ColosSUMO: Evaluating Cooperative Driving Applications with Colosseum

Gabriele Gemmi*, Pedram Johari*, Paolo Casari†, Michele Polese*, Tommaso Melodia*, Michele Segata†

*Institute for the Wireless Internet of Things — Northeastern University, Boston, MA
†Department of Information Engineering and Computer Science — University of Trento, Italy

* n.surname@northeastern.edu † name.surname@unitn.it

***Abstract*—The quest for safer and more efficient transportation through cooperative, connected and automated mobility (CCAM) calls for realistic performance analysis tools, especially with respect to wireless communications. While the simulation of existing and emerging communication technologies is an option, the most realistic results can be obtained by employing real hardware, as done for example in field operational tests (FOTs). For CCAM, however, performing FOTs requires vehicles, which are generally expensive, may require significant manpower, and lead to unacceptable safety issues. Mobility simulation with hardware-in-the-loop (HIL) serves as a middle ground, but current solutions lack flexibility and reconfigurability. This work thus proposes ColosSUMO as a way to couple Colosseum, the world's largest wireless network emulator, with the SUMO mobility simulator, showing its design concept, how it can be exploited to simulate realistic CCAM scenarios, and its flexibility in terms of communication technologies.**

***Index Terms*—Connected and automated mobility; hardware-in-the-loop; simulation**

I. Introduction and Related Work

The research in cooperative, connected and automated mobility (CCAM) has seen a constant growth in performance evaluation tools and frameworks in the last years [1]–[11]. This is to be expected, as CCAM applications require thorough testing prior to market introduction, but such tests cannot be performed using real vehicles and hardware until the very last system design phases due to practical, economic, and safety reasons While simulation-based platforms provide the highest degree of flexibility and reproducibility, the quality of the outcome depends on the input scenarios and employed models.

A methodology serving as a middle ground between field operational tests (FOTs) and simulations is emulation with hardware-in-the-loop (HIL), where some components of the evaluation system are run as software or through simulations, and others are physical components (e.g., radios) interfacing to the software sub-systems. HIL emulation has been successfully applied across various domains, notably in the testing of vehicle engine components [12].

This hybrid technique is also common in research on communication technologies, notably in CCAM. For instance, the VENTOS simulator [1], while primarily focused on simulation studies, also provides HIL capabilities for devices compliant with the IEEE802.11p standard. VENTOS's main limitations are the choice of devices, and that lack of communication channel modeling, which limits the reproducibility of the experiments. Another example is the work in [3], where the authors make it possible to test individual communication devices. A more recent approach [2] enables a single device under test to generate signals for multiple vehicles. However, achieving accurate channel simulations with this approach requires specific hardware components. Additionally, this tool is not readily available to the research community.

The limitations of existing HIL platforms are manifold. Firstly, they lack flexibility in terms of communication technologies. CCAM applications will not rely on a single technology, but rather on a collection that may include one or more of 5G/6G cellular V2X (C-V2X) and IEEE 802.11p [5]. For a HIL platform it should be fundamental to enable application effectiveness tests in the presence of different communication technologies, potentially considering their co-existence or mutual interference issues. Moreover, hardware components limit scalability, meaning that it becomes unfeasible to consider more than a handful of devices in such platforms. Thus, considering realistic CCAM scenarios with several vehicles might simply be impossible. In addition, this limits accessibility: even if a framework supports several HIL devices, research groups would still be required to cover the procurement and installation costs of their desired hardware, plus the overhead for testbed management and maintenance.

In the field of 5G and 6G research, we find Colosseum [13], an open testbed with hardware-in-the-loop used for the emulation of wireless networks. Colosseum embeds 128 software defined radios (SDRs) connected through an FPGA-based channel emulator, each of which can be configured to run a specific technology along with a communication and application stack of choice. The testbed is installed at Northeastern University in Boston and it is remotely accessible. Colosseum offers flexibility, a large number of nodes, and open accessibility to a vast community of researchers. It also features highly realistic channel modeling, as each channel is modeled using Finite Impulse Response (FIR) filters and is updated up to 1000 times per second. The resulting set of tool-chains can provide high-fidelity digital twins of virtually any real-world wireless scenario [14].

