Towards Cooperative Maneuvering Simulation: Tools and Architecture

Patrick Pirri, Claus Pahl, Nabil El Ioini, Hamid R. Barzegar
Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
{ppirri, claus.pahl, nabil.elioini, hamid.barzegar}@unibz.it

2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC) | 978-1-7281-9794-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCNC49032.2021.9369507

***Abstract*—Cooperative Maneuvering is a well-known application in the field of Cooperative, Connected, and Automated Mobility (CCAM). However, extensions are desired to offer more efficient, reliable, and safer solutions. With the introduction of the latest 5G technology, such features can be finally supported with the backup of real-time measurements of multiple vehicles on the road. Projects like the European 5G-CARMEN, address such challenges by designing and implementing edge and cloud based services that take advantage of 5G's high performance and reliability. This paper addresses the simulation of a Cooperative Maneuvering function in the context of the 5G-CARMEN project. Cooperative Maneuvering allows safer and more efficient navigation among multiple vehicles by applying optimal driving strategies. The simulation focuses on the scenario where a vehicle is on the entry lane of a highway and it needs to perform a lane change. Therefore, vehicles depending on their distance and location are advised to apply appropriate actions to make this lane change possible. Additionally, to find the best simulator for a Cooperative Maneuvering implementation a systematic survey of four simulators presents.**

***Index Terms*—Vehicular communication, Automated Mobility, Cooperative Maneuvering, V2X, V2V, CCAM, 5G-CARMEN.**

I. Introduction

The vision of intelligent and self-driving vehicles are more relevant to the automotive industry than ever before. While vehicles become always more wirelessly connected to improve the driving experience, there is also demand of platforms to support safer, smarter, and greener transportation systems. In order to realize such platforms, there is an increasing need for wireless communication among vehicles and an infrastructure that relies heavily on real-time measurements by vehicle on-board sensors. With the recent deployment of the new fifth-generation (5G) technology, a reliable wireless communication between Vehicles-to-Everything (V2X) can be guaranteed [1]. Projects such as **5G-CARMEN**[1] [2] and **5G-CroCo**[2] [3], have received funding from the European Union to address the challenges of Connected, Cooperative and Automated Mobility (CCAM). CCAM goes beyond by enabling vehicles not just to communicate, but also to interact with other vehicles, influence their behavior, and take Cooperative Maneuvering (CM) decisions so everyone can benefit from it.

CCAM has a direct impact on the reduction of traffic, the reduction of pollution, and the most significant has the potential for safer roads. However, thanks to the ultra-Reliable Low-Latency Communication (uRLLC) now available in 5G, it is more feasible and practical. CCAM platform implicitly improves road safety, traffic efficiency, autonomous driving [4] and has an explicit impact on fuel consumption, green driving without the need to extend existing road or communications infrastructure [5].

5G-CARMEN aims to develop an automotive sector along the highway corridor from Bologna, Italy to Munich, Germany via Austria, where vehicles communicate over the latest 5G technology. There are several use cases where CCAM has a high expected benefit, but 5G-CARMEN limited their possible use cases on four [6]:

- Cooperative Maneuvering: to guarantee safe and efficient navigation among multiple vehicles, intelligent strategies in situations like lane changing, overtaking, entering/exiting highways must be taken, to optimize traffic flow and minimize traffic congestion.
- Situation Awareness: humans are limited by their perception of the road situation, various dangers could remain unnoticed and compromise the safety of passengers. Situation awareness has two scenarios: i) State sharing of cooperative vehicles to alert other vehicles to sudden changes on the road and ii) Alert vehicles of the arrival of an emergency vehicle.
- Video Streaming: an expectation while being in an autonomous vehicle is the consumption of multimedia content. The internet traffic for such actions demand a constant wireless service continuity. It is essential to ensure to the costumer a high-quality service, even in inter-operator situations or crossing of borders, which could cause connection interruptions.
- Green Driving: in addition to safe and efficient driving obtained by CM and situation awareness, the 5G-CARMEN project focuses also on air quality and air pollution. A scenario would be the use of the electric drive mode by hybrid vehicles to reduce emissions in critical areas on the road.

