink communications, as described in the remainder of this section.

NR-V2X introduces significant advancements over LTE-V2X, tailored to support a broader range of use cases. Notably, NR-V2X adds support for groupcast and unicast communication modes, as well as higher numerologies, which can be configured individually for each Sidelink Bandwidth Part (BWP). Initially introduced in Release

15 for the Uu interface, BWPs were extended to the PC5 interface in Release 16, reducing power consumption and computational complexity by enabling devices to operate within a narrower bandwidth. This minimizes the need for continuous full-bandwidth processing, reducing sampling rate requirements and optimizing resource allocation for more efficient sidelink communication. All sidelink channels, reference signals, and synchronization signals are transmitted within a single Sidelink BWP, ensuring consistency in numerology across the sidelink. The Sub-Carrier Spacing (SCSs) supported by NR-V2X are 15 kHz, 30 kHz, 60 kHz, and 120 kHz, representing the numerologies 0, 1, 2, and 3, respectively. NR-V2X operates in two frequency ranges: Frequency Range 1 (FR1) from 410 MHz to 7.125 GHz and Frequency Range 2 (FR2) from 24.25 GHz to 52.6 GHz. The maximum bandwidth per carrier is 100 MHz in FR1 and 400 MHz in FR2, significantly exceeding the 20 MHz bandwidth supported by LTE-V2X. The frame structure of NR-V2X is organized into 10 ms radio frames, divided into 10 subframes of 1 ms each. Unlike LTE-V2X, the number of slots per subframe in NR-V2X varies based on the SCS, where larger SCS values result in shorter slot durations and higher time resolution. For instance, a 15 kHz SCS results in one slot per subframe, while a 60 kHz SCS allows for four slots. Each slot comprises 14 OFDM symbols for normal Cyclic Prefix (CP) or 12 symbols for extended CP, with extended CP specifically supported for 60 kHz SCS to accommodate more considerable delay spreads. The Transport Block (TB) remains the smallest unit of data transmission over the Physical Sidelink Shared Channel (PSSCH), with Modulation and Coding Schemes (MCS) including QPSK, 16-QAM, 64-QAM, and 256-QAM. For control information, NR-V2X transmits Sidelink Control Information (SCI) over the Physical Sidelink Control Channel (PSCCH). Notably, NR-V2X also introduces the Physical Sidelink Feedback Channel (PSFCH) for Hybrid Automatic Repeat Request (HARQ) feedback, a feature absent in LTE-V2X. In NR V2X, sidelink communication has been enhanced by introducing a split Sidelink Control Information (SCI) structure, which optimizes resource management and signaling efficiency. The SCI is divided into two formats: 1st-stage SCI and 2nd-stage SCI, each serving distinct purposes. The 1st-stage SCI, carried on the PSCCH, provides resource priority, time and frequency assignments, the 2nd-stage SCI format, modulation and coding scheme (MCS), and the resource reservation period. Notably, the 1st-stage SCI facilitates the reservation of up to  $N_{\max\_reserve}$  resources (pre-configured as 2 or 3) within a selection window. This feature exempts the UE from sending a 1st-stage SCI in every slot, instead it

specifies the allocation of multiple slots, with some slots carrying only PSSCH data. The 2nd-stage SCI, transmitted via the PSSCH, carries information such as HARQ process ID, redundancy version, new data indicator (NDI), and source/destination IDs, ensuring the reliability and correctness of data transmission.

Resource allocation in NR-V2X sidelink communications includes centralized Mode 1 and distributed Mode 2, analogous to LTE-V2X's Modes 3 and 4. However, NR-V2X's resource allocation mechanisms provided several enhancements as described in the following subsections.

### **2.2.2.1 Mode 1 (Centralized Allocation)**

Mode 1 in NR-V2X builds upon LTE-V2X Mode 3, offering centralized resource allocation managed by a gNB. It supports two scheduling methods: Dynamic Grant (DG) and Configured Grant (CG). DG provides flexibility for non-periodic traffic, dynamically allocating resources based on real-time requests from UEs. However, it introduces delays due to signaling overhead. To address latency-sensitive applications, CG pre-allocates periodic resources based on QoS requirements like latency, reliability, and priority, as specified by the UE. NR-V2X introduces two CG types: Type 1, which activates resources immediately upon configuration for consistent availability, and Type 2, which dynamically activates resources to improve efficiency by reallocating unused capacity. This dual approach ensures a balance between resource efficiency and low-latency communication. Mode 1 also supports HARQ retransmissions, where feedback from receiving UEs is relayed to the gNB for retransmission decisions. Additionally, NR-V2X enhances flexibility by allowing multiple Modulation and Coding Scheme (MCS) options and supporting efficient coordination between UEs in shared resource pools, minimizing collisions. These advancements make Mode 1 well-suited for high-density, low-latency vehicular networks requiring reliable and efficient communication.

### **2.2.2.2 Mode 2 (Distributed Allocation)**

Mode 2 in NR-V2X, like LTE-V2X Mode 4, enables UEs to autonomously select sidelink resources without requiring network coverage. These resources can be pre-configured by the network if the UE was previously under coverage. Mode 2 supports two scheduling schemes:

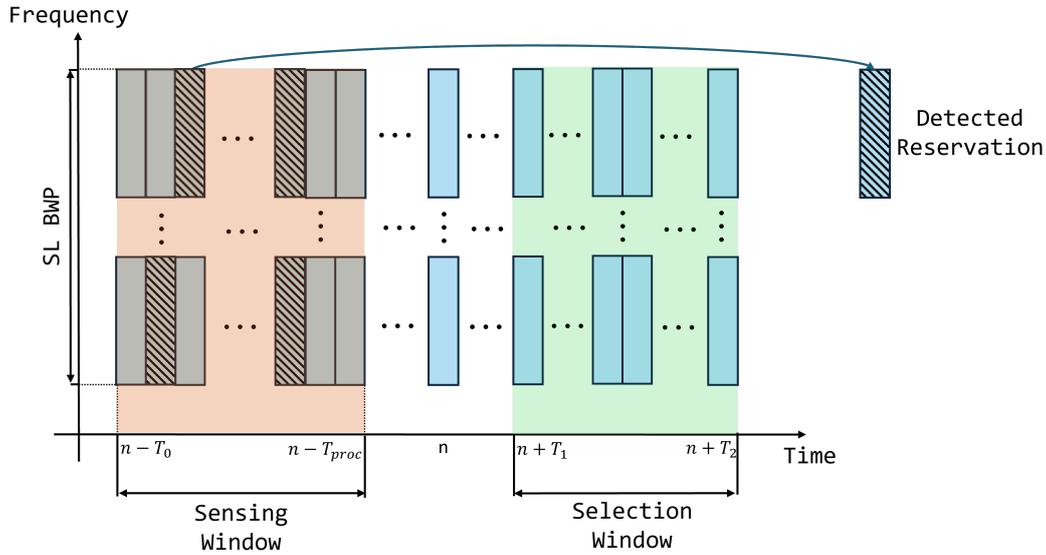


Fig. 2.7 Sensing-based Semi Persistent Scheduling Scheme in NR-V2X.

- **Dynamic scheduling:** under this scheme, resources are selected for individual Transport Blocks (TBs) and their retransmissions, minimizing resource reservation but potentially increasing signaling overhead.
- **Semi-Persistent Scheduling:** under which resources are reserved for multiple consecutive TBs based on a configurable Resource Reservation Interval (RRI), designed to manage resources for periodic and predictable traffic patterns. It allows UEs to minimize the overhead of repeatedly selecting new resources for each Transport Block (TB).

Unlike LTE-V2X Mode 4, which supports a limited set of predefined RRI values, Mode 2 accommodates varying traffic patterns, allowing for any integer RRI between 1 and 99 ms, along with preconfigured options up to 1000 ms. Once a given RRI is selected, each UE defines a selection window over which the occupied sidelink resources will be identified. The selection window is defined within an interval  $[n - T_0, n - T_{proc,0}]$  where  $T_0$  mandates how far back the UE considers resources, which is usually in accordance with the RRI selected.  $T_{proc,0}$  represents the processing time required for the completion of the sensing procedure. During this interval, a list of candidate resources is constructed relying on the information collected from received SCIs, the RSRP associated with those SCIs, and the transmission and retransmission patterns of neighboring UEs. When a given UEs generates a new TB,

it selects sidelink candidate resources obtained from the sensing window under a two-step resource selection process. First, the resource selection is performed, where a selection window is defined based on the packet's delay budget (PDB), priority, and subcarrier spacing (SCS). This window includes the range of slots  $[n + T_1, n + T_2]$  where  $T_1$  represents the processing time measured in the number of slots that the UE requires to identify the candidate resources and select new ones and  $T_2$  represents the maximum transmission delay desired which the PDB typically mandates. The list of candidate resources obtained from the sensing window is further filtered to contain only the ones within the selection window. Second, the UE reserves resources for a specified number of future transmissions based on the selected RRI. For each reserved resource, the time interval between consecutive allocations equals the RRI. After the reservation, the UE includes the reserved resources in its first-stage SCI, informing neighboring UEs of the resources it plans to use for both initial TB transmissions and potential HARQ retransmissions. For each TB transmission (including retransmissions), the Reselection Counter is decremented by 1 and when the counter reaches zero, the UE either selects new resources or reuses the same resources with a probability of  $1 - P$ , where  $P \in [0, 0.8]$ . NR-V2X's Mode 2 supports HARQ retransmissions only for unicast and groupcast communications if the Physical Sidelink Feedback Channel (PSFCH) is pre-configured. Otherwise, blind retransmissions are used. If HARQ retransmissions are used, if a retransmission is positively acknowledged, the UE cancels subsequent retransmissions for that TB, freeing up the reserved resources.

## 2.3 ETSI MEC

As already described at the beginning of the chapter, the Multi-Access Edge Computing (MEC) paradigm emerged as a solution to overcome the inherent limitations of centralized cloud computing in latency-sensitive and bandwidth-intensive applications. Particularly in the context of Vehicle-to-Everything (V2X) communication, MEC allows vehicles to process and exchange large volumes of real-time data with minimal delay by providing computational resources to the edge of the network. By executing critical workloads in local edge nodes rather than remote data centers, latency is significantly reduced, which ensures prompt decision-making and timely operation of safety-critical systems. This paradigm not only addresses the critical

latency and bandwidth constraints of cloud-based solutions but also ensures that contextual information, such as localized sensor data, remains as up to date as possible to improve traffic safety and road efficiency.

MEC's ability to provide localized computation and data services is particularly crucial for Day 2 and Day 3 Cooperative ITS (C-ITS) tasks, where real-time sensor fusion and high-throughput data exchange are required. These tasks include advanced use cases such as the detection and awareness of Vulnerable Road Users (VRUs), cooperative lane merging, or the execution of Cooperative Adaptive Cruise Control (CACC). In such scenarios, the amount of sensor data needing real-time processing is overwhelming to be offloaded to remote cloud infrastructures since it would incur excessive latency and bandwidth overhead at the core network. Instead, edge nodes located at base stations or roadside units can handle data close to the vehicles, analyzing environmental information (e.g., from LIDAR, radar, and cameras) with minimal delay and disseminating information quickly to neighboring vehicles and road infrastructure.

Moreover, MEC supports the integration of diverse communication technologies, such as Dedicated Short-Range Communication (DSRC) and Cellular V2X (C-V2X), acting as a unifying layer that ensures interoperability. This is especially relevant as 5G-based V2X services evolve, making the MEC architecture an enabler that fulfills Ultra-Reliable Low Latency Communication (URLLC) requirements. By abstracting underlying physical and network-layer differences, MEC enables applications and service developers to build and deploy use-case specific solutions (e.g., hazard notification or local map fusion) without being tied to a single network technology, thereby increasing robustness in multi-operator and multi-access scenarios.

At the core of the MEC architecture depicted in Figure 2.8 lies the MEC Host, a base component that combines computational resources with networking capabilities. Each host is equipped with a virtualization infrastructure that supports virtual machines or containers to deploy and manage edge applications dynamically. This infrastructure is crucial for handling diverse workloads efficiently, ensuring that computational resources are allocated dynamically based on application demands. Within the MEC Host, the MEC Platform plays a central role as the middleware layer. It orchestrates interactions between applications and the underlying virtualization infrastructure, creating an environment where developers can leverage standardized APIs to access a range of edge services, including traffic management,

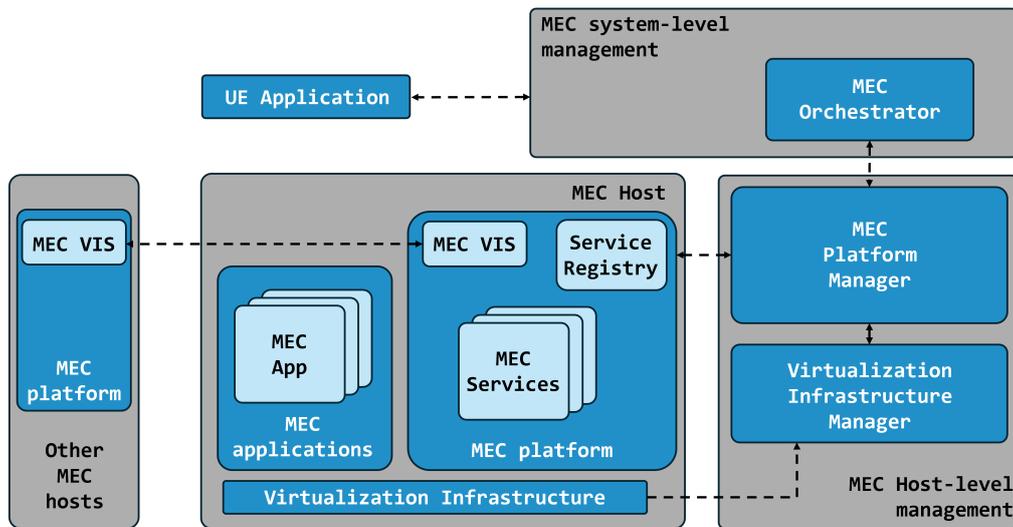


Fig. 2.8 MEC architecture according to [82][83].

location-based analytics, and real-time data processing. The MEC Platform also provides lifecycle management functionalities, such as deploying, scaling, and terminating applications. Furthermore, it performs the critical function of offloading data from the core network to the edge, reducing bandwidth usage and minimizing latency, which is essential for time-sensitive applications. MEC Applications are virtualized software instances hosted on the MEC platform within the MEC Host. These applications run as virtual machines (VMs) or containers provided as part of the application package, leveraging the MEC system's virtualization infrastructure. Once deployed, they can interact with the MEC platform to consume or provide services such as traffic management, real-time analytics, and location-based services. MEC applications are governed by rules and requirements, including resource needs, latency constraints, and dependencies on specific MEC services, which are validated by the MEC system-level management. They can also register runtime information, indicate availability, or prepare for user state relocation to ensure seamless operations. The MEC system-level management maintains operations across the MEC ecosystem including components such as the MEC Orchestrator. This component takes care of managing resource allocation, workload balancing, and policy compliance across multiple MEC hosts. At the host level, the MEC Platform Manager ensures localized control over applications and platform operations, while the Virtualization Infrastructure Manager oversees the efficient utilization of virtualized resources.

The development of MEC for V2X services is based on the need to support multi-access, multi-network, and multi-operator scenarios [83]. These scenarios account for the diverse configurations of V2X applications and involve various operational setups. In the scenarios where services are managed by the Original Equipment Manufacturer (OEM), V2X services may be delivered by multiple Mobile Network Operators (MNOs), either within a single country or across borders, requiring seamless service integration across different operators. Similarly, in the ITS Operator scenario, services are provided to vehicles from various manufacturers, often requiring nationwide coverage. This setup relies on using networks from multiple operators and MEC systems supplied by different vendors to ensure comprehensive service availability. To support these complex environments, MEC V2X Information Services (VIS) enable interoperability between MEC systems and ensure smooth data exchange across multi-operator setups. This is essential to maintain service continuity, particularly during roaming or cross-border transitions. In fact, MEC VIS facilitates service continuity in cross-border scenarios by exposing V2X service provisioning information to both MEC platforms and MEC applications. Thus, service consumers can query or subscribe to updates about the visiting Public Land Mobile Networks (PLMN), enabling UEs to adapt promptly during handovers. In particular, MEC VIS enables MEC applications within different MEC systems to securely exchange V2X-related information. In Section 4.2, a V2X service for cooperative perception is presented that supports cross-border scenarios, using the VIS features of MEC such as interoperability of multi-operators and integration of cross-APIs to ensure seamless continuity of services between countries.

## **2.4 AMQP 1.0 for IP-based V2X communications**

Day 2 services like the Collective Perception Service (CPS) and the up and coming Maneuver Coordination Service (MSC) being currently developed by ETSI are primarily designed to operate over GeoNetworking and BTP networking and layers. However, the evolution of supporting edge infrastructure (MEC) deployments necessitates alternatives that utilize IP-based communications. In particular, entities such as the C-Roads Platform have recognized the need for extending compatibility with modern networking infrastructures and have focused on providing comprehensive guidelines for deploying V2X services over IP networks. These guidelines integrate

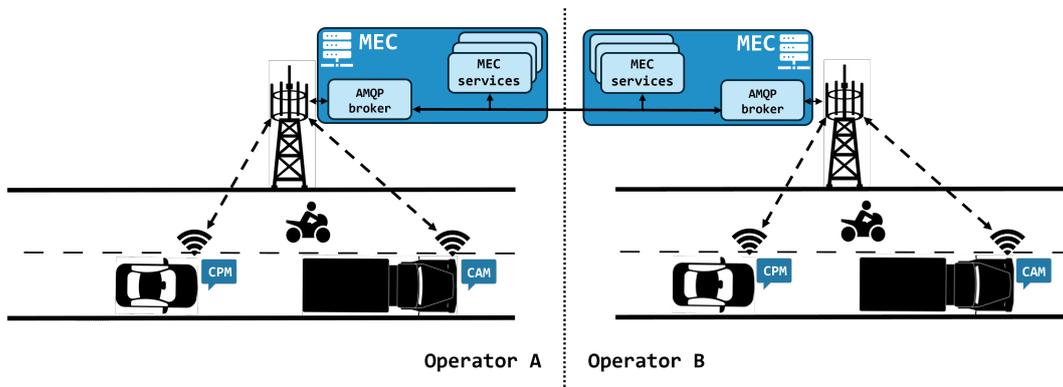


Fig. 2.9 Architecture for the exchange of C-ITS messages over AMQP for V2X services in a multi-operator scenario.

seamlessly with the capabilities of Multi-Access Edge Computing (MEC) architectures, enabling enhanced scalability, flexibility, and interoperability for ITS. For the execution of IP-based communications, C-Roads has defined the Basic Interface (BI) protocol for the communication between C-ITS actors leveraging the Advanced Message Queuing Protocol (AMQP) 1.0.

AMQP 1.0 is a versatile communication protocol designed to facilitate reliable message exchange between distributed systems. It operates at the application layer, providing a standardized solution for asynchronous message-oriented middleware, where messages can be produced and consumed independently, enabling loose coupling and scalability in system design. A key feature of AMQP is its ability to support a range of messaging patterns, including point-to-point and publish/subscribe. In particular, for point-to-point, AMQP operates under the concept of queues which serve as temporary storage buffers where messages are held until a consumer retrieves them. Queues operate in a First In, First Out (FIFO) manner, although priority-based processing is also supported. On the other hand, for the publish/subscribe scheme, AMQP 1.0 relies on topics where a single message can be delivered to multiple recipients based on their subscriptions. Messages published to a topic are routed dynamically to subscribers using routing keys or predefined rules, making topics highly effective for broadcasting information.

A key component of the protocol is the AMQP broker entity, serving as an intermediary that facilitates message exchange between producers and consumers. The AMQP broker is in charge of correctly routing messages providing load balancing, security, and filtering features. Indeed, AMQP subscriptions to a given topic can be

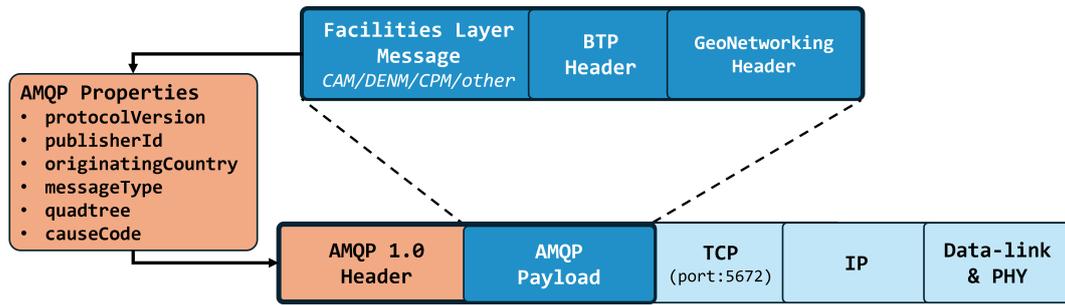


Fig. 2.10 C-ITS message encapsulation scheme as a payload of AMQP 1.0 packets.

further filtered according to specific values or ranges of values within the AMQP properties, either standard or customized, introduced in the AMQP header. In order to better extract the capabilities of AMQP 1.0 for V2X communications requirements, the C-Roads platform has defined in [54] a set of specifications for the exchange of AMQP messages under the BI protocol. These specifications include the encoding format, and mandatory AMQP properties to be included in messages. Besides filtering messages according to cause codes, C-Roads has devised a geographical filtering scheme based on Quadtree tiles. Quadtrees are hierarchical data structures that recursively divide a geographical area into quadrants, representing varying levels of detail through zoom levels. This approach ensures that AMQP brokers only route AMQP messages to C-ITS actors within relevant spatial boundaries according to the quadtree AMQP property value.

The BI architecture is designed to work in conjunction with the C-ITS stack defined in Section 2.1.2, by encapsulating C-ITS messages such as CAMs, DENMS and CPMs into AMQP messages. As showed in Figure 2.10, the BI client inserts the C-ITS message (including the Facilities layer message as well as BTP and GeoNetworking headers) as a payload, extracting the needed information for the AMQP properties setup. In this manner, the BI client (implementing an AMQP client) publishes these messages to an AMQP broker, which forwards them to other C-ITS actors and other AMQP brokers subscribed to the given type of C-ITS messages. Indeed, through inter-broker communications, AMQP 1.0 enables a horizontally scalable architecture for an enhanced execution of ITS applications. Indeed, AMQP brokers can subscribe to brokers in different countries, as shown in Figure 2.9, allowing C-ITS actors to perform cooperative tasks when crossing borders in an interrupted way. It is worth noting that the concept of C-ITS actors defined by C-Roads includes not only road actors (e.g., vehicles, RSUs, and VRUs),

but also MEC services executing ITS applications. Indeed, this architecture envisions use cases where MEC services deployed by entities, such as road operators, gather and process information from C-ITS messages to execute complex tasks such as computing the coordination of automated maneuvers.

## 2.5 Vehicular Micro Clouds

While MEC focuses on deploying computational resources at fixed network edges to enable low-latency, high-bandwidth access for resource-intensive services, the high deployment costs of edge servers have slowed their widespread adoption [124, 42]. In contrast, the rapid evolution of modern vehicles, already equipped with advanced computing power, sensors, and storage, has led to the widespread availability of resources on vehicles, giving birth to a novel paradigm known as Vehicular Micro Clouds (VMC). The root concept of VMCs, first introduced over 10 years ago as Vehicular Cloud Computing (VCC) [86], emerged as a branch of Mobile Cloud Computing, recognizing the availability of not only computational resources but also a comprehensive set of sensors, communication capabilities, and storage within vehicles. By leveraging these resources, VCC allows vehicles to collaborate in tasks such as data aggregation, environmental monitoring, traffic management, and infotainment. From the concept of VCC, many works have explored the clusterization of neighboring vehicles in a hierarchical scheme to share their resources and form autonomous Vehicular Micro Clouds (VMC). The primary objective of VMCs is to create an additional layer of edge computing that reduces the data sent to remote servers by enabling pre-processing and aggregation within the VMC itself. Furthermore, VMCs can provide on-demand services, not only to vehicles within the cluster but also to those outside it, for unexpected events that cannot be effectively addressed with pre-allocated centralized resources.

VMCs are formed by clustering vehicles based on their geographical proximity, movement patterns, or specific application requirements. Clustering algorithms dynamically group vehicles based on factors such as location, speed, direction, and network connectivity. This ensures that clusters are formed and dissolved as needed, adapting to traffic conditions and maintaining efficient resource utilization. Among the adopted schemes for clustering the more common strategy is geographic clustering, where vehicles within a defined area, such as near an intersection, can

form a cluster to share real-time traffic data or collaboratively manage traffic flow. Several works have explored a context-aware clustering scheme, where vehicles with similar objectives, such as navigation assistance (e.g., CACC) or environmental monitoring, can form clusters.

The VMC architecture, depicted in Figure 2.11, can be structured into three main layers: inside-vehicle layer, communication layer, and cloud layer [145]. The inside-vehicle layer focuses on collecting data from various sensors, such as environmental, body, and inertial navigation sensors, which monitor the driver's condition and vehicle status. This information is then relayed to the cloud for further processing. Vehicles are equipped with On-Board Units (OBUs) featuring broadband wireless communication technologies like DSRC (e.g., ITS-G5), LTE-V2X, or NR-V2X to support this exchange. The communication layer facilitates interactions through V2V and V2I/V2N communications. V2V enables vehicles to share emergency warnings and real-time behavioral data (e.g., CAMs and CPMs), while V2I/V2N links vehicles with external infrastructure, such as traffic management systems and cloud networks, to optimize safety and mobility. Finally, the cloud layer provides computational capabilities and data storage. It includes sub-layers, namely, the application layer, the infrastructure layer, and the platform layer. The application layer provides real-time services like Network as a Service (NaaS), Cooperation as a Service (CaaS), Information as a Service (INaaS), and Storage as a Service (STaaS). The infrastructure layer is in charge of handling data storage and computation tasks. The platform layer ensures seamless interoperability between different services enabling the virtualization of computing and storage resources.

As mentioned, VMCs have been thought to work under a hierarchical VCC framework. In this framework, VMCs represent localized clusters that handle tasks within a small area, providing immediate data processing and storage. Additionally, groups of micro clouds may interconnect to form city-wide or region-wide *macro clouds*, enabling more complex applications like large-scale traffic management or urban planning.

The diversity of VMC implementations allows for flexibility in addressing different scenarios and application requirements. Two main types of VMCs can be considered in terms of geographical deployment:

- **Stationary Micro Clouds:** Formed by parked vehicles, these clusters act as stable nodes for data storage and processing. For example, vehicles in parking

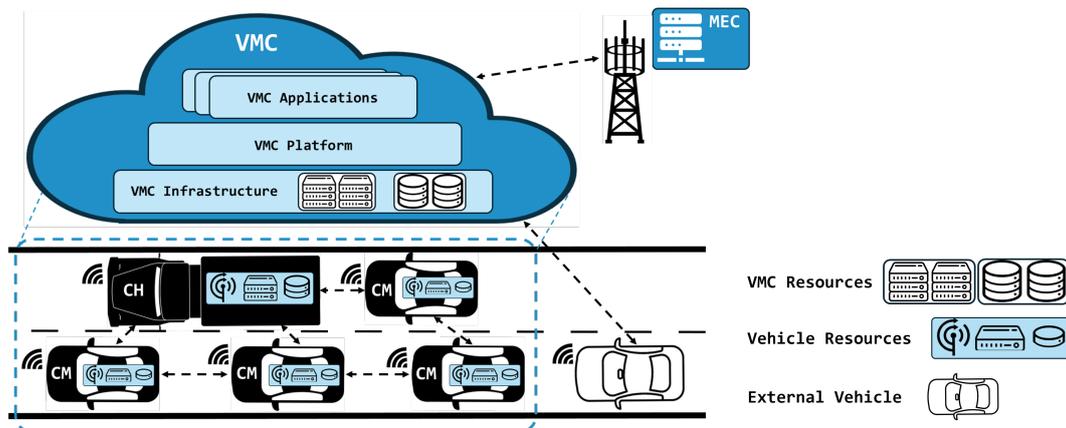


Fig. 2.11 Vehicular Micro Cloud (VMC) architecture. The Cluster Head (CH) and Cluster Members (CMs) aggregate and virtualize their resources through the VMC platform for the deployment of VMC applications. External Vehicles may subscribe to VMC applications without the need to connect to a centralized server (e.g., a MEC server).

lots can provide weather data or monitor environmental conditions in urban areas [66].

- **Mobile Micro Clouds:** These are dynamically formed by moving vehicles. They adapt to changing traffic patterns and are particularly effective for real-time applications such as environment monitoring and hazard detection.

In order for VMC to leverage vehicles to create a virtual computing platform, the following operations need to be supported [86]:

- **Cloud Resource Discovery:** When a vehicle runs an application, it becomes the cloud leader, identifying and recruiting nearby vehicles to form a vehicle cloud. The leader determines resource needs based on the application's requirements. It broadcasts a resource request to nodes within the range, and those willing to share resources respond with a resource reply detailing their capabilities.
- **Cloud Formation:** The cloud leader reviews resource replies and selects members to form the cloud, aiming to optimize resource usage while meeting the application's needs.
- **Task Assignment and Result Collection:** The leader divides the application into tasks and assigns them to members based on their resource availability.

It maintains a cloud table for tracking member IDs and task assignments. Members complete their tasks and return results to the leader.

- **Content Publishing and Sharing:** The leader processes task results to produce output content, which shall be stored or published. Published content is made accessible to the entire VMC, allowing vehicles to request and receive data from the cloud members.
- **Cloud Maintenance:** When a member leaves the cloud, it informs the leader, who replaces it with another node from the resource replies capable of completing the task. The leader redistributes the task and updates the cloud table.
- **Cloud Release:** Given the case where the cloud is no longer needed, or the leader moves out of range, it releases resources by notifying all members and clearing the cloud member list.

One of the main challenges of VMCs derives from the dynamic nature of its member nodes (i.e., vehicles). Indeed, vehicles belonging to a VMC may go out of range in a short span of time, which demands mechanisms to recover data from exiting nodes properly. Additionally, the constant mobility and availability of nodes pose challenges when selecting Cluster Heads (CH), especially for Mobile VMCs. A significant body of research has studied mechanisms for cluster creation and management aiming to reliably gather and aggregate data within VMCs [93][91].

Outside architectural challenges, another primary challenge to implementing VMCs is providing sufficient incentives to potential VMC members to share their resources willingly. Even though modern vehicles are equipped with enough computing power and storage to spare, sharing them implies additional power consumption with a non-negligible economic cost. Therefore, for the creation of VMCs new members shall be rewarded with economical or operational benefits that cover the incurred costs.

Many works have focused on strategies aiming to evaluate the feasibility of the VMC concept with novel cluster formation, cluster management, intra-VMC and inter-VMC routing [93, 92, 99, 64, 106, 91]. However, there is a research gap within the VMC literature on the evaluation of specific use-cases that could leverage the VMC architecture. In this thesis, the focus is given to one of the key services for CCAM applications, Cooperative Perception. As described in Section 2.1.2.1, ETSI

has standardized the service, calling it Collective Perception Service (CPS), to be deployed as a Facilities layer service. As described in Chapter 5, many challenges are faced towards the deployment of ETSI's CPS due to highly redundant information being exchanged among vehicles, product of the fully decentralized operation of the service. Here, an argument is that for efficient use of both computational and communication resources, the Cooperative Perception concept needs to be deployed under a hierarchical scheme, i.e., VMCs.

While DSRC can support low-latency data exchange within short ranges, C-V2X unlocks a more extensive network footprint as well as a more extensive device compatibility. By tapping into the broader ecosystem of cellular-equipped road users which includes not only vehicles but also VRUs, VMC applications can extend their cooperative services beyond vehicle clusters alone. This expanded reach is particularly valuable where infrastructure is sparse or intermittent, as micro-cloud resources could potentially be coordinated among a wider set of connected entities, further enhancing the scalability and resilience of collective perception and other VMC-driven functionalities.

Finally, as the focus lies on VMC applications rather than VMC formation, the research has concentrated on the *Task Assignment and Result Collection* and *Content Publishing and Sharing* operations mentioned earlier. To do so, we have chosen one of the primary CCAM use cases, platooning, as the foundation of our explored VMC architecture scheme. Indeed, platooning provides the ideal substrate for mobile VMC creation, as it involves vehicles traveling in close proximity for extended periods, forming a structured cluster with a predefined hierarchy. Moreover, as platooning is mainly thought to be deployed on highways where infrastructure presence is limited, it represents one of the use cases that would most benefit from the creation of a virtual edge server. Lastly, as one of the most prominent vehicular applications in the literature [52], platooning focuses on reducing fuel consumption and enhancing traffic safety. These economic and safety advantages inherently provide sufficient incentives for platooning vehicles to share their resources and deploy a VMC.

## Chapter 3

# Development of performance evaluation tools for CCAM

The push for safer, more efficient, and eco-friendly vehicles is driving a significant transformation in the automotive industry. A cornerstone of this progress is Vehicle-to-Everything (V2X) communication, which allows vehicles to connect with one another as well as with the infrastructure. Developing and deploying V2X technologies, along with applications aimed at improving road safety, presents considerable challenges. Testing these applications on real vehicles during early stages can introduce safety risks to drivers and other road users, especially considering cases where an application malfunctions or sends incorrect information to the vehicle's systems. Furthermore, testing on large fleets is costly, making it impractical for many scenarios. To overcome these hurdles, simulation and emulation tools play a crucial role in reducing risks and supporting the development of both applications and the underlying communication technologies.

When analyzing the available network simulation tools, OMNeT++ [5] provides a comprehensive set of wireless communication technologies models, an intuitive IDE, and statistics collection facilities. On the other hand, its main counterpart ns-3 [116] offers a similar range of communication models with a closer alignment to real-life UNIX implementations, providing models and APIs that resemble the system-level operations found in production environments. This allows developers to design, test, and update network applications in the simulation environment while maintaining compatibility with deployment-ready implementations for, e.g.,

---

on-board units, by employing functions resembling Linux system calls for packet handling. Furthermore, ns-3 supports seamless integration with external systems and frameworks. This capability further enhances its utility for emulating realistic application scenarios where hybrid systems combine both simulated and physical components. Another simulation framework for modeling communications worth mentioning is MATLAB. While robust for high-level modeling, it falls short in detailed network simulation compared to tools like ns-3 and OMNeT++. Its focus on mathematical abstraction over packet-level fidelity makes it less suitable for realistic protocol interactions or event-driven simulations. Additionally, as opposed to OMNeT++ and ns-3, MATLAB's proprietary nature and licensing costs limit accessibility, making it less ideal for open-source collaboration and research.

For the accurate evaluation of CCAM applications, realistic modeling of both mobility and vehicle sensing capabilities is essential. The base mobility models provided by OMNeT++ and ns-3 are overly simplistic, making them inadequate for representing complex vehicular dynamics and interactions. To address these shortcomings, simulators like SUMO [108] and CARLA [63] have become crucial tools. SUMO (Simulation of Urban Mobility) excels in detailed microscopic traffic modeling, offering efficient simulations of diverse traffic conditions. However, it lacks the ability to model vehicle sensor data, allowing only simple abstracted sensor data to be generated in simulations. Conversely, CARLA has become quite popular in recent years providing high-fidelity 3D environments, vehicle dynamics, and realistic sensor models (e.g., LiDAR, radar, and cameras), making it ideal for autonomous driving research and tasks requiring accurate perception data. While CARLA is decisively superior in terms of sensor modeling, its computational demands make it less efficient than SUMO for large-scale simulations. These trade-offs highlight the need for simulation frameworks that can flexibly incorporate both tools to cover all the diverse aspects of CCAM evaluations, from large-scale evaluations where sensor data can be abstracted to more specific use cases where the detail and accuracy of sensor-generated data and control dynamics are critical.

Among the available simulation frameworks for V2X, Veins [131] stands out as one of the most established platforms. It integrates SUMO and OMNeT++ through a comprehensive implementation of the TraCI (Traffic Control Interface) [143], enabling seamless interaction between traffic and network simulations. In addition to its native IEEE WAVE/802.11p capabilities, Veins has expanded its functionalities with projects like SimuLTE [140] and Simu5G [113], enabling simulations for LTE

and 5G networks. Extensions such as Artery and Vanetza [125, 43] have introduced a fully functional ETSI C-ITS model and a flexible prototyping platform for V2X applications. It is worth noting that using Artery and Vanetza requires manual installation of OMNeT++ libraries (if not using the offered bundled virtualized environment) and belonging to different projects, their latest developments are not always ensured to be compatible with each other.

Another prominent framework, iTETRIS [126], couples ns-3 with SUMO to provide IEEE 802.11p, UMTS, WiMAX, and DVB-H implementations. However, it lacks a fully integrated ETSI C-ITS stack in its original form. Moreover, its focus on access technologies leaves less emphasis on enabling streamlined application development and testing for ETSI-compliant vehicular use cases. WiLabV2Xsim [135], developed on MATLAB, supports access technologies like IEEE 802.11p, LTE-V2X, and NR-V2X. However, it lacks ETSI C-ITS stack support and relies solely on pre-recorded mobility traces, limiting its versatility for real-time testing. Furthermore, it primarily targets sidelink resource allocation analysis, rather than providing a comprehensive platform for developing vehicular applications.

