POLITECNICO DI TORINO
Repository ISTITUZIONALE

MS-VAN3T-CARLA: An Open-Source Co-Simulation Framework for Cooperative Perception Evaluation

(Article begins on next page)

20 March 2024

ms-van3t-CARLA: an open-source co-simulation framework for cooperative perception evaluation

Carlos Mateo Risma Carletti[*†], Claudio Casetti[*], Jérôme Härri[†], Fulvio Risso[*]
[*]Politecnico di Torino - Department of Control and Computer Engineering, Turin, Italy
[†]EURECOM - Communication Systems Department, Sophia-Antipolis, France
E-mail: {carlos.rismacarletti, claudio.casetti, fulvio.risso}@polito.it {risma, jerome.haerri}@eurecom.fr

Abstract—**The CARLA simulator stands out amongst available open-source simulators providing life-like rendering of vehicular scenarios. While several frameworks, such as OpenCDA [1], have extended CARLA capabilities for the evaluation of cooperative perception applications, a significant gap remains in the use of accurate communication models. To address this need, we present an extension of the ms-van3t simulation framework, based on the ns-3 simulator. It integrates CARLA's high-fidelity sensor and vehicle physics, OpenCDA's data fusion and control automation, with ns-3's advanced communication models.**

ms-van3t-CARLA offers a comprehensive tool for researchers working on the evaluation of the effects of vehicular cooperative perception under different communication technologies.

Index Terms—**V2X, Autonomous Vehicles, Cooperative Perception, Local Dynamic Map, ITS, Vehicular Networks, 5G**

I. INTRODUCTION

The capability of data sharing among vehicles is a critical component for bringing automated driving to fruition in the forthcoming years. Communication technologies encompassed within Vehicle-to-Everything (V2X) systems enable this exchange, under telecommunications standards like IEEE 802.11p/bd, 3GPP C-V2X, and NR-V2X introduced in 3GPP Release 16. With the advent of Connected Autonomous Vehicles (CAVs), the vast amount of data gathered by their onboard sensors becomes crucial for the awareness of the context around them. However, awareness limitations appear due to the presence of obstacles and adverse environmental conditions. With the aim of solving this limitation, the concept of cooperative perception allows vehicles to share gathered data between them extending their context awareness.

In addition to Cooperative Awareness Messages (CAMs), the European Telecommunication Standards Institute (ETSI) has made efforts for the standardization of Collective Perception Messages (CPMs) for vehicles to share their sensed data. In order to evaluate the benefits provided by the exchange of such messages, researchers rely on simulation tools, to avoid the excessive cost of trials with a significant number of real vehicles. Thus, it is crucial for simulation frameworks to jointly, accurately model the mobility, generated sensor data, and communication over the wireless channel. Well-established simulation tools, such as SUMO [2], provide great mobility simulation but lack sensor modeling capabilities, allowing only synthetic data to be generated. On the other hand, the CARLA simulator provides mobility simulation with life-like scenario rendering together with sensor models (e.g. LiDAR, radar and cameras) able to produce realistic outputs that can be used for computer vision tasks at simulation run-time. In order to harness CARLA's capabilities, many frameworks provide modular extensions to ease the creation and evaluation of scenarios as is the case with the OpenCDA framework [1]. OpenCDA extends CARLA offering perception, localization, and vehicle control modules abstracting the different elements within the simulation and allowing for a quick development of new modules.

The ms-van3t framework [3], instead, couples ns-3 and SUMO for the simulation of vehicular networks offering multiple communication technology models and a comprehensive implementation of the ETSI C-ITS stack. ms-van3t already provides an implementation of the Collective Perception Service together with a Local Dynamic Map (LDM) [4] facility, relying on the information provided by SUMO for the modeling of sensor-generated data.

In this work, we have extended the capabilities of the ms-van3t framework[1] offering the possibility of replacing SUMO with CARLA for the simulation of vehicle mobility and sensor perception. To achieve this, we have first leveraged the OpenCDA framework to develop an LDM module and extend the adapter devised in [5] to be able to extract not only localization information from CARLA but also perception information from the LDM module. Furthermore, we developed a client module on ns-3 to query the information in order to use it for the mobility of each of the ns-3 simulated nodes and to update the LDM module with all perception data sent over the simulated vehicular network.