A. Background

CCAM over a 5G technology bring together V2X [7], [8] communication in order to enable coordination driving maneuvers and global knowledge of their status, leading to safer and more efficient driving. To have a better understanding how such communication could work in 5G-CARMEN, we need

[1]https://5gcarmen.eu/
[2]https://5gcroco.eu/

to take a closer look at the CM use case. CM among multiple vehicles is needed to ensure safety and efficiency during navigation [9], and it comes in handy for different scenarios such as lane changing, overtaking, and even entering/exiting highways. To have a successful cooperation among vehicles we need to exchange information with minimal delays, which can be done over 5G networks. This wireless communication system can be used to exchange important information like speeds, positions, and intended maneuvers of different vehicles. In the 5G-CARMEN this process is referred as *state sharing*. The goal of CM is with help of state sharing to derive an optimal driving strategy to minimize traffic congestion [10], [11] and to recognize dangerous situations [12], [13]. The main application of such systems in the 5G-CARMEN project is the cooperative lane merging illustrated in Fig. 1. As it often happens on highways, there is the need to merge lanes due to multiple reasons (e.g. constructions, restoration, overtaking, entering/exiting). Cooperative lane merging focus on creating sufficient spatial gap to have enough space for a vehicle that needs to merge [14]. This process focuses also to perform it in a way to avoid congestion and to guarantee a constant traffic flow. The optimal behavior of the vehicles can be realized following either a localized or a centralized cooperative lane merging. In the *centralized* manner, an additional entity called Mobile Edge Computing (MEC) handles the merging request from the incoming vehicle 1. The MEC needs to obtain an overview of the current traffic situation. Therefore, all the vehicles involved in the cluster need to share their state to the MEC. With help of the information, the back-end server can take appropriate actions by generating gaps at ideal positions and finally approve or deny the requested merge. In the *localized* approach all the vehicles need to be aware of their surroundings. By exchanging their status information, a vehicle can communicate to nearby vehicles the intention of merging. Once the gap is created the vehicle 3 in front of the gap confirms to the merging vehicle 1 that it is safe to merge. If the request cannot be satisfied, the vehicle 3 denies the request and another solution must be found. The goal of the localized manner is to not involve the MEC by finding a solution via a Vehicle-to-Vehicle (V2V) communication [15]. Both methods rely heavily on the exploitation of a cellular connection to exchange precise and fast state information. Therefore, the reliability and service continuity [5], [16] of the communication need to be guaranteed to make such processes work correctly.

The aim of the 5G-CARMEN is to develop an automotive sector along the highway corridor from Bologna to Munich, which traverses the borders between Italy, Austria, and Germany. To guarantee the correct function of CM, we need also to address cross-border scenarios. In these scenarios vehicles that are crossing the border need to perform a network re-selection to the MEC of the adjacent country, which causes a short time of disconnectivity [17]–[20]. In the case of a cooperative lane merge illustrated in Fig. 2, the first vehicle has already crossed the border and connected to the Austrian MEC through an eNodeB access point, but the merging

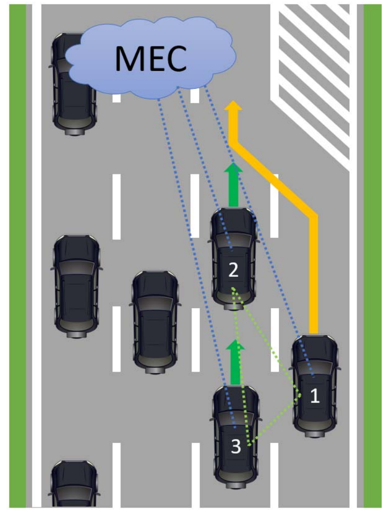

Fig. 1. Cooperative Lane Merging based on State Sharing of vehicles.

vehicle is still connected to the Italian MEC. However, the nearby vehicles need to react accordingly, regardless to which infrastructure they are connected. Therefore, the merging vehicle sends a request-to-merge to the Italian MEC, which is responsible to distribute the appropriate actions to nearby vehicles and if necessary to adjacent MECs. The adjacent MEC is then responsible to send the commands to vehicles that have already passed the border and connected to their network but still take part of the cooperative maneuver.