Colosseum has been widely used for wireless communication

This work was partially supported by the National Telecommunications and Information Administration (NTIA)'s Public Wireless Supply Chain Innovation Fund (PWSCIF) under Award No. 25-60-IF011. This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Security and Rights in CyberSpace" (PE00000014 - program "SERICS").

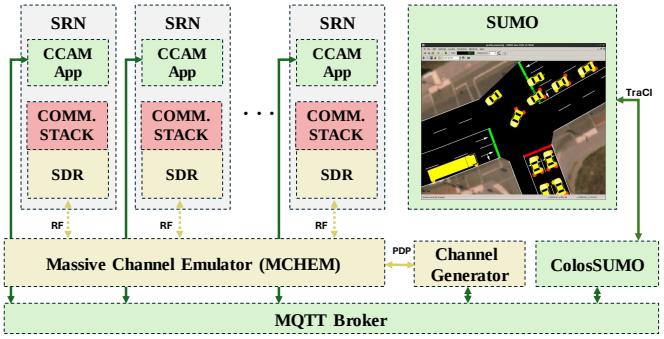

Figure 1: ColosSUMO architecture. Vehicular components are shown in green, while the wireless communication components are in yellow.

research [15]–[17]. However, limited works demonstrate its capabilities in emulating vehicular communications. To bridge this gap, and to address the limitations of existing frameworks, in this work we propose ColosSUMO, a HIL framework to evaluate the performance of CCAM applications that couples Colosseum for network emulation and SUMO for mobility simulation. Section II describes its architecture in detail. This framework, coupled with the real protocol stacks already available on Colosseum and its channel emulation capabilities, will allow researchers to accurately simulate vehicular technology applications. Differently from existing solutions, ColosSUMO enables researchers to perform HIL experiments exploiting the Colosseum testbed rather than setting up their own hardware, and with potentially any communication technology that is supported by SDRs. In addition, ColosSUMO exploits realistic mobility patterns and channel environments, enabling any kind of network or mobility analysis that can be done with modern simulation frameworks. ColosSUMO is available as open source software[1].

In the following sections, we describe its design principles and show its potential through a use case that simulates a platoon of 3 vehicles connected to a 5G Base Station.

II. Architecture

We start the description of ColosSUMO by introducing an architectural overview of Colosseum, so as to better understand the framework we propose in this work. For further details, we refer the interested reader to [13].

As anticipated in the introduction, Colosseum is a HIL testbed that integrates 128 standard radio nodes (SRNs). Each SRN, in turn, includes an x86 server and a SDR, which are used to execute the applications running on a network device, as well as the whole network stack down to the physical layer. The software is deployed through Linux Containers. The SDR, specifically an NI USRP X310, is then connected to the massive channel emulator (MCHEM), an FPGA-based emulator that can model each individual radio channel between any SRN pair. The channel dynamics are modeled by MCHEM with Power Delay Profiles (PDPs) including up to 4 non-zero taps, representing the multipath components of the channel, which are updated with a refresh rate of 1 kHz.

[1]https://github.com/michele-segata/colossumo

ColosSUMO aims at enabling the emulation of CCAM scenarios on Colosseum. The emulated vehicles communicate over a real wireless communication stack, e.g., a cellular vehicle-to-infrastructure stack, with wireless channels emulated by MCHEM. This makes it possible to measure the impact of wireless communications on the vehicle dynamics, as currently achieved through simulation frameworks such as Plexe and Veins [5], [10]. To simulate vehicle dynamics, we employ the SUMO mobility simulator [18]. It features realistic cooperative control algorithms thanks to Plexe extensions and enables the remote control of the simulation through the TraCI interface.