II. RELATED WORKS

One of the most popular open-source simulation frameworks for vehicular networks is Veins [6]. Veins integrates OMNeT++ and SUMO for modeling the network and mobility simulation, respectively. Initially offering the possibility of simulating nodes equipped with IEEE 802.11p, over the years the framework has grown to include new extensions for LTE and 5G models, as well as support for an ETSI C-ITS stack implementation with the Artery framework [7].

The introduction of the CARLA simulator has marked a significant milestone in the field of vehicular simulation. This tool has unlocked a multitude of new possibilities for modeling

---

[1]https://github.com/ms-van3t-devs/ms-van3t

and evaluating Cooperative Autonomous Vehicles (CAVs). Furthermore, with the aim of providing higher modularity in the creation of CARLA scenarios, frameworks such as OpenCDA have emerged, offering a comprehensive set of features including computer vision models and Cooperative Driving Automation (CDA) algorithms. While several CARLA-based frameworks provide capabilities for CDA testing, e.g., OpenCDA, most of them neglect the use of wireless communication models, usually working under unrealistic assumptions.

In order to overcome this limitation of CARLA, some works have integrated Veins together with CARLA. In [8], the authors combine CARLA and OMNeT++ using a ZeroMQ API, focusing on CARLA's vehicle physics model for control decisions based on CAMs received over the 5G network in OMNeT++, without utilizing sensor-generated data. The integration of Artery, SUMO, and CARLA is discussed in [9], where SUMO controls vehicle behavior. The CARLA-ROS bridge is used for incorporating CAMs information into CARLA's simulated perception system for enhanced pose estimation. Although this framework provides functionalities to produce sensor information, it does not provide support for inserting such information in CPMs to be used in cooperative perception use cases. In [5], a CARLA and Veins bridge is presented, using a gRPC adapter for co-simulation, demonstrating the adapter's flexibility for use with other simulators.

This work extends the concept, integrating OpenCDA and ms-van3t (based on ns-3) to offer an interface that combines mobility and sensor data generation with precise communication models. To the best of the authors' knowledge, this is the first framework offering CARLA and ns-3 co-simulation not only using CARLA for the simulation of vehicle mobility but also providing support to retrieve sensor data from CARLA and using such information to encode and send CPMs, in addition to CAMs, over a simulated network on ns-3.

III. SIMULATION FRAMEWORK

ms-van3t extends the ns-3 network simulator offering an implementation of the C-ITS stack together with support for most state-of-the-art ns-3 communication models such as IEEE 802.11p, 3GPP LTE, LTE-V2X, and NR-V2X. Additionally, it incorporates realistic vehicle mobility simulation achieved by integrating SUMO. This integration involves the creation of a network node within ns-3 for each vehicle within SUMO. Each ns-3 node is equipped with a wireless network interface of choice, on which one or multiple applications can be installed on top of the ETSI C-ITS stack.

Following the same philosophy, ms-van3t-CARLA synchronizes the movements of vehicles within CARLA with the mobility state of ns-3 nodes. To achieve such a synchronization Google Protobuf and gRPC Remote Procedure Call are leveraged to bridge ms-van3t with an adapter developed as a module for the OpenCDA framework to which a client can connect and query information from ns-3. We provide below a detailed description of the modules developed to integrate the two frameworks, as shown in Figure 1.
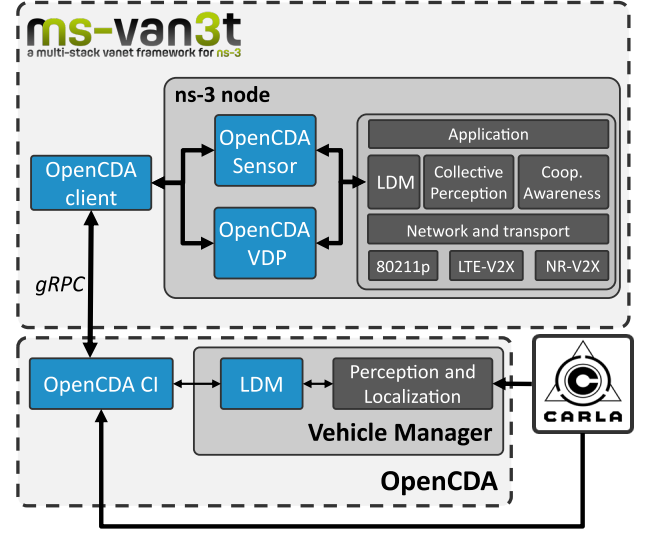

Fig. 1. Architecture of ms-van3t-CARLA illustrating the interaction between all the implemented modules (blue) with the existing ones (grey).