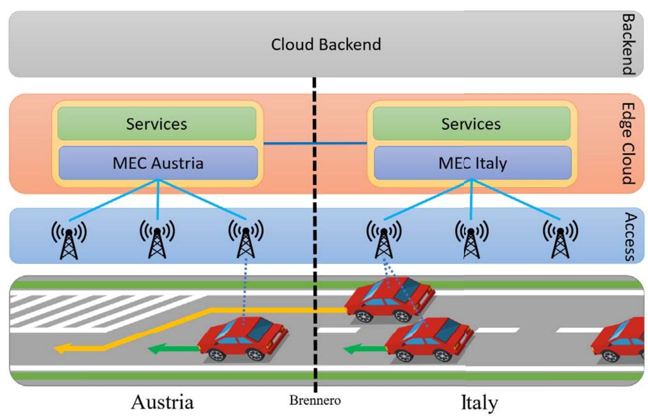

Fig. 2. 5G-CARMEN Architecture in a cross-border scenario.

The main contributions of this study are, firstly to simulate a CM function in a selected traffic simulator. Secondly, to have an overview study about possible simulators for CM that can be utilized by researches in academia.

The remainder of this paper is organized as follows. Section II is a literature review about other CM papers. Section III presents the results of the simulation comparison, section IV and V the system architecture and implementation into the simulator, and section VI concludes the paper.

II. Literature Review

In this section we would like to investigate other papers about CM solutions based on vehicular communication and proposed simulation frameworks.

The V2X communication is a key technology to support the introduction of intelligent transport systems (ITS) and the capabilities of CCAM. The V2X direct communication mode allows direct exchange of real-time information between individual devices over the short-range PC5 interface. PC5 is capable of handling high density traffic and vehicles traveling at high speeds thanks to the enhanced signal design and efficient resource allocation. V2X network communication on the other hand is for long-range communication that allows to exchange information among road users, traffic control centers, public and private cloud services, as well as transport and emergency services [9].

With help of the V2X communication traffic efficiency can be improved by up to 42% over ego-centric driving [7]. The paper mainly presents two CM concepts: the explicit and implicit coordination. In the explicit approach, the vehicle in need of cooperation transmits a cooperation request with a maneuver proposal, and other vehicles accept the proposal or send an alternative plan. However, potential cooperation vehicles need to be first identified and it becomes increasingly complex once the number of vehicles and scenarios increase. To solve this problem in an scenario-independent way an implicit approach was developed. The approach is based on the periodic transmission of possible driving paths (trajectories) with cost values via V2X communication. The trajectories cost indicates the vehicle's necessity for cooperation. The goal is to exchange possible trajectories among the affected vehicles so that the lowest trajectory cost can be found and executed.

Another CM concept was presented in [10] with the goal of reducing traffic congestion via V2V communication. The algorithm is focused on identifying vehicles that are stuck in a deadlock and resolve them by using CM. Vehicles in a deadlock condition cause traffic congestion, due to their inability to accelerate to its maximal speed nor perform a lane change. In the proposed solution the vehicle in front of the deadlock vehicle needs to perform a lane change with help of an assistant vehicle. The assistant vehicle creates a gap by slowing down depending on the current traffic situation.