Fig. 1 shows ColosSUMO's overall architecture. We start by describing it from a high-level perspective. The whole emulation is governed by the ColosSUMO framework, which includes a software component that controls and orchestrates the emulation process, a MQTT broker, and the CCAM applications that run on the emulated vehicle. Specifically, the software component facilitates communication with SUMO to manage the simulation and retrieve mobility data. Concurrently, it ensures seamless interaction with the channel generator, enabling the exchange of simulation updates such as the positions of the vehicles. This integration allows MCHEM to adjust the communication channel in accordance with the evolving simulated scenario. MQTT mediates the exchange of messages between the various components, thanks to a publish-subscribe mechanism that allows multiple subscribers to receive simulation updates in real time. The CCAM applications running on the SRN communicate with ColosSUMO using the same MQTT protocol. This allows the applications to gather information about the vehicle to be shared with others (such as GPS position, radar measurements, speed, etc) as well as to alter the vehicle's behavior, e.g., by passing control information to cooperative adaptive cruise control (CACC) algorithms.

In more detail, to run a ColosSUMO emulation, we first reserve a set of SRNs on Colosseum. Then, we define a SUMO scenario, i.e., a road network plus the vehicles that should be simulated. Vehicles can be defined either using SUMO configuration files or via a scenario script that is given as an input parameter. Note that the number of reserved SRNs allocated at the beginning can not be changed in real time, thus the maximum number of vehicles that concurrently appear in the simulation will be equal the number of reserved SRNs.

ColosSUMO then starts the SUMO simulation and monitors its evolution, including the injection of new vehicles into the simulation. ColosSUMO also provides the SRNs with the application to be run, which is started within the containerized network stack. Such application will then exchange packets through the network stack, and Colosseum will emulate the wireless communication channel depending on the model and on the position of the communication nodes. The network stacks most typically used on Colosseum include OpenAirInterface and srsRAN, two open source implementation of 4G and 5G mobile networks. OpenAirInterface also partially supports 5G sidelink communication, which is of interest for the vehicular networking research community. Additionally, the 802.11 and LoRa protocols can also be executed.

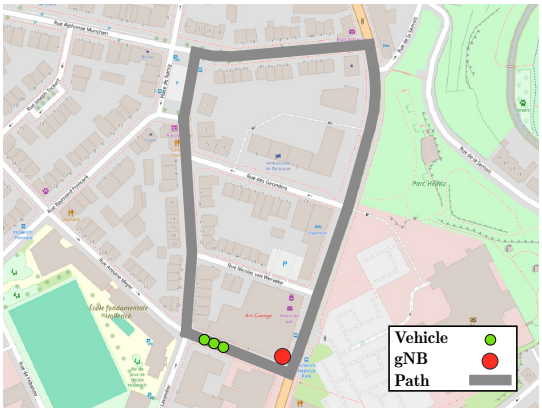

Figure 2: Map of the scenario showing the path of the vehicles, their start position, and the location of the gNB. Image courtesy OpenStreetMap.

III. Validation of the HIL emulator.

To validate ColosSUMO, we set up an experiment with three vehicles traveling in a platoon configuration around three blocks of Luxembourg City. We leverage OpenAirInterface as a software-based 5G stack [19] to coordinate the communication between the vehicles. Every vehicle runs a software user equipment (UE) on its associated SRN. The 5G UEs then connects to a 5G gNodeB (gNB) placed on the corner of one of the blocks. The gNB is deployed in an additional SRN. Figure 2 shows the initial location of the vehicles, their path and the location of the gNB.

Using this configuration, the emulated vehicles can communicate with one another through the cellular infrastructure in real time using a real mobile network stack that exploits SDRs for actual signal transmission and reception. The wireless channel between all the SRNs is then modeled by MCHEM. In this experiment, the channel between every pair of SRN consisted in a PDP with a single tap, generated in real time every $10\,\mathrm{ms}$. The specific channel model uses LiDAR data of Luxembourg City to evaluate the LoS between the transceivers and then computes the pathloss according to the stochastic 3GPP channel model for 5G networks [20].

The leading vehicle of the platoon travels at an average speed of $20\,\mathrm{m/s}$, and continuously changes its speed in the range between $15\,\mathrm{m/s}$ and $25\,\mathrm{m/s}$ every $10\,\mathrm{s}$. While such speeds are not realistic for a city scenario, they allow us to better observe the impact of the network on the dynamics of the vehicles.

