LIMoSim: A Lightweight and Integrated Approach for Simulating Vehicular Mobility with OMNeT++

Benjamin Sliwa, Johannes Pillmann, Fabian Eckermann and Christian Wietfeld
Communication Networks Institute
TU Dortmund
44227 Dortmund, Germany
e-mail: {Benjamin.Sliwa, Johannes.Pillmann, Fabian.Eckermann, Christian.Wietfeld}@tu-dortmund.de

***Abstract*—Reliable and efficient communication is one of the key requirements for the deployment of self-driving cars. Consequently, researchers and developers require efficient and precise tools for the parallel development of vehicular mobility and communication. Although current state-of-the-art approaches allow the coupled simulation of those two components, they are making use of multiple specialized simulators that are synchronized using interprocess communication, resulting in highly complex simulation setups. Furthermore, the compatibility of those simulators requires constant attention as they are developed independently. In this paper, we present a lightweight and integrated approach for simulating vehicular mobility directly in Objective Modular Network Testbed in C++ (OMNeT++) and INET without the need for external tools or Interprocess Communication (IPC). The proposed framework Lightweight ICT-centric Mobility Simulation (LIMoSim) is available as Open Source software and can easily be combined with other third-party extension frameworks for providing vehicular mobility based on well-known microscopical models. In contrast to existing approaches, the amount of necessary preprocessing steps for simulation setups is significantly reduced. The capabilities of LIMoSim are demonstrated by a proof of concept evaluation in combination with the Long Term Evolution (LTE) simulation framework SimuLTE.**

I. Introduction

OMNeT++ [1] is a well-established network simulation framework available as Open Source software for academic usage. Because of its modular approach, it has been extended by many third-party frameworks focusing on specialized communication technologies like LTE and IEEE 802.11p. As we have shown in previous work [2] the performance and robustness of communication systems can be highly improved by integrating knowledge of the users' mobility behavior into the routing decisions. With the deployment of autonomous cars in the near future, information about planned trajectories becomes available in all those vehicles and should be exploited for decision processes in the next generation of intelligent communication systems. With the increasing amount of interactions between mobility and communication, engineers and developers require efficient tools for simulating both aspects at once. Although current state-of-the-art frameworks already allow the coupling of specialized simulators for the two individual aspects, this approach has a number of system-immanent disadvantages (cf. Sec. II) because each of those simulators was developed for a very isolated field of application and not intended to be combined with others. In this paper, we present a lightweight framework for simulating microscopic vehicular traffic based on well-known models. In contrast to existing approaches, the proposed LIMoSim is intended to be used by communication simulators by design. It is seamlessly integrated into OMNeT++ (cf. Fig. 1) and requires no external tools or synchronization through IPC. Furthermore, it is completely compatible to third-party extension frameworks like SimuLTE [3] and INETMANET, which can integrate the novel mobility modules into their simulation scenarios in a transparent way using the widely-used INET framework. The rest of the paper is structured as follows. After discussing the related work, we present the basic architecture of our simulation model and provide a detailed description about the integration of LIMoSim into OMNeT++. In the next section, we describe the simulation setup for a proof of concept evaluation scenario in an LTE-context. Finally, detailed simulation results are presented and discussed.

II. Related Work

Vehicles in Network Simulation (Veins) [4] is a well-established simulation framework for simulating Vehicular Ad-hoc Networks (VANETs) in OMNeT++. It provides an implementation of IEEE 802.11p and acts as an interface to the microscopic traffic simulator Simulation of Urban Mobility (SUMO) [5]. Both simulators are synchronized through IPC using Transmission Control Protocol (TCP) and the dedicated Traffic Control Interface (TraCI) protocol of SUMO. While this approach ensures highly precise simulation results for both communication and vehicular mobility through usage of specialized simulators, it has a number of disadvantages:

- Since the different simulators are developed individually, their *compatibility* needs to be validated with every new