A similar scenario of a highway lane reduction was analyzed in [14]. As it often happens on highways a lane reduction needs to be performed due to restoration work. In this case the entire scenario is first simplified by breaking down the two platoons into modules consisting of three vehicles. The first stage is synchronizing the speeds of both platoons. Then the first vehicle on platoon A, sends a merge request to platoon B. Platoon B starts to create gaps and vehicles are sequentially paired-up (A2B) to merge. Once the gaps are ready, every vehicle on platoon B sends a safe-to-merge message to its paired partner in platoon A.

Ros et. al. [21] surveyed the most important simulation models for wireless signal propagation, communication technologies, and vehicular mobility in the context of vehicular networks. The paper helps researchers to determine the most appropriate tools and models, in order to obtain realistic simulation results and interpretations.

In [22], the free and open-source VENTOS simulator was presented, which is capable of simulating countless realistic traffic scenarios for research purposes. The simulator combines two well-known simulators: SUMO for vehicular traffic mobility models and OMNET++ for wireless communication among the different nodes. VENTOS allows to integrate short-range communication like V2X or V2V that present to be also important in the realization of the 5G-CARMEN project.

Llatser et. al. [23] proposes a new simulation framework for Inter-Vehicle Communication using bidirectionally coupled vehicle and network simulator (Webots and Network Simulator Version 3 (NS-3) [24]), which allows recreating Cooperative Automated Driving scenarios with accurate vehicle dynamics and realistic communication. The results of the evaluation of the proposed framework shows that the computational overhead scales well in larger-scale convoys, whereas it is negligible for smaller conveys.

In [25], the authors proposed a simulation framework for the detection of misbehaving vehicles that send faulty kinematics data (e.g. position, speed, heading, etc.) to the Cooperative Intelligent Transport Systems via V2X communication. These kind of false data injection attacks could cause disruption in the network and indirectly causing safety issues on the road. The proposed framework enables to the research community to develop, test, and compare misbehaving detection algorithms.

Have access to realistic telecommunication data is important for system modeling in academia as well as industries but, because of some Mobile Network Operators (MNO) policy and privacy restrictions, it is not that easy to work with required data to train simulators. Therefore, authors in [26] proposed a data generator application that can provide the required data for different scenarios.

III. Simulator Comparison Overview

To simulate a CM function, a dedicated simulator was needed. Therefore, four different simulators were considered and compared by their requirements, advantages and disadvantages presented in Table I.

- **Carla**[3]: is an open-source simulator to support development, training, and validations of autonomous driving systems. The simulation runs on Unreal Engine and uses the OpenDRIVE standard to define roads and urban settings. To interact with the simulator, Carla offers a powerful, flexible, and constantly growing Application Programming Interface (API) that is handled in Python and C++. Carla consists of a client-server architecture. The Server is responsible of the simulation and the client is responsible of controlling the logic in the simulation by exploiting scripts communicating with the API.

[3] https://carla.org/

TABLE I
SIMULATOR COMPARISON

Simulator	**Carla 0.9.9.4**	**AirSim 1.2.8**	**DuckieTown 5.0.56**	**Metamoto**
Platform	Windows / Linux	Windows / Linux / MacOS	Windows / Linux	Browser
Availability	Free	Free	Free	Paid Subscription
Documentation	Good	Good	Poor	Good
Installation	Clone GitHub repository	Clone GitHub repository	Clone GitHub repository	None
Advantages	Open-source Ideal for traffic simulation Multiple maps	Open-source API for various languages	Open-source Ideal for traffic simulation	Multiple tests & cycles Test unique edge cases Instant & precise feedback
Disadvantages	Forces Python 2.7 / 3.7	Not ideal for traffic simulation	Limited selection of sensors	Not open-source