Several CARLA-based frameworks facilitate Cooperative Driving Automation (CDA) testing, such as OpenCDA [149]. Still, a significant limitation is their neglect of realistic wireless communication models, often operating under oversimplified assumptions. Some studies have explored integrating OMNeT++ with CARLA as a mobility simulator to address this gap. For instance, the work in [51] describes a simulation setup combining CARLA and OMNeT++ using a ZeroMQ-based API. Additionally, an efficient co-simulation scheme bridging CARLA and Veins leveraging a gRPC interface is presented in [96]. Although these approaches use CARLA-generated mobility data for channel modeling and message simulations, they notably lack integration of sensor-generated data. Another effort, presented in [48], proposes a framework integrating Artery with SUMO and CARLA. By utilizing CARLA's SUMO co-simulation capabilities, vehicle behavior is controlled in SUMO and represented as network nodes in Artery via the TraCI API. Similarly, the work in [102] introduces a co-simulation platform combining Artery, CARLA via a Robot Operating System (ROS) network, emphasizing modular architecture and dynamic integration of communication services, which enhances flexibility and scalability for V2X-based Cooperative Driving Automation (CDA) scenarios. Despite generating sensor information, this framework does not support embedding such data into CPMs, which limits its application for cooperative perception scenarios.

In terms of Hardware-in-the-Loop (HIL) testing, existing solutions like the ones presented in [129], [112], and VENTOS [47] provide functional emulation capabilities but face challenges with scalability. OMNeT++-based approaches, such as Artery extensions and Simu5G emulation, simplify vehicular HIL testing but introduce delays or rely on traffic models that struggle to scale in real-time with a large number of vehicles. Furthermore, despite the growing support for ETSI MEC entities in Simu5G, large-scale vehicular emulation remains underexplored.

While there exists a wide range of simulation and emulation tools for V2X and CCAM scenarios, there's a need for a comprehensive, unified solution to address the shortcomings of the existing solutions. This thesis presents a framework designed to cover these gaps, integrating a full ETSI C-ITS stack alongside support for IEEE 802.11p, 3GPP LTE-V2X, LTE, and NR-V2X access technologies. This framework, called ms-van3t (Multi-Stack VANET framework for ns-3), offers seamless integration of SUMO, CARLA, and GNSS trace support with state-of-the-art access technologies models. It uniquely supports the efficient large-scale capabilities of SUMO for real-time simulation and large-scale emulation capabilities, as well as the realism and accuracy of CARLA for high-fidelity mobility, sensor, and control modeling, making it one of the most comprehensive solutions for V2X application development and testing.

### **3.1 The ms-van3t simulation framework**

The ms-van3t is an open-source framework integrating cutting-edge models for V2X communications, allowing simulated connected vehicles to flexibly utilize and switch between these models depending on scenario requirements. This feature, to the best of our knowledge, is unique among simulation tools with comparable support for emerging and established V2X technologies. A key strength of ms-van3t is its fully implemented ETSI C-ITS stack, built from the ground up to optimize ns-3 capabilities. This stack supports the two primary Day 1 Facilities layer services for the exchange of CAM (Cooperative Awareness Messages) containing vehicle dynamic status and attributes [70] and DENM (Decentralized Environmental Notification Messages) for event-driven road updates [71]. Additionally, Day 2 use cases are also supported through the implementation of standard-compliant facilities like the LDM (Local Dynamic Map) and Collective Perception Services for the exchange of

CPMs (described in Section 2.1.2.1). Furthermore, basic services for the exchange of IVIMs (Infrastructure to Vehicle Information Message) and VAMs (Vulnerable road user Awareness Messages) are also supported. Beyond simulation, ms-van3t enables Hardware-in-the-Loop (HIL) testing by emulating vehicle presence. Using SUMO to configure mobility scenarios, it can transmit and receive standard ETSI C-ITS messages through physical interfaces, facilitating real-world integration with V2X applications and actual vehicles.

The key capabilities of ms-van3t can be boiled down into the following:

- **Full implementation of the ETSI C-ITS stack:** offering a complete implementation of the Networking and Transport Layers standards presented in Section 2.1.2.2 as well as the implementation of the most relevant Facilities layer services for Day 1 and Day 2 applications for CCAM.
- **Multiple access technologies models:** ms-van3t enables the comparison of most of the technologies described in Chapter 2, namely, IEEE 802.11p, LTE-V2X and NR-V2X, under varying scenarios to evaluate their strengths and weaknesses.
- **Open-source codebase:** The framework is fully open-source, maintained on a public GitHub repository, and supports most Debian-based Linux distributions.
- **Utilization of Real GNSS Data:** Mobile nodes can leverage pre-recorded GNSS traces, allowing to simulate real-world errors instead of relying solely on synthetic SUMO-generated scenarios.
- **Map-based Visualization:** A web-based visualizer enables easy monitoring of mobile nodes on real maps. This visualization can dynamically incorporate data from real vehicles during emulation.
- **CARLA integration:** ms-van3t extends its capabilities by integrating with the CARLA simulator, enabling high-fidelity modeling of vehicular scenarios. This integration allows for the simulation of cooperative perception, including the generation and processing of Collective Perception Messages (CPMs) and realistic sensor data for CCAM applications.

A table summarizing all the available simulation solutions for V2X applications testing is shown in Figure 3.1 comparing their features with the ones present in

Comparison of ms-van3t features with available vehicular networks simulation/emulation solutions

Tool	Multi-stack	Large scale simulations	Native integration with SUMO	Native integration with CARLA	Pre-recorded GNSS traces	Emulation mode	Supported ETSI messages	Web-based visualizer	Open source
Veins	X 802.11p	✓	✓	✓	X	X	No ETSI stack	X	✓
Artery	X 802.11p	✓	✓	✓	X	X	CAM, DENM, MAPEM, SPATEM, CPM	X	✓
SimuLTE	X LTE	✓	X	X	X	X	No ETSI stack	X	✓
Simu5G	✓ LTE, 5G, NR-V2X Mode 1	✓	X	X	X	✓	No ETSI stack	X	✓
iTETRIS	✓ 802.11p, WiMAX, UMTS, DVB-H	✓	✓	X	X	X	CAM, DENM, LDM	X	✓
WiLabV2Xsim	✓ 802.11p, LTE-V2X Mode 4, NR-V2X Mode 2	✓	X	X	X	X	No ETSI stack	X	✓*
VENTOS	X 802.11p	✓	✓	X	X	✓	No ETSI stack	X	✓
RVE	X 802.11p	✓	✓	X	X	✓	No ETSI stack	X	X
ms-van3t	✓ 802.11p, LTE, Release 14 C-V2X Mode 4, NR-V2X Mode 2	✓	✓	✓	✓	✓	CAM, DENM, IVIM, CPM, LDM	✓	✓

\*Additionally to the MATLAB version, WiLabV2Xsim offers an Octave-based branch in their repository.

Fig. 3.1 Comparison of ms-van3t features with existing vehicular network simulation and emulation solutions.

ms-van3t. The next sections will briefly describe the framework, following the main key features showcased in the table, with a complete dedicated description of the ms-van3t integration with CARLA in Section 3.4.

## 3.2 ms-van3t features and architecture

The ms-van3t platform is built from multiple software modules that interact via specialized interfaces, creating a robust environment for V2X application testing. Its foundation includes SUMO [109] (v1.9.2), an open-source urban mobility simulator that dynamically models the positions of simulated or emulated nodes, and ns-3

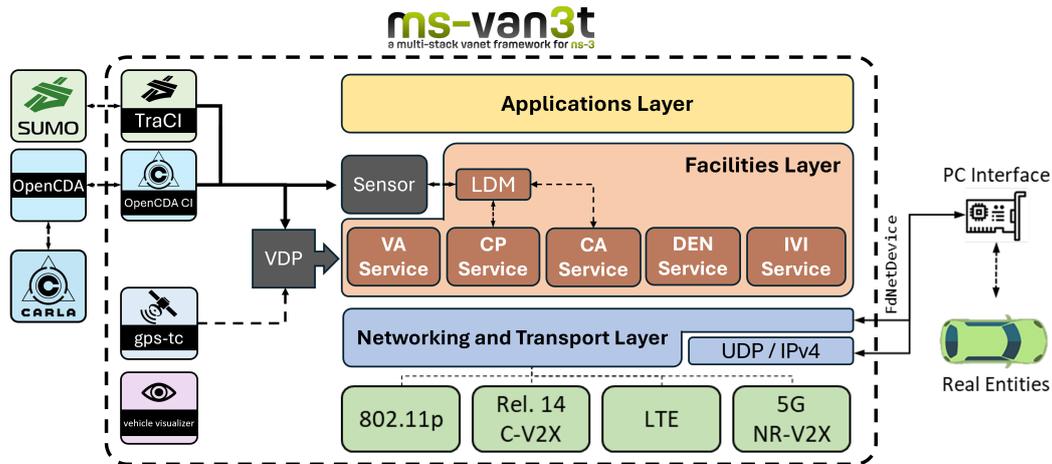


Fig. 3.2 Architecture of MS-VAN3T, featuring simulation and emulation capabilities alongside four state-of-the-art access layer models.

[116] (v3.35), a modular discrete-event simulator. Enhanced with features like the 3GPP Release 16 5G NR-V2X, the ETSI C-ITS stack, and various novel modules, ms-van3t extends ns-3's capabilities. Thanks to a user-friendly installation script, these additional modules are automatically integrated as libraries.

The platform supports large-scale simulations with access to a complete vehicular stack. Simulated vehicles can utilize different access technologies, represented in the architecture (Figure 3.2) by green-highlighted boxes. Vehicle mobility is managed either through SUMO, the default mode, or via pre-recorded GNSS traces, offering flexibility depending on the simulation requirements. Additionally, a web-based visualizer allows vehicles to be displayed on a map, which is particularly advantageous for scenarios utilizing GNSS trace data.

What sets ms-van3t apart from existing solutions is its ability to provide a comprehensive set of features as part of its core system, eliminating the need for additional extensions or plug-ins. Many alternatives available in the literature or market only offer subsets of these capabilities. By contrast, ms-van3t integrates all these functionalities into a unified platform. These comparative features are further detailed in Figure 3.1 and illustrated in Figure 3.2, showcasing the full architecture of this innovative solution.

### 3.2.1 Multi-stack

The simulation environment of ms-van3t is designed to simplify the development of vehicular applications by providing a high-level abstraction layer. This approach allows users to focus on application-specific tasks without delving into the complexities of the lower-layer communication stack, which functions as a "black box." By abstracting the lower layers, ms-van3t ensures ease of use while supporting the evaluation and deployment of vehicular applications across multiple technologies. Currently, ms-van3t supports the simulation of vehicular scenarios using four distinct access technologies, encompassing both established and emerging standards. These are implemented with cutting-edge models, detailed as follows:

1. **IEEE 802.11p:** vehicles are equipped with IEEE 802.11p-compliant On-Board Units (OBUs) for V2V or V2I communications. For V2V scenarios vehicles directly exchange messages whereas for V2I scenarios, Road Side Units (RSUs) equipped with 802.11p interfaces receive vehicle-generated messages and run centralized applications. This technology fully supports the ETSI BTP [69] and GeoNetworking [72] protocols, allowing entities to broadcast messages according to the ETSI ITS-G5 guidelines.
2. **3GPP LTE:** each vehicle in the simulation is equipped with User Equipment (UE) connected via the Uu interface to an eNB. Vehicle-to-Network (V2N) scenarios can be modeled with this technology with a centralized application placed in a remote server. The centralized application running in a simulated remote host is made reachable through the Evolved Packet Core (EPC). Communication involves encapsulating messages in UDP/IP headers and transmitting them as unicast packets, ensuring compatibility with LTE's IP-based addressing.
3. **3GPP Release 14 LTE-V2X:** this model, presented in [65] implements 3GPP's Release 14 specifications for sidelink communications where vehicles are equipped with *PC5* interfaces configured in Transmission Mode 4. The model simulates out-of-coverage V2V communication, where vehicles broadcast their messages leveraging BTP and GeoNetworking for message delivery.
4. **3GPP Release 16 5G NR-V2X:** this extension of the 5G-LENA simulator proposed in [45] models vehicles equipped *PC5* interfaces for Sidelink communications, following the specifications for Transmission Mode 2. Through

this model, vehicles implement the Semi-Persistent Scheduling algorithm described in Section 2.2.2 for resource allocation in out-of-coverage scenarios. Notably, random resource selection, introduced in 3GPP Release 17, is supported in addition to SPS. Similarly to the LTE-V2X model, BTP and GeoNetworking are fully exploited, allowing for the comparison of all the relevant access technologies for V2V communications under the C-ITS stack.

### 3.2.2 Large scale simulations

A standout feature of `ms-van3t` is its robust simulation environment, designed to support the testing of large-scale vehicular communication scenarios. Given the significant costs associated with equipping numerous vehicles for real-world testing, simulation becomes an essential step in the evaluation of V2X applications. `ms-van3t` addresses this by enabling the simulation of an arbitrary number of vehicles, constrained only by the hardware capacity of the device hosting the framework.

A common challenge in large-scale simulations is efficient data collection. Gathering metrics for a substantial number of nodes and processing them for system-wide statistics can be complex and time-consuming. To overcome this, `ms-van3t` incorporates the *MetricSupervisor* module, which automates and simplifies the collection of key metrics. Specifically, this module provides two primary metrics:

- **Packet Reception Ratio (PRR):** as defined by 3GPP TR 36.885 [38], this metric quantifies packet loss caused by collisions or adverse propagation conditions.
- **One-Way Latency:** this measures the time between a packet's transmission and its reception by nearby vehicles. Latency is calculated each time a message is received and averaged over all received packets throughout the simulation.

These metrics are automatically processed and presented as average values at the end of each simulation. Results can be exported as CSV files, streamlining post-simulation data analysis. Additionally, the *MetricSupervisor* is extendable, supporting the collection of other metrics such as Packet Inter-Reception (PIR). It is worth noting that the *MetricSupervisor* operates transparently across all underlying access technologies, simplifying the comparison of different communication protocols.

### 3.2.3 Native integration with SUMO

ms-van3t achieves seamless integration with the SUMO simulator [109], leveraging its robust vehicle mobility models to emulate a variety of realistic urban and highway scenarios. Through this integration, every vehicle in the simulation is equipped with a full ETSI C-ITS stack via ns-3, providing network connectivity for developing and testing V2X applications. This capability is made possible by the `traci` module, adapted from [88], which implements the TraCI interface. This integration establishes a bidirectional coupling between ns-3 and SUMO, enabling both the direct control of the vehicle dynamics from ns-3 as well as the retrieval of relevant vehicle data from SUMO, such as position, velocity, and acceleration. This functionality allows for the simulation of advanced scenarios, including environments with partially or fully autonomous vehicles, where connected vehicles can actively respond to network-controlled dynamics.

Moreover, ms-van3t's TraCI implementation extends its support to Vulnerable Road Users (VRUs), such as pedestrians and cyclist, who can be included in SUMO simulations alongside connected and non-connected vehicles. This inclusion enables the design of complex scenarios involving interactions between vehicles and VRUs. Even if VRU Awareness Messages (VAMs) are a recent addition to ETSI standards [73], their integration into ms-van3t is fully supported with an implementation of the VRU Awareness service.

Finally, a SUMO-based sensor module is offered where, similarly to other solutions for collective perception [90][46] analysis, a simulated radar sensor can be placed on the top of each simulated vehicle. This sensor identifies objects within its line of sight and operates over a configurable range, which defaults to 50 meters. Every 100 milliseconds, the sensor module communicates with SUMO to query the positions of all vehicles within its detection range. During the detection process, vehicles that are obstructed are filtered out, ensuring only relevant objects are considered. Gaussian noise is then applied to the measurements, with the noise level increasing proportionally to the distance between the detected object and the sensor. In addition to spatial information, additional measurements such as speed and heading are recorded. These measurements are also subject to Gaussian noise, which is introduced based on the following mathematical expressions:

$$d_s = d_{gt} + \delta \cdot \frac{d_{gt}}{r_s} \quad (3.1)$$

$$[w_s, l_s] = [w_{gt}, l_{gt}] \cdot \left(1 + \delta \cdot \frac{d_{gt}}{10 \cdot r_s}\right) \quad (3.2)$$

$$h_s = h_{gt} + \theta \cdot \frac{d_{gt}}{r_s} \quad (3.3)$$

$$v_s = v_{gt} + \nu \cdot \frac{d_{gt}}{r_s} \quad (3.4)$$

where the stored measurements, denoted as  $d_s$ ,  $[w_s, l_s]$ ,  $h_s$  and  $v_s$ , correspond to the detected distance, width, length, heading, and speed of an object. These are derived from the respective ground truth values provided by TraCI, represented as  $d_{gt}$ ,  $[w_{gt}, l_{gt}]$ ,  $h_{gt}$  and  $v_{gt}$ . To introduce realism, Gaussian noise is added to each ground truth measurement. This noise, represented by  $\delta$ ,  $\theta$  and  $\nu$ , is generated with a mean of zero and specific standard deviations: 1.0 m for distance, 0.01 rad for heading, and 0.5 m/s for speed. The noise gain, normalized by the sensor range  $r_s$ , is represented as  $d_{gt}/r_s$  ensuring that the noise increases proportionally with the distance of the object from the sensor.

### 3.2.4 Pre-recorded GNSS traces

A distinctive feature of ms-van3t is its ability to utilize pre-recorded GNSS traces for vehicle positioning, alongside SUMO and CARLA, offering flexibility in mobility modeling. Unlike earlier works that integrated GNSS traces with outdated frameworks like ns-2 [104] [57], ms-van3t supports modern technology models and a full C-ITS stack. The `gps-tc` module enables the use of GNSS data by removing SUMO and CARLA from the simulation loop, allowing users to incorporate realistic positioning data and simulate real-world GNSS errors, particularly impactful in urban environments [118].

This module accepts GNSS data in CSV format with customizable fields, assigning each vehicle a mobility model based on its trace, which can be replayed synchronously or with custom delays. To ensure accuracy, it converts geodetic coordinates to Cartesian coordinates using a Transverse Mercator projection centered on the average longitude, minimizing distortion and avoiding UTM-related issues. Efficient projection algorithms from the GeographicLib library [103] are integrated into the framework, ensuring consistent and reliable positioning for V2X simulations.

### 3.2.5 Emulation Mode

Beyond offering a robust simulation environment, ms-van3t extends its functionality to support emulation and interaction with external entities, including real vehicles. This capability facilitates Hardware-in-the-Loop (HIL) testing for V2X applications and enables hybrid scenarios where emulated vehicles—whether driven by SUMO or pre-recorded GNSS traces via the `gps-tc` module—can exchange standard-compliant messages with real vehicles, and vice versa. For instance, a real vehicle can seamlessly communicate with multiple emulated vehicles in its vicinity, enabling developers to evaluate the behavior of V2X applications on several nodes while only equipping a single real vehicle, significantly reducing costs. The architecture supporting this emulation mode, including its connection to physical interfaces, is illustrated in Figure 3.2. The emulation mode is built on the `FdNetDevice`, a network device type in ns-3 that facilitates communication with the external world by routing packets to a physical interface instead of relying on a simulated access layer. This setup allows ms-van3t to handle the transmission and reception of standard-compliant messages such as CAMs, CPMs, DENMs, and IVIMs, enabling real vehicles to participate in the simulated scenario. ms-van3t supports two emulation operational modes:

- **Standard mode:** in this mode, packets generated by the ETSI Application, Facilities, and Network and Transport layers are directly broadcast through the physical interface.
- **UDP mode:** here, packets from the same layers are encapsulated within UDP and IP headers, ready for delivery to a server, either on the Internet or located at an RSU.

It is worth noting that in both modes, ms-van3t can receive standard-compliant messages through the physical interface, facilitating bidirectional communication. A notable use case of ms-van3t's emulation capabilities involves validating real-world vehicular services, such as the Server Local Dynamic Map (S-LDM), as it will be described in Chapter 4.

### 3.2.6 ETSI C-ITS stack

To enable standard-compliant simulations in vehicular communication scenarios, `ms-van3t` incorporates a comprehensive implementation of the ETSI C-ITS stack, described in Chapter 1. The `automotive` module encapsulates the complete ETSI C-ITS stack, encompassing application logic, message encoding and decoding mechanisms, and the networking logic required for vehicular communication as defined by ETSI standards. This robust implementation enables vehicles to exchange standard-compliant messages within simulations, ensuring realistic and interoperable V2X communication scenarios. Validation of this implementation is further supported by interoperability tests presented in Section 3.3.2, demonstrating `ms-van3t`'s ability to communicate with real V2X entities effectively.

One of the key strengths of the ETSI C-ITS stack in `ms-van3t` is its compatibility with multiple access layer technologies, such as IEEE 802.11p, LTE-V2X and NR-V2X. This flexibility ensures that the upper layers of the stack can seamlessly integrate with the MAC and physical layers of state-of-the-art models described earlier. This compatibility allows researchers and developers to test various configurations and technologies within a unified simulation environment.

#### 3.2.6.1 Facilities Layer

The Facilities Layer in `ms-van3t` acts as a middleware, bridging the Application and Transport layers by providing essential services for message management as defined by ETSI standards. Currently, the framework offers a full implementation of the following services:

- Cooperative Awareness (CA) Service [70].
- Decentralized Environmental Notification (DEN) Service [71].
- Infrastructure to Vehicle Information (IVI) Service [75].
- Collective Perception (CP) Service [77].
- Local Dynamic Map (LDM) [68].
- Vulnerable Road User Awareness (VA) Services [73].

To comply with ETSI standards, all messages are encoded and decoded using ASN.1 (Abstract Syntax Notation One), a universal method for defining complex data structures in a platform and language independent manner. This approach allows the automatic generation of encoding and decoding functions from ASN.1 description files using tools like `asn1c` [87]. A significant advantage of this method is that the generated code is not exclusive to ms-van3t, making it reusable in standalone implementations. Building on the generated ASN.1 code, ms-van3t provides user-friendly implementations for the core services, accessible through a variety of intuitive functions. These include:

- Functions to start and stop the dissemination of CAMs, CPMs and VAMs.
- Functions to manage DENMs, including populating their content, defining the destination area, and triggering, updating, or canceling transmissions. Users can also configure repetition intervals for periodic DENM broadcasts, adhering to ETSI specifications.
- Functions to fill-in the content of IVIMs, and control their transmission [75].
- Functions providing an easy interface to set and configure optional containers in CAMs, CPMs, VAMs, DENMs and IVIMs.
- Function to set customized callbacks for applications invoked when messages are received by Facilities layer services during a simulation or emulation session.
- Functions to retrieve, insert, update and delete information stored in the LDM, including interfaces with CA and CP services.

In our implementation, the LDM keeps track of all neighboring connected vehicles through the reception of CAMs from which the implemented LDM interface stores the information. All the detected objects by the sensor module (introduced in 3.2.3), go through a matching process with the already stored elements of the LDM. This matching step recognizes whether a newly sensed object is actually a connected vehicle known from a previous perception cycle, thereby preventing the vehicle from being erroneously registered twice. The CP service queries the implemented LDM interface for available information about detected objects, which is then used to

create CPMs. Furthermore, the implemented CP service applies the generation rules outlined in [77] for redundancy mitigation.

All the generated messages are passed to the Transport Layer, implementing the Basic Transport Protocol (BTP).

### 3.2.6.2 Basic Transport Protocol

The Facilities Layer is bridged by the BTP to efficiently interact with the GeoNetworking Protocol by facilitating the exchange of control messages while ensuring minimal resource overhead. Similarly to the User Datagram Protocol (UDP), BTP operates as a connectionless protocol. According to its standard definition, the BTP module in the ms-van3t framework utilizes ports for multiplexing and demultiplexing messages. This mechanism ensures a seamless flow of data between the Facilities Layer and the GeoNetworking entity, supporting bidirectional communication effectively.

### 3.2.6.3 GeoNetworking

GeoNetworking (GN) is a protocol tailored for the Cooperative Intelligent Transport Systems (C-ITS) Networking Layer, specifically designed to address the complexities of ad-hoc communications. It incorporates geographical addressing and forwarding capabilities, which enable the efficient routing of packets based on geographic information. Within the ms-van3t framework, the GN module supports two primary forwarding schemes for message dissemination:

- **Single-Hop Broadcast (SHB):** This method is used for CAMs, CPMs and VAMs, limiting their broadcast to the first reachable hop, ensuring minimal propagation and low overhead for frequent, short-range updates.
- **Geo-Broadcast (GBC):** This approach is utilized for DENMs and IVIMs. It targets dissemination to a defined geographical area (GeoArea) as specified by the DEN Basic Service, making it particularly suitable for disseminating location-specific alerts and notifications.

The management of network functionalities required for transmitting and receiving GN packets, such as address configuration, updating the ego vehicle's position

vector, and periodic beaconing, relies on geographical data provided by a Vehicle Data Provider (VDP). This provider can be integrated with the SUMO through the TraCI interface, with CARLA with our custom interface detailed in Section 3.4, or linked to GNSS traces via the `gps-tc` module.

### 3.3 Evaluation and results

To comprehensively validate our framework’s simulation and emulation capabilities, we have considered two sets of Key Performance Indicators (KPIs), each devised to evaluate these distinct functionalities. The first set of KPIs selected are meant to compare the performance of the different access technologies offered in the framework. These network-related KPIs are assessed in Section 3.3.1. On the other hand, the framework’s ability to handle real-time emulation of a substantial number of vehicles is validated to demonstrate scalability and practical applicability in Section 3.3.2.

#### 3.3.1 Performance comparison between V2X access technologies

The primary aim of the results presented is not to deliver a comprehensive comparison of access technologies. Instead, the focus is on demonstrating the effectiveness of `ms-van3t` as a versatile framework for evaluating these technologies, particularly in large-scale scenarios and during pre-deployment phases. In this context, a specific scenario is examined involving a custom-designed highway loop created using SUMO, featuring two lanes per direction and a total circumference of 3 km. The technologies analyzed within the framework include LTE, IEEE 802.11p, Release 14 LTE-V2X Mode 4, and Release 16 NR-V2X Mode 2.

To ensure fair comparisons between LTE and the other technologies using direct device-to-device communication, LTE is configured with an eNB strategically positioned at the center of the scenario. This eNB connects to a server functioning as a message relay. In the LTE setup, V2X messages are first transmitted to the eNB and subsequently forwarded as a series of unicast packets to all vehicles within its coverage area.

Parameter	Value
Number of simulations for each point	10
Duration of each simulation	200 s
Vehicle density	3 veh/km → 33 veh/km
<b>IEEE 802.11p</b>	
Transmission power	23 dBm, 30 dBm, 33 dBm
Center frequency	5.9 GHz @ 10 MHz
Physical data rate	3 Mbit/s
Propagation Loss Model	WINNER B1 (urban microcell)
<b>LTE-V2X</b>	
Transmission power	23 dBm, 30 dBm, 33 dBm
Center frequency	5.9 GHz @ 10 MHz
MCS	20
Probability of keeping the resources in SPS	0
Resource Reservation Interval	20 ms
Propagation Loss Model	WINNER B1 (urban microcell)
<b>NR-V2X</b>	
Transmission power	23 dBm, 30 dBm, 33 dBm
Center frequency	5.9 GHz @ 10 MHz
MCS	20
Probability of keeping the resources in SPS	0
Resource Reservation Interval	20 ms
Propagation Loss Model	WINNER B1 (urban microcell)
<b>LTE</b>	
Transmission power (eNB)	30 dBm
Transmission power (UE)	10 dBm
Center frequency	2.12 GHz @ 25 MHz (DL), 1.93 GHz @ 25 MHz (UL)
MCS	Adaptive [60]
Propagation Loss Model	Friis

Table 3.1 Main simulation parameters for the evaluation of the network KPIs.

The evaluation of the technologies centers on two key performance metrics: one-way latency and Packet Reception Ratio (PRR). The latter, synonymous with Packet Delivery Ratio (PDR) in common terminology, is used here following the naming convention of the 3GPP Technical Reports [38]. PRR serves as a measure of network reliability, reflecting packet loss caused by adverse propagation conditions or channel congestion.

To calculate PRR, the methodology adheres to the specifications outlined in 3GPP TR 36.885 [19]. A baseline distance around the transmitting vehicle, varying between 100 and 200 meters, is defined. For each transmitted message, PRR is computed as the ratio of the number of vehicles within this distance that successfully

receive the message ( $X$ ) to the total number of vehicles within the same distance ( $Y$ ):

$$PRR_i = \frac{X_i}{Y_i}, \quad i = 1, \dots, n_{Tx} \quad (3.5)$$

The resulting PRR values for all  $n_{Tx}$  transmitted messages are then averaged to produce the final metric. Intuitively, a PRR value of 1 signifies flawless communication with no packet loss, reflecting high stability. Conversely, a PRR of 0 indicates that no communication is achievable.

Each metric has been averaged across 10 simulations, each lasting 200 seconds and characterized by unique traffic patterns. Vehicle density was varied extensively, spanning from 3 vehicles/km to approximately 33 vehicles/km. Additionally, three transmission power levels—20 dBm, 26 dBm, and 30 dBm—were tested, leveraging the framework’s capability to adjust the transmission power of the On-Board Units (OBU) or User Equipment (UE).

To ensure consistency, IEEE 802.11p, LTE-V2X, and NR-V2X were configured to operate within the Dedicated Short-Range Communications (DSRC) spectrum at 5.9 GHz, while LTE utilized band 1 at 2.1 GHz. Specific configurations included a physical data rate of 3 Mbit/s for IEEE 802.11p, and a channel bandwidth of 10 MHz, a Resource Reservation Interval (RRI) of 20 ms, a probability of resource retention of 0, and a Modulation and Coding Scheme (MCS) of 20 for LTE-V2X and NR-V2X. These parameter settings were selected as they yielded optimal performance for the technologies in the evaluated scenario. A detailed summary of the simulation parameters is provided in Table 3.1.

A key parameter influencing latency performance for LTE-V2X and NR-V2X is the Resource Reservation Interval (RRI). Within the context of the C-V2X Semi-Persistent Scheduling (SPS) mechanism, the RRI determines the periodicity at which a vehicle reserves the same resources for a series of consecutive transmissions [94]. Following a set number of reservations, these resources can be reselected based on a configurable probability between 0.2 and 1. The probability of retaining the same resources is the inverse of this value, which may range from 0 to 0.8.

The analysis of one-way latency, as illustrated in Figure 3.3, is plotted as a function of the vehicle density. IEEE 802.11p emerges as the one with best performance, consistently achieving latencies around 0.5 ms. NR-V2X demonstrates latency values around 10 ms, with a slight peak of 12 ms under heavy congestion. LTE-V2X consis-

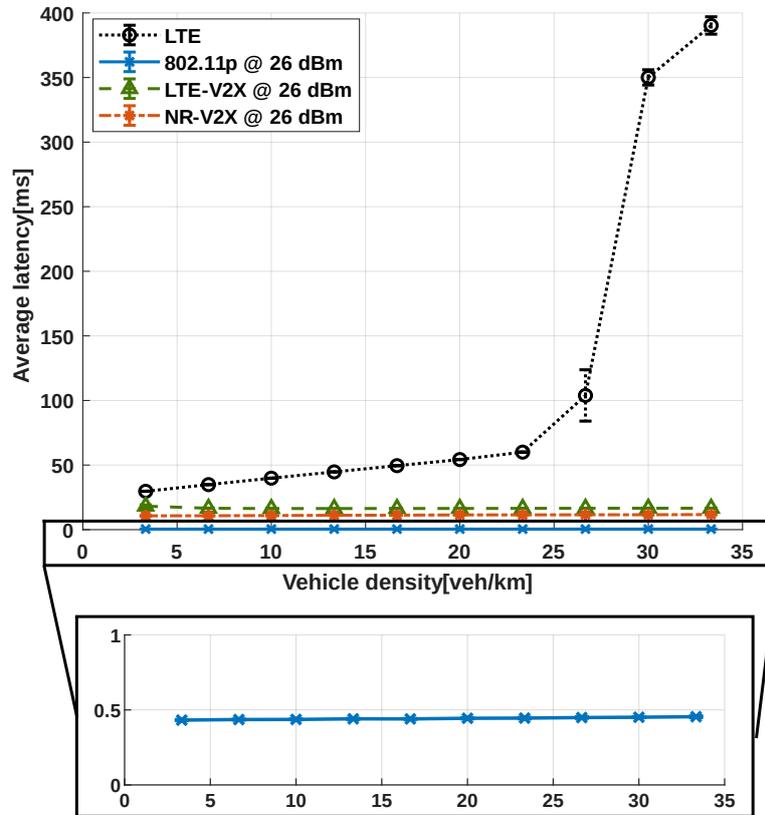


Fig. 3.3 One-way latency as a function of the vehicle spatial density. IEEE 802.11p, LTE-V2X, and NR-V2X show a stable and low latency. Vehicles communicating with LTE, instead, experience a higher latency, reaching 390 ms under very high traffic congestion.

tently maintains latency around 16 ms across most densities. However, LTE displays significantly poorer performance at higher densities, with latencies exceeding 350 ms. This is attributed to the fact that the eNB is working as a bridge between UEs and creating a broadcast storm effect from relayed traffic in high-density scenarios.

The average Packet Reception Ratio (PRR), depicted in Figure 3.4, reflects a similar pattern. While LTE suffers a pronounced performance decline at high vehicle densities, it outperforms other technologies at densities up to 23.33 veh/km, leveraging the eNB's higher transmission power. Conversely, LTE-V2X shows slight susceptibility to congestion, yet all access technologies maintain PRR values exceeding 0.97, signifying stable performance.

Focusing on a vehicle density of 20 veh/km, the PRR as a function of baseline distance is illustrated in Figure 3.5. At a 200 m baseline, cellular technologies

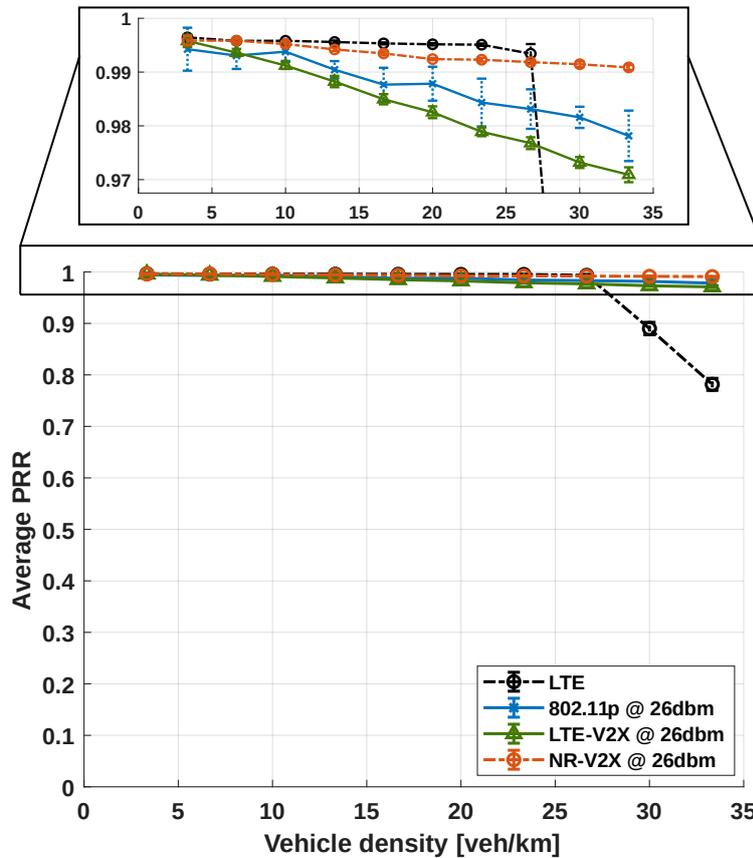


Fig. 3.4 PRR as a function of the vehicle density, for a baseline distance of 150 m and transmission power level of 26 dBm.

(LTE-V2X, NR-V2X, and LTE) excel, delivering over 98% PRR for LTE-V2X and over 99% for LTE and NR-V2X. IEEE 802.11p, at a transmission power of 30 dBm, achieves similar PRR values to LTE-V2X, just below 98%. However, its performance drops to nearly 80% at 26 dBm, reflecting its limitations in propagation range. This indicates that DSRC-based IEEE 802.11p requires higher transmission power to reach vehicles effectively within the baseline. Notably, IEEE 802.11p can deliver over 99% of packets at shorter baselines (e.g., 100 m) and with sufficient transmission power (>26 dBm). This performance aligns with findings in [115], where IEEE 802.11p demonstrated slightly lower PRR than LTE-V2X at comparable baseline distances.

It is important to mention that the observed results are influenced by the propagation loss models employed in the simulations. In particular, the NR-V2X module uses a default model based on [39] while for 802.11p a Log-distance model is used by

default. To ensure a fair comparison, the Friis model was adopted for LTE, while the WINNER B1 model (urban microcell, per [38]) was applied to LTE-V2X, NR-V2X, and IEEE 802.11p. Additionally, default PHY layer settings (e.g., SNR and RSSI sensitivity) were preserved across technologies, except for the standardized path loss models for IEEE 802.11p, LTE-V2X, and NR-V2X.