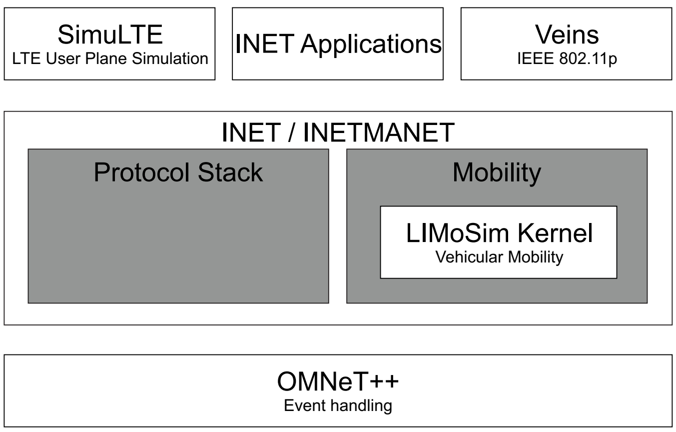

Fig. 1. Integration of the proposed framework into the family of OMNeT++ extensions. Since the the kernel of LIMoSim is embedded into the INET mobility module, the integration is transparent for the application-specific extensions frameworks like SimuLTE and can be used without requiring additional adjustments.

release. When we started to work on LIMoSim, Veins was not compatible to the newest SUMO version.
- For mobility-aware communication applications, TraCI is the bottleneck for the development process. The protocol needs to be extended for every new information type that should be shared between the simulators.
- Simulation setups have a high complexity because different tools have to be executed simultaneously. This aspect is even more dramatic for massive simulation scenarios that are executed on multiple servers in parallel. Additionally, even the generation of SUMO-only scenarios from OpenStreetMap (OSM)-data is quite complex, as several preprocessing steps are required.
- SUMO itself is rather designed for being used in a static way using precomputed data (for example routing paths). This does not match well with highly-dynamic vehicular applications, where the mobility behavior influences communication processes and vice versa.

Since the IPC-approach using SUMO is the current state-of-the-art way for simulating vehicular mobility in OMNeT++, those disadvantages are propagated to third-party extension frameworks like INETMANET and SimuLTE as well, if they want to make use of vehicular motion.

Consequently, we think the OMNeT++ community could benefit from a more diverse way for simulating vehicular mobility depending on the application scenario. While the current approach using Veins with SUMO is fitting for many IEEE 802.11p applications, there are scenarios where Veins could be coupled with our proposed framework to avoid the SUMO overhead and the need for IPC. Additionally, the complexity of many LTE and Mobile Ad-hoc Network (MANET) scenarios can be significantly reduced by using LIMoSim only.

III. Integration of LIMoSim into OMNeT++

LIMoSim focuses on highly dynamic traffic scenarios where all decision processes and routes are determined at runtime. Its main field of application is the simulation of medium-sized city scenarios, where intelligent vehicles interact with other traffic participants through means of communication. In this context, the simulation of vehicular mobility is considered as a service for the simulation of communication systems.

The simulator consists of two main components: the simulation kernel with the different elements of the microscopic mobility models and the User Interface (UI) part for standalone vehicle simulation and the road editor for easy generation of new mobility scenarios. It furthermore features live visualization of statistical data in dynamically updated plots and contains export functions for vector graphics. In this paper, we focus on the simulation kernel of LIMoSim, as it is the only component that is linked to OMNeT++ and provide descriptions of the hierarchical mobility model and the event handling mechanism in the following subchapters.