- **AirSim**[4]: is an open-source simulator mainly for the simulation of drones and cars. The simulator is also built on Unreal Engine, and its goal is to develop a platform for artificial intelligence research. Similar to Carla, AirSim exposes APIs to interact with the simulation and to retrieve data. The APIs are accessible for several languages, including C++, Python, C#, and Java.
- **DuckieTown**[5]: is a fast, open, and incredibly customizable simulator written in pure Python/OpenGL. According to the website DuckieTown is focused to train and test machine learning, reinforcement learning, imitation learning, or even classical robotics algorithms. DuckieTown offers a variety of different small maps with a few objects. The simulator is ideal to run multiple tests without a huge impact on the provided hardware.
- **Metamoto**[6]: is a paid cloud platform to train, test, debug, and validate autonomous systems software. The platform is split into three components i) Designer to create scenarios inside virtual scenes ii) Director to run simulations across a spectrum of scenarios and iii) Analyzer to replay simulations and assess the performance of the vehicle software. Metamoto is useful if there is the need of high performance to run multiple tests in order to create a robust and secure software. To test unique edge cases, Metamoto offers the possibility to create several environmental representations by changing multiple parameters. All three component's services are accessible via an API that communicates with the underlying system.

In comparison to the other tested simulators, Carla is the ideal option for the implementation of CM. Carla is quick and easy to setup, offers great documentation, and it provides helpful built-in functions. Carla is a comprise between a heavyweight and lightweight application, and it also provides a good performance for users with limited hardware capabilities. Additionally, Carla offers a Traffic Manager (TM) which allows to control multiple vehicle's speed, distance to leading vehicle, and lane changes.

[4]https://microsoft.github.io/AirSim/
[5]https://duckietown.org/
[6]https://metamoto.com/

Fig. 3. The Carla simulator playground (Town 04).

IV. CARLA ARCHITECTURE

The Carla simulator consists of a scalable client-server architecture (illustrated in Fig. 4) that communicate over TCP. The client connects Carla to the server, which with help of the Unreal Engine 4 and Carla Plugins runs the simulation. The simulator is in charge of computing the physics and rendering the scenes of the simulation. Thanks to the client-server architecture, it is possible to create multiple clients in the same or in different nodes. However, this could result to potential issues, since not all the components are manged by the server. For example, if multiple clients spawn multiple pedestrians, this could lead to collisions inside the simulation. Once the client is connected to the server, it can retrieve data and send commands using scripts through the Carla API. All the functionalities are available for Python and C++. Python offers an easy-to-use communication and on the other hand C++ provides a lower-level fast-performance communication. One of the core concepts of Carla is the world and client. After the client connected to the server, a simulation world needs to be loaded on which the client can spawn different actors (e.g. vehicles). From there on the client is able to constantly retrieve data and send commands with help of the world object. The client contains the TM, which is a module that lies on top of the C++ API. The goal of the TM is to recreate urban traffic to imitate real-life scenarios.

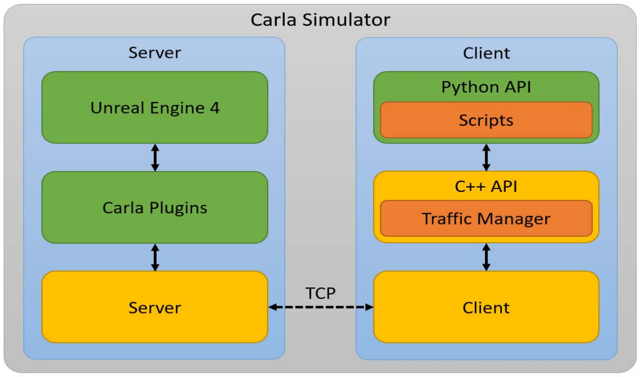

Fig. 4. Carla Simulator System Architecture Pipeline.