A. OpenCDA Modules

1) LDM

The perception module in OpenCDA has been leveraged to implement an LDM facility within the framework in order to keep track of the objects detected on each simulation step. The perception module provided by OpenCDA offers the possibility of extracting pre-processed data from the post-fusion output of a given number of cameras and a LiDAR sensor deployed on the simulated vehicle.

As envisioned by ETSI, the LDM module we have implemented in OpenCDA has the purpose of matching both locally perceived objects on every sensor period and perceived objects received from other connected vehicles through CPMs, storing tracks in the facility's database. The matching task is treated as an assignment problem, optimally pairing current detections with existing tracks or creating new ones, using a linear assignment algorithm considering Intersection Over Union and distance. The speed and acceleration of objects are estimated using a Kalman filter with a constant acceleration model, enhancing track accuracy and reducing false negatives. The LDM database includes metadata like path history, detection timestamps, and flags indicating whether an entry is a connected vehicle or detected object and whether it is in perception range and tracked long enough for CPM inclusion.

The LDM handles non-local perceptions in two ways: for new connected vehicles (from CAMs), matching occurs once, using for future updates the unique CAM ID. For new detected objects (from CPMs), each perception undergoes matching as a new local detection.

2) OpenCDA CI

The OpenCDA Control Interface (OpenCDA CI) module creates a gRPC server, which is based on the concept presented in [5]. This server enables communication with OpenCDA

using a set of Protobuf-defined messages. In line with OpenCDA's simulation configuration, which includes a set of Connected Autonomous Vehicles (CAVs) with perception capabilities and a set of background vehicles, the OpenCDA CI is designed to provide LDM information specifically for CAVs. In contrast, only mobility information is available for background vehicles.

The OpenCDA CI is in charge of encoding the information extracted from CAVs' LDM into the defined Protobuf messages format, taking as input the actor ID of the vehicle that is updating its perception data. Furthermore, in addition to executing the *ticks* of the CARLA simulation, commanded from ms-van3t, this module manages the updates of the perception module and LDM simulated within OpenCDA. It is worth mentioning that in order to maintain the synchronization of both simulations, CARLA is configured to run in *synchronous mode* so that every step represents a fixed time duration equal to the time duration between the mobility steps within the ms-van3t simulation.

Lastly, the OpenCDA CI provides the possibility of applying the control input of the simulated CAVs by an application or C-ITS facility defined in ms-van3t. To this end, desired acceleration or speed inputs are translated into CARLA-compliant control inputs, i.e., throttle, and brake. For background vehicles instead, the control is handled by CARLA's traffic manager.

B. ms-van3t modules

1) OpenCDA Client

The OpenCDA Client module provides the implementation of all the gRPC interfaces with the OpenCDA CI. This module takes charge of launching the OpenCDA simulation, establishing the connection with the OpenCDA CI, and commanding each mobility simulation step. With the aim of providing further flexibility on simulation setup, the OpenCDA client allows for a variable rate of background vehicles to be equipped with wireless communication devices. This feature allows to evaluate the tradeoff on channel load between the number of vehicles broadcasting CAMs and the number of vehicle detections to be introduced in CPMs.

Due to the impossibility of creating new network nodes at simulation run-time within ns-3, a pool of idle nodes is created at the beginning of the simulation from which the OpenCDA client assigns a given CARLA actor to a given ns-3 node and installs the user-defined application and C-ITS stack. Thus, every connected vehicle within the ns-3 simulation is linked with a CARLA actor inheriting the actor's ID allowing the application, and all layers of the C-ITS stack, on each ns-3 node to request mobility information to the OpenCDA client. Lastly, for every simulated message, the OpenCDA client provides an interface to the measurements module of ms-van3t on which, according to a configured baseline distance and mobility state, Packet Reception Ratio and one-way latency are computed.