A. Simulation of Vehicular Mobility

The *LIMoSimCar.ned* module extends the *MovingMobilityBase.ned* module of the INET framework and acts as a logical structure for the different mobility-related submodules. Its hierarchical structure is illustrated in Fig. 2. A strategic model is used to determine the current destination point, which is the motivation for the vehicles movement. This node is

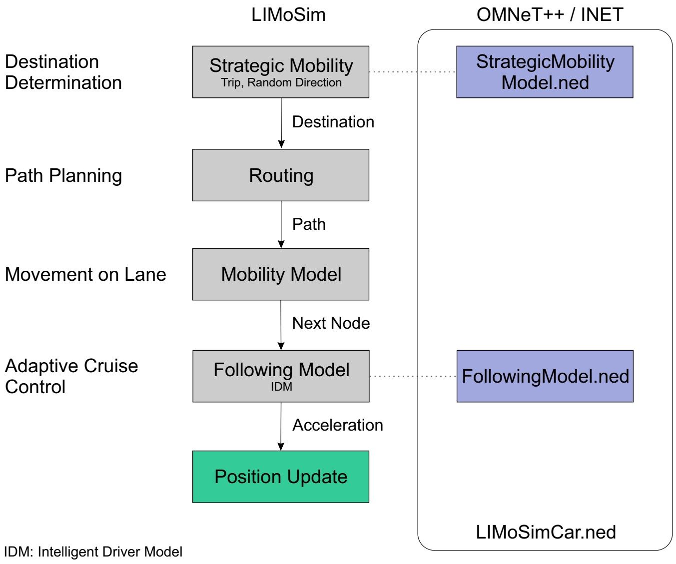

Fig. 2. Hierarchical model for microscopic traffic simulation encapsulated in the *LIMoSimCar.ned* module. OMNeT++-interface modules are available for the top and the bottom layer of the model.

then passed to a routing algorithm that computes the optimal path with respect to a defined criterion. All nodes of the path are handled sequentially by the general mobility model. Required adjustments to the mobility behavior because of encounters with other traffic participants are handled by the *Following Model*, which controls the distance between vehicles depending on their velocity by acceleration and deceleration. Finally, the position is updated with the calculated speed. In the current version of LIMoSim, we use the Intelligent Driver Model (IDM) [6] as a well-known following model for providing a realistic representation of vehicular mobility and combine it with the dedicated *lane-changing model* Minimizing Overall Braking Induced by Lane change (MOBIL) [7]. It should be noted that since the regular version of this model is not intersection-aware, the IDM implementation of LIMoSim treats traffic signals like static vehicles if the light is yellow or red. The integration of further models is planned for later releases. Since the individual mobility control modules

```
*.ue.mobilityType = "LIMoSimCar"
*.ue.mobility.map = "map.osm"
*.ue.mobility.strategicModel = "Trip"
*.ue.mobility.strategicModel.trip = "677230875,
275672221,3569208993,477807"
*.ue.mobility.way = "337055293"
*.ue.mobility.segment = 4
*.ue.mobility.lane = 0
*.ue.mobility.offset = 1m
```

Fig. 3. Example .ini configuration for using LIMoSim in an LTE-scenario. The trip model routes the car to a sequential list of a destinations. The actual node ids can be obtained from the UI part of LIMoSim or the raw OSM-files. The configuration can be either done manually by setting the parameters of the mobility modules or automatically using a generated XML file.

are interfaced through a *Simple Module*, the respective parameters can be set for the *.ned-file* without requiring external configuration files. An example configuration is shown in Fig. 3, where a User Equipment (UE) is assigned with the `LIMoSimCar` mobility type and positioned on a specified way segment. Furthermore, the `Trip` strategic mobility model is configured to sequentially approach multiple destination points (cf. Fig. 5). The ids originate from the OSM data model and can be obtained using UI of LIMoSim.