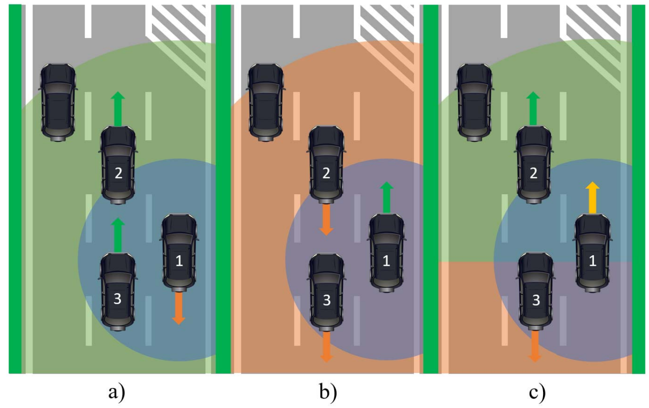

Fig. 5. Implementation of three Cooperative Maneuvering functions.

V. System Implementation

Cooperative Maneuvering (CM) is a complex undertaking which needs to adapt to various factors different vehicles could encounter. The infrastructure that controls all the vehicles in the simulation, needs to make sure once a cooperative lane merge is calculated, to guarantee a safe travel continuity for all involved vehicles. To allow such system, the infrastructure needs to be constantly aware of the surroundings of the vehicle that needs to perform a cooperative lane merge. The introduction of the 5G plays an important role, because it provides to the infrastructure a fast and reliable information exchange of multiple vehicles simultaneously. Once the involved vehicles are individuated a cooperative solution is executed. In this short-term study, three types of solutions were considered and implemented in the Carla simulator illustrated in Fig. 5. There are four general steps in the functions below: (1) send a request-to-merge to the infrastructure, (2) identify nearby vehicles located on the merging lane, (3) resolve the request by accelerating/decelerating affected vehicles, (4) execute the lane change once the nearby lane is free.

a) *Acceleration*: vehicles that are detected by vehicle 1 located within the green area and on the merging lane need to accelerate. Meanwhile, vehicle 1 that sent a merging request to the infrastructure needs to decelerate. Once vehicle 1 doesn't detect any vehicles on the merging lane within the blue area (safe-to-merge area), it can execute a safe lane change behind vehicle 2 and 3.

b) *Deceleration*: actions are completely reversed in comparison to scenario a). Vehicles that are within the detection radius and on the merging lane need to decelerate, while the merging vehicle 1 accelerates. This leads to vehicle 1 executing a lane change in front of vehicle 2 and 3.

c) *Acceleration & Deceleration*: is a combination of function a) and b) where the detection area is split in half. Vehicles on the upper half, need to accelerate. Vehicles on the lower half, need to decelerate. During this process vehicle 1 needs to keep a speed less than vehicle 2, but greater than vehicle 3. Once both vehicles 2 and 3 adapted to the appropriate changes, vehicle 1 is able to execute a lane change in the created gap between vehicle 2 and 3.

TABLE II
Carla Simulation Settings

Parameters	Value
Automatic Lane Change	Off
Default Speed	30
Acceleration Speed	60
Deceleration Speed	0
Radius of Detection Area	30 m
Radius of Safe-to-Merge Area	7 m

For the implementation of the three scenarios illustrated in Fig. 5 a Python script was developed. *Town04* of Carla was used to represent a highway realistic environment with multiple lanes (see Fig. 3). In order to imitate real-life traffic situations, all spawned vehicles were subscribed to the TM, which allowed us to change the speed of vehicles and to execute lane changes in appropriate moments. For the most part the default settings of the TM were adopted to also guarantee a certain level of safeness. To determine vehicles within a certain distance, Carla offers build-in functions to retrieve the locations of vehicles. Furthermore, with the vehicle's location the current road and lane could be determined of nearby vehicles in order to send commands to vehicles on the merging lane. Table II presents other simulation parameters.

Fig. 6. Town 04 of the Carla Simulator.