To display the potential of ColosSUMO in the analysis of CCAM, we implement a CACC application on each vehicle, using a platooning algorithm provided by Plexe. This algorithm takes advantage of the control data transferred over the 5G SA network to maintain a constant inter-vehicle gap of $5\,\mathrm{m}$. Additionally, in order to evaluate how wireless channels affect the behavior of the vehicles, we implemented a simple fallback mechanism based on delay monitoring. Vehicles exchange platooning control data every $100\,\mathrm{ms}$. Upon receiving a packet, a vehicle checks the end-to-end delay by comparing the L4 timestamp with the current time.[2] If this value exceeds $300\,\mathrm{ms}$, the vehicle experiencing the delay will switch to a classic adaptive cruise control (ACC) with a time headway spacing policy of $1.2\,\mathrm{s}$. The vehicle will continue to use the ACC unless the measured delay falls below a $100\,\mathrm{ms}$ threshold consistently for at least $5\,\mathrm{s}$. In this case, the vehicle will switch back to cooperative driving control using the CACC. We remark that this is implementation is just a proof-of-concept, as a proper fallback mechanism requires more sophisticated algorithms [5].

[2]All SRNs in Colosseum are synchronized to a local Stratum-1 NTP server, with a synchronization accuracy not less than $5\,\mu\mathrm{s}$.

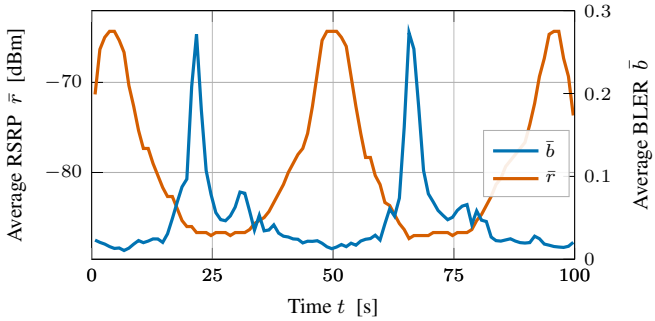

Figure 3: State of the wireless channel between the three vehicles and the gNB.

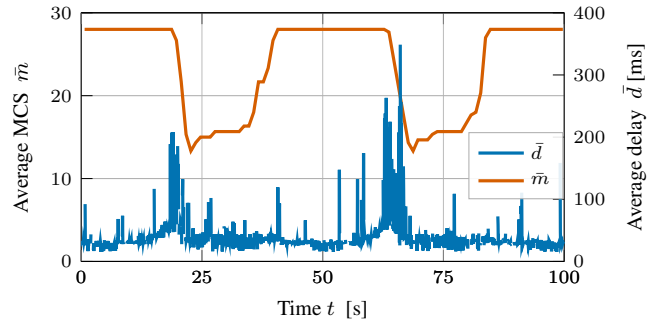

Figure 4: Average modulation and coding scheme (MCS) and average end-to-end delay.

IV. Metrics and Results

During the platooning experiments, different metrics are collected from all the different components. In particular, multiple metrics are collected every second across multiple layers from the OpenAirInterface stack. The vehicular application collects the end-to-end delay for each control message, while ColosSUMO logs the position of the vehicles every $100\,\mathrm{ms}$.

Fig. 3 shows two metrics: the average reference signals received power (RSRP), $\bar{r}$, and the average Uplink block error rate (BLER), $\bar{b}$ computed by averaging the RSRP and BLER metrics extracted from the wireless network stack at the gNB. We observe that the trend of the metrics are roughly related to the distance between the platoon and the gNB. At $t = 0\,\mathrm{s}$, the RSRP equals $-70\,\mathrm{dBm}$ and increases up to $-65\,\mathrm{dBm}$ as the platoon approaches the gNB. As the cars move away from the gNB, the RSRP decreases to $-87\,\mathrm{dBm}$ at $t = 30\,\mathrm{s}$, which corresponds to the farthest point from the gNB on the path.