2) OpenCDA VDP

ms-van3t offers a VDP (Vehicle Data Provider) module allowing any application and all the layers of the C-ITS stack to access the necessary mobility information in the same way independently of how the mobility is simulated. Keeping the same structure, the CARLA VDP implements all the necessary methods to retrieve specific information from the CARLA simulation through the OpenCDA client. This module allows for facilities such as the Collective Perception Service to retrieve the mandatory data to be inserted on each message, e.g., the location and speed of the sender vehicle and the characteristics of the sensors used for the detection of objects. The OpenCDA VDP is created at application start-up with the ID assigned by the OpenCDA client that is unique to that node for the entire simulation. The VDP has access to the OpenCDA client and queries the actor-specific information using that ID.

3) OpenCDA Sensor

The OpenCDA Sensor module is in charge of retrieving perceived information from the OpenCDA LDM of a given CAV and providing the information of received V2X messages back to it. ms-van3t's LDM implementation offers the possibility of storing the perception of objects and connected vehicles in the same format as the OpenCDA counterpart. Similarly to the VDP, this LDM implementation is designed to be mobility-model agnostic, i.e., it can be integrated with the two supported simulation environments, OpenCDA and SUMO. Thus, no modifications are required within the application or C-ITS facilities that interact with the data stored in it.

The OpenCDA Sensor module works by periodically synchronizing the LDM facilities implemented in both frameworks. This synchronization allows OpenCDA's LDM to handle matching and data fusion algorithms, while its ns-3 counterpart ensures rapid data retrieval during simulations. Thus, simulations rely on fewer API calls compared to only using OpenCDA's LDM making simulation more efficient. This synchronization is performed in 3 steps. First, all new perceptions received within the synchronization period through V2X messages (i.e., CAMs and CPMs) are extracted from the ns-3 LDM and sent to the OpenCDA LDM. Second, the OpenCDA LDM computes the matching of all the newly received perceptions following the mechanism described in Section III-A1. In the third step, the OpenCDA sensor retrieves the updated copy of the OpenCDA LDM state and synchronizes it with the ns-3 LDM. Since some existing perceptions in the first step might have been matched with existing perceptions in the second step, the OpenCDA Sensor deletes all the leftover perceptions stored in the ns-3 LDM.

IV. SIMULATION SETUP

For scenario creation, the configuration must be defined in two separate files, one for the mobility setup and another for the communication setup. The mobility setup is done through a YAML file, following OpenCDA's approach, where users specify spawn points and destinations for background vehicles and CAVs, along with simulated perception system settings
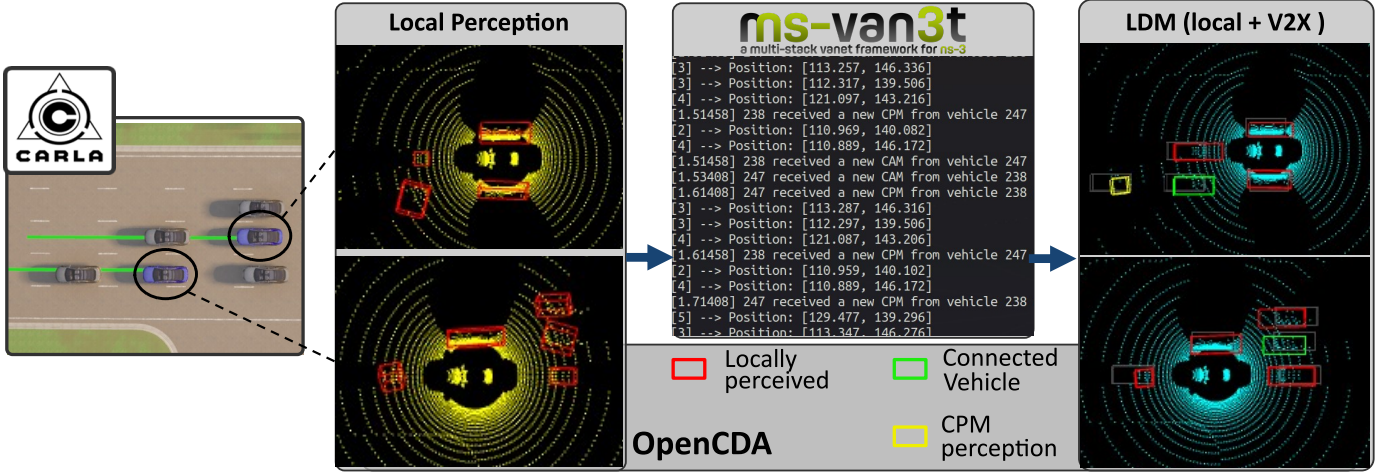
Fig. 2. LDM representation on top of OpenCDA's LiDAR view, illustrating the different types of perceptions according to CAM, CPM or local detection.

for each CAV, including RGB cameras and LiDAR sensors. Regarding the communication setup, the configuration of the simulation is done on an ns-3 simulation script written in C++.