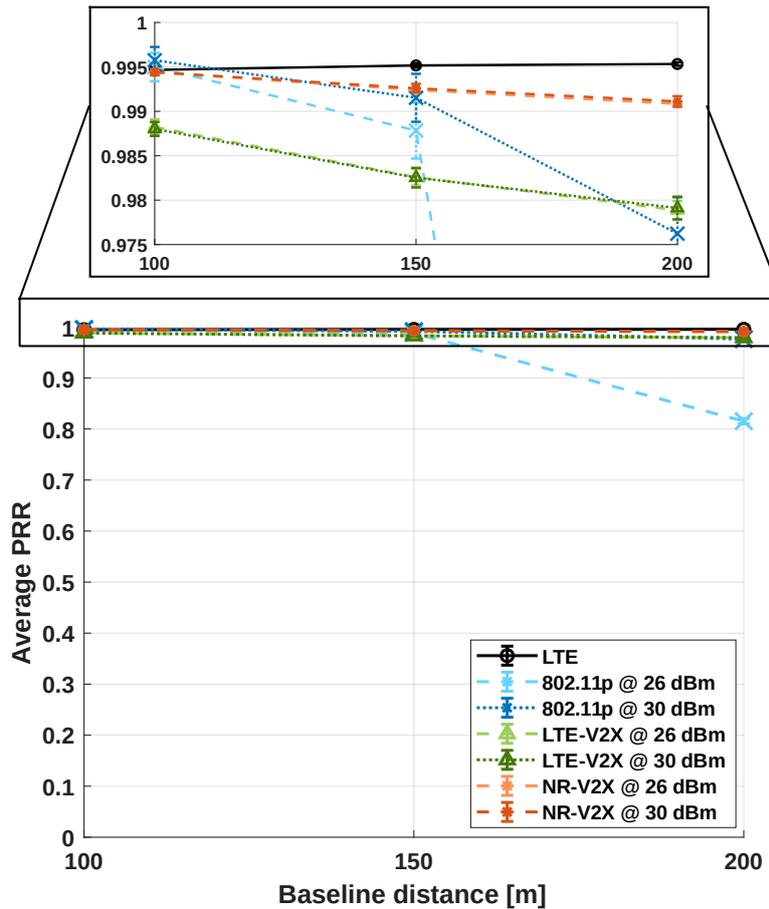


Fig. 3.5 PRR as a function of the baseline distance from the sender, with at the top the zoom in of the region between 0.97 and 1, where the advantages of using cellular-based technologies are evident. Vehicle density: 20 veh/km.

To summarize, IEEE 802.11p stands out for delivering the lowest latency among the evaluated options, however, in terms of Packet Reception Ratio (PRR), cellular-based technologies surpass it. It is fair to say that LTE-V2X Mode 4 and NR-V2X Mode 2 offer a well-balanced compromise between latency and PRR, making them favorable choices for many scenarios. NR-V2X demonstrates superior performance compared to LTE-V2X, providing slightly better PRR, reduced latency, and enhanced robustness against channel congestion. Meanwhile, LTE excels in scenarios with

low vehicle densities, where its infrastructure-based communication allows for the highest PRR. However, as vehicle density increases, LTE experiences a significant performance decline, highlighting its limitations in high-density environments.

Finally, the study showcases the utility of the *ms-van3t* framework for simulating and emulating V2X scenarios. Thanks to the *MetricSupervisor* module, it enables seamless collection and export of performance metrics across various access-layer technologies.

### 3.3.2 Performance evaluation of emulation capabilities

To evaluate the interoperability of *ms-van3t* with real nodes, multiple tests were conducted relying on a commercial ETSI C-ITS stack. The chosen stack was a recent release of the V2X SDK from Cohda Wireless, commonly used with their MK5 and MK6 boards. This SDK includes a self-contained Linux-based virtual machine environment, capable of emulating vehicles on a PC and running Cohda V2X applications before actual deployment in vehicles. Leveraging this virtual environment, we successfully executed the Cohda Wireless stack on standard laptops, bypassing the need to directly use the MK5 hardware.

For the tests, we have set up two laptops in our laboratory to simulate a V2X communication scenario:

- **Cohda SDK laptop:** running Ubuntu 18.04 LTS, it hosted the Cohda Wireless virtual environment to emulate a vehicle and handle the transmission and reception of V2X packets over a physical interface.
- **ms-van3t laptop:** running Ubuntu 20.04 LTS, used to execute *ms-van3t* in emulation mode.

The laptops were interconnected via a Gigabit Ethernet connection through our lab's switched LAN, forming the experimental setup shown in Figure 3.6.

Although no wireless channel was established between the two devices, the experimental setup allowed us to successfully validate *ms-van3t*'s ability to interact with real nodes using standard-compliant messages. Notably, transitioning from an Ethernet connection to a wireless one would be straightforward. For instance, the switched LAN could easily be replaced with a pair of embedded boards offering



Fig. 3.6 Laboratory setup for the interoperability tests, performed to validate the emulation feature available in ms-van3t.

IEEE 802.11p connectivity, similar to the approach described in our previous work [123].

As an initial interoperability test, a simple scenario with a limited number of vehicles was configured in ms-van3t. From this test, both message reception from and transmission to the ETSI-compliant commercial stack worked accordingly. Specifically, the virtual vehicle running the Cohda Wireless SDK received and successfully decoded all emulated messages from the ms-van3t vehicles and vice versa, validating ms-van3t's C-ITS stack implementation.

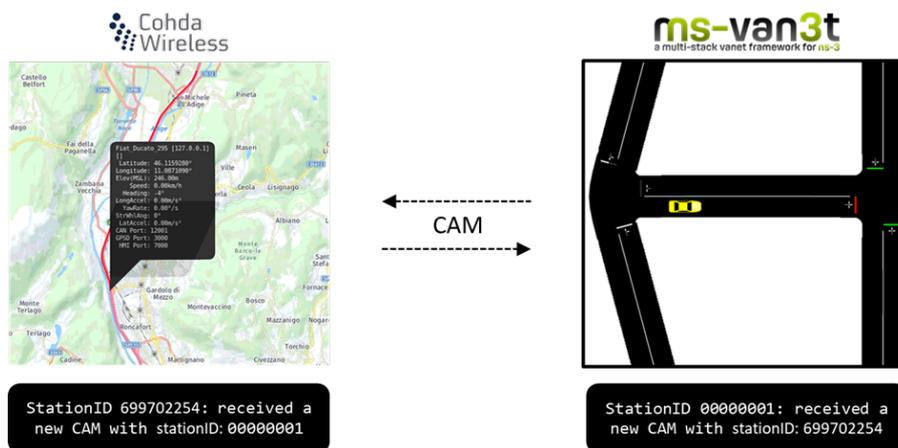


Fig. 3.7 Communication between ms-van3t (configured to emulate one vehicle with stationID equal to 1) and a commercial ETSI C-ITS stack (configured to emulate a vehicle with stationID 699702254).

Figure 3.7 illustrates the experimental setup, with the Cohda emulator’s GUI and its output during a CAM exchange test displayed on the left, and the SUMO GUI alongside the ms-van3t output shown on the right. It is important to note that the Cohda Wireless stack was configured with security protocols disabled. This adjustment was necessary because ms-van3t currently supports only non-secured GeoNetworking messages. This configuration allowed seamless communication between the Cohda stack and ms-van3t, facilitating the validation of message exchange functionality under these constraints.

In addition to the initial interoperability test, we assessed ms-van3t’s ability to emulate a large number of vehicles in real-time. This evaluation built upon the setup described in Section 3.3.1, configuring a more complex scenario within ms-van3t. Within this scenario, one of the simulated vehicles was designated as the Device Under Test (DUT). During the tests, every message received by the DUT from other simulated vehicles followed the following path: after being transmitted over the simulated wireless link, the message was processed by the simulated MAC layer and then forwarded to a physical network interface. The primary objective was to evaluate the framework’s performance in emulating a high-density vehicle environment, where numerous vehicles transmit messages to the DUT, enabling Hardware-in-the-Loop (HIL) testing. Although no direct wireless channel was established between the physical network devices, ns-3 computed the propagation loss and channel congestion using its well-known models. Specifically, we utilized the IEEE 802.11p model detailed in Section 3.2, with all emulated OBUs configured to transmit at a power level of 23 dBm. This approach ensured realistic emulation of wireless communication dynamics in the absence of physical wireless connections.

To test the real-time emulation capabilities of ms-van3t, an ad-hoc application was developed for the Cohda SDK device. This application sent Cooperative Awareness Messages (CAMs) to ms-van3t while also receiving messages from all emulated vehicles. During the tests, the application calculated the average Packets-Per-Second (PPS) received over the entire 1,000-second emulation duration. Vehicle density in ms-van3t has been increased from 20 to 150 vehicles across tests. The measured PPS values were then compared to a set of expected values, derived from prior simulations of the system, which did not consider real-time constraints. These expected values from simulations represent the ideal PPS achievable under optimal conditions in real-time emulation. As the vehicle density increases, any significant deviation

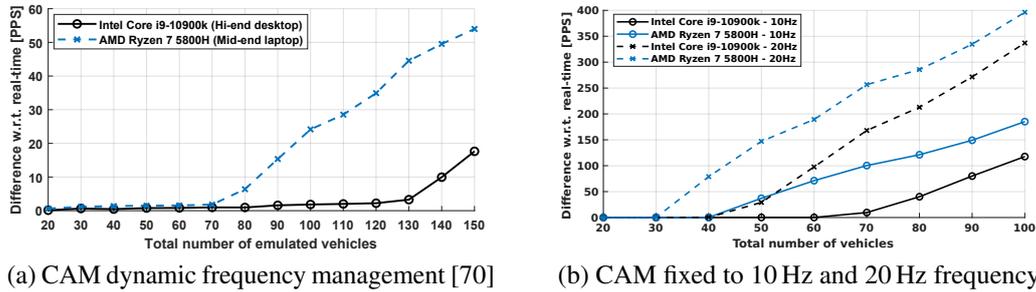


Fig. 3.8 Difference between the achieved PPS on the Cohda stack and the expected PPS, for different CPUs and as a function of the total number of vehicles emulated by ms-van3t.

between observed and expected values indicates the point where ms-van3t began to exceed its real-time emulation capacity.

The two main factors influencing the real-time performance are the number of emulated nodes and the amount of traffic generated by each. In order to demonstrate the effects of these two factors, in addition to increasing the number of emulated nodes for each test, we have also considered two different traffic generation scenarios. The first scenario represents the ETSI-compliant dynamic periodicity for disseminating CAMs according to the rules defined in [70]. In the second scenario, instead, the CAM generation frequency has been fixed, considering a frequency of 10Hz and 20Hz on each emulated node. Different from the first scenario, with a fixed frequency, the total number of scheduled packets presents a linear behavior with respect to the number of emulated nodes.

Figure 3.8 illustrates the performance of ms-van3t under the different vehicle densities. The x-axis represents the total number of vehicles emulated by ms-van3t, while the y-axis shows the difference between the achieved Packets-Per-Second (PPS) on the Cohda stack (receiving packets from ms-van3t) and the expected PPS derived from simulations.

The experiments extended to two different hardware setups, a mid-range AMD Ryzen 7 5800H laptop and a high-performance Intel Core i9-10900K desktop PC, to explore how computing resources affect ms-van3t's capacity to emulate vehicles in real-time. These tests also evaluated how the system performs under different CAM frequency configurations, including dynamic frequency management (as per ETSI standards) and fixed frequencies of 10 Hz and 20 Hz. The fixed frequencies,

while challenging, provided insights into worst-case scenarios where traffic loads are consistently high.

On the AMD Ryzen laptop, ms-van3t could emulate up to 70 vehicles in real-time with dynamic CAM frequency, dropping in performance to 40 and 30 vehicles for fixed frequencies of 10 Hz and 20 Hz, respectively. The high-performance Intel Core i9 desktop nearly doubled these numbers, handling 120 vehicles for dynamic generation and 60 and 40 vehicles at 10 Hz and 20 Hz fixed frequencies. These results highlight that even standard hardware is capable of supporting a significant number of vehicles in emulation. The higher capacity of the Intel desktop reflects the impact of hardware capabilities, revealing that ms-van3t showcases a performance increase with improved computing resources.

The achieved performance compares favorably with other tools in the literature. For instance, the HIL-exclusive framework presented in [112] supported a maximum of 85 vehicles, while ms-van3t achieves comparable results and offers broader simulation functionalities. Tools like VENTOS [47], while also incorporating emulation, lack extensive evaluations for high-vehicle-density scenarios, with their studies limited to four vehicles.

A key observation lies in how the traffic volume heavily influences real-time emulation. Indeed, doubling the packet frequency significantly reduces the number of vehicles the system can emulate and accelerates performance degradation as the load increases. Moreover, the tests revealed that single-core CPU performance is the most critical factor for real-time emulation. With this consideration, as faster CPUs become available, ms-van3t could handle even larger emulated scenarios. Parallelizing the framework's code to better utilize multiple CPU cores is another promising possibility for future enhancements.

Beyond emulation limits, ms-van3t demonstrated good flexibility in enabling communication between emulated vehicles and real-world entities. Vehicles emulated by ms-van3t can connect with physical devices, such as real vehicles or Road Side Units (RSUs). Essentially, users can configure the following type of setups:

- **One-to-one mapping:** where each simulated vehicle links to a dedicated network interface for fine-grained control.
- **Many-to-One Mapping:** where all simulated vehicle traffic is routed through a single or limited number of interfaces for a large-scale integration.

For instance, a single IEEE 802.11p device connected via Ethernet can act as a bridge between the emulated and real-world vehicles, seamlessly merging the simulated and physical environments as demonstrated in the measurements performed for the results in Figure 3.8.

### **3.4 Extending ms-van3t for realistic perception and automation modelling with CARLA simulator**

Evaluating the benefits of CCAM applications demands for simulation frameworks to accurately and jointly model mobility, sensor data generation, and wireless channel communication. The ms-van3t framework was initially integrated with SUMO, which excels at large-scale, efficient mobility simulation. However, a significant limitation of SUMO is its inability to model sensors, which restricts the simulation of cooperative perception scenarios. Conversely, the CARLA simulator represents a more advanced solution, offering not only mobility simulation but also life-like scenario rendering and realistic sensor models, such as LiDAR, radar, and cameras. These features enable CARLA to produce sensor outputs suitable for the evaluation of real-time computer vision tasks.

To expand ms-van3t's capabilities, CARLA has been incorporated to simulate vehicle mobility and sensor perception. The integration leverages the OpenCDA framework to develop an LDM (Local Dynamic Map) facility module and a gRPC-based Control Interface, enabling the extraction of CARLA's localization and perception data. Additionally, a corresponding gRPC-based client module was developed within ns-3 to query data from CARLA. In this manner, the data retrieved from CARLA updates both ns-3's node mobility and the LDM facility, thereby enabling the simulated vehicular network to transmit the sensor-perception data.

Remarkably, this is the first framework to offer CARLA and ns-3 co-simulation in a manner that goes beyond vehicle mobility simulation. Indeed, ms-van3t supports the retrieval of data from CARLA to be then used for the encoding and dissemination of CPMs over the ns-3 simulated network. As mentioned in Section 3.1, this section provides a detailed description of the architecture developed for the integration, which has been used for the evaluations described in Chapter 5 towards the analysis of a Vehicular-Micro-Cloud-based CCAM application.

Bringing ns-3 and CARLA together in ms-van3t posed several challenges. First, ns-3's C++ foundation had to be carefully bridged to CARLA's Python API, requiring an efficient interface layer for data exchange between the two without introducing unnecessary delays or synchronization issues. Second, ns-3's event-based, single-process structure created the need for a robust scheme where each simulator could pause or resume in a coordinated way, preventing the faster-moving system from overrunning the slower one. This synchronous execution was crucial to keep simulation time consistent and reduce performance bottlenecks as the vehicle count grew. Finally, since ms-van3t already included a SUMO-based sensor model and an LDM module, adapting the CARLA client was necessary to ensure that the newly integrated components aligned with existing data flows and sustained the expected behavior from the framework's established modules. The rest of this chapter will explore the specific implementation strategies employed to address these challenges.

### 3.5 Architecture of ms-van3t's CARLA integration

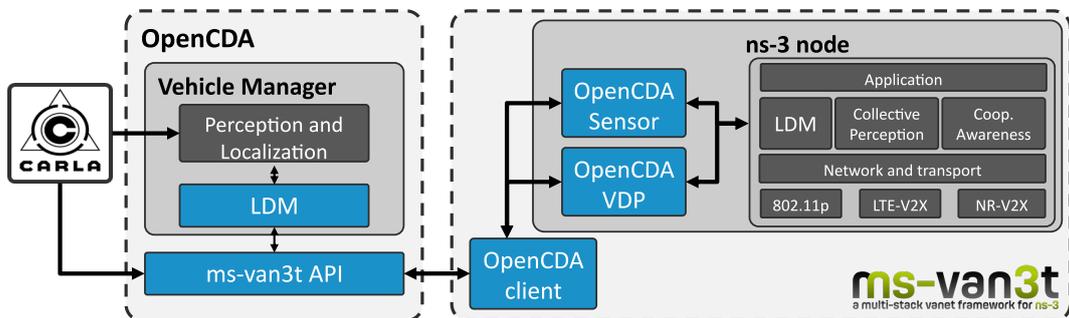


Fig. 3.9 Architecture of ms-van3t-CARLA.

As detailed in Section 3.2.3, ms-van3t offers a native integration with SUMO, allowing seamless interaction between the traffic and network simulations through the TraCI interface. Similarly, it supports simulating node mobility with pre-recorded GNSS traces, described in Section 3.2.4.

To facilitate consistent mobility management across mobility models, ms-van3t employs the Vehicle Data Provider (VDP) class. This abstraction layer allows mobility information to be queried uniformly, regardless of the underlying mobility handler.

Expanding on this approach, ms-van3t-CARLA integrates CARLA's vehicle movements with ns-3 nodes, synchronizing their mobility states. Google Protobuf and gRPC are utilized to establish a connection between ms-van3t and a Control Interface, developed as a module within the OpenCDA framework. This interface allows clients to query mobility data and integrate it seamlessly with ns-3. Inspired by the architecture outlined in [96], which bridges CARLA with the Veins framework, our integration incorporates a gRPC client in ms-van3t. This client facilitates the transmission of both mobility and perception information from OpenCDA to applications operating atop the ETSI C-ITS stack of each connected node.

A detailed explanation of the developed modules in each framework is provided, aligning with the architecture depicted in Figure 3.9. This description aims to illustrate the cohesive integration between the frameworks and their functional components.

### 3.5.1 OpenCDA Modules

#### 3.5.1.1 LDM

The perception module within OpenCDA has been utilized to implement a Local Dynamic Map (LDM) facility that tracks detected objects during each simulation step. This implementation aligns with ETSI's vision for the LDM, which involves matching locally perceived objects (detected during each sensor period) and objects received from other connected vehicles through Cooperative Perception Messages (CPMs) with existing tracks stored in the LDM's database. The object-matching process is formulated as an assignment problem whose objective is to optimally assign detections from the current frame to existing tracks or, when necessary, create new tracks. To execute the assignment task, the algorithm provided by the SciPy Python package is employed. This algorithm, as detailed in [59], solves the assignment problem using a cost matrix. The matrix is constructed based on two key metrics: the Intersection Over Union (IoU) between new detections and existing tracks and the spatial distance ( $d$ ) between them. Thus, for  $N$  objects stored in the LDM and  $M$  objects obtained in a given detection period, each value  $c_{n,m}$  of the  $N \times M$  cost matrix  $C$  is computed as follow:

$$c_{n,m} = \begin{cases} IOU_{n,m} & \text{if } IOU_{n,m} > 0 \\ d_{n,m} - 1000 & \text{if } IOU_{n,m} = 0 \text{ and } d_{n,m} \leq 3 \\ -1000 & \text{otherwise} \end{cases} \quad (3.6)$$

The constructed cost matrix  $C$  is then used as input to the Scipy's *linear assignment* function, which employs the Hungarian algorithm [105], finding the optimal pairing of detections and LDM tracks that minimizes the overall cost.

Additionally, in order to compute a speed and acceleration estimation for each object, the LDM facility uses a Kalman filter considering a constant acceleration model. Thus, for each track, the Kalman filter is updated before attempting to match the new detections in order to decrease false negatives and to increase the probability of re-matching objects that might become momentarily out of sight.

Each entry of the LDM database contains several metadata values in order to support Collective Perception and Cooperative Awareness services needs:

- **Path history:** a list of the last 10 positions detected.
- **Timestamp** of the latest detection
- **Connected flag:** indicating if such entry is about a connected vehicle or not (e.g., a legacy non-connected vehicle or another type of detected object). If this flag is set to true, this entry is not considered to be included in a CPM.
- **On-sight flag:** indicating if the object (or connected vehicle) is currently under perception range. If this flag is set to false, this entry is not considered to be included in a CPM.
- **Tracked flag:** indicating if the object has been on sight for at least 10 perception periods (i.e., detections have been matched with this LDM object for at least 1 second). If this flag is set to false, this entry is not considered to be included in a CPM (due to high speed and acceleration uncertainty).

The LDM provides 2 methods to handle new non-local perceptions, one for new connected vehicles, i.e., received from a CAM, and new detected object perceptions, i.e., received from a CPM. In the case of a newly connected vehicle, the matching process with local perceptions is done only for the first time the information is received, and no matter the outcome, the LDM entry adopts the connected vehicle's

unique CAM ID for future updates. In addition to receiving their CAMs, connected vehicles are also detected by sensors. In such cases, we overwrite sensor-detected data with the vehicle's CAM information because it is typically more accurate, thus avoiding confusion between the same vehicle's CAM-based and sensor-based stored information. On the other hand, for detected object perceptions from CPMs, each one goes through the matching process described before as if it were a new local perception.

### 3.5.1.2 OpenCDA Control Interface

The OpenCDA Control Interface (OpenCDA CI) module functions as a gRPC server, utilizing Protobuf-defined messages to facilitate communication with OpenCDA, as inspired by the concept in [96]. Its design aligns with OpenCDA's simulation setup, which integrates both Connected Autonomous Vehicles (CAVs) equipped with perception systems and background vehicles. The module specifically extracts Local Dynamic Map (LDM) information from CAVs, whereas for background vehicles, the information is limited to mobility data.

CARLA operates in synchronous mode, ensuring each simulation step corresponds to a fixed time interval. This interval matches the mobility step duration within the `ms-van3t` simulation. However, given that computer vision algorithms used for sensor perception require significant processing time, the OpenCDA CI ensures synchronization by completing a step only after all CAVs have updated their perception status. The OpenCDA CI plays a crucial role in this process, encoding the perception data extracted from CAVs' LDM into Protobuf messages using the actor ID of the vehicle whose perception data is being updated. Beyond perception updates, `ms-van3t` directs CARLA's simulation ticks and ensures the coordination of OpenCDA's simulated systems.

Additionally, the OpenCDA CI provides a mechanism to apply control inputs for simulated CAVs as dictated by applications or C-ITS facilities defined in `ms-van3t`. For background vehicles, control responsibilities are always handled by CARLA's traffic manager, further distinguishing the operational scope of the module for different vehicle categories.

## 3.5.2 ms-van3t modules

### 3.5.2.1 OpenCDA Client

All gRPC interfaces with the OpenCDA CI are implemented in the OpenCDA Client module, enabling seamless integration with the OpenCDA simulation. This module is responsible for initiating the simulation, establishing connectivity with the OpenCDA CI, and controlling each mobility simulation step. A notable feature of the OpenCDA Client is its ability to support a variable penetration rate, allowing a flexible proportion of background vehicles to be equipped with wireless communication devices. This flexibility is crucial for evaluating the impact of Collective Perception on factors like channel load and KPIs such as perception accuracy and Age of Information (AoI).

Given the limitations of ns-3 in dynamically creating network nodes during simulation runtime, the OpenCDA Client preemptively generates a pool of nodes at the simulation's start. These nodes are then assigned to CARLA actors as needed, pairing each ns-3 node with a corresponding CARLA actor based on the actor's ID. This association enables the installed user-defined applications and C-ITS stack to interact with the OpenCDA Client for mobility information. To maintain accurate channel modeling, the OpenCDA Client updates the mobility state of every ns-3 nodes at each mobility simulation step. Finally, the OpenCDA client offers a dedicated interface to the measurements module of ms-van3t described in Section 3.2.2, enabling the evaluation of communication performance for each simulated message. Using a configured baseline distance and the mobility state of vehicles, this interface facilitates the computation of PRR and one-way latency metrics.

### 3.5.2.2 OpenCDA VDP

As highlighted in Section 3.5, ms-van3t includes a Vehicle Data Provider (VDP) module that standardizes access to mobility information from all applications and layers of the C-ITS stack, irrespective of the underlying mobility simulation platform. In line with this structure, the CARLA VDP is implemented to retrieve required data from the CARLA simulation via the OpenCDA client. This module plays a vital role in enabling services such as the LDM and CP service by providing simulation data needed for message generation. In particular, it facilitates access to information

like the location and speed of the transmitting vehicle and the specifications of the sensors used to detect objects.

The OpenCDA VDP is instantiated at the start of the application with a unique ID assigned by the OpenCDA client. This ID remains constant for the duration of the simulation, enabling consistent and precise querying of actor-specific data. Through this ID, the VDP interacts with the OpenCDA client to obtain relevant information about a specific actor.

### 3.5.2.3 OpenCDA Sensor

The OpenCDA Sensor module manages the flow of perception information for a given CAV, retrieving data from the OpenCDA LDM and relaying information from received V2X messages back to it. The LDM implementation in `ms-van3t` mirrors OpenCDA's format, enabling the seamless storage of object and connected vehicle perceptions. Designed to be mobility-model agnostic, `ms-van3t`'s LDM implementation integrates with both CARLA and SUMO simulations without requiring changes to the applications or C-ITS facilities interacting with it.

To ensure efficiency and consistency, the OpenCDA Sensor module periodically synchronizes the LDM in OpenCDA with one on `ns-3`. This approach allows OpenCDA's LDM to focus on computationally intensive tasks like data matching and fusion while leveraging the `ns-3` LDM interfaces for fast data retrieval during simulations. By reducing the number of OpenCDA client calls compared to using only OpenCDA's LDM, this synchronization approach significantly enhances simulation efficiency. The synchronization process is executed in three steps:

- **V2X data extraction:** during each synchronization period, all new perceptions received via V2X messages (e.g., CAMs and CPMs) are extracted from the `ns-3` LDM and transferred to the OpenCDA LDM.
- **Data Matching:** the OpenCDA LDM processes these perceptions, performing data matching according to the mechanism detailed in Section 3.5.1.1.
- **State Update and Cleanup:** after the matching step, the updated state of the OpenCDA LDM is retrieved and synchronized with the `ns-3` LDM. Any perceptions in the `ns-3` LDM that were matched during this process are removed to maintain consistency.

### 3.6 Simulation Setup

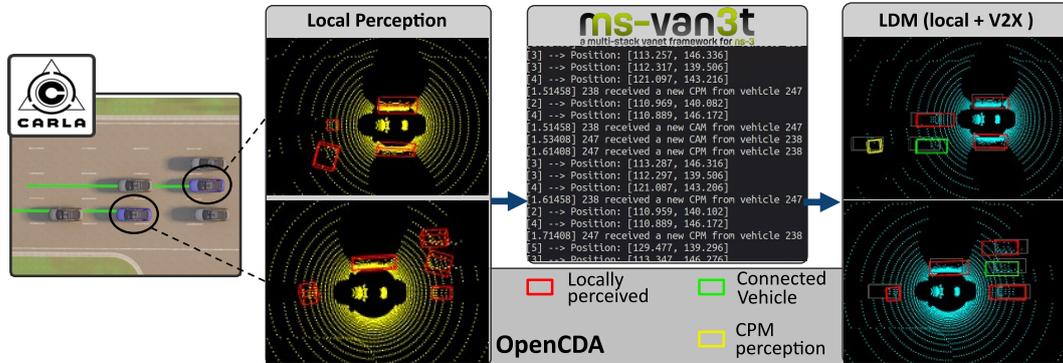


Fig. 3.10 LDM representation over the LiDAR view offered in OpenCDA, showcasing the different types of perceptions according to CAM, CPM or local detection.

Scenario creation with ms-van3t’s CARLA integration involves configuring two separate files: one for mobility setup and another for communication setup. The mobility configuration is defined in a YAML file, following OpenCDA’s format. In this file, users specify spawn points and destinations for both background vehicles and Connected Autonomous Vehicles (CAVs). Additionally, it allows users to configure simulated perception systems for each CAV, including settings for sensors such as RGB cameras and LiDAR. On the other hand, communication setup is handled within an ns-3 simulation script written in C++.

To enhance user understanding of CAV awareness, OpenCDA’s LDM module provides a Bird’s Eye View (BEV) of its database. This visualization highlights different perception sources—local detection, CAM, and CPM—alongside CARLA’s ground truth. Figure 3.10 illustrates a basic scenario where two CAVs travel along a road amidst several non-connected vehicles. Each CAV independently processes its sensor data, producing a set of detected objects. Differences in detections arise due to varying Points of View (POV), as reflected in the local perception images. Through the OpenCDA sensor, perception data from CAVs is updated in the ns-3 LDM and transmitted over the simulated wireless channel in ETSI-standard CPMs. Similarly, each CAV uses the OpenCDA VDP to encode its mobility data into CAMs. These messages are received and stored in the ns-3 LDM, which later synchronizes with the OpenCDA LDM.

As depicted on the right side of Figure 3.10, the OpenCDA LDM processes and fuses data received via V2X messages. This fusion combines the diverse

POVs to create enhanced context awareness. While this example involves only two CAVs, scaling to larger numbers of simulated vehicles allows for deeper analysis. Specifically, the effects of CAMs and CPMs on Key Performance Indicators (KPIs) like Packet Reception Ratio can be examined using ns-3 communication models, providing valuable insights into system performance under varying conditions.

Leveraging ms-van3t's CARLA integration, the joint evaluation of vehicle control, sensor-generated data and wireless communications enables the combined analysis of all these aspects in the execution of CCAM applications. As a matter of fact, thanks to these capabilities, an extensive evaluation of a Cooperative perception scheme based on Vehicular Micro Clouds has been performed, presented in Chapter 5.

## **Chapter 4**

# **Leveraging Connected Vehicle Resources through MEC-Based Services**

In the evolution of C-ITS, one of the key advancements is the transition of computational resources from a centralized cloud infrastructure to the network edge. Multi-access Edge Computing (MEC) represents the foundational framework for this paradigm shift, enabling low-latency, high-bandwidth services that are essential for safety-critical and time-sensitive applications. By deploying computational resources closer to the vehicles, MEC serves as a critical enabler for resource-intensive cooperative perception services, where real-time data processing and sharing among vehicles and infrastructure are paramount. Indeed, the capabilities provided by MEC allow for the implementation of centralized cooperative perception services able to provide timely instructions to safely perform highly automated maneuvers.

Even though centralized services for managing automated maneuvers necessitate handling significant volumes of data from vehicles, only a pre-processed subset of these data is usually effectively used. Typically, only the information about several vehicles or a specific road segment, such as a "map" of a given area needs to be provided to autonomous control systems. Thus, the ability to efficiently process and deliver data to both centralized systems and vehicles is essential, demanding high reliability and low latency to guarantee timely responses.

Given the central role of MEC in bringing computation to the edge, it provides a logical starting point for developing and evaluating cooperative perception services. The service presented in this chapter leverages MEC infrastructure to deliver cooperative perception capabilities to subscribing vehicles. This approach lays the groundwork by demonstrating the feasibility and performance of edge-based cooperative perception, which addresses key challenges such as latency, reliability, and scalability.

## 4.1 Review on LDM concept and centralized implementations

Since the SAFESPOT Project introduced the Local Dynamic Map (LDM) concept [9], efforts by organizations like ETSI and the International Organization for Standardization (ISO) have focused on its standardization. As described in Section 2.1.2.1, ETSI defines the LDM [68] as an Information Support facility that stores context information, accessible to C-ITS applications on demand. According to ETSI's specifications, the LDM operates on ITS Stations (ITS-S), such as vehicles or Road-Side Units (RSUs). It combines static and dynamic data to interpret surroundings. To enhance the interpretation possible with onboard sensors, the LDM also interfaces with services like the CA Service [70], the CP Service [77], and the DEN Service [71], all of which act as LDM data providers. This allows the LDM to store information from CAMs, CPMs, and DENMs shared by other ITS-S in an interoperable LDM Data Object format, accessible to registered applications for tasks such as tracking vehicles or road events. Although the distributed exchange of CPMs enables vehicles to share sensor data, this significant exchange of data introduces challenges like increased channel utilization [133], and computational burdens, as demonstrated in [141], where storing data from multiple vehicles impacts the LDM's update efficiency.

The Mobile Edge Computing (MEC) paradigm makes the deployment of centralized LDM services at the network edge possible, enhancing computational capabilities and the support of diverse use cases. For example, the authors in [119] proposed an LDM module using SQL-based databases to store CAMs and DENMs, detect potential collisions, and notify vehicles via DENMs. Similarly, [50] focused

on modular LDM services, providing data for intersection control applications, while [139] emphasizes data serialization (e.g., JSON) and forwarding via MQTT. These implementations, however, rely solely on SQL-based databases and often cater to specific applications, limiting the reuse of stored information for broader services.

Other centralized LDM solutions explore integrating information about detected objects. For instance, [107] proposes a service consolidating raw sensor data from multiple vehicles to offload computational demands. However, this approach sacrifices scalability, as raw data is significantly larger than post-processed CPMs. In contrast, [85] utilizes a graph-based Neo4j database to store CPM-derived object data and examines scalability under a V2X message rate of 10 Hz.

Although the mentioned solutions effectively offer tailored LDM services for specific use cases, there remains a gap in providing MEC services that combine generalized, latency-critical, on-demand data retrieval with scalable and efficient storage capabilities. Aiming to address this gap, this chapter introduces a 5G-enabled Multi-Access Edge Computing (MEC) component for centralized cooperative perception, offering an interoperable interface to other MEC services as well as vehicles for the execution of automated maneuvers.

## **4.2 Server-Local Dynamic Map for enhanced collective perception**

The Server Local Dynamic Map (S-LDM) is a centralized LDM service developed as part of the 5G-CARMEN project [34] featuring a custom memory-efficient database that stores both the latest kinematic data and path history of all subscribed vehicles and non-connected objects within a specific geographical area. This design enables the system to provide other MEC services only with filtered and processed data essential for executing highly automated maneuvers effectively.

For connected vehicles, the S-LDM utilizes standardized CAMs, extracting information such as vehicle position, speed, heading angle, and the state of steering and exterior lights. Meanwhile, for non-connected objects detected by vehicle sensors, the S-LDM supports the encoding of this information as Virtual CAMs. These Virtual CAMs, representing a preliminary implementation of CPMs within

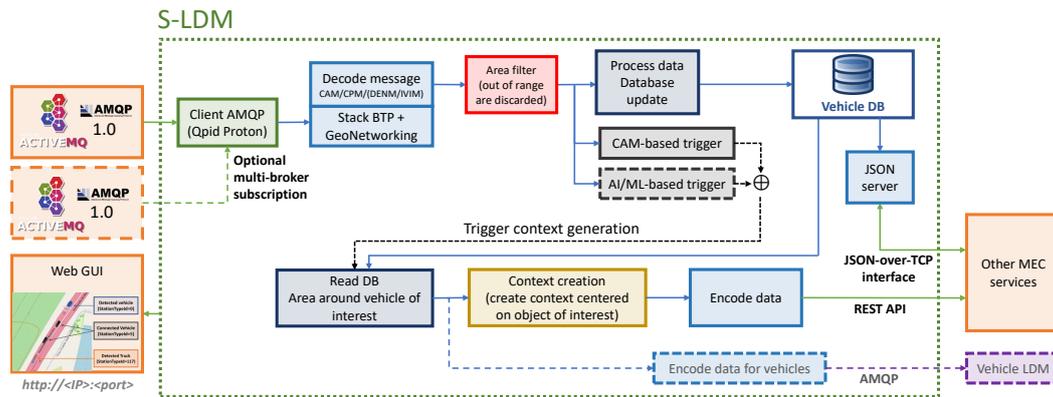


Fig. 4.1 S-LDM service internal architecture.

the 5G-CARMEN EU project, contain the absolute position of detected objects as if they sent standard CAMs.

The S-LDM's ability to provide a comprehensive vision of the road enables more accurate traffic flow predictions, a critical factor in reducing traffic congestion and mitigating road incidents. By centralizing the collection and analysis of vast datasets, the system empowers C-ITS applications to anticipate and manage traffic challenges more effectively.