In the same figure, we also observe that the RSRP tightly correlates with the BLER. In fact, as the RSRP sharply decreases and drops below $-80\,\mathrm{dBm}$ at $t = 20\,\mathrm{s}$ and at $t = 60\,\mathrm{s}$, a sudden BLER spike occurs, as the current MCS needs to be adapted to the worse channel conditions experienced, as discussed next.

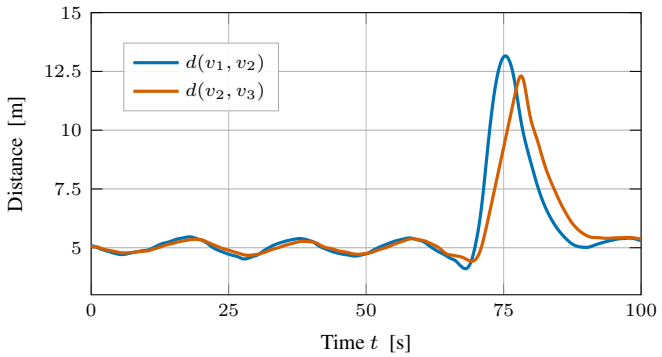

Figure 5: Distance between the vehicles.

Fig. 4 shows two additional metrics: the first one is the average MCS, $\bar{m}$, computed in the same way as the other metrics from the wireless network stack; the second one is the end-to-end delay measured at each vehicle and averaged, $\bar{d}$. We can observe that the MCS actually decreases when the RSRP decreases at $t = 20\,\mathrm{s}$ and $t = 70\,\mathrm{s}$, confirming our observation related to the results in Fig. 3. We also notice that, during this low-RSRP transient, the delay experienced at the upper layers (and specifically at the application layer) increases dramatically and exceeds $200\,\mathrm{ms}$. This is due to the high number of HARQ retransmissions performed at the link layer during the BLER spike. In fact, by analyzing the network stack, we observe up to 100 retransmissions per seconds during these transients.

Fig. 5 finally shows the behavior of the platoon during the experiment. In normal conditions, i.e., as long as the delay experienced by the CACC application is below the $300\,\mathrm{ms}$ target, the distance oscillates between $4.5\,\mathrm{m}$ and $5.5\,\mathrm{m}$, according to the acceleration/deceleration pattern of the leading vehicle. However, at $t = 70\,\mathrm{s}$, when the RSRP reaches its minimum value and the delay exceeds $300\,\mathrm{ms}$, the CACC algorithm is deactivated, and the vehicles switch to radar-based ACC. We observe the consequences of this fact as the distance between the vehicles increasesto $12\,\mathrm{m}$, before going back to normal once the CACC algorithm is re-activated at $t = 75\,\mathrm{s}$. These preliminary results show the potential of ColosSUMO for the analysis of CCAM applications using realistic communication stacks but without the need for custom hardware for HIL-based performance evaluation.

V. Conclusions and Future Work

HIL evaluation frameworks are becoming more and more common as constitute are a middle ground between FOTs and simulations offering the best of the two worlds. ColosSUMO, the contribution of this work, offers the possibility of simulating cooperative driving algorithms using SUMO and exploit Colosseum as the HIL emulation platform for the network and application stack. Different from existing solutions, the use of Colosseum, which is available for both for academia and industry, enables this type of analysis with no need to buy dedicated hardware equipment. In this paper, we show how ColosSUMO can be used to evaluate a simple CCAM application on top of the 5G stack, gaining insight both on the network and on the vehicles' dynamics. However, thanks to the fact that Colosseum employs SDRs, it will enable researchers to consider other communication technologies as well, such as IEEE 802.11p or any other that will be proposed in the future.

Part of our future work includes showcasing such capabilities by comparing the performance of vehicular applications considering different communication technologies, as well as more realistic and complex vehicular scenarios. Finally, we plan to make ColosSUMO available to the Colosseum user community at large through well-defined APIs.