To aid the user's understanding of each CAV awareness, OpenCDA's LDM module provides a Birds Eye View (BEV) of the database, visually distinguishing different perception inputs (local detection, CAM, CPM) together with CARLA's ground truth. Figure 2 showcases a basic simulation scenario. In this scenario, two CAVs are traveling along a road together with several non-connected vehicles. In this case, each CAV processes its sensor data to provide a set of detected objects. As can be seen from the local perception images, each CAV produces, for each detected vehicle, different perceptions due to the different Points Of View (POV). Through the OpenCDA sensor, the ns-3 LDM is updated with the perception data to be sent over the simulated wireless channel encoded in ETSI standard-compliant CPMs. Additionally, each CAV encodes the information provided by the OpenCDA VDP in CAMs. Once these messages are received, the information is stored on the ns-3 LDM to be later synchronized with the OpenCDA LDM. Afterwards, as can be seen on the right of Figure 2, the OpenCDA LDM performs the matching and fusion of data received from the V2X messages achieving an enhanced context awareness by combining the different POVs. Although this is a simple scenario with only two connected cars, when a higher number of simulated vehicles are involved, the effects of CAMs and CPMs on KPIs such as Packet Reception Rate can be evaluated thanks to ns-3 communication models.

V. CONCLUSIONS AND OUTLOOK

This paper introduces a simulation framework integrating ms-van3t with OpenCDA, combining CARLA's high-fidelity vehicle dynamics and sensor data with ns-3's robust communication models. This integration facilitates realistic analysis of cooperative perception in vehicular networks under different V2X technologies. Significantly, our work lays the foundation for the evaluation of AI/ML-enabled autonomous driving applications within various edge computing architectures, leveraging realistic sensor data. This framework provides a key tool for developing 5G-based Cooperative, Connected, and Automated Mobility (CCAM), offering new insights into the interplay between vehicular and communication technologies.

REFERENCES

[1] R. Xu, H. Xiang, X. Han, X. Xia, Z. Meng, C.-J. Chen, and J. Ma, "The opencda open-source ecosystem for cooperative driving automation research," 2023.

[2] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo." IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[3] M. Malinverno, F. Raviglione, C. Casetti, C.-F. Chiasserini, J. Mangues-Bafalluy, and M. Requena-Esteso, "A multi-stack simulation framework for vehicular applications testing," 2020. [Online]. Available: https://doi.org/10.1145/3416014.3424603

[4] ETSI, "ETSI EN 302 895 V1.1.1 (2014-09) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)," European Telecommunications Standards Institute, Standard ETSI EN 302 895 V1.1.1, 2014.

[5] T. Hardes, I. Turcanu, and C. Sommer, "Poster: A case for heterogenous co-simulation of cooperative and autonomous driving," in 2023 IEEE Vehicular Networking Conference (VNC), 2023, pp. 151–152.

[6] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," IEEE Transactions on Mobile Computing, vol. 10, no. 1, pp. 3–15, 2011.

[7] R. Riebl, H.-J. Günther, C. Facchi, and L. Wolf, "Artery: Extending veins for vanet applications," in 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015, pp. 450–456.

[8] C. Ayimba, V. Cislaghi, C. Quadri, P. Casari, and V. Mancuso, "Copy-cav: V2x-enabled wireless towing for emergency transport," Comput. Commun., vol. 205, no. C, p. 87–96, may 2023. [Online]. Available: https://doi.org/10.1016/j.comcom.2023.04.009

[9] C. Anagnostopoulos, C. Koulamas, A. Lalos, and C. Stylios, "Open-source integrated simulation framework for cooperative autonomous vehicles," in 2022 11th Mediterranean Conference on Embedded Computing (MECO), 2022, pp. 1–4.