VI. Conclusion and Future Work

In this paper we highlighted an important function of the 5G-CARMEN related to the improvement of efficiency and safety of vehicles. To find the ideal simulator for the implementation of a CM, we compared four simulators by their requirements, advantages, and disadvantages. Although the development of CM represents a critical first step in bringing cooperation to a simulation, numerous other important factors need to be identified and addressed in order to produce reliable solutions. Future work in the improvement of a CM function can be done with the consideration that not all the vehicles are connected to the infrastructure. Therefore, important information needs to be exchanged among vehicles that are connected so that non-connected vehicles can be detected and respected. Considering this paper focused on a simple implementation of a CM function using the basic utilities of the Carla simulator, more sophisticated tools of the Carla library (e.g. on-board sensors) could be used, which would drastically improve the safeness and performance of a CM function.

Acknowledgement

This work has been performed in the framework of the EU Horizon 2020 project 5G-CARMEN co-funded by the EU under grant agreement No. 825012. The views expressed are those of the authors and do not necessarily represent the project. The Commission is not liable for any use that may be made of any of the information contained therein.

References

[1] S. A. A. Shah, E. Ahmed, M. Imran and S. Zeadally, "5G for Vehicular Communications," in IEEE Communications Magazine, vol. 56, no. 1, pp. 111-117, Jan. 2018, doi: 10.1109/MCOM.2018.1700467.

[2] Barzegar, Hamid R and El Ioini, Nabil and Le, Van Thanh and Pahl, Claus, "5G-CARMEN: Service Continuity in 5G-Enabled Edge Clouds," in 8th European Conference On Service-Oriented And Cloud Computing, 2020.

[3] D. Hetzer, M. Muehleisen, A. Kousaridas and J. Alonso-Zarate, "5G Connected and Automated Driving: Use Cases and Technologies in Cross-border Environments," 2019 European Conference on Networks and Communications (EuCNC), Valencia, Spain, 2019, pp. 78-82, doi: 10.1109/EuCNC.2019.8801993.

[4] T. Haramaki and H. Nishino, "A Safe Driving Support System Based on Distributed Cooperative Edge Computing," 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taichung, 2018, pp. 1-2, doi: 10.1109/ICCE-China.2018.8448585.

[5] Barzegar, Hamid R. & Le, Van & El Ioini, Nabil & Pahl, Claus. (2020). Service Continuity for CCAM Platform in 5G-CARMEN. 1764-1769. 10.1109/IWCMC48107.2020.9148380.

[6] Danilo Amendola et al., "5G CARMEN Use Cases and Requirements", 5G for Connected and Automated Road Mobility in the European UnioN, Europe, 2019. Web. 4 Aug. 2020. https://5gcarmen.eu/wp-content/uploads/2020/03/5G_CARMEN_D2.1_FINAL.pdf.

[7] I. Llatser, T. Michalke, M. Dolgov, F. Wildschütte and H. Fuchs, "Cooperative Automated Driving Use Cases for 5G V2X Communication," 2019 IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 2019, pp. 120-125, doi: 10.1109/5GWF.2019.8911628.

[8] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer and A. Kovacs, "Enhancements of V2X communication in support of cooperative autonomous driving," in IEEE Communications Magazine, vol. 53, no. 12, pp. 64-70, Dec. 2015, doi: 10.1109/MCOM.2015.7355568.

[9] R. Weber, J. Misener and V. Park, "C-V2X - A Communication Technology for Cooperative, Connected and Automated Mobility," Mobile Communication - Technologies and Applications; 24. ITG-Symposium, Osnabrueck, Germany, 2019, pp. 1-6.

[10] N. Wang, X. Wang, P. Palacharla and T. Ikeuchi, "Cooperative autonomous driving for traffic congestion avoidance through vehicle-to-vehicle communications," 2017 IEEE Vehicular Networking Conference (VNC), Torino, 2017, pp. 327-330, doi: 10.1109/VNC.2017.8275620.

[11] M. A. Togou et al., "A Distributed Blockchain-based Broker for Efficient Resource Provisioning in 5G Networks," 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 2020, pp. 1485-1490, doi: 10.1109/IWCMC48107.2020.9148565.