The S-LDM has been deployed and extensively evaluated within the 5G-CARMEN infrastructure. The test results included in this chapter focus on real-world measurements conducted on MEC platforms of domestic and cross-border network operators. V2X-equipped vehicles, provided by Stellantis, have been leveraged in these evaluations, proving the effectiveness of the system in real traffic conditions. High scalability and cross-border operational capabilities are central to the S-LDM's design, aligning with the objectives of the 5G-CARMEN project. As a centralized service, the S-LDM is designed to serve a large number of vehicles, enhancing the richness of road-related data. To test such capabilities, an extensive scalability study conducted as part of this research work will showcase the service's ability to handle high traffic and diverse operational demands.

## 4.3 Service description and deployment architecture

### 4.3.1 S-LDM description

The S-LDM features a modular architecture comprising sub-modules that interact seamlessly, from message reception via AMQP to generating context data for MEC services. Figure 4.1 outlines its internal structure and operations that can be divided into the following modules:

- **AMQP client:** The AMQP client module in the S-LDM uses a multi-threaded design to manage connections to multiple AMQP 1.0 brokers efficiently. It incorporates the Apache Qpid Proton library for fast and reliable message handling.

Each S-LDM instance filters incoming messages based on a configurable geographical coverage area, leveraging Quadkeys for efficient pre-filtering. Currently, rectangular coverage areas are used, aligning well with Quadkeys' structure, but future updates plan to include circular and elliptical areas. During initialization, the S-LDM calculates and caches the minimal Quadkey set for its coverage area, expediting future restarts with unchanged parameters. If the area is modified, the cache updates accordingly.

Quadkey pre-filtering ensures that AMQP clients receive only messages from vehicles slightly beyond the defined area, improving performance by reducing the load on the Area Filter module and enhancing overall efficiency.

- **Message decoder:** The S-LDM decodes ETSI C-ITS messages, including CAMs and DENMs, using a custom, efficient stack designed for high update rates. Messages are decoded automatically, whether they include full GeoNetworking and BTP layers or are "pure" facilities C-ITS messages, ensuring compatibility with varying implementations.

Non-connected objects are addressed via Virtual CAMs, a preliminary solution to simulate CPM functionality. These messages encode data about detected objects from vehicle sensors into a CAM format, relying on a custom C-ITS type ID for their proper classification. This approach enables the S-LDM to track non-connected objects as if they were transmitting standard CAMs, ensuring comprehensive coverage until CPMs are standardized.

The system supports high-frequency CAM reception at 20 Hz, exceeding ETSI's 10 Hz limit, to match onboard sensor rates and enable fine-grained updates for automated maneuvers.

- **Area Filter:** Once a message is decoded, it is processed by the Area Filter module, which determines whether the originating vehicle's position falls within the S-LDM's coverage area. The position is extracted either from the GeoNetworking layer (if present) or directly from the CAM message. Messages from road users outside the designated area are discarded.

Although Quadkey pre-filtering reduces the number of irrelevant messages, the Area Filter module remains essential. This is because the S-LDM is designed for flexibility, allowing it to handle messages even without Quadkey pre-filtering.

- **Main Database:** The S-LDM processes incoming data to update a custom, in-memory, thread-safe database that serves as its core component. This database is optimized for rapid insertions and lookups, ensuring high performance essential for latency-critical applications. It uses a hierarchical structure of hash tables, enabling simultaneous read and write operations while minimizing locking conflicts, making it highly efficient for managing large numbers of vehicles.

The in-memory design offers significant speed advantages over disk-based databases, which, though capable of data recovery after service restarts, are less relevant in high-mobility V2X scenarios where vehicle data quickly becomes outdated. The database supports core operations like insertion, deletion, single and geographical lookups, and tracks detailed data for each vehicle, including position, speed, and status updates. It also maintains a Path History (PH) for vehicles, optimized to store only significant updates based on configurable thresholds, reducing memory usage while retaining valuable historical data.

- **CAM-triggered context creator:** A key functionality of the S-LDM is detecting triggering conditions such as road situations requiring MEC service intervention or a vehicle signaling a high-level automated maneuver request. Upon detecting such a condition, the S-LDM computes a context around a reference node, capturing information on all vehicles and non-connected objects

within a configurable radius. This context is shared with MEC services for centralized maneuver management, leveraging 5G network capabilities.

Currently, triggering detection is deterministic, analyzing fields in received CAM messages (e.g., detecting lane merge intentions from turn indicator signals). Future updates will integrate AI/ML-based prediction, complementing the deterministic approach to enhance detection accuracy.

- **REST API:** The S-LDM enables efficient data sharing with MEC services through two key interfaces. First, it uses a REST API for periodic updates, acting as a client to send pre-processed JSON data to REST servers exposed by other services. These updates occur at configurable intervals and continue until the service replies with a specific response indicating an error, successful processing with a request for continued updates, or a request to stop further updates. Second, the S-LDM provides an on-demand JSON-over-TCP interface. Through this lightweight TCP server, other services can query the database directly by specifying parameters such as location or station IDs of interest in JSON format. The S-LDM then returns a JSON file containing relevant vehicle or object data.
- **Web-based GUI:** The S-LDM features a web-based GUI for real-time road monitoring, allowing users and road operators to view and interact with the local dynamic map stored in the database. The GUI provides detailed information about vehicles and non-connected objects, including IDs and headings, and uses the Station Type field in CAM messages to classify and display different vehicle types with distinct icons.

The S-LDM has been designed to be deployed within the architecture standardized by the C-Roads project [35]. As detailed in Section 2.4, the C-Roads architecture leverages the AMQP 1.0 protocol to efficiently transmit messages between vehicles, MEC services, and centralized entities, utilizing V2I and V2N communications to support Cooperative ITS (C-ITS). Within this architecture, messages from vehicles, such as CAMs and CPMs, are directed into AMQP broker queues or topics. Vehicles and MEC services can subscribe to these queues to receive relevant information, such as warnings, road signage updates, or maneuver management instructions. Additionally, the architecture also incorporates application properties. These properties enable advanced filtering and routing of messages, allowing MEC services to subscribe based on specific criteria, such as geographic area or country of origin. In the

considered system, Quadkeys [127] are used to define the location of vehicles, enhancing the precision of message filtering. Quadkeys are hierarchical base-4 strings that represent progressively smaller geographical areas, based on dividing the Earth's Mercator-projected map into rectangular tiles. For the S-LDM underlying architecture, Quadkeys are processed by AMQP brokers, efficiently filtering messages from vehicles within specific areas and improving performance by excluding messages outside the service's area of interest. Importantly, AMQP brokers apply these filters at an AMQP level, transparently and independently of the specific payload type.

The S-LDM, aligned with C-Roads specifications, subscribes to one or more AMQP 1.0 brokers to receive ETSI-compliant messages, including layers like Facilities, GeoNetworking, and Basic Transport Protocol [32]. Notably, this design ensures platform independence, enabling reliable communication over IEEE 802.11p/802.11bd, C-V2X, or 5G networks. Cross-OEM compatibility is simplified as vehicle manufacturers only need to implement an AMQP 1.0-compliant client using publicly available standards.

The 5G-CARMEN project has focused on an architecture leveraging a 5G network layer to support latency-critical automotive applications, such as decentralized cooperative maneuvers through sensor sharing. Unlike 4G, 5G ensures the low latency and reliability needed for services like the S-LDM [33]. Regarding the AMQP broker deployment, the architecture employs Apache ActiveMQ instances providing an open-source solution ensuring reliability in single-broker scenarios.

Figure 4.2 shows the S-LDM integrated within the described architecture for C-ITS when considering a single broker scenario.

The S-LDM incorporates the Apache Qpid Proton library for AMQP message reception and a custom ETSI ITS decoding stack, based on the implementation offered on ms-van3t (described in Section 3.2.6), enabling rapid updates to its internal database. Its flexibility allows deployment on a variety of MEC platforms, whether Virtual Machine-based or container-based.

Field-tested in real V2X scenarios, the S-LDM has been integrated as a containerized application within the MEC platforms of three 5G-CARMEN network operators—TIM (Italy), Magenta Telekom (Austria), and Deutsche Telekom AG (Germany). Each MEC platform included a local AMQP broker to minimize latency, operating within a Kubernetes-orchestrated edge platform developed by 5G-CARMEN. This edge platform, adhering to ETSI MEC, ETSI NFV, and 3GPP

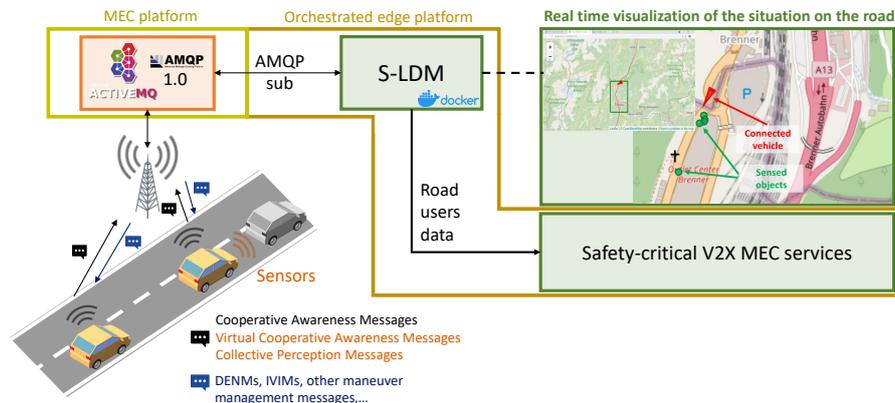


Fig. 4.2 The S-LDM as part of the 5G-CARMEN architecture, in turn based on C-Roads standards.

standards, supported 5G-enabled CAD applications, facilitating collaboration and service continuity across domains. Designed using cloud-native principles, the orchestrated edge platform integrates 5G and MEC/NFV, enabling end-to-end data plane control and application-specific orchestration. This architecture allows S-LDM and other services to extend their functionality across multiple edges and collaborate with instances in different domains.

### 4.3.2 Deployment and scalability options

To address scalability and support cross-border use cases, each S-LDM instance is designed to cover a specific portion of the road, filtering out messages from vehicles outside its defined coverage area. Cross-border scenarios are particularly challenging due to the need for seamless road coverage when transitioning between MEC platforms operated by different network providers in neighboring countries.

Solutions such as local breakout and fast network reselection, enabled by 5G technologies like Equivalent Public Land Mobile Network (E-PLMN), help reduce latency and interruption times during border crossings, as demonstrated by 5G-CARMEN [33]. However, MEC services must also be scalable and interoperable, considering the deployment of multiple S-LDM instances collaborating and processing the overlapping portions of messages to ensure comprehensive road coverage and handle a high volume of connected vehicles.

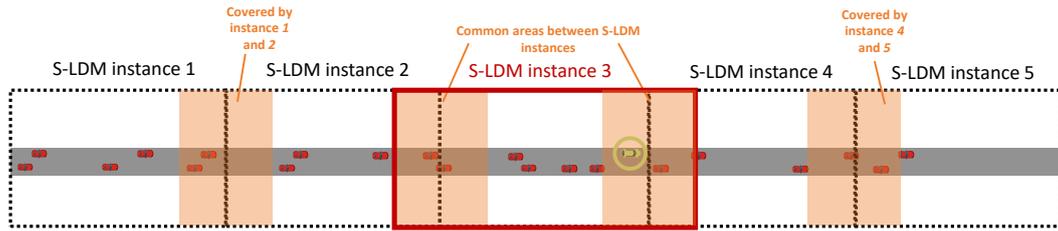


Fig. 4.3 Multiple S-LDM architecture for cross-border and in-country scalability.

To achieve seamless coverage, multiple S-LDM instances are deployed with adjacent, overlapping coverage areas of interest. This ensures that neighboring instances provide a complete view of the road, even for vehicles near the borders of their coverage zones. The coverage areas can be easily configured during the S-LDM instantiation phase within MEC platforms, enabling flexible deployment to meet cross-border requirements.

During the instantiation phase within a MEC platform, the coverage area of each S-LDM instance can be easily configured to ensure seamless road coverage. As shown in Figure 4.3, overlapping coverage areas between neighboring S-LDM instances create an *extended area* (e.g., the red area for *instance 3*), while the core area unique to a single instance is termed the *base area* (e.g., the region between the two dotted black lines for *instance 3*).

Thanks to the overlapping areas illustrated in Figure 4.3, vehicles traveling near the borders of coverage areas, such as the highlighted car, can access complete contextual information about nearby road users, including those in neighboring S-LDM instances (e.g., the preceding red car in *instance 4*). This mechanism ensures robust context generation and uninterrupted service across adjacent S-LDM instances.

As mentioned in Section 4.3.1 the S-LDM is designed to detect triggering conditions from received messages for the provision of pre-processed information to other services. Considering the presence of overlapping coverage areas, triggering conditions may occur within them for which two scenarios are considered:

- **In-country:** the S-LDM instance with the triggering vehicle located within its *base area* will be the only one to transmit data to the corresponding MEC services.

- **Cross-border:** considering the example where *instance 3* operates in Austria and *instance 4* in Italy, both S-LDM instances will send data to their respective MEC services (e.g., those located in Austria and Italy).

To handle cross-border scenarios, the S-LDM relies on multi-broker subscriptions, enabling instances from different operators to process messages from vehicles connected to both sides of the borders, leveraging 5G inter-operator connectivity, demonstrated and validated in 5G-CARMEN through 5G inter-operator connectivity. Additionally, S-LDM instances can dynamically adjust their coverage areas or deploy new instances in response to increased vehicular traffic, network load, or resource demands (e.g., due to abnormal traffic, a coverage area can be split in two, each managed by a separate instance to balance the load). Platforms like the orchestrated edge framework in 5G-CARMEN [130] support effective scaling and service continuity to address these requirements.

It is important to mention that the S-LDM's modular structure and flexible deployment options enable it to support various centralized use cases by efficiently delivering filtered and processed information:

- **Centralized maneuver management:** The S-LDM provides real-time, relevant vehicle data to maneuver management services, enabling safety-critical tasks while minimizing data processing overhead. It can be deployed on MEC platforms of mobile network operators (MNOs) or road operator MEC servers, supporting integration with car manufacturer-developed coordination services.
- **Smart traffic light control:** Deployed in a city mobility center's MEC or cloud, the S-LDM supplies data to optimize traffic light phases, reducing congestion and emissions. It also supports distributed virtual traffic light systems by providing a comprehensive road view to vehicles.
- **Centralized platooning:** The S-LDM may provide the required data for Cooperative Adaptive Cruise Control (CACC) algorithms executed on MEC platforms. For this specific maneuver management use case, a service like the S-LDM has the potential to enhance platoon safety by providing additional data on interfering vehicles nearby. Incorporating detected object information into the context enables the creation of safer platooning algorithms that are robust against intrusions from legacy non-connected vehicles [148].

## 4.4 On-road evaluation

Following an in-lab evaluation campaign, the S-LDM was integrated as a container within the 5G-CARMEN orchestrated edge platform [130], deployed on the MEC platforms of the three participating network operators. This deployment enabled road test campaigns to validate the S-LDM in real-world V2X scenarios, leveraging a 5G Non-Standalone (NSA)  $Uu$  link for continuous data reception on connected and non-connected objects via AMQP 1.0. While the in-lab tests provided valuable insights into the S-LDM's performance under controlled conditions, they are not the primary focus of this thesis. Instead, the road tests, conducted using Stellantis vehicles equipped with enhanced V2X OBUs based on L3 Pilot [36] Maserati prototypes, will be the main focus for evaluating the system's capabilities. The considered OBUs supported both short-range ( $PC5$ ) and long-range ( $Uu$ ) communication, with the 5G New Radio (NR) modem ensuring high data rates and leveraging 5G-CARMEN's cross-border features. The integration was configured for high-frequency data transfer (20 Hz), exceeding standard V2V communication rates [70], ensuring continuous service, and minimizing latency to the MEC platforms and S-LDM. This focus on real-world testing provides a more comprehensive validation of the S-LDM's performance and functionality in practical V2X scenarios, such as cross-border scenarios.

The road tests were designed with three primary objectives:

- Assess the S-LDM's ability to receive CAMs and virtual CAMs at a high frequency of 20 Hz via AMQP 1.0, supported by a 5G network.
- Evaluate the S-LDM's performance in handling data from real-world connected vehicles and non-connected objects when deployed across different MEC platforms.
- Analyze the disconnection time caused by MNO network reselection at the border from the S-LDM's perspective.

To evaluate the performance of the S-LDM in cross-border scenarios, several test runs were conducted with two Maserati vehicles (referred to as "Vehicle 1" and "Vehicle 2") along the Brennerpass border on the E45/A22 motorway between Austria and Italy. For these experiments, two S-LDM instances were deployed on

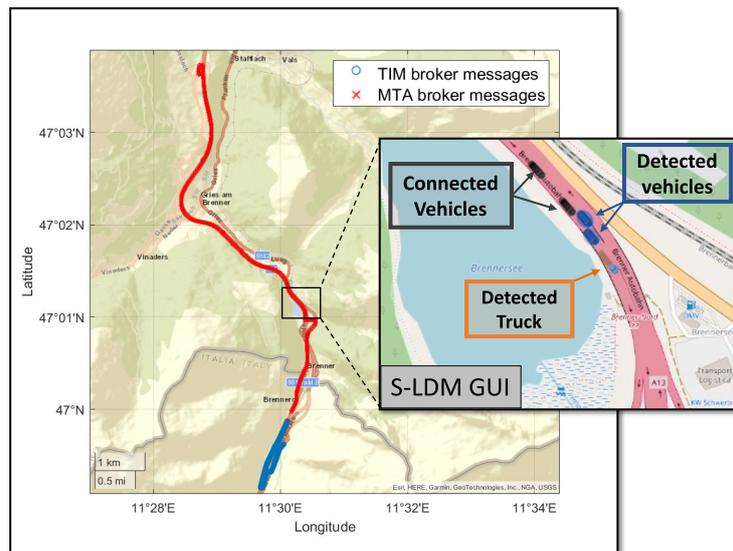


Fig. 4.4 The path followed by one of the two Stellantis vehicles during the cross-border road tests from Austria to Italy is shown. Messages received by the S-LDM from the TIM network via the AMQP broker are represented in blue, whereas those received from the MTA network via its corresponding broker are represented in red.

MEC platforms managed by TIM in Italy and MTA in Austria, respectively. These instances were configured as subscribers to two AMQP 1.0 brokers: one deployed on the Italian MEC platform and the other on the Austrian MEC platform. Both S-LDM instances were set to cover overlapping areas around the border to ensure that both instances covered the entire border region.

The tests reported in this document were conducted on the E45 section of the road between Nöblach (Austria) and Brennerpass (Italy), focusing on the Austria-to-Italy direction as a representative scenario. The path followed by one of the Maserati vehicles is shown in Figure 4.4. This map uses the Italian S-LDM instance as a reference and displays, in blue, the positions recorded from CAMs received from the Italian broker and, in red, those obtained via CAM messages from the Austrian broker (leveraging the multi-broker subscription for cross-border scenarios).

The overall outcome of the cross-border tests was favorable. The service successfully processed real-time information to track connected vehicles and detected objects on both sides of the border by handling messages from the respective AMQP brokers. While multiple test runs were performed, this document focuses on the most significant one to provide a detailed and more intuitive outline of the results. Figure 4.5 illustrates the *normalized message delay* as a function of time since the

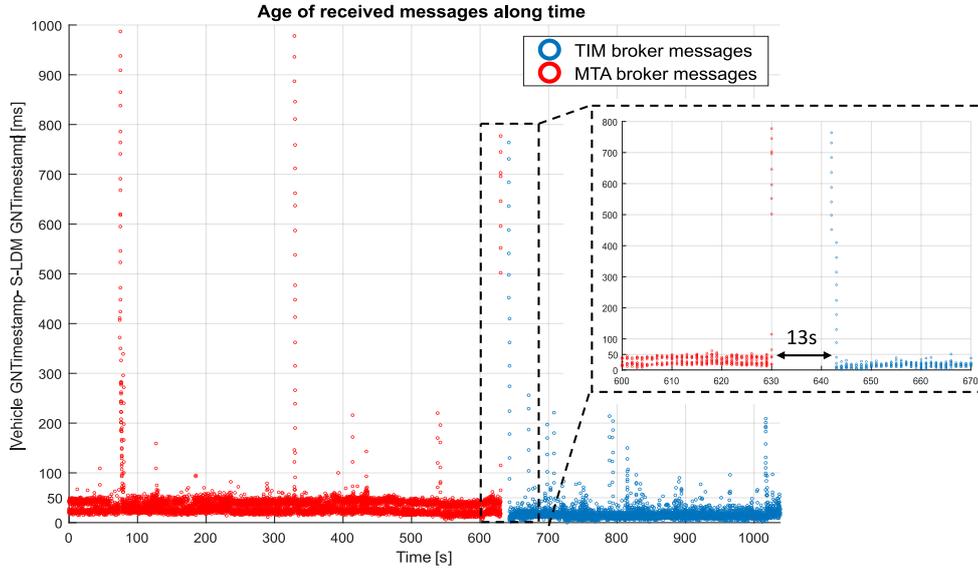


Fig. 4.5 Normalized message delay (i.e., the delay between the transmission of messages from vehicles and their reception by the S-LDM, adjusted by subtracting the minimum delay observed during the entire test) over the duration of a single test run for a single vehicle. The S-LDM instance deployed on the TIM MEC platform has been used as the reference point.

test's initiation, referencing the S-LDM instance deployed on the TIM MEC platform. The *normalized message delay* quantifies the difference between the delay of each message's transmission and reception by the S-LDM, relative to the minimum delay observed throughout the test. If  $d_i$  denotes the delay for packet  $i$ , the normalized message delay for the same packet is computed as:

$$\hat{d}_i = d_i - \min_i d_i \quad (4.1)$$

In this manner, the minimum normalized delay corresponds to a value of 0. It must be mentioned that using a normalized delay was necessary due to the lack of time synchronization between the MEC platforms of the different MNOs, which prevented consistent measurements when using absolute delay values.

As observed, the normalized delay remains mostly stable throughout the test duration, with a few sparse peaks. The observed peaks, lasting less than a second, and are likely caused by jitter and delay inherent to the underlying 5G NSA network. These effects are expected to diminish significantly when transitioning to a full-fledged 5G SA architecture. Additionally, it can be seen that the average normalized

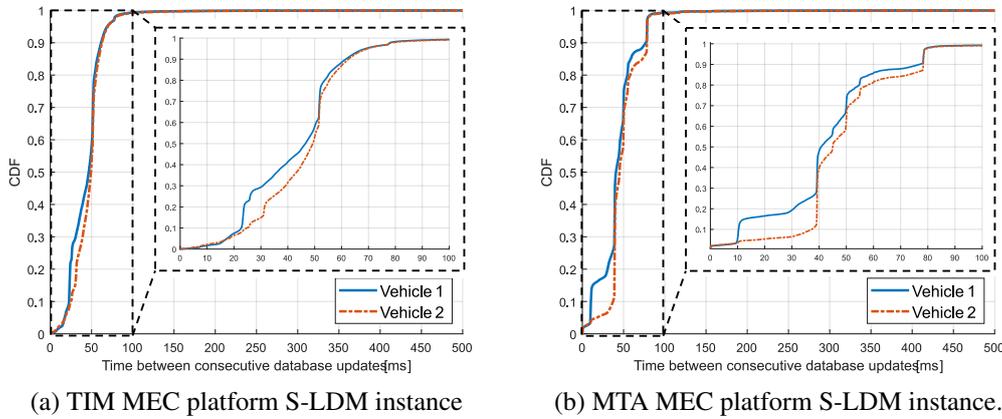


Fig. 4.6 Cumulative Distribution Function of the time between consecutive database updates for each of the Maserati vehicles during the most significant Austria-to-Italy cross-border test.

delay for the MTA broker is slightly higher overall. This behavior is anticipated, as the data shown in Figure 4.5 is based on the perspective of the S-LDM instance in Italy. Consequently, messages from the MTA broker in Austria incur an additional inter-MEC communication delay.

A disconnection interval of less than 14 seconds was observed when transitioning from Italy to Austria, and this duration remained consistent across all test runs along the border. Specifically, when focusing on the Austria-to-Italy direction, the S-LDM exhibited an average disconnection time due to fast network reselection of  $13.3729\text{ s}$ , with a standard deviation of  $2.0203\text{ s}$ , and a 95% confidence interval of  $\pm 0.7062\text{ s}$ .

It was also possible to measure the time interval between two consecutive database updates for a given connected vehicle or detected object. This metric, referred to as the *Instantaneous Update Rate*, was collected to understand how often the database was updated, accounting for the processing times and network conditions. The Cumulative Distribution Function (CDF) of the *Instantaneous Update Rate* of the S-LDM, corresponding to both the TIM and MTA instances during the most significant Austria-to-Italy cross-border test, are shown in Figure 4.6. As observed, approximately 80% of the time, for a given vehicle, the database is updated with a periodicity of 50 ms or less (with lower values attributable to network jitter). This aligns with the 20 Hz CAM message frequency. The results are similar for both the S-LDM instance on the TIM MEC platform and the one on the MTA MEC

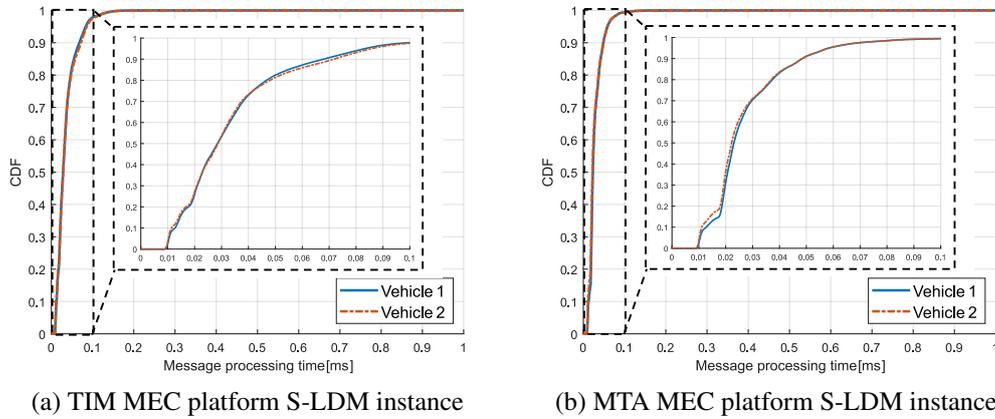


Fig. 4.7 Cumulative Distribution Function of the message processing times for each of the Maserati vehicles during the most significant Austria-to-Italy cross-border test.

platform, demonstrating that the S-LDM can effectively store and provide a real-time map of the road even in cross-border scenarios.

Finally, the overall message processing times during the same cross-border test were evaluated. The CDFs corresponding to both S-LDM instances are shown in Figure 4.7. As observed, 90% of messages are processed in under 72 microseconds on the S-LDM instance deployed on the TIM MEC platform and in under 50 microseconds on the instance deployed on the MTA MEC platform. The performance differences can be attributed to the distinct hardware configurations supporting the two MEC platforms.

Additionally, the average processing time was approximately 35 microseconds for the S-LDM in Italy and 28 microseconds for the S-LDM in Austria. Overall, these results validate the S-LDM as a service able to support message frequencies above the maximum of 10Hz considered by ETSI [70], successfully supporting the 20Hz frequency adopted by 5G-CARMEN towards the support of advanced automated maneuvers.

## 4.5 Scalability analysis

Scalability remains a critical challenge for most V2X MEC services, particularly considering the rising market penetration of connected vehicles. To address this,

S-LDM instances are designed to manage scalability by covering only a limited road segment. This design anticipates future deployments where multiple instances work together to cover more extensive areas, such as entire motorways or urban districts. These instances may also dynamically adjust their coverage zones in real-time, based on factors like the number of active vehicles producing messages or the current resource usage.

To validate the system's ability to handle this demand, a comprehensive set of scalability tests have been conducted. These test have focused on the two following aspects:

- **Real-time data storage:** i.e., the capability of storing information from received messages without introducing performance bottlenecks.
- **Data provision to other services:** i.e., the capability of concurrently processing stored data to be sent to other requesting MEC services.

All tests have been conducted using the testbed illustrated in Figure 4.8, deployed on an OpenStack platform running Ubuntu Server 20.04 LTS. The setup included a Kubernetes-orchestrated cluster, deployed with the *kubeadm* toolbox within the *S-LDM VM*, to emulate the behavior of services running on MEC in the 5G-CARMEN on-road architecture. To emulate connected vehicles disseminating CAMs at 20 Hz, as specified by 5G-CARMEN, an ad-hoc emulator on the *Producer VM* was employed. This emulator used pre-recorded packet traces generated offline by an *ms-van3t* simulation, configured with a scenario based on the A22/E45 motorway, to realistically emulate a large number of vehicles.

Resource usage of each pod and container in the cluster was monitored and logged using *kubect1* tool. During each test, the number of emulated vehicles was incrementally increased by 10 every 15 seconds, reaching a maximum of 750 vehicles over a total test duration of 1125 seconds. Additionally, a basic REST server mimicking an automated driving centralized Maneuvering Service (MS) was deployed on a third *MS VM*, completing a testbed configuration similar to the on-road setup used for cross-border tests within the 5G-CARMEN architecture.

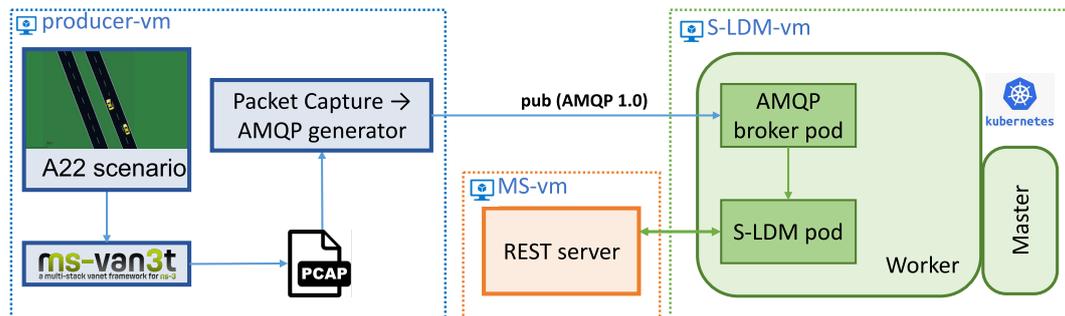


Fig. 4.8 Laboratory testbed setup for S-LDM scalability analysis.

### 4.5.1 Real-time data storage scalability

As outlined in Section 4.3.2, S-LDM instances are designed to ensure comprehensive road coverage, with each instance's coverage beginning where the previous one ends. In cross-border scenarios, however, it becomes necessary for coverage areas to overlap with adjacent ones. This overlap ensures seamless service continuity across borders. Consequently, at these border regions, S-LDM instances need to handle messages from multiple AMQP brokers.

To assess the performance of the architecture involving S-LDM and one or multiple AMQP broker services, three distinct scenarios were devised and evaluated during the tests:

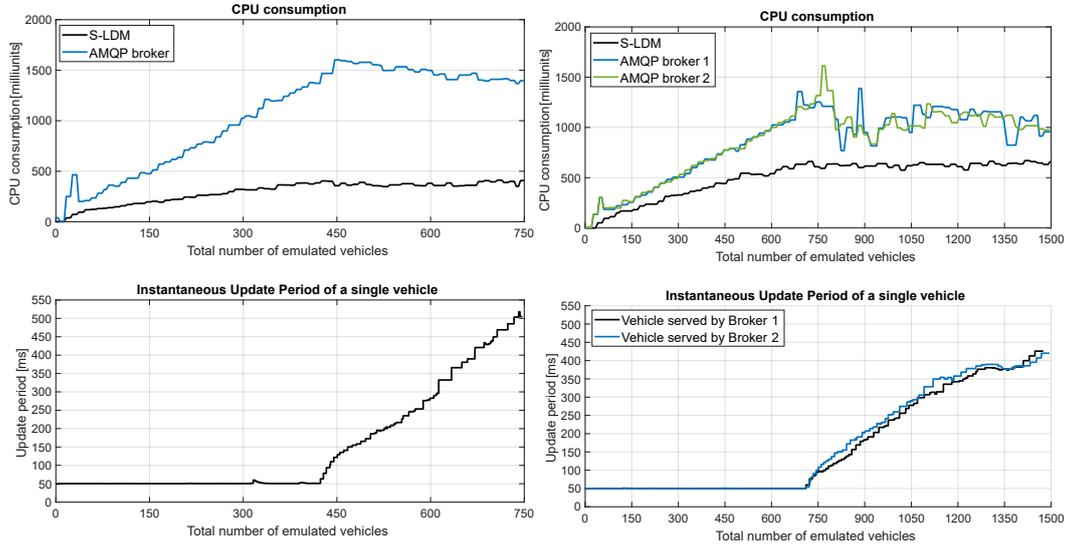
1. One S-LDM instance subscribed to a single AMQP broker.
2. One S-LDM instance subscribed to two AMQP brokers.
3. Two S-LDM instances subscribed to a single AMQP broker.

In test 1, which represents a simple in-country scenario (discussed in Section 4.3.2), CPU consumption for both S-LDM and AMQP components increases linearly up to 430 emulated vehicles, as illustrated in the top part of Figure 4.9(a). Beyond this threshold, CPU consumption stabilizes for the S-LDM and gradually decreases for the AMQP broker. This behavior impacts the timing of database updates by the S-LDM for individual vehicles. Initially stable at 50 ms (consistent with the 20 Hz CAM periodicity of emulated vehicles), the update interval begins to increase significantly when the number of emulated vehicles surpasses 430, as shown in the bottom part of Figure 4.9(a). This phenomenon results from a buffering behavior at

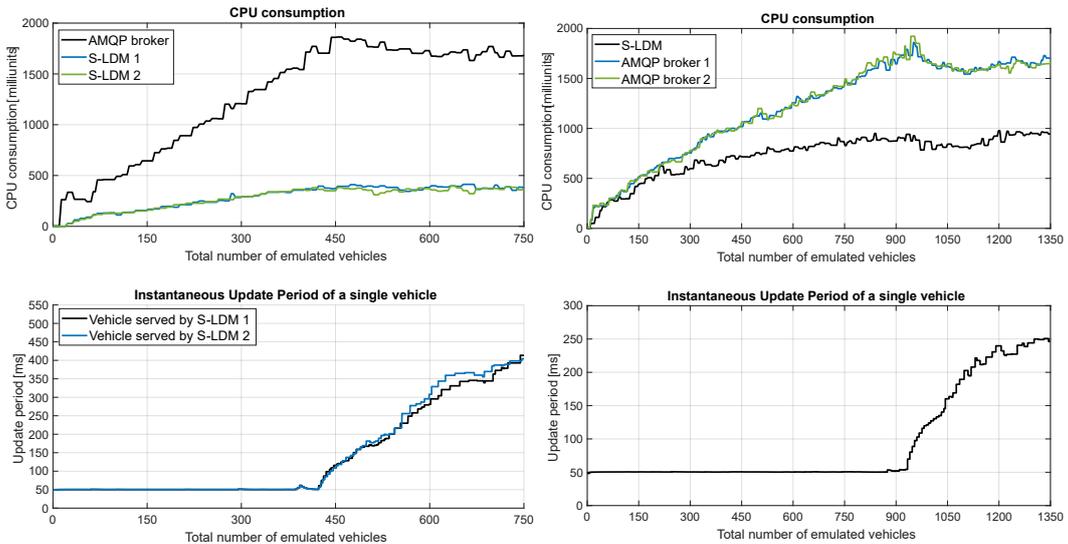
the AMQP level. When a very large number of producers send data to the S-LDM via AMQP 1.0, messages are transmitted in "bulks." This buffering introduces delays for individual messages and causes multiple messages to be grouped into the same bulk, leading to bursts of consecutive database updates within intervals shorter than a millisecond. It is important to mention that for the sake of clarity, Figure 4.9 only includes database update intervals longer than a millisecond. The results indicate that a single S-LDM instance, connected to a single AMQP broker in an in-country scenario, can manage up to 430 vehicles without any performance degradation. These results can be used as a guide for deployment strategies, enabling the definition of appropriate coverage areas for multiple instances. Considering vehicles traveling at 130 km/h with 36 m spacing on a two-lane-per-direction motorway, a single S-LDM instance can reliably cover an approximate road length of 3.9 km.

In scenarios involving multiple AMQP broker subscriptions, the S-LDM must run multiple AMQP client threads, i.e., one for each broker. To evaluate the performance of the S-LDM under these conditions, such as in cross-border scenarios, test 2 was conducted. As mentioned, this test was configured with a single S-LDM instance subscribed to two AMQP brokers. The number of emulated vehicles sending messages increased by 10 every 15 seconds, reaching a maximum of 750 vehicles per broker. Consequently, the total number of vehicles served by the database at any given time was doubled compared to test 1, while each AMQP client thread handled the same number of vehicles as in the single-broker scenario. Figure 4.9(b) shows the CPU consumption of all pods relative to the number of vehicles served by each component. Since each broker managed the same number of vehicles per step as in test 1, both AMQP brokers demonstrated a linear increase in CPU usage at a rate similar to the single-broker case. Intuitively, the S-LDM exhibited a linear CPU usage increase at twice the rate, reaching a peak of 660 CPU milliunits. Beyond a total of 710 emulated vehicles, CPU consumption for all components stopped to showcase a linear behavior, as shown in Figure 4.9(b). This coincided with the database update periodicity for a vehicle served by a given AMQP broker, also depicted at the bottom of Figure 4.9(b). The results confirm that the system can serve up to 710 vehicles without performance degradation, approximately twice the number compared to the case relying on a single broker.