B. *Embedding LIMoSim-Events into OMNeT++*

The Discrete Event Simulation (DES)-coupling mechanism is illustrated in Fig. 4. Since LIMoSim can also be used in a standalone mode, its objects are not aware of their OMNeT++-environment. The integration into the event handling mechanism is performed by a virtual event queue that does not take OMNeT++ events into consideration. In contrast to the IPC-based approach, there is no need for real DES-synchronization, as only a single queue is used. If an event e is scheduled, the *event mapping singleton* creates a new OMNeT++ *cMessage* m and stores a map entry for original event. m is then inserted into the OMNeT++ event queue. Once the *handleMessage()* method for m is called, e is retrieved from the map entry and handled by the respective object. With this approach, the actual event handling in OMNeT++ is transparent for all LIMoSim objects. Moreover, LIMoSim objects can use the event handling mechanism of OMNeT++ without even requiring an actual OMNeT++ module. As a consequence, objects that only influence the vehicular traffic but do not contain communication modules (like interference traffic or traffic signals) do not need to be modelled in OMNeT++.

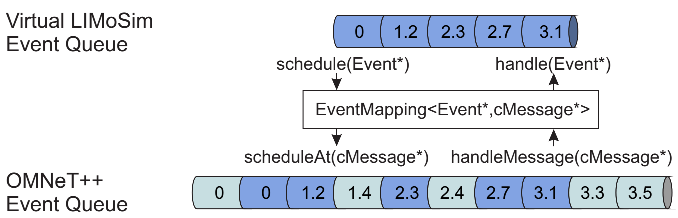

Fig. 4. Coupling of the event queues for mobility and communication simulation. LIMoSim objects are not aware of their execution environment and use a virtual event queue, which is mapped to the event queue of OMNeT++.

IV. Proof of Concept Evaluation

In this section, we present the setup and the results of the proof of concept evaluation. The default parameters for the scenarios are defined in Tab. I. Additional LTE-parameters are set according to the Handover (HO) example of SimuLTE.

TABLE I
Simulation Parameters of the Reference Scenario

Simulation parameter	**Value**
Strategic mobility model (UE)	Trip
Strategic mobility model (interference traffic)	Random Direction
Number of interference cars	100
Following model	IDM
Lane change model	MOBIL
Speed factor (driver behavior)	1 ± 0.2
Carrier frequency	1800 [MHz]
eNode B transmission power	46 [dBm]
eNode B antenna	omnidirectional

A. *LTE-scenario with Real-world Map Data*

As a realistic scenario, real-world map data from OSM is used. An LTE-enabled car is monitored while it is driving around the campus area of the TU Dortmund University using a *Trip* strategic mobility module. 100 other cars act as interference traffic with *Random Direction* mobility and influence the mobility behavior of the considered car. The area is covered by three different Evolved Node Bs (eNBs), which have been positioned according to network provider information. Fig. 5 provides an illustration of the described simulation setup. Due to the mobility of the car, multiple HOs are required at runtime. For the considered car, the current velocity, acceleration and the Received Signal Strength Indicator (RSSI) of the LTE link are measured over the trip duration and shown in Fig. 6.

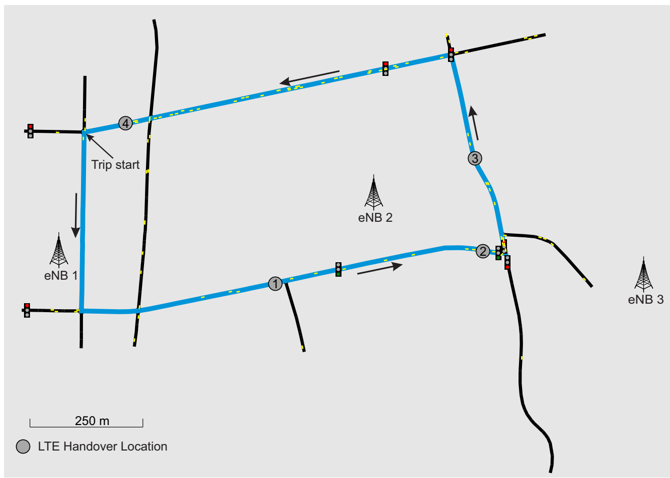

Fig. 5. Example scenario using OSM location data. 100 cars are simulated in the TU Dortmund University area. The map has been exported directly from the UI part of LIMoSim using its vector graphic export function.