[12] T. Batz, K. Watson and J. Beyerer, "Recognition of dangerous situations within a cooperative group of vehicles," 2009 IEEE Intelligent Vehicles Symposium, Xi'an, 2009, pp. 907-912, doi: 10.1109/IVS.2009.5164400.

[13] M. A. Togou et al., "DBNS: A Distributed Blockchain-Enabled Network Slicing Framework for 5G Networks," in IEEE Communications Magazine, vol. 58, no. 11, pp. 90-96, November 2020, doi: 10.1109/MCOM.001.2000112.

[14] J. Ploeg et al., "Cooperative Automated Maneuvering at the 2016 Grand Cooperative Driving Challenge," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 4, pp. 1213-1226, April 2018, doi: 10.1109/TITS.2017.2765669.

[15] F. Chou, S. E. Shladover and G. Bansal, "Coordinated merge control based on V2V communication," 2016 IEEE Vehicular Networking Conference (VNC), Columbus, OH, 2016, pp. 1-8, doi: 10.1109/VNC.2016.7835933.

[16] H. R. Barzegar, N. E. Ioini, V. T. Le and C. Pahl, "Wireless Network Evolution Towards Service Continuity in 5G enabled Mobile Edge Computing," 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), Paris, France, 2020, pp. 78-85, doi: 10.1109/FMEC49853.2020.9144852.

[17] N. Slamnik-Kriještorac, H. C. Carvalho de Resende, C. Donato, S. Latré, R. Riggio and J. Marquez-Barja, "Leveraging Mobile Edge Computing to Improve Vehicular Communications," 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2020, pp. 1-4, doi: 10.1109/CCNC46108.2020.9045698.

[18] Gand, F., Fronza, I., El Ioini, N., Barzegar, H. R. and Pahl,C. (2020). A Lightweight Virtualisation Platform for Cooperative, Connected and Automated Mobility. In 6th Intl Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS).

[19] Pahl, C., Fronza, I., El Ioini, N. and Barzegar, H. R. (2019). A Review of Architectural Principles and Patterns for Distributed Mobile Information Systems. In 14th Intl Conf on Web Information Systems and Technologies.

[20] Lu, Meng & Blokpoel, Robbin & Schindler, Julian & Maerivoet, Sven & Mintsis, Evangelos. (2018). ICT Infrastructure for Cooperative, Connected and Automated Transport in Transition Areas. 10.5281/zenodo.1456600.

[21] Francisco J. Ros, Juan A. Martinez, Pedro M. Ruiz, "A survey on modeling and simulation of vehicular networks: Communications, mobility, and tools," Computer Communications, Volume 43, 2014, Pages 1-15, ISSN 0140-3664, doi: https://doi.org/10.1016/j.comcom.2014.01.010.

[22] Amoozadeh, Mani & Ching, Bryan & Chuah, Chen-Nee & Ghosal, Dipak. (2019). VENTOS: Vehicular Network Open Simulator with Hardware-in-the-Loop Support. Procedia Computer Science. 151. 61-68. 10.1016/j.procs.2019.04.012.

[23] I. Llatser, G. Jornod, A. Festag, D. Mansolino, I. Navarro and A. Martinoli, "Simulation of cooperative automated driving by bidirectional coupling of vehicle and network simulators," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 1881-1886, doi: 10.1109/IVS.2017.7995979.

[24] Van Thanh Le and Nabil El Ioini and Hamid R. Barzegar and Claus Pahl, "A Multi-domain Network Simulator based on NS-3," in 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, 2020.

[25] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa and P. Urien, "Simulation Framework for Misbehavior Detection in Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 6631-6643, June 2020, doi: 10.1109/TVT.2020.2984878.

[26] Pomalo, Marco and Le, Van Thanh and El Ioini, Nabil and Pahl, Claus and Barzegar, Hamid R., "A Data Generator for Cloud-Edge Vehicle Communication in Multi Domain Cellular Networks," in 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS) December 2020.