Moving to test 3, this configuration focused on evaluating AMQP performance when multiple S-LDM instances subscribe to it. For this scenario, all emulated vehicles were set to travel within an area covered by two overlapping S-LDM



(a) One S-LDM instance subscribed to a single AMQP broker. (b) One S-LDM instance subscribed to two AMQP brokers.



(c) Two S-LDM instances subscribed to a single AMQP broker. (d) One S-LDM instance subscribed to two AMQP brokers, with two AMQP clients per broker.

Fig. 4.9 (Top): CPU consumption of S-LDM and AMQP broker pods in milliunits as a function of the number of vehicles being emulated. (Bottom): *Instantaneous Update Rate* of a reference vehicle (i.e., reference stationID) as the total number of vehicles managed by a single S-LDM instance increases.

instances, representing a boundary zone between adjacent instances. The CPU consumption of all pods as a function of the number of vehicles served by each service is shown at the top of Figure 4.9(c). Keeping the trend, results indicate a linear increase in CPU usage for all services up to 430 emulated vehicles, consistent with the findings of test 1. Beyond this point, resource usage stabilizes, and performance begins to decline. One key observation from this test is that the CPU usage of the AMQP broker is minimally affected when serving multiple S-LDM instances, showing only a slight increase of 16% compared to test 1. This suggests that the AMQP broker itself is not a significant performance bottleneck, even in scenarios with overlapping S-LDM coverage areas. When combined with the findings of the previous tests, the results point to a performance bottleneck at the AMQP client level within the S-LDM. This bottleneck is likely due to limitations of the Qpid Proton library used for the AMQP client module, rather than the broker itself. Although the current architecture supports a substantial number of connected vehicles per S-LDM instance, future performance enhancements could involve exploring alternative AMQP 1.0 client implementations to address this limitation.

An additional performance test was conducted to benchmark S-LDM performance when multiple AMQP client threads are used for each AMQP broker. This test forks from the scenario of test 2, with two AMQP client threads connected to each of the two AMQP brokers. Each client thread subscribed to messages from an area half the size of the one covered in test 2, effectively dividing the area into smaller regions leveraging the S-LDM's Quadkey-based filter. In this manner, the two client threads covered the full area previously considered in test 2. As shown in the top part of Figure 4.9(d), CPU consumption across all services increased linearly with the number of vehicles, up to 930 emulated vehicles. Beyond this point, resource usage stabilized, and the database update periodicity for a given vehicle, shown in the bottom part of Figure 4.9(d), remained stable and efficient up to the same threshold.

The results demonstrate that using multiple AMQP client threads per broker allows the S-LDM to handle significantly more vehicles without performance degradation. Specifically, this configuration increased the manageable vehicle count by almost 30% compared to test 2, where only one AMQP client thread per broker was used. This finding highlights the scalability advantage of deploying multiple client threads per broker to optimize S-LDM performance.

### 4.5.2 Data provision to other services scalability

The tests described in this section aimed to evaluate the service's performance in terms of resource consumption while providing road context to other latency-critical MEC services. In particular, a custom REST server application was deployed on the *MS VM* to simulate the behavior of a centralized maneuver management service. This REST server was configured to accept POST requests from the S-LDM instance upon detecting a triggering condition and respond after processing 10 POST requests, one every 50 ms, with a `rsp_type` field set to STOP.

Following a similar philosophy to previous tests, the setup involved a gradual increase in the number of vehicles triggering context data transmission to the REST server. Starting with one emulated vehicle, the number increased every 50 seconds by one vehicle, up to a maximum of 20 vehicles, for a total test duration of 1000 seconds per test. For this specific experiment, a CAM-based trigger was used to detect when a vehicle activated its right-turn indicator, simplifying data collection and ensuring controlled triggering conditions. To introduce additional complexity, each test iteration included five additional non-triggering vehicles (i.e., 15, 20, and 25 surrounding vehicles) placed within the 150-m context range of the triggering vehicles. These non-triggering vehicles were generated simultaneously with the triggering vehicles and configured to maintain the same speed, ensuring they remained within the context range throughout the test duration.

The results for S-LDM CPU consumption, in relation to both the number of triggering vehicles and the number of vehicles within a 150 m context range of each triggering vehicle, are presented in Figure 4.10. The findings show that CPU consumption increases by approximately 1000 CPU milliunits for every new vehicle triggering data transmission to the REST server. This increase arises from the S-LDM creating a new REST client thread for each service request, which retrieves data from the database and sends POST requests containing the associated context information. Interestingly, the number of vehicles within the transmitted context does not affect CPU consumption. However, as seen in Figure 4.10(a), when the total number of served vehicles approaches the maximum of 430 vehicles identified in the previous section (as observed with a single AMQP client and broker), performance begins to decline. This is evident in scenarios with 15 triggering vehicles, each accompanied by 25 non-triggering vehicles within the context range. Again, the performance degradation is linked to delays in AMQP message reception, leading to a reduction in

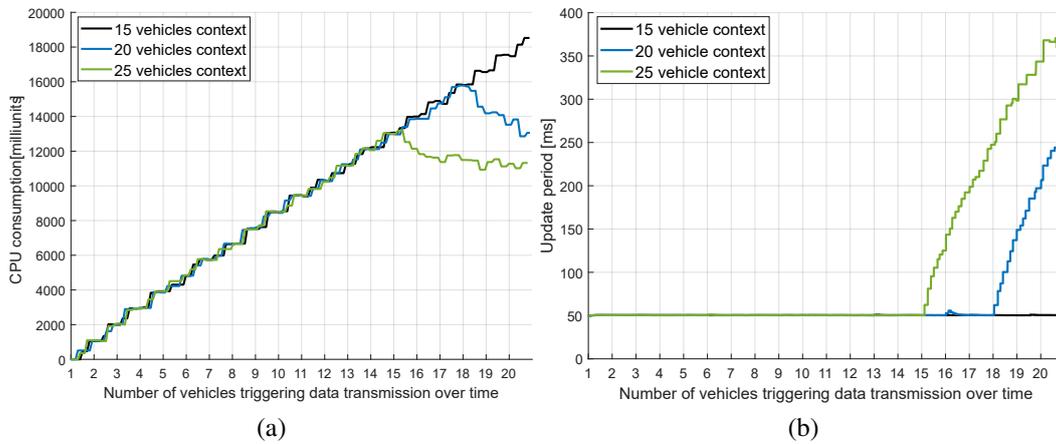


Fig. 4.10 (a) CPU consumption of S-LDM pods, measured in milliunits, with respect to the number of vehicles requesting context information and the number of vehicles located within the context range of each requesting vehicle. (b) *Instantaneous update periodicity* of a reference vehicle (i.e., reference stationID), with respect to the number of requesting vehicles managed by a single S-LDM instance and different numbers of vehicles in the triggered context.

CPU consumption. It is worth mentioning that the total number of emulated vehicles depends on the number of non-triggering vehicles within the context range. For instance, in a scenario with 18 triggering vehicles and 20 non-triggering vehicles surrounding each of them, the total number of emulated vehicles aligns closely with the scenario involving 15 triggering vehicles and 25 non-triggering ones, resulting in similar behavior. This confirms that the observed performance decrease is influenced only by the total number of served vehicles, not by the number of triggering vehicles or the size of their context range.

This behavior stems from the triggering conditions being reliant on information within CAMs, specifically the status of exterior lights found in the *lowFrequencyContainer*. The *lowFrequencyContainer* is included in CAMs every 500 ms, a timing that aligns with the duration required to transmit all POST requests while delivering the context information for a requesting vehicle.

As depicted in Figure 4.10(b), when buffering delays occur at the AMQP client, the detection of triggering conditions in the CAM messages is also delayed. This results in fewer REST client threads being active simultaneously at any given moment, thereby impacting the system's ability to handle context data transmission efficiently.

Beyond establishing useful thresholds for deploying multiple S-LDM instances, the tests in this section also highlight valuable insights for designing a smart scaling policy, as previously suggested in Section 4.3.2. Such a policy could leverage a combination of predicted road traffic conditions (enabling proactive scaling actions) and real-time CPU usage metrics. While the number of vehicles triggering road context transmissions increases the S-LDM's CPU consumption, it has been demonstrated that this does not impact database performance. Instead, performance limitations are tied to the total number of vehicles managed by a single AMQP client instance. Future iterations of the S-LDM could enhance single-instance performance by dynamically spawning multiple AMQP clients for each broker subscription, running in separate threads. This adjustment could be triggered when performance degradation is detected or anticipated, such as when approaching the maximum vehicle capacity of a single AMQP client instance. This proactive approach would ensure sustained performance and scalability under varying traffic conditions.

This chapter addresses practical design considerations while also raising broader questions beyond present-day implementation design. Examining the interplay between coverage variations, traffic density, and vehicle mobility, it explores how systemic constraints influence the performance and adaptability of cooperative mobility services. These findings, while motivated by specific deployment scenarios under the 5G-CARMEN proposed architecture, offer a foundation for more general deployments into robust vehicular edge architectures, serving as a leap forward towards the evolution of CCAM.

## **Chapter 5**

# **Leveraging Connected Vehicle Resources through Micro Cloud Services**

As described throughout this thesis, the concept of Local Dynamic Maps (LDMs) has been standardized by ETSI as a powerful facility to represent the dynamic environment surrounding vehicles, offering guidelines for integrating data from onboard sensors, Cooperative Awareness Messages (CAMs) [70], and Cooperative Perception Messages (CPMs) [77]. While these capabilities are foundational for enabling cooperative perception, existing research highlights several challenges, particularly in the exchange and processing of CPMs.

The exchange of redundant data through CPMs, when no control mechanisms are applied, has been shown to cause increased channel load and elevated Age of Information (AoI) within the LDM due to computational overhead [134], [142]. Matching newly received CPMs with existing objects in the LDM and updating their states over time, while critical for effective perception, faces scalability issues as the number of objects and data sources grows [120]. Additionally, redundant CPMs can overload the matching process, introducing high end-to-end delay and reducing the utility of the shared information [142].

Efforts to address these challenges using centralized approaches, such as aggregating vehicle perceptions at edge servers, show promise due to the computational power of MEC infrastructure [151]. Solutions like the S-LDM presented in Chap-

ter 4, have proven the potential of relying on the computing power available at the MEC for enhancing cooperative perception. This centralized approach demonstrated significant potential in assisting automated vehicular maneuvers by aggregating and providing shared situational awareness. However, the reliance on MEC infrastructure highlighted limitations, such as vulnerability to connectivity disruptions, especially in cross-border scenarios.

In the realm of decentralized computing, vehicular micro-clouds have gained attention for task allocation and resource optimization. Algorithms have been proposed to minimize task completion times within heterogeneous vehicular clusters [98], optimize multi-objective metrics like latency and energy consumption using genetic algorithms [100], [144], and balance resource usage across micro-clouds [61]. Despite their contributions, these works focus on general computational tasks without addressing the specific constraints of cooperative perception, such as data synchronization and redundancy control.

Building upon these concepts, distributed algorithms have been explored for clustering perceived objects and assigning responsibilities among Connected Autonomous Vehicles (CAVs) [95]. While promising in reducing redundancy, fully distributed approaches lack effective synchronization mechanisms, necessitating all CAVs to independently match and track perceptions from multiple CPMs. The Platoon Local Dynamic Map (P-LDM) instead adopts a hierarchical scheme, enabling synchronized reference IDs across platoon members and reducing computational burdens associated with fully distributed systems.

Platoons, with their stable topology and predictable communication patterns, have been leveraged as clusters of computational resources, demonstrating benefits over general cluster-based schemes [137], [147]. Advanced concepts propose offloading non-platoon tasks to idle resources within platoon members [114] or using platoons for federated learning to enhance cooperative applications [110]. However, gaps remain in addressing cooperative perception in mixed traffic scenarios, particularly with non-connected Human Driven Vehicles (HDVs) [111].

The authors in [149] present a framework utilizing the CARLA simulator to demonstrate the benefits of cooperative perception for platoon joining maneuvers. While valuable, their approach relies on raw sensor data under ideal communication conditions, which is infeasible when relying solely on Vehicle-to-Vehicle (V2V) communications.

To address these limitations, this chapter introduces the Platoon-Local Dynamic Map (P-LDM), a lightweight micro-cloud service tailored for platoons. The P-LDM enables object-level perception sharing, distributing and offloading perception tasks required for executing and planning platooning maneuvers among all platoon members. This approach reduces redundancy, minimizes computational load, and synchronizes context representation effectively, addressing the constraints identified in prior works. The P-LDM represents a significant step toward robust cooperative perception in real-world scenarios, ensuring efficiency and reliability even in connectivity-limited environments.

## 5.1 Platoon-Local Dynamic Map: micro-cloud service for cooperative perception

One of the ITS applications most studied by the research community is platooning, which is a cooperative Automated Driving System (ADS) application for which huge efforts have been made to enable its deployment on real highways [6]. In particular, for the European market, the ENSEMBLE project has brought together the main truck manufacturers of the region to realize a communication protocol for multi-brand truck platooning [138]. ENSEMBLE has laid the foundation for standardization bodies, such as ETSI, to create a new facilities layer service for the C-ITS set of standards. Platoons are, in essence, clusters of vehicles that organize themselves to travel close behind one another with the aim of saving fuel, reducing traffic, and increasing road safety. The availability of context information exchanged through V2V messages is key for the platoon to safely and timely react to potential disturbances (e.g., from outsiders).

As illustrated in Figure 5.1, a problem arises for vehicles in a platoon because of their close proximity, resulting in the gathering and sharing of highly redundant data that needs to be processed by each Platoon Member (PM) to align the locally gathered data with the shared data. Although this redundancy allows for enhancement of the view of the road, the incurred processing load to match the information together with the necessary communication overhead results in diminishing returns within the platoon. To address this issue, the concept of vehicular micro-cloud can be employed, which allows the distribution of the computing resources of the PMs. Under this

micro-cloud approach, a master node (the Platoon Leader in our case) takes on the role of coordinating and distributing the contextual information responsibility among all PMs by assigning a subset of all Platoon Perceived Objects to each.

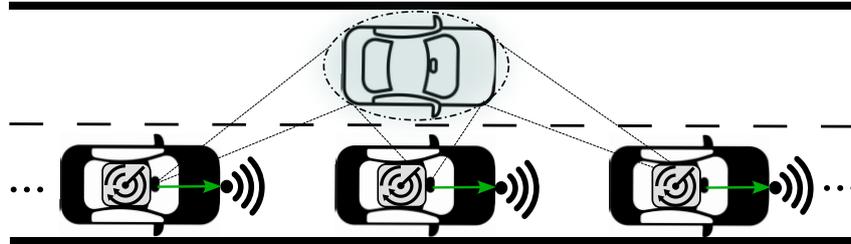


Fig. 5.1 Scheme showcasing multiple platooning vehicles detecting and broadcasting redundant sensing information about the same non-connected vehicle.

The service investigated in this chapter, the Platoon Local Dynamic Map (P-LDM), aims to reduce redundant perceptions being shared and balance the processing load within the platoon's micro-cloud architecture. The P-LDM creates a single aggregated dataset of context information surrounding a platoon, where the Platoon Leader assigns a subset of all Platoon Perceived Objects to each Platoon Member. Each PM is responsible for updating the portion of data assigned to it by exchanging custom V2V messages, specifically defined for this service. By distributing the processing responsibilities across multiple PMs, the system efficiently balances the computational load within the platoon's micro-cloud. This approach not only minimizes redundant perception sharing but also reduces the overall computational and communication overhead. As a result, it enables higher update rates and decreases the Age of Information (AoI) for all perceptions, thus enhancing the platoon's real-time situational awareness.

## 5.2 System Model

### 5.2.1 System overview

In the considered system, all vehicles belonging to a platoon are equipped with communication devices for V2X communications and a C-ITS stack for the dissemination of CAMs and CPMs. All Connected Vehicles (CVs) are equipped with a GNSS device for positioning, a set of cameras (facing front, left, back, and right

of the vehicle), and a LiDAR sensor. The cameras and LiDAR sensor are used for object detection leveraging sensor fusion. Longitudinal platoon control is performed using the CACC controller defined in [122, Chapter 7] for constant spacing. The set of messages defined by the ENSEMBLE project [138] is used for the exchange of the data necessary for the controller.

Based on available context information, the Platoon Leader (PL) is responsible for all platoon-wide maneuvers, including allowing new Platoon Members (PMs) to join or notifying that a PM has left the platoon. In accordance with a near future scenario, the system considers only a given percentage of vehicles to be CVs, defined by the V2X technology penetration rate, establishing a *mixed* traffic scenario. For the environmental awareness of all vehicles, each CV relies on CAMs to perceive other CVs within the communication range and on CPMs to perceive non-connected vehicles outside its perception range.

For all the vehicles under perception range, the data collected by each CV's sensor is processed to create an object list. This list is then used to update the Vehicle's Local Dynamic Map (V-LDM), which includes a list of objects and CVs along with their path history. The V-LDM can be updated with data from onboard sensors and information received from other CVs through CAM and CPM transmissions. The information stored in the V-LDM is used by all CVs to generate CPMs, following the specifications outlined in [77].

Within ETSI's vision, each vehicle independently manages all the information received from nearby vehicles and independently shares all the locally perceived objects. Under such a paradigm, in scenarios where many CVs perceive the same objects (e.g., a vehicle platoon), most of the information on received CPMs does not increase context awareness. Moreover, unless discarded, each CV must process and match the different points of view from all nearby vehicles repetitively. To achieve a more efficient use of the available communication and computational resources for vehicles belonging to a platoon, as depicted in Figure 5.2, the P-LDM unifies the multiple views of all PMs into a single synchronized database of the entire platoon. However, the processing of this unified database is not centralized and is instead distributed among the PMs. Thus, in our approach, the entire set of Platoon Perceived Objects (PPOs), i.e., the set of objects under the perception range of the platoon as a whole, is aggregated in a coordinated way by all PMs. This coordinated aggregation and distributed processing allows the entire platoon to efficiently manage the PPOs

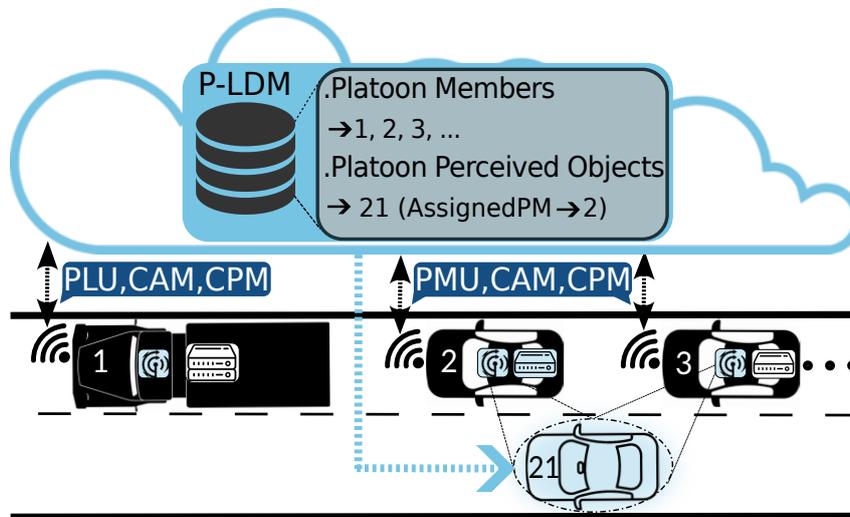


Fig. 5.2 P-LDM architecture and perception aggregation scheme.

while reducing redundant computational effort on individual vehicles by splitting the database management processing among all PMs. The PL serves as the master node in charge of managing the P-LDM service, overseeing the distribution of processing tasks, and offloading the data aggregation to the rest of the members.

The P-LDM contains information about all PPOs, i.e., objects currently being perceived by one or more PMs, and information about all CVs in the platoon vicinity. Finally, the P-LDM contains a *PM state map* with information on each PM, including which PPOs are currently under their detection range, which PPOs they are assigned with, and their available free resources. To simplify the management of the P-LDM database, each entry is stored with a Platoon ID representing the platoon-wide identity of each vehicle in the context (PL, PMs, CVs and PPOs). The Platoon ID used for CVs outside the platoon is the ITS-Station (ITS-S) ID, allowing for interoperability with CAMs. Instead, the Platoon ID used for PPOs is the object ID given by the first PM assigned with such PPO.

### 5.2.2 Message Protocols

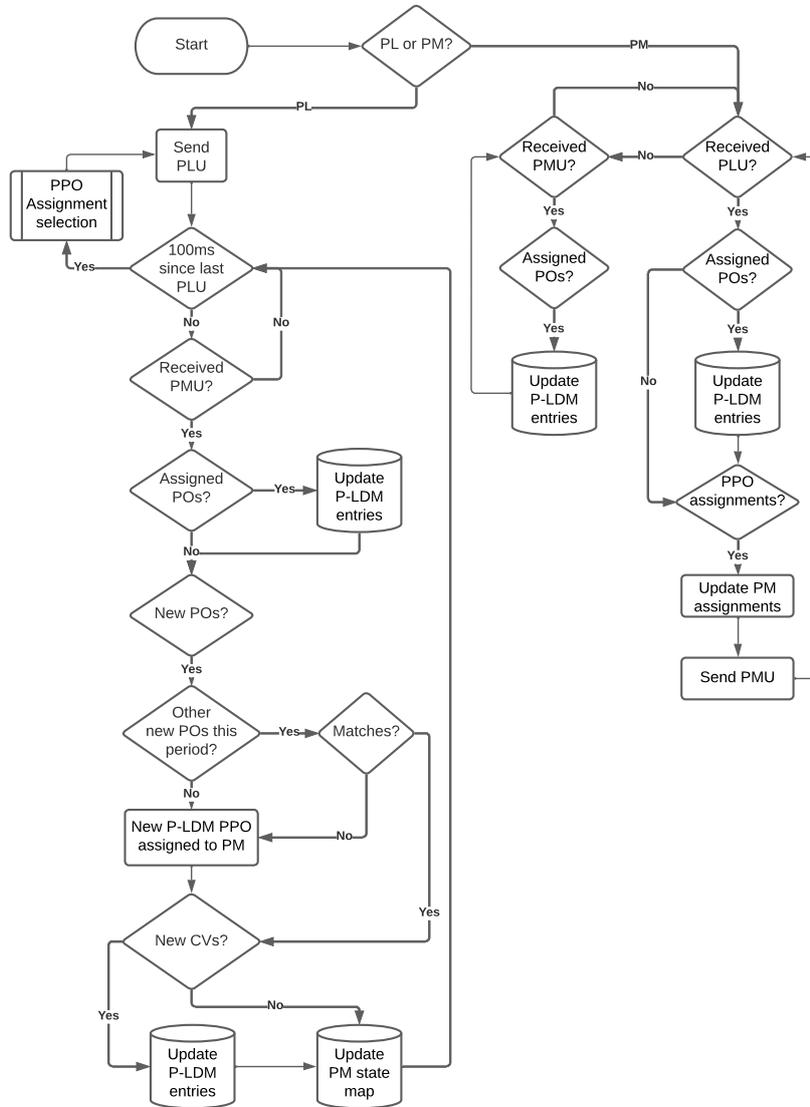


Fig. 5.3 P-LDM protocol flowchart.

The data aggregation is achieved by synchronizing the context databases of all PMs leveraging, in addition to CAMs, a set of two newly defined messages, the *Platoon Leader Update* (PLU) and the *Platoon Member Update* (PMU). This message set is used for perception data exchange and management data exchange for the PPO assignment procedure. The PLU is sent periodically by the PL, carrying the following information containers:

- **Platoon Leader Container:** geographical, dynamic information of the PL.
- **Platoon Members:** list of all PMs' ITS Station Identifier (ITS-S ID) and Platoon ID.
- **Connected Vehicles:** list of each CV not belonging to the platoon, containing their ITS-S ID
- **Assigned Perceived Objects:** a list containing geographical and dynamic information about all PPOs that have been self-assigned by the PL.
- **Platoon Perceived Objects Assignments:** a list specifying the PPOs' Platoon ID and the PM assigned with each of the PPOs.

The PMU, instead, is sent by each PM after the reception of a PLU carrying the following information containers:

- **Assigned Perceived Objects:** a list containing geographical and dynamic information about all the PPOs that have been assigned to this PM.
- **New Perceived Objects:** a list with information about all the Perceived Objects 'seen' by this PM (i.e., appearing in its V-LDM) but not yet included in the P-LDM.
- **New Connected Vehicles:** list of ITS-S IDs of the CVs that are 'seen' by this PM (i.e. appearing in its V-LDM) but not yet included in the P-LDM.
- **PM State:** containing the PM's available resources for the P-LDM and the list of PPO IDs of all the PPOs this PM currently detects (assigned to this PM or not).

For the management of the P-LDM service, the PL periodically sends a PLU to all PMs, every 100 ms. At service start-up, the PL issues a PLU with the Platoon ID of all PMs and all CVs outside the platoon, together with the data of PPOs currently self-assigned by the PL. At PLU reception, each PM matches the new PPO data with its locally perceived objects and for all replicas, it updates the local object's ID with the P-LDM platoon ID.

After the first P-LDM update period, i.e., PLU sent by PL and all PMUs received from all PMs, the information about all PPOs stored in the P-LDM is only the one

reported on the *Assigned Perceived Objects* of each PMU and PLU. Thus, upon reception of a PLU, all PMs update the assignments on their P-LDM and wait for the rest of the PMUs to complete the P-LDM database update. For the mechanism to function properly, all PMUs need to be received by the entire platoon before a new PLU is issued. Thus, each PMU is scheduled in a predefined slot according to the PM's platoon position to avoid collisions. This process is illustrated in Figure 5.3, which presents a flowchart detailing the steps involved in the protocol's operation and data exchange mechanisms.

### 5.2.3 Object data processing

In order to match the objects for ID synchronization, each object of the V-LDM and P-LDM is translated into a 2D box defined by 4 corners and a central point. Each object of the V-LDM is compared to each object of the P-LDM computing the Intersection-over-Union (IoU) and the distance between the central points of each object. The IoU metric measures the overlap between two 2D boxes and is defined as the ratio of the area of their intersection to the area of their union. The IoU ranges from 0 to 1, where 0 indicates no overlap and 1 indicates a perfect overlap. Considering the possible time difference between the detection of all compared objects, the positions of all objects are recomputed, considering speed and heading, to match the predicted position at the time of the matching process. For an IoU higher than 0 and a central point distance lower than 3 m, the two objects are considered to be the same, and the local object ID in the V-LDM is synchronized with the Platoon ID. This process is visually outlined in Figure 5.4, showcasing the thresholds and criteria used for object matching. The reason behind the usage of the central point in addition to the IoU as a threshold to match the objects is due to the shape of vehicles, often being long rectangles in their 2D representation, thus resulting in low IoU values for small lateral misalignments, which can create a high number of false negatives during matching.

When locally perceived objects stored in the PM's V-LDM are not matched to any of the PPOs in the P-LDM, the PM appends their information to the *New Perceived Object* container of the PMU, using a newly generated Platoon ID that is not currently in use. Once all new PPOs are matched with its locally Perceived Objects, each PM checks which of the PPOs *in sight* have been assigned to it by the

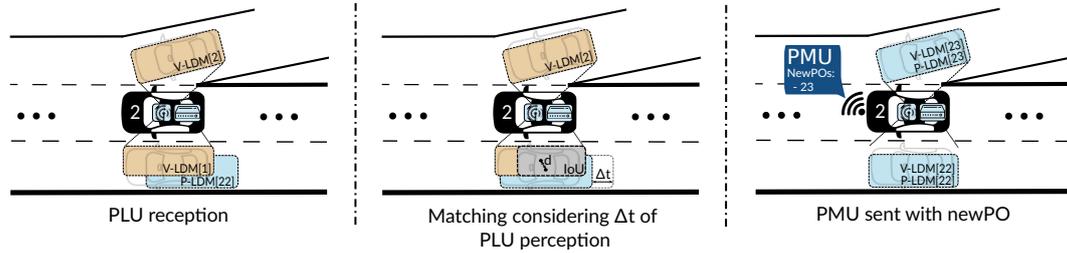


Fig. 5.4 P-LDM object matching and ID synchronization scheme.

PL, if any, and adds the PPO data to the *Assigned Perceived Objects*. After the PLU reception procedure is finished, each PM sends a PMU with the updated information.

On each PMU reception, the PL updates the P-LDM with the updated perceptions extracted from the *Assigned Perceived Objects* container and checks for new PPOs in the *New Perceived Objects* to be added to the P-LDM. Before storing a new PPO in the P-LDM, the PL checks for duplicates that might be found in other PMUs received during the same update period, storing the perception of the closest PM.

Before starting a new update period, the PL begins the *assignment selection procedure* in which it updates, if necessary, the PPOs assignments. The *assignment selection procedure* seeks to jointly achieve the following platoon-wide objectives:

- **Minimum computational cost**
- **Maximum computational fairness**
- **Maximum perception robustness**

The assignment selection problem is extensively analyzed in Section 5.4 for which the proposed algorithms are described in Section 5.5.

## 5.2.4 Quantifying object perception

The P-LDM is designed as Facilities layer service within the ETSI's C-ITS stack, working together with facilities such as the Cooperative Awareness and Collective Perception services. In order to provide compatibility with ETSI's standardized messages and to take full advantage of the information being exchanged on them, the P-LDM is updated with the information through such messages.

For the management of the information of all the CVs outside the platoon, the PL stores all the incoming information found in their CAMs, and whose ITS-S ID is used as the Platoon ID. On the other hand, for the management of PPOs, each PM updates the PL with the context information of its assigned PPOs, for which not only the sensor information of that PM but also the sensor information of other PMs is used through the exchange of CPMs.

By fusing the perceptions of multiple PMs, the overall perception accuracy is enhanced by leveraging a higher granularity of perception updates as well as multiple viewpoints from which the objects are sensed. This solves problems caused by sudden obstacles in front of the vehicle assigned with an object.

All the CPM perceptions received for a PPO are stored by the assigned PM in its local P-LDM copy. For each assigned PPO, a *path history* is created, which contains the last 10 received perceptions. To reduce the computational workload and delay caused by matching and combining objects from incoming CPMs, PMs only handle CPM objects with IDs assigned by the PL. Additionally, since ID synchronization is only guaranteed for CPMs from other PMs, all CPMs from CVs outside the platoon are discarded. This policy is in line with the objective of maintaining a continuous awareness of the surroundings, which relies on information from vehicles that are consistently within communication range, i.e., Platoon Members.

Each PM continuously aggregates all the perceptions of an assigned PPO by computing a weighted average of the received bounding box  $\mathbf{B}'$  compared to the one stored in the P-LDM ( $\mathbf{B}$ ). The resulting weighted average between the 2 bounding boxes is the one obtained by computing the weighted average of each of the variables that define them: center point (stored in a cartesian format from the ego-vehicle point of reference), width, length, and yaw angle. With the aim of achieving a trade-off between prioritizing newer perceptions and prioritizing perceptions perceived from a better viewpoint, the weights are computed by dividing the perception confidence  $C$  by the perception age  $A$ . The perception confidence of an object in our system is the classification confidence obtained by the computer-vision AI model used (in our case, the YOLOv5 [101]), which provides a value between 0 and 100. On the other hand, the perception age is the difference between the time at which the object has been detected and the current time, measured in milliseconds, with only values in the range from 1 to 100 ms considered (i.e., a value of 0 is treated as 1). Thus, each new PPO perception represented by the bounding box  $\pi_{PPO}$  is converted into

the weighted average of the received perception and the already stored perception according to the following expression:

$$\pi_{PPO} = \frac{\frac{C}{A} \cdot \mathbf{B} + \frac{C'}{A'} \cdot \mathbf{B}'}{\frac{C}{A} + \frac{C'}{A'}} \quad (5.1)$$

where  $C$  and  $C'$  represent the confidence values of the stored and new perceptions, and  $A$  and  $A'$  indicate the detection age of the stored and new perception, respectively.

### 5.3 Preliminary service evaluation

A preliminary proof-of-concept evaluation of the service has been performed on the ms-van3t framework. The scope of this evaluation was to assess the potential benefits of the P-LDM service compared to a fully distributed cooperative perception scheme. In particular, this evaluation sought to quantify the potential of the service for distributing the processing load of platoon members as well as the potential to provide an extended and robust representation of the context.

For this evaluation, the P-LDM service has been implemented on ms-van3t, including the message protocols described in Section 5.2.2 together with a simplistic algorithm for the assignment selection procedure mentioned in Section 5.2.3 which will later be further studied and formalized in Section 5.4.

---

#### Algorithm 1 Perception Duration Improvement Selection

---

- 1:  $t \leftarrow$  current time
  - 2: Get PPOs list
  - 3: **for** each  $PPO_i$  in P-LDM **do**
  - 4:     Get all PMs perceiving  $PPO_i$
  - 5:     **for** each  $PM_j$  perceiving  $PPO_i$  **do**
  - 6:         Compute predicted distance of  $PM_j$  from PPO at  $\tau = t + 1s$
  - 7:     **end for**
  - 8:     Pre-assign PPO to closest PM at time  $\tau$  in PMs state list
  - 9: **end for**
- 

The algorithm for the *assignment selection procedure* developed for this evaluation is a two-step process aiming to distribute the computational load on all PMs as well as to increase the perception duration of each PPO inside the P-LDM, i.e., increasing the time during which a PPO is continuously 'seen' by the platoon. The

**Algorithm 2** Computational Load Distribution Selection

---

```

1: PMS  $\leftarrow$  PMs State Map  $\triangleright$  |PMS| = PMS length
2: Read all PPO assignments
3: Order PM states from most to least loaded
4: for  $i = 0$  to |PMS| do
5:   Read Assigned PPOs for  $PM_i$  (A-PPOs)
6:   for  $k = |PMS|$  downto  $i$  do
7:     Read Assigned PPOs for  $PM_k$  (A-PPOs)
8:     if  $PM_i$  A-PPOs  $<$   $PM_k$  A-PPOs + 1 then
9:       Break
10:    end if
11:    for each  $PPO_j$  assigned to  $PM_i$  do
12:      if  $PM_k$  is perceiving  $PPO_j$  then
13:        Read Available CPU (A-CPU), Total CPU (T-CPU)
14:         $PPOload \leftarrow \frac{T-CPU - A-CPU}{A-PPOs}$ 
15:        if  $PM_k$  A-CPU  $>$  PPOload then
16:          Assign  $PPO_j$  to  $PM_k$ 
17:          Update PMS list
18:        end if
19:      end if
20:    end for
21:  end for
22: end for
23: Update P-LDM

```

---

process prioritizes computational load distribution over perception duration, which necessitates the two-step approach. Initially, a preliminary assignment selection is performed to achieve the longest perception duration for each PPO. From this preliminary result, the final assignments are determined, emphasizing an even distribution of the computational load by balancing the number of PPOs assigned to each PM.

In the first step, detailed in Algorithm 1, the PL evaluates each PPO and determines which PM, among those currently perceiving the PPO, will be closest to the PPO's predicted position after one second, assuming constant velocity. Since all PMs move at the same speed, the PPO's relative position is calculated by treating the PMs as static and considering only the PPO's relative speed. If the currently assigned PM is not predicted to be the closest, the PPO is preliminarily reassigned to the nearest PM.

Once the preliminary selection is complete, the PL redistributes the assignments based on the current computational load of each PM. Algorithm 2 outlines this

mechanism, where the PL evaluates each PM's load, starting with the most heavily loaded. For each PM  $PM_i$ , the algorithm checks for a less loaded PM  $PM_k$  that is also perceiving one of  $PM_i$ 's assigned PPOs. If  $PM_k$  has sufficient available CPU, the PPO is reassigned to  $PM_k$ . To determine CPU availability, the algorithm calculates the average CPU load per PPO (line 14) and verifies whether  $PM_k$  can handle an additional PPO. The reassignment is executed only if the difference in the number of assigned PPOs between  $PM_i$  and  $PM_k$  exceeds one, as verified in line 8.

### 5.3.1 Simulation setup

All the simulations have been performed on a highway scenario, for which a custom 3-lane highway loop has been designed in SUMO with a total circumference of 3 km. The CACC platooning protocol was out of the scope of this evaluation, hence all simulation measurements have been performed on an artificially controlled platoon. Platoon sizes of 10 and 20 PMs have been considered to evaluate the impact on the service performance. Beside platoon members, a variable percentage of simulated vehicles have been considered to be NR-V2X-enabled, according to a pre-configured market penetration rate. Indeed, the penetration rate has an impact on the density of vehicles without V2X capabilities which in our scenario translates to an impact on the number of potential objects to be detected and shared by connected vehicles. As our service has been envisioned to be used by a platooning application deployed in the near future, i.e., at low market penetration rates, we consider values of 10% and 50%. All communications are configured to take place using NR-V2X mode 2 on a 20 MHz channel at a central frequency of 5.9 GHz, with a subcarrier spacing of 30 kHz, using the Modulation and Coding Scheme (MCS) index 14, and a transmission power of 23 dBm.