References

[1] M. Amoozadeh et al., "VENTOS: Vehicular Network Open Simulator with Hardware-in-the-Loop Support," *Procedia Computer Science*, vol. 151, Jan. 2019.

[2] J. Wang and Y. Zhu, "A Hardware-in-the-Loop V2X Simulation Framework: CarTest," *Sensors*, vol. 22, no. 13, Jul. 2022.

[3] C. Obermaier et al., "Fully Reactive Hardware-in-the-Loop Simulation for VANET Devices," in *21st IEEE International Conference on Intelligent Transportation Systems (ITSC 2018)*, Maui, HI: IEEE, Nov. 2018.

[4] M. Segata et al., "On the Feasibility of RIS-enabled Cooperative Driving," in *14th IEEE Vehicular Networking Conference (VNC 2023)*, Istanbul, Turkey: IEEE, Apr. 2023.

[5] M. Segata et al., "Multi-Technology Cooperative Driving: An Analysis Based on PLEXE," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, Aug. 2023.

[6] V. Cislaghi et al., "Simulation of Tele-Operated Driving over 5G Using CARLA and OMNeT++," in *14th IEEE Vehicular Networking Conference (VNC 2023)*, Istanbul, Turkey: IEEE, Apr. 2023.

[7] T. Hardes et al., "Poster: A Case for Heterogenous Co-Simulation of Cooperative and Autonomous Driving," in *14th IEEE Vehicular Networking Conference (VNC 2023)*, Istanbul, Turkey: IEEE, Apr. 2023.

[8] A. Memedi et al., "Impact of Vehicle Type and Headlight Characteristics on Vehicular VLC Performance," in *10th IEEE Vehicular Networking Conference (VNC 2018)*, Taipei, Taiwan: IEEE, Dec. 2018.

[9] B. Schiel et al., "Poster: Informing V2I Deployment Decisions Using Commercial Hardware-in-the-loop Testing," in *14th IEEE Vehicular Networking Conference (VNC 2023)*, Istanbul, Turkey: IEEE, Apr. 2023.

[10] C. Sommer et al., "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, Jan. 2011.

[11] M. Drago et al., "MilliCar: An Ns-3 Module for MmWave NR V2X Networks," in *Workshop on ns-3 (WNS3 2020)*, ACM, Jun. 2020.

[12] R. Isermann et al., "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *Control Engineering Practice*, vol. 7, no. 5, May 1999.

[13] L. Bonati et al., "Colosseum: Large-Scale Wireless Experimentation Through Hardware-in-the-Loop Network Emulation," in *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN 2021)*, Los Angeles, CA: IEEE, Dec. 2021.

[14] D. Villa et al., "Colosseum as a Digital Twin: Bridging Real-World Experimentation and Wireless Network Emulation," *IEEE Transactions on Mobile Computing*, In press 2024.

[15] B. Keshavamurthy and N. Michelusi, "Learning-Based Spectrum Sensing and Access in Cognitive Radios via Approximate POMDPs," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, Jun. 2022.

[16] C. Puligheddu et al., "SEM-O-RAN: Semantic and Flexible O-RAN Slicing for NextG Edge-Assisted Mobile Systems," in *Proceedings of IEEE INFOCOM 2023 (preprint available as arXiv:2212.11853 [cs.NI])*, New York Area, NJ, USA, May 2023.

[17] M. Polese et al., "ColO-RAN: Developing Machine Learning-Based xApps for Open RAN Closed-Loop Control on Programmable Experimental Platforms," *IEEE Transactions on Mobile Computing*, vol. 22, no. 10, Nov. 2023.

[18] P. Alvarez Lopez et al., "Microscopic Traffic Simulation using SUMO," in *21st IEEE International Conference on Intelligent Transportation Systems (ITSC 2018)*, Maui, HI: IEEE, Nov. 2018.

[19] F. Kaltenberger et al., "OpenAirInterface: Democratizing innovation in the 5G Era," *Computer Networks*, vol. 176, 2020.

[20] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," Technical Report (TR) 38.901, Jun. 2018.