The graphs for velocity and acceleration show typical inner-city characteristics and mirror the traffic dynamics of the scenario. Due to the map topology, the traffic signals and the other traffic participants, the current street condition is under constant change. Therefore, a lot of braking and accelerating is required, causing a highly dynamic velocity behavior. The RSSI of the LTE signal shows plausible characteristics depending on the distance to the current serving eNB. Four handovers occur during the considered time (for the actual handover locations cf. Fig. 5). With HO2, the device attaches from eNB2 to eNB3 and shortly afterward back to eNB2 with HO3. This *ping pong* behavior can be considered as a motivation for further research in the interdependency of communication and mobility, as the handovers could have probably been avoided using a mobility-aware decision approach. In earlier work [8], we have performed a similar case-study using a different toolchain with SUMO and an analytical LTE model, which achieved similar results for the handover behavior.

B. *Example Behavior of the IDM model*

In order to analyze the effects of the IDM model on the traffic dynamics, we consider a typical street scenario. Fig. 7 visualizes the acceleration behavior of multiple cars in a space-time diagram. The cars start in a jam situation, which is then

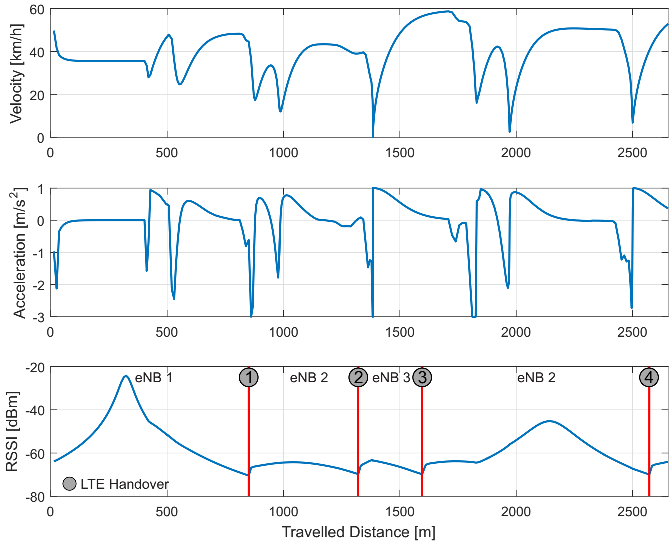

Fig. 6. Temporal behavior for velocity, acceleration and RSSI of the considered vehicle. The graphs show typical inner-city mobility characteristics and plausible LTE-behavior.

resolved into free traffic. After about 700m the cars encounter a car with engine failure that causes another traffic jam as only a single lane is used. Different driver behavior types can be identified depending on the slope of the curve. The results are confirmed by the analytical evaluation in [9].

V. Conclusion

In this paper, we presented the novel framework LIMoSim[1] for simulating microscopic vehicular mobility directly in OMNeT++. In contrast to existing approaches that treat vehicular mobility and communication separately and require IPC for the synchronization of different simulation tools, our proposal brings these aspects together in an integrated way. The actual mobility simulation relies on well-known models in order to guarantee the required accuracy. The easy integration is especially attractive for LTE and MANET simulations, which can make use of vehicular mobility without requiring a complex simulation setup. The capabilities of LIMoSim were demonstrated with a proof-of-concept evaluation in an LTE-context using real-world map data from OSM.

In future work, we want to couple LIMoSim with Veins for the simulation of IEEE 802.11p networks. LIMoSim could serve as an alternative to SUMO providing a lightweight solution for simulating vehicular motion without the IPC-overhead. Furthermore, we want to integrate strategic mobility models that are closer to human decision making like Small Worlds In Motion (SWIM) [10], which has already been applied to OMNeT++ in [11] and utilize the framework for the simulation of indoor robotic networks in a logistical context. Moreover, we want to exploit the visualization capabilities of the UI-part of LIMoSim to enable live visualization of OMNeT++ key performance indicators.