To evaluate the benefits of the P-LDM service, its KPIs were compared against two baseline scenarios:

- **Full Cooperative Perception (Full CP):** This baseline represents a fully distributed scenario. The PL manages the platoon using information received from all neighboring vehicles, including CVs outside the platoon, which is stored in its own V-LDM. All the information is processed entirely by the PL itself. Similarly, each Platoon Member (PM) must locally process all incoming data, which may lead to a significant computational load.

- **Platoon Cooperative Perception (Platoon CP):** This baseline serves as an intermediate scenario between Full CP and the P-LDM service. Here, the PL and PMs process only the information of detected objects shared exclusively by other PMs. As a result, PMs handle a reduced portion of incoming data, which is stored in their own V-LDM. Since the shared information originates from within the platoon, it remains contextually more relevant and is likely available for a longer duration.

The KPIs considered for each scenario are as follows:

- **Total Number of PPOs:** This refers to the total number of Platoon Perceived Objects stored in the LDM. A higher total indicates greater situational awareness.
- **Number of Perceived Objects Handled by Each PM:** This metric represents the number of Perceived Objects that each PM processes periodically for object matching and fusion. It is directly proportional to the computational load on each PM.
- **Platoon Perception Duration:** This measures the time during which a given object is continuously perceived by the platoon. It reflects the robustness of the aggregated perception, mitigating limitations caused by individual vehicle sensor ranges and sensor blockages.
- **Perception Age:** This indicates the time elapsed between the detection of a PPO by a sensor and the current time.

All results were obtained from ten simulations, each lasting 600 seconds, with a vehicle density of 30 vehicles per kilometer for each scenario. Measurements for all KPIs were collected every 100 ms, from the perspective of the PL. Finally, the results are presented from the PL's point of view, as it utilizes the context data for managing platoon maneuvers. It is important to note that in the P-LDM scenario, the same data is also available to all PMs in addition to the PL.

### 5.3.2 Preliminary results

Regarding the *Total Number of Perceived Objects* stored in the LDM, the Cumulative Distribution Function (CDF) for each scenario is shown in Figure 5.5. The results

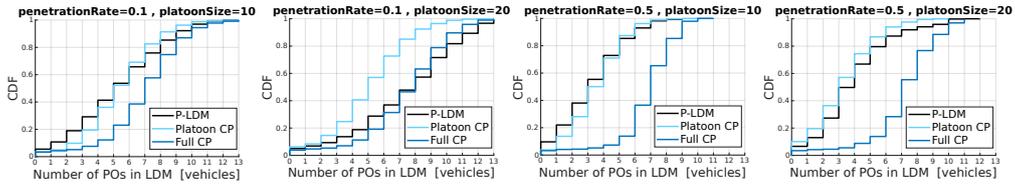


Fig. 5.5 Cumulative Distribution Function of the Number of Perceived Objects stored in the LDM at a given point in time.

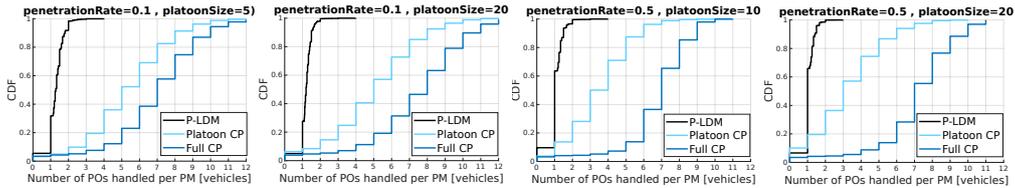


Fig. 5.6 Cumulative Distribution Function of the Average Number of Perceived Objects handled by each PM, in the LDM at a given point in time.

indicate that the Full CP scenario generally achieves a higher number of perceptions in the LDM across all cases, except for the scenario with a platoon size of 20 and a market penetration rate of 10%. In this particular case, the P-LDM yields nearly identical results because the aggregated perception range of the platoon approaches the communication range. The superior performance of Full CP arises from its inclusion of objects perceived by vehicles outside the platoon, which becomes more significant at higher penetration rates where more connected vehicles are present.

However, in both the Full CP and Platoon CP scenarios, all perceived objects must be processed by each PM. In contrast, the P-LDM assigns objects to specific PMs, significantly reducing their computational load. This improvement is evident in the CDF of the number of perceived objects handled per PM, shown in Figure 5.6. With P-LDM, each PM processes no more than 2 PPOs for 99% of the time.

Furthermore, the average *Platoon Perception Duration*, depicted in Figure 5.7 for a platoon size of 10 vehicles and a market penetration rate of 50%, demonstrates a substantial advantage of the P-LDM. Specifically, the duration for which perceptions remain in the LDM is ten times longer compared to both Full CP and Platoon CP.

When comparing these results with the total number of perceptions shown in Figure 5.5, it becomes clear that while Full CP records a higher number of perceptions, these perceptions are sporadic. This is because objects perceived by vehicles outside the platoon are often not updated after those vehicles move out of

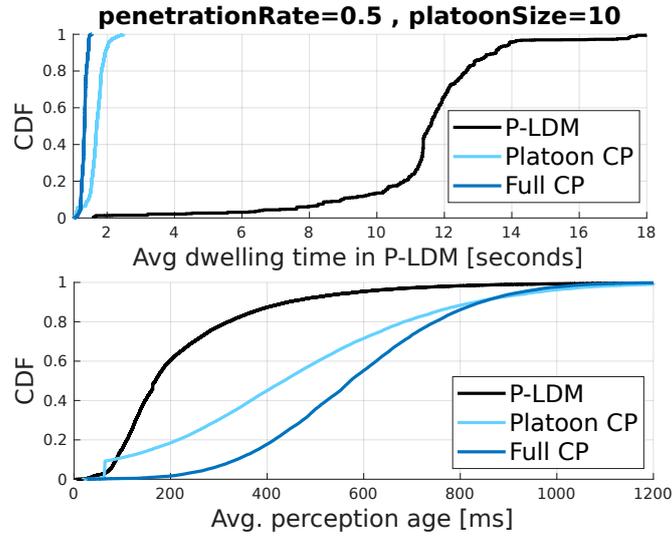


Fig. 5.7 Cumulative Distribution Function of the Average Platoon Perception Duration of Objects and the Average Age of Perceived Objects in the LDM.

communication range. In contrast, the P-LDM maintains consistent and reliable perception updates within the platoon.

Finally, when evaluating the *age of the perceptions* stored in the LDM, the P-LDM again outperforms the two baseline scenarios. This is illustrated in the CDF of the average perception age at a given point in time, shown in Figure 5.7.

The results indicate that, with P-LDM, perceptions remain no older than 400 ms for 90% of the time, whereas in the other two baselines, this value increases to 800 ms. It is important to note that the reported values represent the average across all perceptions at a given moment. This includes perceptions that are no longer being updated, such as objects that have moved outside the platoon's perception range but remain stored in the database until it is cleaned (the database is cleaned every 2 seconds to avoid retaining outdated, i.e., inaccurate, perceptions).

These results demonstrate how the service effectively delivers a more stable and up-to-date view of the road compared to the standard cooperative perception scheme. Additionally, it leverages redundant context data as an advantage by aggregating it efficiently without overloading all platoon members.

While this evaluation has demonstrated the potential of the P-LDM in distributing computational load and enhancing perception robustness, there remains a need for a more sophisticated algorithm capable of reliably optimizing the assignment selection

objectives jointly. Furthermore, the evaluation of computational load distribution was based on the number of PPOs assigned to PMs. However, more detailed assessments are required to accurately quantify the impact of the number of assigned PPOs on the actual computational load experienced by a PM. It also needs to be mentioned that this evaluation lacked a platooning protocol implementation, leaving the question of how the enhanced perception might influence the execution of platooning maneuvers unexplored.

In the following sections, we will further investigate the assignment selection problem by formalizing it as an optimization problem and developing the necessary optimization algorithms. Additionally, we will utilize a more realistic and detailed simulation framework to evaluate not only the computational load distribution but also the potential benefits of enhanced perception in improving platooning maneuvers within the micro-cloud.

## 5.4 Assignment Problem Formulation

As described in Section 5.2, the P-LDM contains a database with the entire context under the platoon perception range. Within this context, each non-connected vehicle stored as Platoon Perceived Objects (PPOs) is periodically assigned to a given Platoon Member (PM). For a given  $N$  number of PMs and  $M$  number of PPOs, we denote as  $\mathbf{P}$  a  $N \times M$  binary matrix whose elements  $p_{n,m}$  are set to 1 if PM  $n$  is perceiving PPO  $m$ , and 0 otherwise, according to the following expression :

$$p_{n,m} = \begin{cases} 1 & \text{if } d_{n,m} \leq r_n(1 - 0.9\omega_o) \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

where  $d_{n,m}$  represents the distance between PM  $n$  and PPO  $m$ ,  $r_n$  the perception range of PM  $n$  and  $\omega_o$  the conditional occlusion factor of PPO  $m$  [132].

We denote as  $\mathbf{X}$  the set of assignments.  $\mathbf{X}$  is thus given as a  $N \times M$  binary matrix whose elements  $x_{n,m}$  are set to 1 if a certain PPO  $m$  is assigned to PM  $n$  [100], 0 otherwise:

$$x_{n,m} = \begin{cases} 1 & \text{if } PM_n \text{ is assigned with } PPO_m \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

The following constraints apply:

- A PPO can be assigned to only one PM (Responsible PM):

$$\sum_{n \in N} x_{n,m} = 1 \quad (m = 1, \dots, M) \quad (5.4)$$

- The number of PPOs assigned to a given PM  $n$ , can be up to  $M$ :

$$\sum_{m \in M} x_{n,m} \leq M \quad (n = 1, \dots, N) \quad (5.5)$$

- A PM can only be assigned with a PPO that is currently being perceived by it:

$$x_{n,m} \leq p_{n,m} \quad (n = 1, \dots, N), (m = 1, \dots, M) \quad (5.6)$$

where  $p_{n,m}$  is the entry at the  $n^{\text{th}}$  row and  $m^{\text{th}}$  column of the binary Perception matrix  $P$ .

We denote  $\mathbf{R}$  as the PPO redundancy vector, whose elements are redundancy factors  $R_m$ , each for a given PPO  $m$ , representing the number of PM  $n$  currently perceiving it:

$$R_m = \sum_{n \in N} p_{n,m} \quad (5.7)$$

The redundancy factor of a given PPO represents the number of available sources that can provide information about it to a PM, which, in our architecture, translates into the number of detections received within an update period from CPMs. The computational cost of handling assignment  $X$  for each PM  $n$  is described by:

$$C_{n,m}(X) = [\alpha_n + \gamma_n \cdot R_m] \cdot x_{n,m} \quad (5.8)$$

$$C_n(X) = \sum_{m \in M} [\alpha_n + \gamma_n \cdot R_m] \cdot x_{n,m} \quad (5.9)$$

where  $\alpha_n$  is the computational load factor of PM  $n$  due to a PPO  $m$  assigned to it and  $\gamma_n$  the additional computational load factor due to the detection of a PPO  $m$  received through CPMs (thus multiplied by the redundancy factor  $R_m$ ).

Thus, the total computational cost  $C(X)$  of handling the  $X$  assignment is given by:

$$C(X) = \sum_{n \in N} C_n(X) \quad (5.10)$$

For which the following constraint applies:

$$C_n(X) \leq \epsilon_n^T \quad (5.11)$$

where  $\epsilon_n^T$  the total available computational resources that PM  $n$  can allocate.

The computational fairness cost  $L(X)$ , can be described by the difference between the computational cost of each PM  $n$  and the average computational cost of the other  $PM_{i \neq n}$ :

$$L(X) = \sum_{n \in N} \left| C_n(X) - \frac{1}{1-N} \sum_{i \neq n} C_i(X) \right| \quad (5.12)$$

The probability of a given PM  $n$  to lose track of a given PPO  $m$  assigned to it, before the following assignment period  $t + \Delta t$ , is assumed to be proportional to the distance between PM  $n$  and PPO  $m$  at  $t + \Delta t$ :

$$P(p_{n,m}(t + \Delta t) = 0 | p_{n,m}(t) = 1) \propto d_{n,m}(t + \Delta t) \quad (5.13)$$

where

$$d_{n,m}(t + \Delta t) = d_{n,m} + v_{n,m} \cdot \Delta t + a_{n,m} \cdot \Delta t \quad (5.14)$$

with  $v_{n,m}$  and  $a_{n,m}$  denoting the speed and acceleration difference between PM  $n$  and PPO  $m$ , respectively.

The perception robustness cost  $D(X)$  of using assignment  $X$  is defined as the sum of the probabilities of each PPO  $m$  going out of perception range of the PM  $n$  assigned with it, before a new assignment is made:

$$D(X) = \sum_{n \in N} D_n(X) \quad (5.15)$$

where

$$D_n(X) = \sum_{m \in M} P(p_{n,m}(t + \Delta t) = 0) \cdot x_{n,m} \quad (5.16)$$

$$= \sum_{m \in M} d_{n,m}(t + \Delta t) \cdot x_{n,m} \quad (5.17)$$

In our system, the optimal assignment  $\hat{X}$  is the one that jointly minimizes the computational load, the computational fairness and perception robustness costs:

$$\min \quad \{C(X), L(X), D(X)\} \quad (5.18)$$

$$\text{subject to: (5.4), (5.5), (5.6), (5.11)} \quad (5.19)$$

In the following section, a detailed description of the proposed algorithms to perform the online assignment computation.

## 5.5 Proposed Heuristic Algorithms

Due to the highly dynamic nature of the information stored in the P-LDM, the optimal PPO assignment changes after every new P-LDM update period, requiring a quick algorithm to compute it. Furthermore, because conflicting objectives must be jointly optimized, the solution of the optimization problem is represented by a pareto-optimal set of assignments instead of a single one. The pareto-optimal solutions are the ones where no individual solution can be considered better than another when considering all objectives simultaneously.

In order to achieve an online computation of the Platoon Perceived Object assignment, the challenge of dealing with integer decision variables necessitates the use of heuristic algorithms due to the high inefficiency of exhaustive methods. Many works on similar multi-objective problems have highlighted the efficacy of Genetic Algorithms (GAs) in finding optimal or near-optimal solutions. Genetic Algorithms are adaptive heuristic search algorithms based on the evolutionary behavior of natural selection and genetics. Their ability to handle multiple objectives and constraints makes them particularly suitable for complex optimization tasks such as ours [44, 121, 144, 100].

Among the various Genetic Algorithms, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) is one of the most prominent and widely utilized. NSGA-II

is popular for its efficiency in handling multi-objective problems through its fast non-dominated sorting approach and its ability to maintain a diverse set of solutions. This algorithm enhances the search process by prioritizing non-dominated solutions (pareto-optimal), thereby effectively balancing the trade-offs between competing objectives [62].

Even though the NSGA-II provides a close to optimal pareto-front of PPO assignments, due to the high number of possible solutions for platoons of more than 10 vehicles in a reasonably congested traffic state (at least twice the number of PPOs), the convergence time becomes unsuitable for our service. Therefore in order to update the PPO assignment after every P-LDM update period, there is a need for an algorithm able to compute a solution of the pareto-optimal front in less than 100ms.

To this end, we have devised a greedy algorithm for the online PPO assignment selection. The idea behind the algorithm is that, instead of assigning PMs to PPOs, PPOs are assigned to PMs in order of cost. Two algorithms have been devised that differ in how the PPOs are ordered before assigning them and evaluating the different performances. The greedy algorithm, depicted in Algorithm 1, orders all PPOs from the least detected (i.e., the one that is detected by the least number of PMs) to the most detected.

The algorithm takes as input the current perception matrix  $\mathbf{P}$  as defined in Equation 5.2, the distance matrix  $\Delta$  with the values computed as defined in Equation 5.14, the vectors with computational load factor  $\alpha$  and computational cost coefficient due to redundancy  $\gamma$  of each PM. Additionally, weight coefficients  $\omega_c$  and  $\omega_d$  are taken as input for the computation of the combined computational and perception robustness costs of a given assignment. As the PPOs get assigned to PMs, a vector  $\mathbf{Cv}$  with the accumulated CPU costs of each PM is updated to compute the fairness cost.

At the beginning of the assignment process, a matrix  $\mathbf{R}$  of size  $2 \times M$  where each row contains the index of each PPO with the number of PMs currently perceiving it. Once the matrix  $\mathbf{R}$  is created, its rows are sorted according to the number of PMs perceiving each PPO obtaining  $\mathbf{R}'$ . For every PPO  $m$  in  $\mathbf{R}'$ , two vectors are defined  $\mathbf{Z}$  and  $\mathbf{N}'$ . The vector  $\mathbf{N}'$  is a subset of  $N$ , containing the indexes  $n'$ , which correspond to the PMs that perceive PPO  $m$ . On the other hand,  $\mathbf{Z}$  is the vector whose elements  $\mathbf{Z}_{n'}$  represent the combined computational and robustness cost of the PPO  $m$  seen by PM  $n'$ . Vectors  $\mathbf{Z}$  and  $\mathbf{A}$  (which have the same dimension) are then combined to

**Algorithm 3** Greedy Algorithm

---

```

1: procedure GREEDY( $\mathbf{P}, \Delta, \boldsymbol{\alpha}, \boldsymbol{\gamma}, w_c, w_d, \boldsymbol{\varepsilon}_n^T$ )
2:    $(N, M) \leftarrow$  dimensions of  $P$ 
3:    $\mathbf{R} \leftarrow [m, \sum_{n \in N} P_{n,m}]$  for  $m \in M$ 
4:    $\mathbf{C} \leftarrow$  CPU cost matrix from  $(\mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ 
5:    $\mathbf{D} \leftarrow$  Perception robustness cost matrix from  $(\mathbf{P}, \Delta)$ 
6:    $\mathbf{Cv} \leftarrow$  initialize CPU cost vector of size  $N$ 
7:    $\mathbf{X} \leftarrow$  initialize assignment matrix  $N \times M$ 
8:    $\mathbf{R}' \leftarrow \text{Sort}(R)$  – Sorted by  $\sum_{n \in N} P_{n,m}$ 
9:   for each object  $m \in R'$  do
10:     $\mathbf{N}' \leftarrow [n \text{ for } n \text{ in } N \text{ if } P_{n,m} = 1]$ 
11:     $\mathbf{Z} \leftarrow [w_c \cdot C_{n',m} + w_d \cdot D_{n',m}]$  for  $n' \in \mathbf{N}'$ 
12:     $\Upsilon \leftarrow [\mathbf{N}', \mathbf{Z}]$ 
13:     $\Upsilon' \leftarrow \text{Sort}(\Upsilon)$  ▷ Sorted by values of  $Z$ 
14:     $c \leftarrow n'$  of  $\Upsilon'[0]$  ▷ PM with lowest  $C(X), D(X)$ 
15:    if  $\sum_{m \in M} X_{c,m} = 0$  then
16:       $X_{c,m} = 1$  ▷ Assign object
17:       $Cv_c \leftarrow C_{c,m}$ 
18:    else
19:       $\mathbf{L} \leftarrow$  additional fairness costs for each  $n'$  in  $\Upsilon'$ 
20:       $\mathbf{L}' \leftarrow \text{Sort}(\mathbf{L})$ 
21:       $l \leftarrow n'$  of  $\mathbf{L}'[0]$  ▷ PM with lowest  $L(X)$ 
22:       $L_c \leftarrow \mathbf{L}'$  of member  $c$  ▷  $L(X)$  of PM from  $\Upsilon'[0]$ 
23:      if  $(Z_l - \Upsilon'[0]) > (L_c - \mathbf{L}'[0])$  then
24:         $X_{c,m} = 1$  ▷ Assign to lowest  $C(X), D(X)$ 
25:         $\mathbf{Cv}_c \leftarrow C_{c,m}$ 
26:      else
27:         $X_{l,m} = 1$  ▷ Assign to lowest  $L(X)$ 
28:         $\mathbf{Cv}_l \leftarrow C_{l,m}$ 
29:      end if
30:    end if
31:  end for
32: return  $X$ 
33: end procedure

```

---

create a matrix  $\Upsilon$ , i.e. each row  $\Upsilon_{n'}$  contains the index of the PM and its associated cost. This matrix is computed in Line 12 and then sorted in ascending order ( $\Upsilon'$ ) according to the cost values in Line 13. The first PPO is assigned to the PM with the lowest cost (i.e.,  $\Upsilon'[0]$ ). For the following assignments, the fairness cost ( $L(X)$ ) is computed according to the current computational cost vector ( $Cv$ ). The fairness cost of each potential assignment is stored in a vector  $L$  and sorted in ascending order ( $L'$ ) in Line 20. In line 23, the algorithm compares the Computational and Robustness cost difference with the Fairness cost difference. The former is the difference in assignment cost if the object is assigned to the node with the lowest  $C(X), D(X)$  cost versus assigning it to the PM that achieves the best fairness, whilst the latter is the difference in fairness cost if the object is assigned to the PM achieving the best fairness versus the node with the lowest  $C(X), D(X)$  cost. If the difference in  $C(X), D(X)$  cost for the PM with best fairness is greater than the difference in fairness cost for the PM with the best  $C(X), D(X)$  cost, then the PPO is assigned to the PM with the lowest  $C(X), D(X)$ . Otherwise, it is assigned to the PM providing the lowest fairness cost (i.e.,  $L'[0]$ ).

The idea of sorting PPOs according to the number of PMs currently detecting them, starting from the least perceived, comes from the fact that the least detected object has fewer candidates to assign it to. Hence, this strategy avoids assigning objects to PMs that are not close enough to the object and cannot ensure a robust perception. However, this strategy has one downside, namely that the most perceived object will yield the highest computational cost due to its high redundancy factor, as explained in Section 5.4. Indeed, assigning the objects with the highest redundancy factor as last, results in poor granularity in the computational cost of assignments, thus lowering the possible achievable fairness.

In order to analyze the trade-off between the different approaches, we have considered two possible versions of our greedy algorithm for the performance evaluation presented in this work, which we have called *least2most* and *most2least*. They only differ by the sorting order performed in Line 8 of Algorithm 1.

For what regards the computational complexity of both *least2most* and *most2least* can be analyzed as follows. The outer loop, iterating over each PPO in  $\mathbf{R}$ , thus being performed  $M$  times. Within this loop, the process for computing  $\blacksquare$  has a complexity of  $\mathbf{O}(N)$ . Additionally, sorting assignable PMs incurs a complexity of  $\mathbf{O}(N \log(N))$ . Lastly, the computation of the fairness costs vector  $L$ , a comparison between all

PMs' computational and distance cost needs to be performed, rendering a complexity of  $\mathbf{O}(N^2)$ . Finally, the overall computational complexity for both algorithms is  $\mathbf{O}(MN^2)$ .

### 5.5.1 Performance Evaluation

To evaluate the performance of our proposed algorithms comprehensively, we have developed a custom python implementation of the NSGA-II algorithm in addition to the greedy algorithm described. Our custom implementation includes all the functionalities of the NSGA-II algorithm with tailored implementations for population initialization, crossover, and mutation to ensure that our constraints are met. In particular, in our implementation, all members of the population (i.e. matrix  $X$  defined in Section 5.4) are such that every PPO is assigned to a single PM. For the initial simulation campaign, we considered a scenario with a platoon size of 10 Platoon Members and a total of 25 Platoon Perceived Objects. For each simulation, the position of PPOs is randomly generated and then arranged so that each one is perceived by at least one PM. Each computational load factor associated with assigning a PPO to a PM (i.e., the assigned values of  $\alpha_n$  defined in (5.8)) has been randomly selected from the set [0.05, 0.1, 0.2] to account for different computational capabilities amongst PMs. In a similar manner, the cost factor due to the redundancy of a PPO (i.e., the assigned values of  $\gamma_n$  defined in Eq. 5.8) has been randomly selected from the set [0.01, 0.05, 0.1]. For the configuration of the NSGA-II algorithm, a population of 100 assignments was used with a mutation rate of 0.1 and a total of 100 generations. Being the output of the NSGA-II, a Pareto front of non-dominated solutions, for each simulation, only the solution with the lowest weighted average (with equal weights) is considered.

The results in terms of optimization performance are shown in Figure 5.8, showcasing the box plots of all the optimization cost outcomes for all the 100 simulations considered. As our objective is to minimize all the costs defined in Eq. 5.18, the lowest values reported represent the best assignments. As depicted in Figure 5.8a, the NSGA-II provides the best assignment compared to the greedy algorithms across all the objectives of interest, providing the best assignment of the three algorithms. As expected, the *most2least* algorithm achieves a lower fairness cost, depicted in Figure 5.8c, compared to the *least2most* counterpart. These results showcase the benefits of assigning the objects that are perceived by most PMs first, as those are the

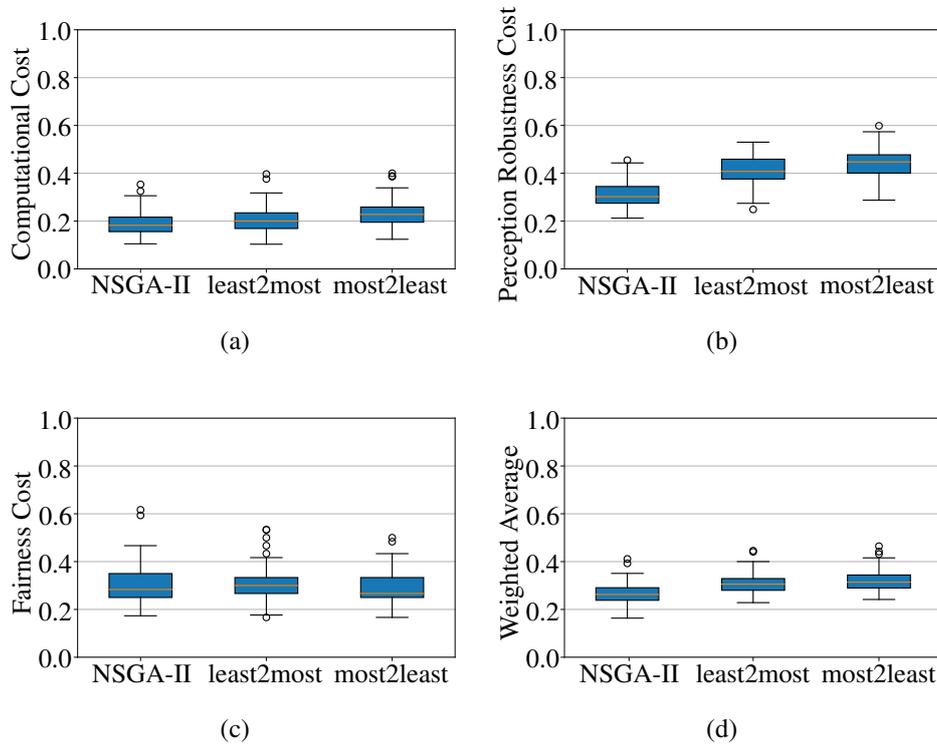


Fig. 5.8 Optimization cost achieved by each algorithm for each of the considered costs (a) CPU cost, (b) robustness cost, (c) fairness cost and (d) weighted average of the three costs in Equation (5.18) with equal weights.

more 'expensive' ones, hence, the ones requiring the best possible assignment. When considering the perception robustness cost, shown in Figure 5.8b, the *least2most* algorithm archives a slightly better performance compared to the *most2least*. As anticipated, the *least2most* achieves a better assignment for the PPOs perceived by the least PMs, statistically further away than the rest of PPOs. As in our assignment problem, the objective is to jointly minimize all three computational, robustness, and fairness costs; the weighted average is shown in Figure 5.8d. Intuitively, the NSGA-II algorithm provides the best performance, with the *least2most* being the best of the greedy algorithm solutions.

As hinted at the beginning of the section, even though the NSGA-II provides the best results of the considered heuristics, the performance showcased is achieved at the expense of a considerable number of iterations. As can be seen in Figure 5.9, the NSGA-II takes over 30 seconds on average to converge, which renders it unsuitable

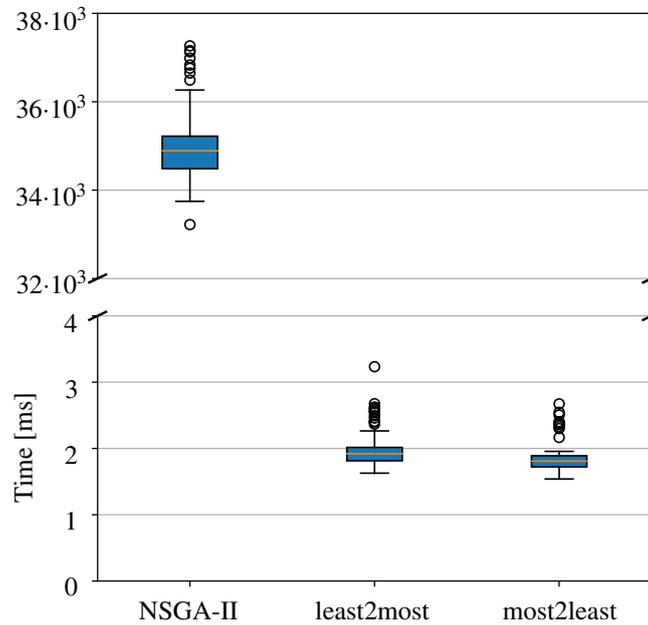


Fig. 5.9 Execution time of each algorithm for a scenario with 10 Platoon Members and 25 Platoon Perceived Objects.

for the target assignment period of 100ms. On the other hand, both greedy algorithms managed to produce an assignment selection in under 2ms, on average.

### 5.5.2 Scalability

Based on the performance evaluation, the NSGA-II achieved better results than the two greedy counterparts. However, as the computation time for the NSGA is over a full minute on average for a scenario with 10 PMs and 25 PPOs, the greedy algorithms become the only viable solution for the online assignment of PPOs. We have performed a scalability analysis in order to evaluate the performance of all algorithms in terms of their optimization performance as well as their execution time under more demanding scenarios.

Considering a platoon with 10 PMs, our first evaluation consisted of increasing the number of PPOs to be assigned. The results depicted in Figure 5.10 showcase the average execution time for each algorithm with respect to the number of PPOs. The average execution time for each algorithm shows a linear behavior for an increasing

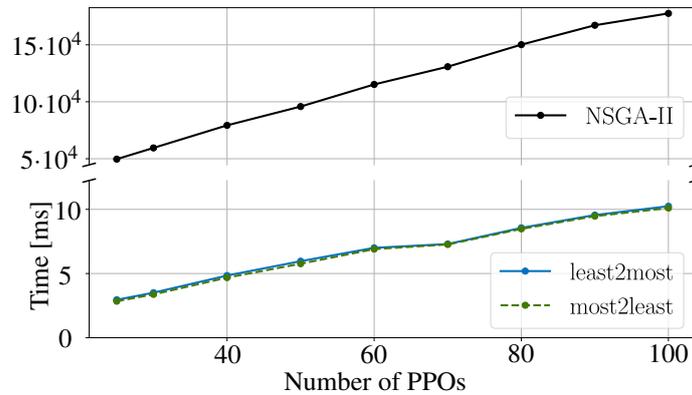


Fig. 5.10 Average execution time of each algorithm for 10 Platoon Members and an increasing number of Platoon Perceived Objects.

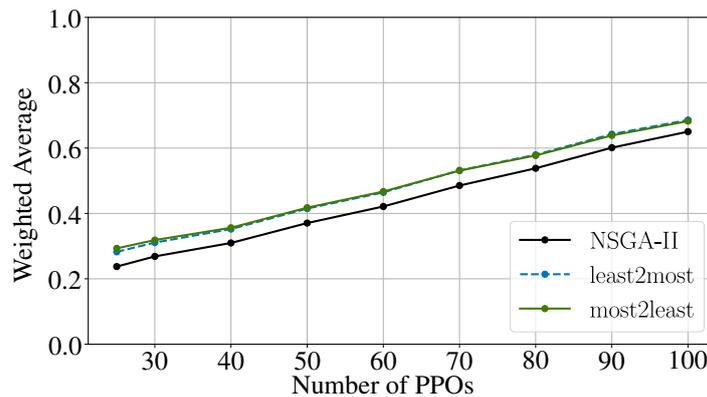


Fig. 5.11 Weighted average of PPO assignment cost of each algorithm for 10 Platoon Members and an increasing number of Platoon Perceived Objects.

number of PPOs to be assigned for all algorithms. Extending the results shown in Figure 5.9, both greedy algorithms achieve identical execution times for all the scenarios.

In terms of cost, when the number of PPOs is increased for a given number of PMs, the NSGA-II algorithm outperforms both greedy algorithms, as shown in Figure 5.11. However, when the number of PMs and PPOs is increased simultaneously, the NSGA-II algorithm performs worse than both greedy algorithms. This is evident from the results presented in Figure 5.12, which show the outcomes of doubling the number of PPOs along with increasing the number of PMs. These results reflect the limitation of the NSGA-II for extensive search spaces, which

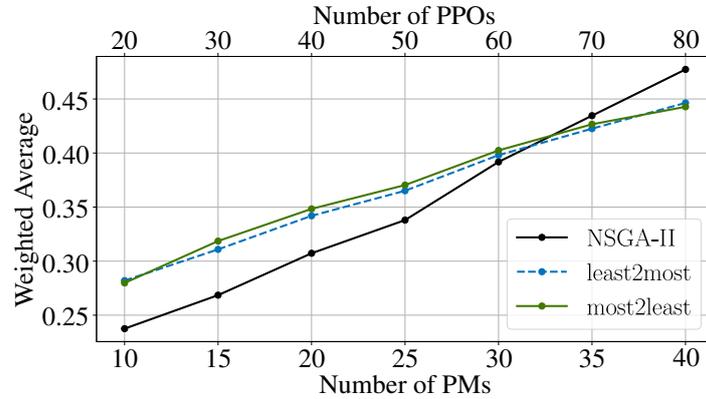


Fig. 5.12 Weighted average of PPO assignment cost of each algorithm for an increasing number of Platoon Members, where the number of Platoon Perceived Objects is twice the number of Platoon Members.

require a bigger population and more generations to achieve good enough results. Indeed, when both the number of PM and PPOs are increased, the search space grows exponentially, reflected in the execution time for all the algorithms, depicted in Figure 5.13. Thus, from these scalability results, we have found that our custom greedy algorithm can outperform the NSGA-II with the configuration considered in our simulations, which were selected for a trade-off between performance and execution time for all scenarios. The scalability results for the NSGA-II algorithm are shown here only as a reference, since its execution time was unacceptably high even for the base scenario described in Section 5.5.1. As pointed out, the Genetic Algorithm needs different configurations for different search spaces to obtain the best trade-off between execution time and performance. To simplify the comparison, we have always considered the same population size, number of generations, and mutation rate, which have resulted in the Genetic Algorithm being over-dimensioned for simple cases and under-dimensioned for the more complex ones. Indeed, the results of our scalability analysis show that for complex scenarios with 40 PMs and 80 PPOs to be assigned, our custom greedy algorithms not only outperform the NSGA-II algorithm but also achieve execution times lower than 30ms, below the target of 100ms needed by the P-LDM.

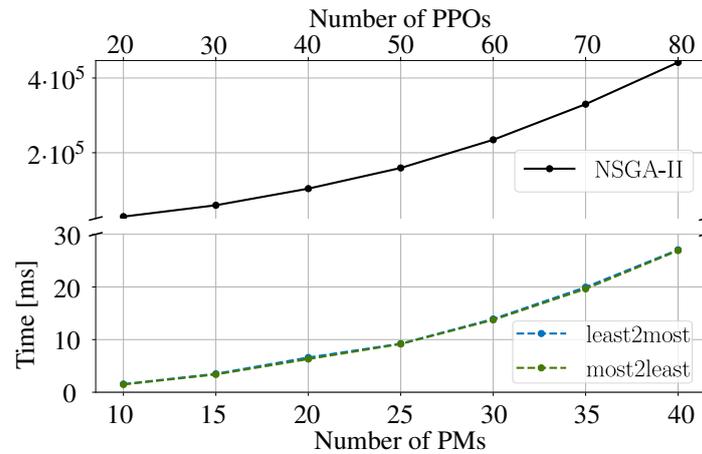


Fig. 5.13 Average execution time of each algorithm for an increasing number of Platoon Members, where the number of Platoon Perceived Objects is twice the number of Platoon Members.

## 5.6 Service Validation

To validate our service in conjunction with our assignment selection cost model, we have implemented our system using ms-van3t. In particular, we have leveraged ms-van3t's CARLA integration as presented in Section 3.4, which relies on CARLA for mobility, perception, and control physics modeling, OpenCDA as middleware for sensor fusion and vehicle control libraries, and ms-van3t for modeling wireless communications among vehicles using a full C-ITS stack.

In addition to the presented service, we have implemented the platooning protocol presented in [138] for the ENSEMBLE project, leveraging 2 newly defined messages:

- **Platoon Management Messages (PMM):** messages for handling join requests and join responses for the management of a given platoon.
- **Platoon Control Messages (PCM):** messages containing all necessary information for the Platoon Members' Cooperative Adaptive Cruise Control (CACC) controller. This message is transmitted with a frequency of 20Hz.

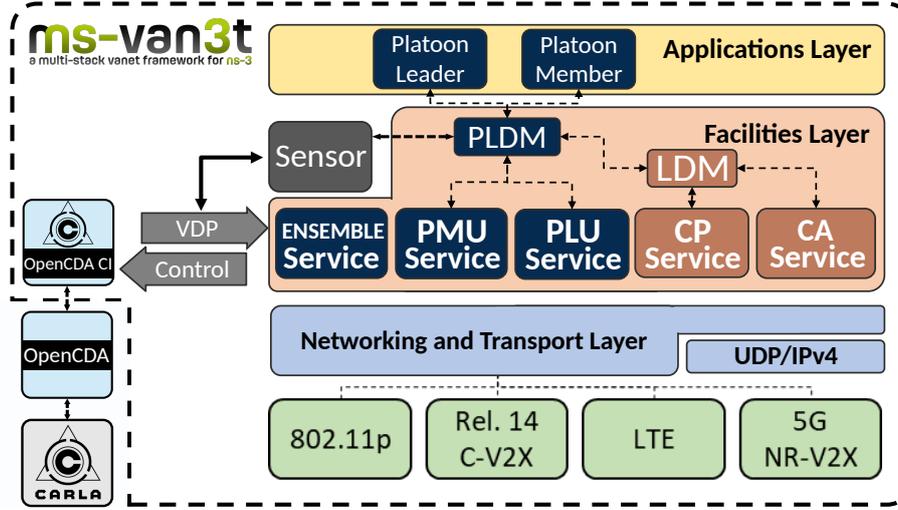


Fig. 5.14 Architecture of the ms-van3t-CARLA framework leveraged in our validation simulations.

### 5.6.1 Simulation setup

All the Platoon Members in our simulations are controlled by the CACC controller that we have implemented in ms-van3t based on the one defined in [122], commonly leveraged in other simulation frameworks such as Plexe [128]. The CACC controller works under a constant spacing policy to secure string stability along the platoon, relying on the dynamic information of the platoon leader and the preceding platoon member. According to the information available about other PMs received through PCMs, and the information about PPOs found in the LDM (or P-LDM) database, each  $PM_i$  computes a desired acceleration according to the following formulas:

$$\begin{aligned} \ddot{x}_i = & (1 - C_1)\ddot{x}_{i-1} + C_1\ddot{x}_\ell - (2\xi - K_1\alpha)\omega_n\dot{\varepsilon}_i \\ & - \alpha\omega_n C_1(\dot{x}_i - \dot{x}_\ell) - \omega_n^2\varepsilon_i \end{aligned} \quad (5.20)$$

$$\varepsilon_i = x_i - x_{i-1} + L_i + \Delta x_{des} \quad (5.21)$$

$$\dot{\varepsilon}_i = \dot{x}_i - \dot{x}_{i-1} \quad (5.22)$$

$$\alpha = \xi + \sqrt{\xi^2 - 1} \quad (5.23)$$

where  $\dot{x}_\ell$  and  $\ddot{x}_\ell$  are the speed and acceleration of the PL while  $\dot{x}_{i-1}$  and  $\ddot{x}_{i-1}$  are the speed and acceleration of the preceding PM (if  $i = 1$  then  $PM_{i-1} \rightarrow PL$ ). The speed and distance error,  $\dot{\varepsilon}_i$  and  $\varepsilon_i$  respectively, are computed according to a

desired  $\Delta x_{des}$  inter-distance between members and a given  $L_i$  PM length. Lastly,  $C_1$ ,  $\xi$ , and  $\xi \omega_n$  represent the weight coefficient for the PL's speed and acceleration, the damping ratio, and the controller bandwidth, respectively.

Once the desired acceleration is computed, it is then passed to the OpenCDACI module (shown in Figure 5.14) where it gets translated into a control (throttle, brake and steering) input leveraging OpenCDA's PID controller.

A simulation campaign has been performed to jointly evaluate the key aspects of the cooperative perception concept that our system aims to improve:

- **Computational load**
- **Perception Robustness**
- **Platoon awareness**

To this end, we have designed a custom CARLA map consisting of a 4-lane 3-km strip of highway with an entrance ramp merging with the main highway at the 1.2 km mark. In addition to platooning vehicles, a total of 60 background vehicles are evenly distributed on the strip to achieve a 20 vehicle/km density. Lastly, an intruder vehicle is spawned on the entrance ramp set to interfere with the platoon execution. In terms of the cooperative perception scheme, the two scenarios considered for the simulations, already introduced in Section 5.3.1, were:

- **P-LDM:** representing the scenario on which each PM runs the P-LDM service and attempts to match only the objects with the IDs assigned to it.
- **Platoon Cooperative Perception (Platoon CP):** representing a scenario in which the PL and PMs process the information of all objects detected by other PMs.

To evaluate the trade-offs on both schemes, two different ITS penetration rates have been considered for the simulated background vehicles, namely, 0.1 and 0.5. The ITS penetration rate determines the ratio of background vehicles equipped with a communication device for the exchange of V2X messages. Hence, a penetration rate of 0.1 yields a scenario in which only 10% of vehicles traveling on the highway are V2X-enabled, representing a near-future case. The remaining background vehicles

are potential PPOs for the platoon. Regarding the V2X-enabled vehicles communication setup, they are equipped with 802.11p OBUs on a 20 MHz channel at a central frequency of 5.9 GHz, configured with a physical data rate of 12 Mbit/s, and a transmission power of 26 dBm. As for the perception configuration, all V2X-enabled vehicles are equipped with four RGB cameras, whose feeds are processed by the YOLOv5 object detection model. Additionally, a LiDAR sensor is configured, and its output is used for late fusion to achieve 3D object detection. Furthermore, the platoon sizes were set to 10, 15, and 20, respectively, to analyze the effect of the platoon size on the perception range and platoon reaction to an intruder vehicle. The intruder vehicle, representing a human-driven vehicle, has been simulated to systematically interfere in the middle of the platoon by cutting in between two members (depicted in phase 3 of Figure 5.15), relying on the ground truth information from the CARLA simulation for the maneuver. Instead, all platoon members rely on the information from platoon control messages and the LDM or P-LDM databases, with which, at every CACC controller step, a check is performed to verify the vehicle or perceived object in front to adjust its acceleration accordingly. Lastly, the PPO assignment selection for the P-LDM scenario has been performed with the *least2most* algorithm detailed in Section 5.5.

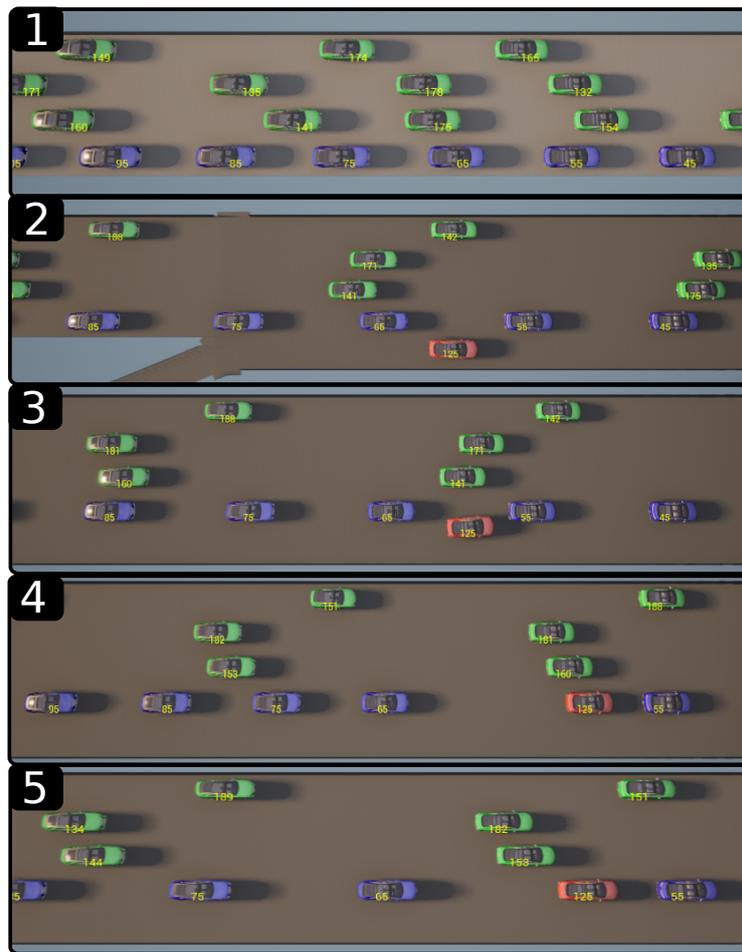


Fig. 5.15 Scenario timeline showing the different phases of the considered simulation scenario: 1. Normal Platooning state (PMs colored in blue and background vehicles colored in green). 2. Intruder vehicle (colored in red) approaches the platoon. 3. Intruder vehicle starts the cut-in maneuver between two PMs. 4. PMs react to the intruder interference by slowing down to create a safe gap. 5. The platoon recovers its normal state, adjusted to the presence of the intruder.

### 5.6.2 Perception computational load analysis

One of the main purposes of the P-LDM is to distribute the processing power dedicated to cooperative perception among all the members of a platoon of vehicles. Indeed, upon reception of a CPM, each PM attempts to match each received object with the ones stored in its own LDM database. As the number of objects stored in the LDM and the number of objects included on each CPM increases, so does

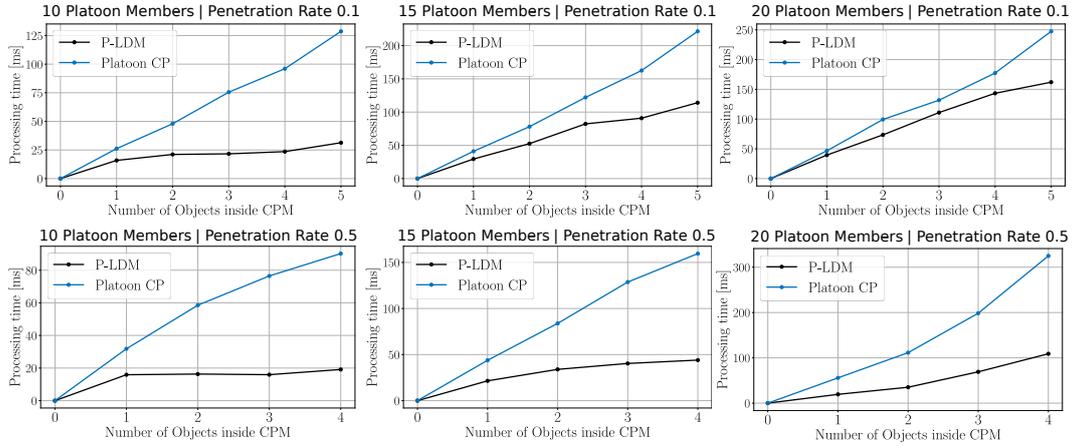


Fig. 5.16 Average processing time of each received message (CPMs) for each number of Platoon Perceived Objects included in each message.

the processing time needed to perform such matching. The LDM matching process considered for this evaluation considers the same scheme detailed in Section 3.5.1.1.

In our framework, each simulation utilizes two primary processes in addition to the CARLA server: one for ns-3 and one for OpenCDA. Consequently, all vehicle perception processing within the simulation is performed sequentially by OpenCDA, using the same hardware resources. This sequential processing approach prevents real-time simulation and makes CPU usage an unsuitable metric for evaluating our service. Therefore, we have chosen processing time as the metric to evaluate the computational load of each scenario. Specifically, we focus on the processing time of CPMs, as they are the most demanding task in our cooperative perception pipeline, apart from local perception. For each simulation in our campaign, the processing time of each CPM is measured from the moment the message is received and passed on to the message handler running in OpenCDA until all objects have been updated in the database. The time measured includes the creation of the bounding box, the matching process according to the intersection over union, and the update or creation of the Kalman filter for each PPO received.

The processing time of a CPM at time  $t$  depends on the number of objects stored in the vehicle's database at that time instant and the number of objects included on the received CPM, translating into the dimensions of the matching matrix. However, in the case of the P-LDM, thanks to the ID synchronization of objects and the assignment scheme used, only the objects assigned by the Platoon Leader to the PM receiving the CPM are processed to be matched, while the rest are discarded. As can

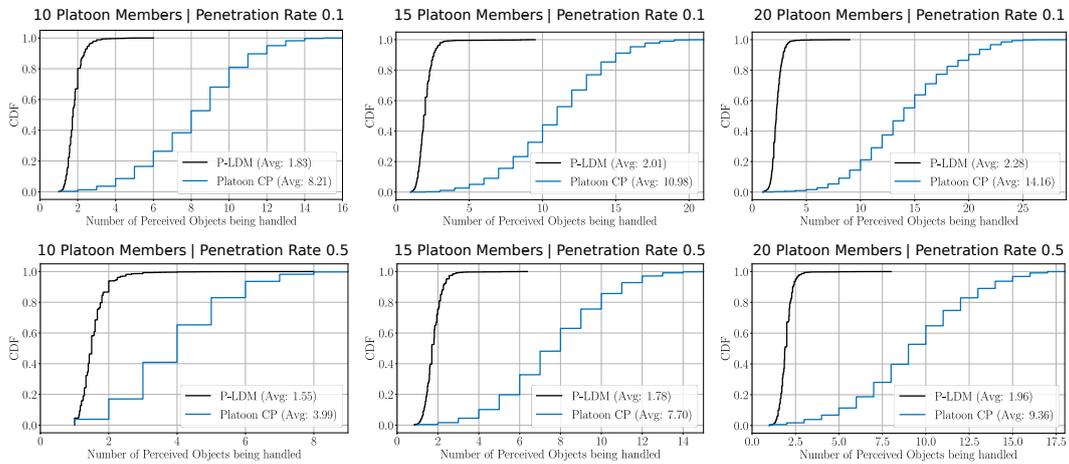


Fig. 5.17 Cumulative Distribution Function of the number of Platoon Perceived Objects being handled, i.e., being processed for each CPM reception.

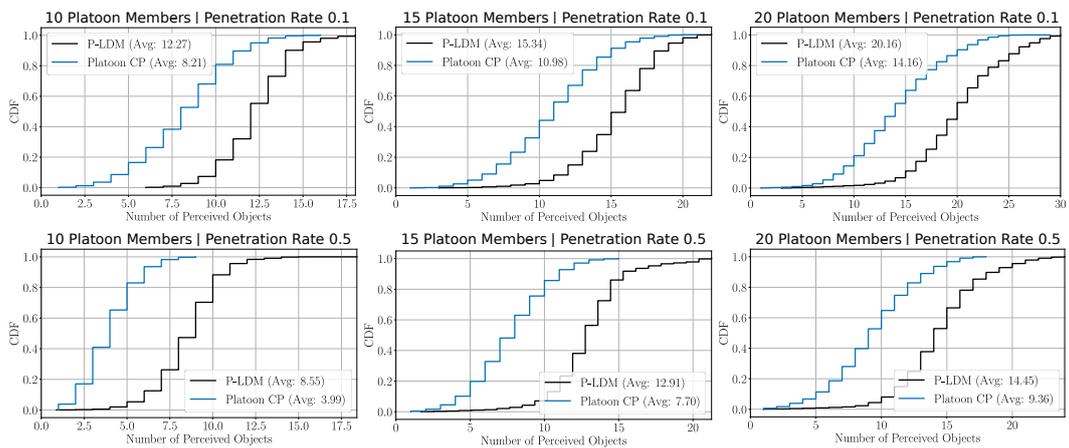


Fig. 5.18 Cumulative Distribution Function of the total number of Platoon Perceived Objects stored in the database at a given point in time.

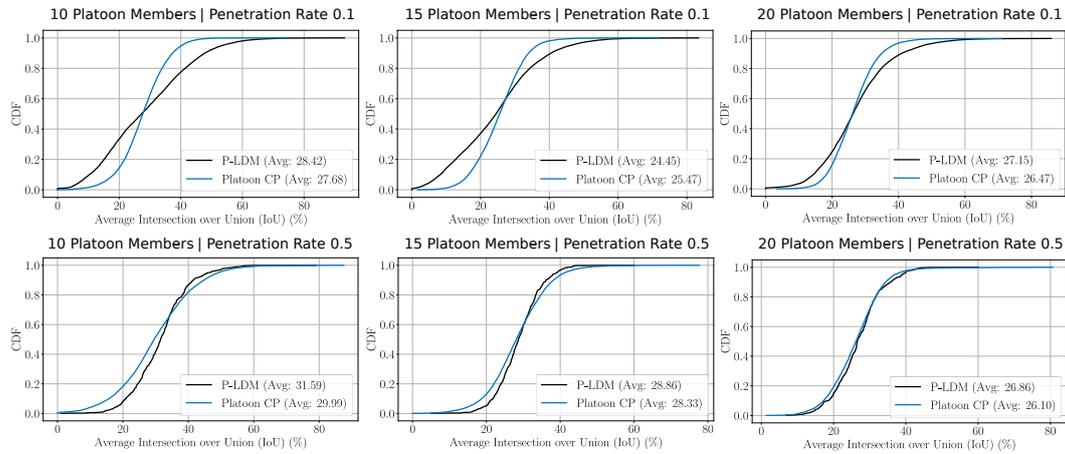


Fig. 5.19 Cumulative Distribution Function of the average Intersection Over Union (IoU) of all objects stored in the database at a given point in time.

be seen from Figure 5.16, in the Platoon CP scenario, the processing time of CPMs increases linearly with the number of objects included in each CPM and the number of objects being stored and maintained by the receiving PM. On the other hand, when leveraging the P-LDM, the processing time increases linearly up to the average number of assigned objects, reported in Figure 5.17 for each platoon size considered and a penetration rate of 0.1 (representing the most demanding case). Indeed, for CPMs with more objects than the ones assigned to each PM, the processing time increases with a flatter behavior compared to the Platoon CP counterpart, where all objects are processed. As the number of Platoon Members increases, the processing time for each CPM becomes more dependent on the total number of objects stored in the database as opposed to the number of objects present on each CPM.

Intuitively, for a higher penetration rate, the number of objects stored in the database decreases for both scenarios, as showcased in Figure 5.18. As a result, when considering the P-LDM, each PM is assigned with fewer PPOs, achieving lower processing times for the same number of objects included in CPMs. On the other hand, for the Full CP scenario, there is no significant difference in the processing time for equally sized CPMs when the penetration rate is increased. It is worth noting that even if there are fewer perceived objects for higher penetration rates, both connected and detected vehicles are considered for the matching process. Indeed, when changing the penetration rate, the combined sizes of PPOs and CVs stored in the databases remain stable.

### 5.6.3 Perception robustness

The cooperative perception concept has been devised with the objective of expanding the local view of each CV for the execution of the so-called Day-2 and beyond [58] set of applications, such as platooning. Indeed, as the vehicle 'view' of the road broadens, maneuvers are executed in a more timely and safe manner. As outlined at the beginning of this chapter, many challenges arise in keeping an updated view of the road through the distributed exchange of CPMs without creating unnecessary redundancy, resulting in both communication and computational overhead.

For our simulation campaign, in addition to measuring the performance of the P-LDM in terms of computational load, we have measured the achieved robustness of the cooperative perception of the road. The two Key Performance Indicators (KPIs) we have chosen for perception robustness are the total number of PPOs stored in the database and the average IoU of all PPOs. For each simulation step (50 milliseconds), the database of each Platoon Member has been compared to the ground truth extracted from CARLA to compute the average IoU of all perceived objects stored. For both P-LDM and Platoon CP, each entry in the database is retained for up to 2 seconds. If not updated by a V2X message or local perception, it is deleted. Such criterion has been taken considering that all perceived objects in both scenarios are, in fact, vehicles that are highly dynamic. Indeed, overly outdated perceptions can hinder the execution of applications relying on them as, in our case, for platoon control.

The results for the different numbers of platoon members and penetration rates are shown in Figure 5.19, where the Cumulative Distribution Function of the measured average IoU is depicted for both P-LDM and Platoon CP scenarios. When considering a penetration rate of 0.1, the distribution of Platoon CP initially provides higher IoU values than the P-LDM for all platoon sizes. This behavior is obtained due to local perceptions being updated with a higher frequency compared to the P-LDM messages, resulting in 'fresher' (more accurate) perceptions. In the Platoon CP scenario, the average perception accuracy is biased by the high accuracy of local perception, as most stored perceptions are indeed local at any given time. On the other hand, the P-LDM entries are updated only by the assigned PM. As a result, objects are updated solely through P-LDM messages, even if they are perceived locally. Thus, the average of all the perceptions at a given point in time does not contain the bias from local perception, resulting in a flatter distribution. Moreover,

to ensure computational fairness alongside perception robustness, the assignment algorithm sometimes allocates objects to PMs that are not the closest. This is particularly noticeable for objects located at the edges of the platoon, which, together with the skipped updates when an object assignment is changed from one PM to another, results in lower average accuracy measurements by the P-LDM for 50% of the time. Nevertheless, for all the considered cases, with the exception of 15 Platoon Members and a penetration rate of 0.1, the P-LDM achieves a slightly higher perception accuracy overall. It is important to note that while the P-LDM does not achieve a significant increase in perception accuracy, it does significantly increase the number of perceived objects stored in the database, as demonstrated in Figure 5.18, without sacrificing accuracy.

When analyzing the cases with a penetration rate of 0.5, the overall accuracy shows similar values for both scenarios. In the case of Platoon CP, the local perception bias fades as there are fewer local perceived objects to generate it. Indeed, both distributions show similar trends, with even the P-LDM showing a sharper distribution product of fewer objects to be assigned to PMs, allowing for a better assignment in terms of perception robustness.

#### **5.6.4 Platooning awareness**

After analyzing the benefits of leveraging the P-LDM in terms of computational load and perception robustness, it is now necessary to evaluate how they translate into the effective execution of platooning maneuvers. To this end, we have configured our simulations to include an intruder vehicle to intervene with platoon control execution by cutting in between two members. The awareness of this intruder representing a legacy, non-connected, and human-driven vehicle relies solely on the sensor perception of PMs. Thus, the platoon's timely and safe reaction will depend on the achieved cooperative perception.

The results for the platoon speed profile is shown in Figure 5.20 for a platoon size of 10 members leveraging the P-LDM on the left and leveraging the Platoon CP scheme on the right. At the beginning of each 60-second simulation, the platoon is already formed, and all vehicles are stationary. The PL, programmed to reach a desired platoon speed of 20 m/s, starts speeding up, and all PMs adjust their control inputs accordingly to match the PL's speed profile, as depicted in Figure 5.20 from

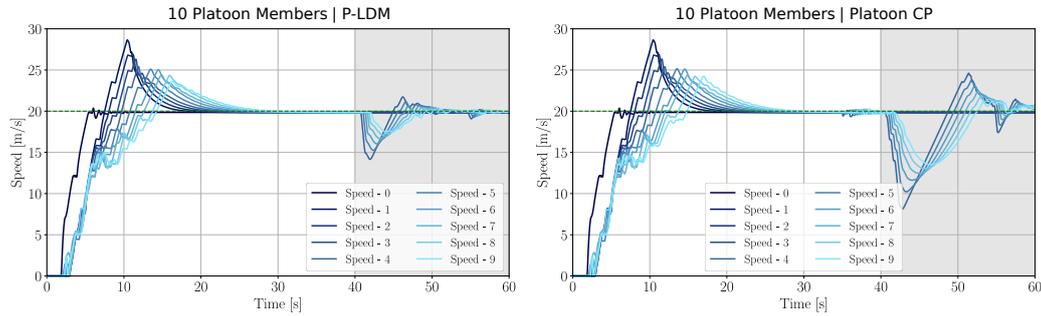


Fig. 5.20 Instantaneous speed of each Platoon Member over time in m/s. The green dashed line at 20 m/s represents the desired platoon speed. The grey background signals the presence of the intruder within the platoon.

0 to 30 seconds of simulation runtime. Once the entire platoon reaches the desired speed (phase 1 in Figure 5.15), the intruder vehicle traveling along the merging entrance ramp adjusts its speed to arrive at the merge lane after 35 seconds (phase 2 in Figure 5.15), simultaneously with the platoon. After an adjusting period, in which the intruder reaches a suitable spot between 2 PMs, the cut-in maneuver is performed (phase 3 in Figure 5.15), highlighted with a grey background in Figure 5.20. As hinted at the beginning of the section, all PMs perform an obstacle check on each controller step. For this check, either the PM's LDM (in the Platoon CP scenario) or P-LDM is queried to provide all potential vehicles to be found in front of it. When a vehicle different from the preceding PM is found to be the immediate vehicle in front, all the information needed for the CACC controller calculations is taken from the LDM/P-LDM instead of the dedicated Platoon Control Messages. After the intruder has started interfering with the platoon, and as soon as the PM behind is aware of it, the platoon performs the necessary adjustments to restore the original distance between members as if the intruder were a new member (phases 4 and 5 in Figure 5.15). It is worth mentioning that the intruder is configured to follow the platoon until the end of the simulation, locking its speed onto the one of the Platoon Leader. As can be seen, when the intruder starts interfering with the platoon, the preceding vehicle and the rest of the platoon members reduce their speed until the desired inter-distance of 10 meters with the intruder is restored. Since the desired acceleration computed by the CACC controller depends not only on the distance with the vehicle in front but also on the speed, the speed estimation of the intruder perception is key for the correct platoon reaction. When comparing the behavior relying on the P-LDM with the one of the Platoon CP, it can be observed that the

initial speed reduction by the PMs is much more abrupt in the Platoon CP. The reason behind these different behaviors is the different speed estimation achieved in the two scenarios, with the P-LDM deriving a more accurate value by keeping track of the intruder longer than the Platoon CP does. In the Platoon CP scenario the intruder's speed estimation is mostly computed through local perceptions with sporadic updates from received CPMs. Given that the speed estimation in our system is done using a Kalman filter with position samples, the longer the object is tracked, the more accurate the speed estimation is going to be. Thus, since the P-LDM perception of the intruder is tracked for a longer time, the speed estimation is more accurate at the moment the cut-in maneuver is performed by the intruder, and the PM does not overreact.

It is worth mentioning that the results shown in Figure 5.20 correspond to a single simulation, and it is shown here for descriptive purposes. In order to provide a more comprehensive view of the results, we have aggregated the results into three KPIs to compare the platoon reaction to an intruder: the average speed difference between PMs and PL, the minimum distance measured between PMs and the minimum Time To Collision between PMs. All the KPIs are computed with the measurements from the moment the cut-in maneuver begins (at around 40 seconds) until the end of the simulation. Intuitively, the results for the mentioned KPIs are more influenced by the platoon size compared to the penetration, with the latter affecting the distribution of vehicles to be detected and not affecting the perception of a single vehicle (the intruder in this case). For this reason, we only present results for a penetration rate of 0.1.

The average speed difference between PMs and PL has been computed with the values of all speed differences between each PM and the PL along the duration of the intrusion manoeuvre, denoting not only the speed adjustment at the intrusion but also the duration of the adjustment until the PMs recover the nominal speed. For the case of 10 platoon members, the P-LDM manages to achieve a much lower speed difference, resulting in a safer, faster reaction. Indeed, the minimum measured distance between PMs is not lower than 7 meters on average, as opposed to 3.5 meters for the Platoon CP scenario. However, as the platoon size increased, the P-LDM scenario highlights a remarkable performance deficit compared to Platoon CP for the same number of members. Instead, due to the fact that local perception is the main source for the Platoon CP, very similar results are obtained for all platoon sizes. The reason behind such a worse performance of the P-LDM with a higher

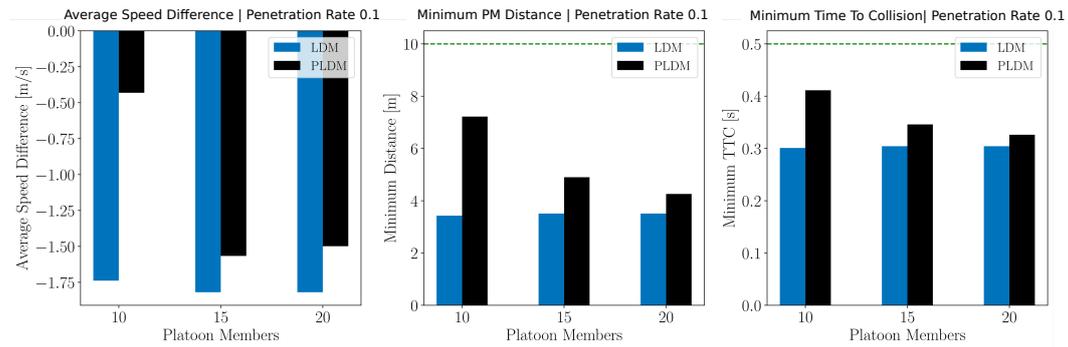


Fig. 5.21 Average member speed difference with platoon leader at intruder cut-in, average member inter-distance at intruder cut-in and average Time To Collision between members at intruder cut-in.

number of PMs lies in the higher number of candidates for the assignment of the intruder object, resulting in many more assignment changes during the considered intrusion. Such changes in assignment, from one PM to another, result in the control process using delayed perceptions of the intruder. Although an abrupt change in performance is obtained for the average speed difference, which is affected by the duration of the platoon reaction, the minimum distance and minimum time to collision measurements show a more linear performance decrease. Thus, when leveraging the P-LDM, a safer platoon reaction is achieved compared to the Platoon CP baseline for all platoon sizes, albeit with slower re-adjustment when platoon sizes are increased.

# Chapter 6

## Conclusions

As the trend toward vehicles becoming increasingly automated, the approach to enhancing their safety is shifting from isolated, vehicle-centric solutions to a holistic, system-wide vision. This transformation is driven by the idea that no single vehicle, however advanced, can ensure optimal safety and efficiency on its own. Instead, the future of transportation lies in seamless collaboration between vehicles, infrastructure, and road users that has given birth to the concept of Cooperative, Connected, and Automated Mobility (CCAM). By enabling real-time communication and coordination, CCAM has the potential to revolutionize road safety, streamline traffic flow, and reduce environmental impact, marking a fundamental shift in how we design and utilize transportation systems.

The evolution of CCAM deployments demands robust communication protocols to support V2X applications across diverse scenarios. To establish a solid foundation, Chapter 2, provided a detailed overview of the key protocols and technologies, including DSRC and cellular-based technologies. A comparative analysis of these technologies underscored the advantages and limitations of each, with DSRC demonstrating lower latency in single-hop scenarios and cellular technologies like LTE-V2X and NR-V2X excelling in scalability and long-range communication. The exploration of the ETSI MEC framework, with its IP-based communication protocols, such as AMQP 1.0, revealed its critical role in enabling seamless V2X communication across heterogeneous environments.

Realistic and scalable simulation tools are indispensable for the design and evaluation of innovative solutions for CCAM before their effective deployment. In order

to obtain meaningful insights on the potential benefits of vehicular applications, simulation tools need to model not only the mobility aspects but the wireless communication exchange as well. These motivations have given birth to *ms-van3t*, a state-of-the-art simulation and emulation framework tailored to large-scale V2X scenarios. This open-source tool incorporated advanced features, such as pre-recorded GNSS traces and compliance with ETSI C-ITS standards, enabling the realistic evaluation of various V2X access technologies, including IEEE 802.11p, LTE-V2X, and NR-V2X. Performance evaluations demonstrated that NR-V2X achieved the lowest one-way latency (approximately 10 ms) and the highest Packet Reception Ratio (PRR) under high-density traffic conditions, making it a robust candidate for future V2X applications. To further enhance the realism of simulations, the integration of *ms-van3t* with the CARLA simulator was developed, as detailed in Chapter 3. This co-simulation framework uniquely enabled the generation and exchange of Collective Perception Messages (CPMs), facilitating the study of cooperative perception services in realistic vehicular scenarios.

The proliferation of connected vehicles has resulted in an immense wealth of contextual data, offering unprecedented opportunities to enhance traffic safety and efficiency through cooperative perception services. Effectively leveraging this data requires architectures capable of handling vast volumes of information while maintaining the low latency and high reliability demanded by safety-critical applications. Multi-access Edge Computing (MEC) provides a crucial framework by bringing computational resources closer to vehicles, reducing latency, and enabling real-time data processing. However, MEC's reliance on centralized infrastructure can face challenges when scaling to meet the dynamic and data-intensive requirements of vehicular systems. To address these challenges, Chapter 4 introduced the Server Local Dynamic Map (S-LDM), a 5G-enabled MEC service that integrates real-time and historical data into a centralized, dynamic map of the road environment. Acting as a middleware entity, the S-LDM efficiently filters and processes data to deliver low-latency, context-specific information to MEC applications. Results from cross-border on-road tests highlighted the S-LDM's reliability in dense traffic conditions, with its scalability demonstrated by its capacity to support up to 1,000 vehicles per instance.

The concept of vehicular micro-clouds not only complements centralized MEC-based solutions like the Server Local Dynamic Map (S-LDM) but also addresses critical gaps in scenarios where network coverage is limited or unreliable. This is

---

particularly relevant in high-speed or remote environments, such as tunnels, cross-border regions, or sparsely connected highways, where reliance on centralized infrastructure may become impractical as we learned in Section 4.4. Building on this insights, Chapter 5 proposed the Platoon-Local Dynamic Map (P-LDM), a micro-cloud service tailored specifically for platooning scenarios. By aggregating and synchronizing perception data from platoon members, the P-LDM provided a unified and up-to-date view of the surrounding environment, significantly reducing dependence on infrastructure-based solutions.

At the heart of the P-LDM's functionality lies a novel optimization problem for perceived object assignment, designed to minimize computational overhead while ensuring fairness and accuracy in data sharing among platoon members. Two heuristic algorithms were developed to solve this problem in real-time, achieving assignment costs within 10% of the optimal solutions provided by the computationally intensive NSGA-II genetic algorithm. The P-LDM demonstrated scalability and efficiency, managing up to 25 platoon members and 100 perceived objects with average computation times below 50 ms per assignment. The P-LDM's implementation within the ms-van3t-CARLA simulation framework enabled highly realistic evaluations. The system achieved a reduced computational load on individual vehicles by up to 40% compared to fully distributed cooperative perception schemes while keeping comparable perception accuracy. These improvements translated into enhanced consistency and reliability, outperforming ETSI's conventional protocols, even in challenging high-speed scenarios. Additionally, the P-LDM's impact on platooning maneuvers was significant, with a 30% reduction in time-to-collision during interactions with non-platooning vehicles for a platoons of 10 members.

In conclusion, the integration of centralized and vehicular micro cloud paradigms through solutions like the S-LDM and P-LDM represents a comprehensive approach to advancing cooperative perception, addressing the critical need for scalable, reliable, and low-latency vehicular communication systems. Together, these solutions may provide a combined framework that enhances situational awareness, improves decision-making, and facilitates the safe execution of complex maneuvers such as platooning. This thesis not only advances the state of CCAM technologies but also establishes a robust foundation for the deployment of intelligent transportation systems capable of harnessing cooperative perception at scale through vehicular micro clouds. Extensive validation efforts, combining both simulation and real-world trials,

demonstrate the practical applicability and transformative potential of this paradigm, paving the way for safer, more efficient, and interconnected vehicular ecosystems.

Finally, future research directions should focus on developing adaptive, learning-based algorithms that can better manage computational loads and provide robust performance under varying operational conditions. Additionally, further analysis towards secure, privacy-aware communication protocols and the seamless integration of centralized and distributed architectures will be essential for realizing the full potential of CCAM in complex, real-world deployments.

# References

- [1] Car2car communication consortium. URL <https://www.car-2-car.org/>.
- [2] C-roads platform. URL <https://www.car-2-car.org/>.
- [3] Fcc reallocates transportation safety spectrum for wi-fi use, endorses c-v2x for auto safety. URL <https://www.engage.hoganlovells.com/knowledgeservices/news/fcc-reallocates-transportation-safety-spectrum-for-wi-fi-use-endorses-c-v2x-for-auto-safety>.
- [4] ns-3, a discrete event network simulator for internet systems. URL <https://www.nsnam.org/>.
- [5] Omnet++ - discrete event simulator. URL <https://omnetpp.org/>.
- [6] ITS Deployment Evaluation Executive briefing. URL <https://www.connectedautomateddriving.eu/blog/platooning-becomes-a-reality-in-europe/>.
- [7] Etsi en 302 665 v1.1.1 - intelligent transport systems (its); communications architecture. Standard, European Telecommunication Standard Institute, 2010.
- [8] Etsi es 202 663 v1.1.0 - intelligent transport systems (its); european profile standard for the physical and medium access control layer of intelligent transport systems operating in the 5 ghz frequency band. Standard, European Telecommunication Standard Institute, 2010.
- [9] SAFESPOT D8.1.1 Final Report. In *SAFESPOT Final Rep—Public Version*, July 2010. URL [http://www.safespot-eu.org/documents/D8.1.1\\_Final\\_Report\\_-\\_Public\\_v1.0.pdf](http://www.safespot-eu.org/documents/D8.1.1_Final_Report_-_Public_v1.0.pdf).
- [10] Etsi ts 102 637-1 v1.1.1 - intelligent transport systems (its); vehicular communications; basic set of applications; part 1: Functional requirements. Technical specification, European Telecommunication Standard Institute, 2010.
- [11] Etsi en 302 663 v1.2.0 - intelligent transport systems (its); access layer specification for intelligent transport systems operating in the 5 ghz frequency band. Standard, European Telecommunication Standard Institute, 2012.