[1] Available at https://github.com/BenSliwa/LIMoSim

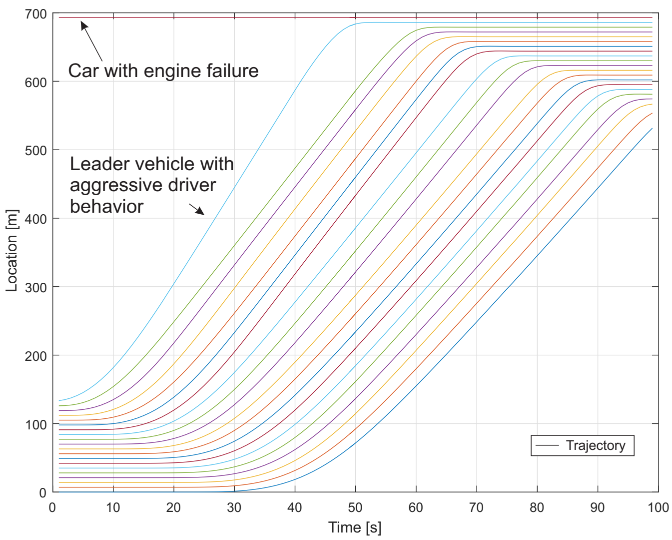

Fig. 7. Space-Time diagram for an inner-city scenario: the vehicles start in a jam situation, move into free traffic and are stopped by an obstacle in about 700m.

Acknowledgment

Part of the work on this paper has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 “Providing Information by Resource-Constrained Analysis”, project B4 and “European Regional Development Fund” (EFRE) 2014-2020 in the course of the “InVerSiV”project under grant number EFRE-0800422 and has been conducted within the AutoMat (Automotive Big Data Marketplace for Innovative Cross-sectorial Vehicle Data Services) project, which received funding from the European Union’s Horizon 2020 (H2020) research and innovation programme under the Grant Agreement No 644657.

References

[1] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools ’08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 60:1–60:10.

[2] B. Sliwa, D. Behnke, C. Ide, and C. Wietfeld, “B.A.T.Mobile: Leveraging mobility control knowledge for efficient routing in mobile robotic networks,” in *IEEE GLOBECOM 2016 Workshop on Wireless Networking, Control and Positioning of Unmanned Autonomous Vehicles (Wi-UAV)*, Washington D.C., USA, Dec 2016.

[3] A. Virdis, G. Stea, and G. Nardini, *Simulating LTE/LTE-Advanced networks with SimuLTE*. Cham: Springer International Publishing, 2015, pp. 83–105.

[4] 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, Jan 2011.

[5] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO - simulation of urban mobility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[6] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Phys. Rev. E*, vol. 62, pp. 1805–1824, Aug 2000.

[7] A. Kesting, M. Treiber, and D. Helbing, “General lane-changing model MOBIL for car-following models,” *Transportation Research Record*, pp. 86–94, 2007.

[8] J. Pillmann, B. Sliwa, J. Schmutzler, C. Ide, and C. Wietfeld, “Car-to-cloud communication traffic analysis based on the common vehicle information model,” in *IEEE Vehicular Technology Conference (VTC-Spring) Workshop on Wireless Access Technologies and Architectures for Internet of Things (IoT) Applications*, Jun 2017.

[9] M. Treiber and A. Kesting, “Traffic flow dynamics,” *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[10] A. Mei and J. Stefa, “SWIM: A simple model to generate small mobile worlds,” in *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*. IEEE, Apr. 2009, pp. 2106–2113.

[11] A. Udugama, B. Khalilov, A. bin Muslim, A. Foerster, and K. Kuladinithi, “Implementation of the SWIM mobility model in OMNeT++,” in *OMNeT++ Community Summit 2016*, September 2016.