- [12] Etsi en 302 636-2 v1.2.1 - intelligent transport systems (its); vehicular communications; geonetworking; part 2: Scenarios. Standard, European Telecommunication Standard Institute, 2013.
- [13] Etsi ts 101 539-1 v1.1.1 - intelligent transport systems (its); v2x applications; part 1: Road hazard signalling (rhs) application requirements specification. Technical specification, European Telecommunication Standard Institute, 2013.
- [14] Etsi ts 101 539-3 v1.1.1 - intelligent transport systems (its); v2x applications; part 3: Longitudinal collision risk warning (lcrw) application requirements specification. Technical specification, European Telecommunication Standard Institute, 2013.
- [15] Etsi en 302 636-1 v1.2.1 - intelligent transport systems (its); vehicular communications; geonetworking; part 1: Requirements. Standard, European Telecommunication Standard Institute, 2014.
- [16] Etsi en 302 636-3 v1.1.2 - intelligent transport systems (its); vehicular communications; geonetworking; part 3: Network architecture. Standard, European Telecommunication Standard Institute, 2014.
- [17] Ieee 1609.0-2013 - ieee guide for wireless access in vehicular environments (wave) - architecture. Standard, Institute of Electrical and Electronics Engineers, 2014.
- [18] Intelligent transport systems (its); cross layer dcc management entity for operation in the its g5a and its g5b medium. Technical specification, 2015.
- [19] 3gpp tr 36.885, v14.0.0, technical specification group radio access network; study on lte-based v2x services; release 14. Technical requirement, 3rd Generation Partnership Project, 2016.
- [20] Ieee 1609.2-2016 (revision of ieee std 1609.2-2013) - ieee standard for wireless access in vehicular environments—security services for applications and management messages. Standard, Institute of Electrical and Electronics Engineers, 2016.
- [21] Ieee 1609.3-2016 (revision of ieee std 1609.3-2010) - ieee standard for wireless access in vehicular environments (wave) – networking services. Standard, Institute of Electrical and Electronics Engineers, 2016.
- [22] Ieee 1609.4-2016 (revision of ieee std 1609.4-2010) - ieee standard for wireless access in vehicular environments (wave) – multi-channel operation. Standard, Institute of Electrical and Electronics Engineers, 2016.
- [23] Ieee 802.11-2016 - ieee standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. Standard, Institute of Electrical and Electronics Engineers, 2016.

- [24] Etsi en 302 636-5-1 - intelligent transport systems (its); vehicular communications; geonetworking; part 5: Transport protocols; sub-part 1: Basic transport protocol. Standard, European Telecommunication Standard Institute, 2017.
- [25] 3gpp tr 21.914 v14.0.0 - 3rd generation partnership project; technical specification group services and system aspects; release 14 description; summary of rel-14 work items (release 14). Technical requirement, 3rd Generation Partnership Project, 2018.
- [26] Etsi ts 101 539-2 v1.1.1 - intelligent transport systems (its); v2x applications; part 2: Intersection collision risk warning (icrw) application requirements specification. Technical specification, European Telecommunication Standard Institute, 2018.
- [27] Etsi ts 102 687 v1.2.1 - intelligent transport systems (its); decentralized congestion control mechanisms for intelligent transport systems operating in the 5 ghz range; access layer part. Technical specification, European Telecommunication Standard Institute, 2018.
- [28] 3gpp tr 21.915 v15.0.0 - 3rd generation partnership project; technical specification group services and system aspects; release 15 description; summary of rel-15 work items (release 15). Technical requirement, 3rd Generation Partnership Project, 2019.
- [29] Etsi en 302 636-4-1 v1.4.1 - 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. Standard, European Telecommunication Standard Institute, 2019.
- [30] 3gpp tr 21.916 v0.6.0 - 3rd generation partnership project; technical specification group services and system aspects; release 16 description; summary of rel-16 work items (release 16). Technical requirement, 3rd Generation Partnership Project, 2020.
- [31] Etsi en 303 613 v1.1.1 - intelligent transport systems (its); lte-v2x access layer specification for intelligent transport systems operating in the 5 ghz frequency band. Standard, European Telecommunication Standard Institute, 2020.
- [32] 5G-CARMEN D2.2, 5G-CARMEN Preliminary System Architecture and Interfaces Specifications. May 2021. URL <https://5gcarmen.eu/wp-content/uploads/2021/10/D2.2-18May2021.pdf>.
- [33] 5G-CARMEN D3.3, Intermediate report on 5G Technological Enablers for CCAM. July 2021. URL <https://5gcarmen.eu/wp-content/uploads/2021/10/D3.3-July-2021.pdf>.
- [34] Home page - 5G-CARMEN, 2022. URL <https://5gcarmen.eu/>.
- [35] The C-Roads Platform, 2022. URL <https://www.c-roads.eu/platform.html>.

- [36] Home page - L3 Pilot, 2022. URL <https://l3pilot.eu>.
- [37] Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 5: Enhancements for next generation v2x. *IEEE Std 802.11bd-2022 (Amendment to IEEE Std 802.11-2020 as amended by IEEE Std 802.11ax-2021, IEEE Std 802.11ay-2021, IEEE Std 802.11ba-2021, IEEE Std 802.11-2020/Cor 1-2022, and IEEE Std 802.11az-2022)*, pages 1–144, 2023. doi: 10.1109/IEEESTD.2023.10063942.
- [38] 3GPP. Technical Specification Group Radio Access Network; Study on LTE-based V2X Services; Release 14. Technical Report (TR) 36.885, 3rd Generation Partnership Project (3GPP), 04 2015. Version 14.0.0.
- [39] 3GPP. Study on evaluation methodology of new Vehicle-to-Everything (V2X) use cases for LTE and NR. Release 15. Technical Report (TR) 37.885, 3rd Generation Partnership Project (3GPP), 06 2019. Version 15.3.0.
- [40] 3GPP. 3rd generation partnership project; technical specification group services and system aspects; release 17 description; summary of rel-17 work items (release 17). Technical Report TR 21.917, 3rd Generation Partnership Project (3GPP), December 2022. URL <https://www.3gpp.org>.
- [41] 3GPP. 3rd generation partnership project; technical specification group services and system aspects; release 18 description; summary of rel-18 work items (release 18). Technical Report TR 21.918, 3rd Generation Partnership Project (3GPP), September 2024. URL <https://www.3gpp.org>.
- [42] A10 Networks. Deployment challenges in multi-access edge computing (mec), 2019. URL <https://www.a10networks.com/blog/deployment-challenges-in-multi-access-edge-computing-mec/>. Accessed: 2024-12-12.
- [43] Adrian Abunei, Ciprian-Romeo Comşa, and Ion Bogdan. Implementation of etsi its-g5 based inter-vehicle communication embedded system. In *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4, 2017. doi: 10.1109/ISSCS.2017.8034921.
- [44] Ahmed A. Al-Habob, Octavia A. Dobre, Ana García Armada, and Sami Muhaidat. Task scheduling for mobile edge computing using genetic algorithm and conflict graphs. *IEEE Transactions on Vehicular Technology*, 69(8):8805–8819, 2020. doi: 10.1109/TVT.2020.2995146.
- [45] Zoraze Ali, Sandra Lagén, Lorenza Giupponi, and Richard Rouil. 3gpp nr v2x mode 2: Overview, models and system-level evaluation. *IEEE Access*, 9: 89554–89579, 2021. doi: 10.1109/ACCESS.2021.3090855.

- [46] Christoph Allig and Gerd Wanielik. Extending the vehicular network simulator artery in order to generate synthetic data for collective perception. *Advances in Radio Science*, 17:189–196, 09 2019. doi: 10.5194/ars-17-189-2019.
- [47] Mani Amoozadeh, Bryan Ching, Chen-Nee Chuah, Dipak Ghosal, and H. Michael Zhang. Ventos: Vehicular network open simulator with hardware-in-the-loop support. *Procedia Computer Science*, 151:61–68, 2019. ISSN 1877-0509. doi: <https://doi.org/10.1016/j.procs.2019.04.012>. URL <https://www.sciencedirect.com/science/article/pii/S1877050919304739>. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.
- [48] Christos Anagnostopoulos, Christos Koulamas, Aris Lalos, and Chrysostomos Stylios. Open-source integrated simulation framework for cooperative autonomous vehicles. In *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–4, 2022. doi: 10.1109/MECO55406.2022.9797115.
- [49] Boris Atanassow, Katrin Sjöberg, Marcus Larsson, Edoardo Mascalchi, Joost van Doorn, Arne Ehlers, Tobias Werle, Roman Alieiev, Antoine Schmeitz, and Dehlia Willemsen. Platooning protocol definition and communication strategy. H2020-ART-2016-2017/H2020-ART-2017-Two-Stages D2.8, Volvo AB; CLEPA; NXP; ZF; MAN; TNO, 2022. URL [www.platooningensemble.eu](http://www.platooningensemble.eu). Approved by EC on 13-10-2022.
- [50] Giuseppe Avino, Paolo Bande, Pantelis A. Frangoudis, Christian Vitale, Claudio Casetti, Carla Fabiana Chiasserini, Kalkidan Gebru, Adlen Ksentini, and Giuliana Zennaro. A mec-based extended virtual sensing for automotive services. *IEEE Transactions on Network and Service Management*, 16(4): 1450–1463, 2019. doi: 10.1109/TNSM.2019.2931878.
- [51] Constantine Ayimba, Valerio Cislighi, Christian Quadri, Paolo Casari, and Vincenzo Mancuso. Copy-cav: V2x-enabled wireless towing for emergency transport. *Comput. Commun.*, 205(C):87–96, may 2023. ISSN 0140-3664. doi: 10.1016/j.comcom.2023.04.009. URL <https://doi.org/10.1016/j.comcom.2023.04.009>.
- [52] Ali Balador, Alessandro Bazzi, Unai Hernandez-Jayo, Idoia de la Iglesia, and Hossein Ahmadvand. A survey on vehicular communication for cooperative truck platooning application. *Vehicular Communications*, 35:100460, 2022. ISSN 2214-2096. doi: <https://doi.org/10.1016/j.vehcom.2022.100460>. URL <https://www.sciencedirect.com/science/article/pii/S2214209622000079>.
- [53] Leonardo Gomes Baltar, Markus Mueck, and Dario Sabella. Heterogeneous vehicular communications-multi-standard solutions to enable interoperability. In *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, 2018. doi: 10.1109/CSCN.2018.8581726.

- [54] C-Roads. C-ITS IP Based Interface Profile. Technical Specifications Version 2.0.8, C-Roads, 2023.
- [55] CAR 2 CAR Communication Consortium. White paper on Additional Investigation of ITS-G5 and Sidelink LTE-V2X Co-Channel Coexistence Methods, 2021. URL [https://www.car-2-car.org/fileadmin/documents/General\\_Documents/C2CCC\\_TR\\_ITS-G5\\_LTE-V2X\\_Coexistence\\_Methods.pdf](https://www.car-2-car.org/fileadmin/documents/General_Documents/C2CCC_TR_ITS-G5_LTE-V2X_Coexistence_Methods.pdf). Available online: <https://www.car-2-car.org/>.
- [56] CAR 2 CAR Communication Consortium. Next Generation V2X – IEEE 802.11bd as fully backward compatible evolution of IEEE 802.11p. Public White Paper 2098, CAR 2 CAR Communication Consortium, 2023. Release: n.a.
- [57] Chao Chen and Yanmin Zhu. An evaluation of vehicular networks with real traces. In *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pages 716–717, 2012. doi: 10.1109/ICPADS.2012.110.
- [58] CAR2CAR Communication Consortium. C-its: Europe’s path to connected, cooperative & automated mobility, 2022. URL [https://www.car-2-car.org/fileadmin/documents/General\\_Documents/Car\\_2\\_Car\\_Communication\\_Consortium\\_-\\_Europe\\_s\\_Path\\_to\\_Connected\\_Cooperative\\_and\\_Automated\\_Mobility\\_.pdf](https://www.car-2-car.org/fileadmin/documents/General_Documents/Car_2_Car_Communication_Consortium_-_Europe_s_Path_to_Connected_Cooperative_and_Automated_Mobility_.pdf).
- [59] David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016. doi: 10.1109/TAES.2016.140952.
- [60] CTTC. The LENA ns-3 LTE Module Documentation. Documentation, Centre Tecnològic de Telecomunicacions de Catalunya, 2014.
- [61] Joahannes B. D. da Costa, Allan M. de Souza, Rodolfo I. Meneguette, Eduardo Cerqueira, Denis Rosário, Christoph Sommer, and Leandro Villas. Mobility and deadline-aware task scheduling mechanism for vehicular edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 24(10):11345–11359, 2023. doi: 10.1109/TITS.2023.3276823.
- [62] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.
- [63] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017.
- [64] Falko Dressler, Gurjashan Singh Pannu, Florian Hagenauer, Mario Gerla, Takamasa Higuchi, and Onur Altintas. Virtual edge computing using vehicular micro clouds. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 537–541, 2019. doi: 10.1109/ICCNC.2019.8685481.

- [65] Fabian Eckermann, Moritz Kahlert, and Christian Wietfeld. Performance analysis of c-v2x mode 4 communication introducing an open-source c-v2x simulator. pages 1–5, 09 2019. doi: 10.1109/VTCFall.2019.8891534.
- [66] Mohamed Eltoweissy, Stephan Olariu, and Mohamed Younis. Towards autonomous vehicular clouds. In Jun Zheng, David Simplot-Ryl, and Victor C. M. Leung, editors, *Ad Hoc Networks*, pages 1–16, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [67] ETSI. ETSI ES 202 663 V1.1.0 (2010-01) - Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band, January 2010.
- [68] ETSI. ETSI EN 302 895 V1.1.1 (2014-09) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM). Standard ETSI EN 302 895 V1.1.1, European Telecommunications Standards Institute, 2014.
- [69] ETSI. ETSI EN 302 636-5-1 V2.2.1 (2019-05) - Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol. Standard ETSI EN 302 636-5-1, European Telecommunications Standards Institute, 2019.
- [70] ETSI. ETSI EN 302 637-2 V1.4.1 (2019-04) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Standard ETSI EN 302 637-2 V1.4.1, European Telecommunications Standards Institute, 2019.
- [71] ETSI. ETSI EN 302 637-3 V1.3.1 (2019-04) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. Standard ETSI EN 302 637-3 V1.3.1, European Telecommunications Standards Institute, 2019.
- [72] 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. Standard ETSI EN 302 636-4-1, European Telecommunications Standards Institute, 2020.
- [73] ETSI. ETSI TS 103 300-3 V2.1.1 (2020-11) - Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 3: Specification of VRU awareness basic service; Release 2. Standard ETSI TS 103 300-3 V2.1.1, European Telecommunications Standards Institute, 2020.
- [74] ETSI. Intelligent Transport Systems (ITS); Pre-standardization study on co-channel co-existence between IEEE- and 3GPP-based ITS technologies in the 5 855 MHz - 5 925 MHz frequency band. Technical Report TR 103

- 766 V1.1.1, ETSI, September 2021. URL [https://www.etsi.org/deliver/etsi\\_tr/103700\\_103799/103766/01.01.01\\_60/tr\\_103766v010101p.pdf](https://www.etsi.org/deliver/etsi_tr/103700_103799/103766/01.01.01_60/tr_103766v010101p.pdf). Published September 2021.
- [75] ETSI. ETSI TS 103 301 V2.1.1 (2021-03) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services; Release 2. Standard ETSI TS 103 301 V2.1.1, European Telecommunications Standards Institute, 2021.
- [76] ETSI. Etsi ts 122 185 v17.0.0 (2022-04) - lte - service requirements for v2x services (3gpp ts 22.185 version 17.0.0 release 17). Technical specification, European Telecommunication Standard Institute, 2022.
- [77] ETSI. ETSI TS 103 324 V2.1.1 (2023-06) - Intelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Collective Perception Service; Release 2. Standard ETSI TS 103 324 V2.1.1, European Telecommunications Standards Institute, 2023.
- [78] ETSI. Multi-access Edge Computing (MEC); Framework and Reference Architecture. Technical Report ETSI GS MEC 003 V3.2.1, ETSI Industry Specification Group (ISG) MEC, April 2024. URL [https://www.etsi.org/deliver/etsi\\_gs/MEC/001\\_099/003/03.02.01\\_60/gs\\_MEC003v032101p.pdf](https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.02.01_60/gs_MEC003v032101p.pdf).
- [79] European Telecommunications Standards Institute (ETSI). ETSI TS 102 942 V1.1.1 (2012-06) - Intelligent Transport Systems (ITS); Security; Access Control. Technical Specification TS 102 942 V1.1.1, European Telecommunications Standards Institute (ETSI), June 2012. URL [https://www.etsi.org/deliver/etsi\\_ts/102900\\_102999/102942/01.01.01\\_60/ts\\_102942v010101p.pdf](https://www.etsi.org/deliver/etsi_ts/102900_102999/102942/01.01.01_60/ts_102942v010101p.pdf).
- [80] European Telecommunications Standards Institute (ETSI). ETSI TS 102 940 V2.1.1 (2021-07) - Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management; Release 2. Technical Specification TS 102 940 V2.1.1, European Telecommunications Standards Institute (ETSI), July 2021. URL [https://www.etsi.org/deliver/etsi\\_ts/102900\\_102999/102940/02.01.01\\_60/ts\\_102940v020101p.pdf](https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/02.01.01_60/ts_102940v020101p.pdf).
- [81] European Telecommunications Standards Institute (ETSI). ETSI TS 102 941 V2.2.1 (2022-11) - Intelligent Transport Systems (ITS); Security; Trust and Privacy Management; Release 2. Technical Specification TS 102 941 V2.2.1, European Telecommunications Standards Institute (ETSI), November 2022. URL [https://www.etsi.org/deliver/etsi\\_ts/102900\\_102999/102941/02.02.01\\_60/ts\\_102941v020201p.pdf](https://www.etsi.org/deliver/etsi_ts/102900_102999/102941/02.02.01_60/ts_102941v020201p.pdf).
- [82] European Telecommunications Standards Institute (ETSI). ETSI GS MEC 003 V3.2.1 (2024-04) GROUP SPECIFICATION Multi-access Edge Computing (MEC); Framework and Reference Architecture. Group Specification GS MEC 003 V3.2.1, European Telecommunications Standards Institute (ETSI), April 2024. URL <https://www.etsi.org>.

- [83] European Telecommunications Standards Institute (ETSI). ETSI GS MEC 030 V3.3.1 (2024-06) GROUP SPECIFICATION Multi-access Edge Computing (MEC); V2X Information Services API. Group Specification GS MEC 030 V3.3.1, European Telecommunications Standards Institute (ETSI), June 2024. URL <https://www.etsi.org>.
- [84] Mario H. Castañeda Garcia, Alejandro Molina-Galan, Mate Boban, Javier Gozalvez, Baldomero Coll-Perales, Taylan Şahin, and Apostolos Kousaridas. A tutorial on 5g nr v2x communications. *IEEE Communications Surveys Tutorials*, 23(3):1972–2026, 2021. doi: 10.1109/COMST.2021.3057017.
- [85] Mikel García, Itziar Urbieto, Marcos Nieto, Javier González de Mendibil, and Oihana Otaegui. ildm: An interoperable graph-based local dynamic map. *Vehicles*, 4(1):42–59, 2022. ISSN 2624-8921. doi: 10.3390/vehicles4010003. URL <https://www.mdpi.com/2624-8921/4/1/3>.
- [86] Mario Gerla. Vehicular cloud computing. In *2012 The 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 152–155, 2012. doi: 10.1109/MedHocNet.2012.6257116.
- [87] GitHub. asn1c [online], 2021. <https://github.com/vlm/asn1c>.
- [88] GitHub. ns3 module for bidirectional coupling with sumo, 2021. <https://github.com/vodafone-chair/ns3-sumo-coupling>.
- [89] Fabio Giust, Vincenzo Sciancalepore, Dario Sabella, Miltiades C. Filippou, Simone Mangiante, Walter Featherstone, and Daniele Munaretto. Multi-access edge computing: The driver behind the wheel of 5g-connected cars. *IEEE Communications Standards Magazine*, 2(3):66–73, 2018. doi: 10.1109/MCOMSTD.2018.1800013.
- [90] Hendrik-jorn Gunther, Oliver Trauer, and Lars Wolf. The potential of collective perception in vehicular ad-hoc networks. In *ITST 2015*, pages 1–5, 2015. doi: 10.1109/ITST.2015.7377190.
- [91] Florian Hagenauer, Christoph Sommer, Takamasa Higuchi, Onur Altintas, and Falko Dressler. Vehicular micro clouds as virtual edge servers for efficient data collection. In *Proceedings of the 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services (CarSys '17)*, pages 31–35, New York, NY, USA, October 2017. ACM. doi: 10.1145/3131944.3133937. URL <https://doi.org/10.1145/3131944.3133937>.
- [92] Florian Hagenauer, Christoph Sommer, Takamasa Higuchi, Onur Altintas, and Falko Dressler. Vehicular micro cloud in action: On gateway selection and gateway handovers. *Ad Hoc Networks*, 78:73–83, 2018. ISSN 1570-8705. doi: <https://doi.org/10.1016/j.adhoc.2018.05.014>. URL <https://www.sciencedirect.com/science/article/pii/S1570870518302464>.

- [93] Florian Hagenauer, Takamasa Higuchi, Onur Altintas, and Falko Dressler. Efficient data handling in vehicular micro clouds. *Ad Hoc Networks*, 91:101871, 2019. ISSN 1570-8705. doi: <https://doi.org/10.1016/j.adhoc.2019.101871>. URL <https://www.sciencedirect.com/science/article/pii/S1570870518309107>.
- [94] Abolfazl Hajisami, Jim Lansford, Aasif Dingankar, and Jim Misener. A tutorial on the lte-v2x direct communication. *IEEE Open Journal of Vehicular Technology*, 3:388–398, 2022. doi: 10.1109/OJVT.2022.3201432.
- [95] Bassel Hakim, Ahmed A. Elbery, Mohamed Hefeida, Waleed S. Alasmay, Khaled H. Almotairi, and Aboelmagd Noureldin. Stc: Spatial and temporal clustering for cooperative perception system. *IEEE Transactions on Vehicular Technology*, pages 1–11, 2024. doi: 10.1109/TVT.2024.3376544.
- [96] Tobias Hardes, Ion Turcanu, and Christoph Sommer. Poster: A case for heterogenous co-simulation of cooperative and autonomous driving. In *2023 IEEE Vehicular Networking Conference (VNC)*, pages 151–152, 2023. doi: 10.1109/VNC57357.2023.10136319.
- [97] Syed Faraz Hasan. *Intelligent Transportation Systems*. Springer International Publishing, 2018. ISBN 3-319-64056-9.
- [98] Ghaith Hattab, Seyhan Ucar, Takamasa Higuchi, Onur Altintas, Falko Dressler, and Danijela Cabric. Optimized assignment of computational tasks in vehicular micro clouds. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking, EdgeSys '19*, page 1–6, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362757. doi: 10.1145/3301418.3313937. URL <https://doi.org/10.1145/3301418.3313937>.
- [99] Takamasa Higuchi, Joshua Joy, Falko Dressler, Mario Gerla, and Onur Altintas. On the feasibility of vehicular micro clouds. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 179–182, 2017. doi: 10.1109/VNC.2017.8275621.
- [100] J Jijin, B-C Seet, and P Chong. Multi-objective optimization of task-to-node assignment in opportunistic fog ran. *Electronics*, 9(3), 2020. ISSN 2079-9292. doi: 10.3390/electronics9030474. URL <https://www.mdpi.com/2079-9292/9/3/474>.
- [101] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, Imyhxy, Lorna, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, Tkianai, YxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, November 2022.
- [102] Mohannad Jooriah, Daryna Datsenko, João Almeida, Ana Sousa, João Silva, and Joaquim Ferreira. A co-simulation platform for v2x-based cooperative

- driving automation systems. In *2024 IEEE Vehicular Networking Conference (VNC)*, pages 227–230, 2024. doi: 10.1109/VNC61989.2024.10575990.
- [103] Charles Karney. GeographicLib 2.1 documentation [online], 2022. <https://geographiclib.sourceforge.io/>.
- [104] Sofiane Khalfallah and Bertrand Ducourthial. Bridging the gap between simulation and experimentation in vehicular networks. In *2010 IEEE 72nd Vehicular Technology Conference - Fall*, pages 1–5, 2010. doi: 10.1109/VETEFC.2010.5594402.
- [105] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955. doi: 10.1002/nav.3800020109.
- [106] Lei Liu, Chen Chen, Qingqi Pei, Sabita Maharjan, and Yan Zhang. Vehicular edge computing and networking: A survey. *Mobile Networks and Applications*, 26(3):1145–1168, 2021. ISSN 1572-8153. doi: 10.1007/s11036-020-01624-1. URL <https://doi.org/10.1007/s11036-020-01624-1>.
- [107] Qiang Liu, Tao Han, Jiang, Xie, and BaekGyu Kim. Livemap: Real-time dynamic map in automotive edge computing, 2020.
- [108] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. IEEE, 2018. URL <https://elib.dlr.de/124092/>.
- [109] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wiessner. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, 2018. doi: 10.1109/ITSC.2018.8569938.
- [110] Xiaoting Ma, Junhui Zhao, and Yi Gong. Joint scheduling and resource allocation for efficiency-oriented distributed learning over vehicle platooning networks. *IEEE Transactions on Vehicular Technology*, 70(10):10894–10908, 2021. doi: 10.1109/TVT.2021.3107465.
- [111] Yitao Ma, Qiang Liu, Jie Fu, Kangmin Liufu, and Qing Li. Collision-avoidance lane change control method for enhancing safety for connected vehicle platoon in mixed traffic environment. *Accident Analysis & Prevention*, 184:106999, 2023. ISSN 0001-4575. doi: <https://doi.org/10.1016/j.aap.2023.106999>. URL <https://www.sciencedirect.com/science/article/pii/S0001457523000465>.
- [112] Babak Mafakheri, Pierpaolo Gonnella, Alessandro Bazzi, Barbara Masini, Michele Caggiano, and R. Verdone. Optimizations for hardware-in-the-loop-based v2x validation platforms. 02 2021. doi: 10.1109/VTC2021-Spring51267.2021.9448667.

- [113] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Viridis. *IEEE Access*, 8:181176–181191, 2020. doi: 10.1109/ACCESS.2020.3028550.
- [114] Meysam Nasimi, Mohammad Asif Habibi, and Hans D. Schotten. Platoon-assisted vehicular cloud in vanet: Vision and challenges, 2020. URL <https://arxiv.org/abs/2008.10928>.
- [115] Tien Viet Nguyen, Patil Shailesh, Baghel Sudhir, Gulati Kapil, Libin Jiang, Zhibin Wu, Durga Malladi, and Junyi Li. A comparison of cellular vehicle-to-everything and dedicated short range communication. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 101–108, 2017. doi: 10.1109/VNC.2017.8275618.
- [116] nsnam. ns-3 network simulator, 2021. <https://www.nsnam.org/>.
- [117] European Conference of Postal and Telecommunications Administrations (CEPT). Ecc recommendation (08)01: Use of the band 5855-5925 mhz for intelligent transport systems (its), n.d. URL <https://docdb.cept.org/download/4232>. Accessed: [insert access date].
- [118] Simon Ollander, Florian A. Schiegg, Friedrich-Wilhelm Bode, and Marcus Baum. Dual-frequency collaborative positioning for minimization of gnss errors in urban canyons. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8, 2020. doi: 10.23919/FUSION45008.2020.9190612.
- [119] A. Olmos, Francisco Vázquez-Gallego, R. Sedar, V. Samoladas, Fermín Mira, and J. Alonso-Zarate. *An Automotive Cooperative Collision Avoidance Service Based on Mobile Edge Computing*, pages 601–607. 09 2019. ISBN 978-3-030-31830-7. doi: 10.1007/978-3-030-31831-4\_43.
- [120] Christoph Pilz, Peter Sammer, Esa Piri, Udo Grossschedl, Gerald Steinbauer-Wagner, Lukas Kuschnig, Alina Steinberger, and Markus Schratte. Collective perception: A delay evaluation with a short discussion on channel load. *IEEE Open Journal of Intelligent Transportation Systems*, 4:506–526, 2023. doi: 10.1109/OJITS.2023.3296812.
- [121] Waqas Rahman and Eui-nam Huh. Content-aware qoe optimization in mec-assisted mobile video streaming. *Multimedia Tools and Applications*, 82:1–33, 04 2023. doi: 10.1007/s11042-023-15163-w.
- [122] R Rajamani. *Vehicle Dynamics and Control*. 01 2006. ISBN 0-387-26396-9. doi: 10.1007/0-387-28823-6.
- [123] Francesco Raviglione, Marco Malinverno, and Claudio Casetti. Open source testbed for vehicular communication. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '19*, page 405–406, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367646. doi: 10.1145/3323679.3326623. URL <https://doi.org/10.1145/3323679.3326623>.

- [124] Counterpoint Research. The role of multi-access edge computing (mec) in 5g, 2024. URL <https://www.counterpointresearch.com/insights/5g-mec/>. Accessed: 2024-12-12.
- [125] Raphael Riebl, Hendrik-Jörn Günther, Christian Facchi, and Lars Wolf. Artery: Extending veins for vanet applications. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 450–456, 2015. doi: 10.1109/MTITS.2015.7223293.
- [126] Michele Rondinone, Julen Maneros, Daniel Krajzewicz, Ramon Bauza, Pasquale Cataldi, Fatma Hrizi, Javier Gozalvez, Vineet Kumar, Matthias Röckl, Lan Lin, Oscar Lazaro, Jérémie Leguay, Jérôme Härri, Sendoa Vaz, Yoann Lopez, Miguel Sepulcre, Michelle Wetterwald, Robbin Blokpoel, and Fabio Cartolano. itetris: A modular simulation platform for the large scale evaluation of cooperative its applications. *Simulation Modelling Practice and Theory*, 34:99–125, 2013. ISSN 1569-190X. doi: <https://doi.org/10.1016/j.simpat.2013.01.007>. URL <https://www.sciencedirect.com/science/article/pii/S1569190X1300018X>.
- [127] Joe Schwartz. Bing Maps Tile System, 2021. URL <https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system>.
- [128] Michele Segata, Stefan Joerer, Bastian Bloessl, Christoph Sommer, Falko Dressler, and Renato Lo Cigno. Plexe: A platooning extension for veins. In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 53–60, 2014. doi: 10.1109/VNC.2014.7013309.
- [129] Ghayoor Shah, Rodolfo Valiente Romero, Nitish Gupta, S M Osman Gani, Behrad Toghi, Yaser Fallah, and Somak Gupta. Real-time hardware-in-the-loop emulation framework for dsrc-based connected vehicle applications. 05 2019.
- [130] Nina Slamnik-Krijestorac, Girma M. Yilma, Marco Liebsch, Faqir Zarrar Yousaf, and Johann Marquez-Barja. Collaborative orchestration of multi-domain edges from a connected, cooperative and automated mobility (ccam) perspective. *IEEE Transactions on Mobile Computing*, pages 1–1, 2021. doi: 10.1109/TMC.2021.3118058.
- [131] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, 2011. doi: 10.1109/TMC.2010.133.
- [132] Leah Strand, Jens Honer, and Alois Knoll. Modeling inter-vehicle occlusion scenarios in multi-camera traffic surveillance systems. In *2023 26th International Conference on Information Fusion (FUSION)*, pages 1–8, 2023. doi: 10.23919/FUSION52260.2023.10224169.

- [133] Gokulnath Thandavarayan, Miguel Sepulcre, and Javier Gozalvez. Redundancy mitigation in cooperative perception for connected and automated vehicles. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020. doi: 10.1109/VTC2020-Spring48590.2020.9129445.
- [134] Gokulnath Thandavarayan, Miguel Sepulcre, and Javier Gozalvez. Redundancy mitigation in cooperative perception for connected and automated vehicles. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020. doi: 10.1109/VTC2020-Spring48590.2020.9129445.
- [135] Vittorio Todisco, Stefania Bartoletti, Claudia Campolo, Antonella Molinaro, Antoine O. Berthet, and Alessandro Bazzi. Performance analysis of sidelink 5g-v2x mode 2 through an open-source simulator. *IEEE Access*, 9:145648–145661, 2021. doi: 10.1109/ACCESS.2021.3121151.
- [136] Viktor Torgunakov, Vyacheslav Loginov, and Evgeny Khorov. A study of channel bonding in ieee 802.11bd networks. *IEEE Access*, 10:25514–25533, 2022. doi: 10.1109/ACCESS.2022.3155814.
- [137] Seyhan Ucar, Takamasa Higuchi, and Onur Altintas. Platoon as a mobile vehicular cloud. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2019. doi: 10.1109/GCWkshps45667.2019.9024445.
- [138] Jacco van de Sluis and Jan FCM de Jongh. V2x communication test tool for scenario-based assessment of truck platooning. In *VEHITS*, pages 135–143, 2023.
- [139] Ricard Vilalta, Ramon Casellas, Roshan Sedar, Francisco Vázquez-Gallego, Ricardo Martínez, Soumya Kanti Datta, Mathieu Lefebvre, Frederic Gardes, Jean-Marc Odinet, Jérôme Härri, Jesús Alonso-Zarate, and Raul Muñoz. Vehicular message exchange in cross-border scenarios using public cloud infrastructure. In *IEEE 35GWF 2020*, pages 251–256, 2020. doi: 10.1109/35GWF49715.2020.9221221.
- [140] Antonio Viridis, Giovanni Stea, and Giovanni Nardini. Simulte - a modular system-level simulator for lte/lte-a networks based on omnet++. In *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*, pages 59–70, 2014. doi: 10.5220/0005040000590070.
- [141] Georg Volk, Quentin Delooz, Florian A. Schiegg, Alexander Von Bernuth, Andreas Festag, and Oliver Bringmann. Towards realistic evaluation of collective perception for connected and automated driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1049–1056, 2021. doi: 10.1109/ITSC48978.2021.9564783.
- [142] Georg Volk, Quentin Delooz, Florian A. Schiegg, Alexander Von Bernuth, Andreas Festag, and Oliver Bringmann. Towards realistic evaluation of collective perception for connected and automated driving. In *2021 IEEE*

- International Intelligent Transportation Systems Conference (ITSC)*, pages 1049–1056, 2021. doi: 10.1109/ITSC48978.2021.9564783.
- [143] Axel Wegener, Michal Piorkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. Traci: An interface for coupling road traffic and network simulators. *Proceedings of the 11th Communications and Networking Simulation Symposium, CNS'08*, 04 2008. doi: 10.1145/1400713.1400740.
- [144] Wenting Wei, Ruying Yang, Huaxi Gu, Weike Zhao, Chen Chen, and Shao-hua Wan. Multi-objective optimization for resource allocation in vehicular cloud computing networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25536–25545, 2022. doi: 10.1109/TITS.2021.3091321.
- [145] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325–344, 2014. ISSN 1084-8045. doi: <https://doi.org/10.1016/j.jnca.2013.08.004>. URL <https://www.sciencedirect.com/science/article/pii/S1084804513001793>.
- [146] Zhenyu Wu, Stefano Bartoletti, Victor Martinez, Vincenzo Todisco, and Alessandro Bazzi. Analysis of co-channel coexistence mitigation methods applied to ieee 802.11p and 5g nr-v2x sidelink. *Sensors*, 23(9):4337, 2023. doi: 10.3390/s23094337. URL <https://doi.org/10.3390/s23094337>.
- [147] Tingting Xiao, Chen Chen, Qingqi Pei, and Houbing Herbert Song. Consortium blockchain-based computation offloading using mobile edge platoon cloud in internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17769–17783, 2022. doi: 10.1109/TITS.2022.3168358.
- [148] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. 2021. doi: 10.48550/ARXIV.2107.06260. URL <https://arxiv.org/abs/2107.06260>.
- [149] Runsheng Xu, Hao Xiang, Xu Han, Xin Xia, Zonglin Meng, Chia-Ju Chen, and Jiaqi Ma. The opencda open-source ecosystem for cooperative driving automation research, 2023.
- [150] Sherali Zeadally, Sherali Zeadally, Ray Hunt, Ray Hunt, Yuh-Shyan Chen, Yuh-Shyan Chen, Angela Irwin, Angela Irwin, Aamir Hassan, and Aamir Hassan. Vehicular ad hoc networks (vanets): status, results, and challenges. *Telecommunication systems*, 50(4):217–241, 2012. ISSN 1018-4864.
- [151] Zijian Zhang, Shuai Wang, Yuncong Hong, Liangkai Zhou, and Qi Hao. Distributed dynamic map fusion via federated learning for intelligent networked vehicles. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 953–959, 2021. doi: 10.1109/ICRA48506.2021.9561612.