Towards a Generic Middleware-Based Interface for Mobile Communication Simulation

Touhid Hossain Pritom, Burkhard Hensel, Mario Franke, and Christoph Sommer
TU Dresden, Faculty of Computer Science, Germany
https://www.cms-labs.org/people/ { touhid.pritom , hensel , franke , sommer }

***Abstract*—Researchers often couple different, formerly independent simulators to form a heterogeneous simulation environment, but often choose tight coupling or an overly generic approach, each of which has its drawbacks. In this paper, we propose a generic middleware-based interface to couple independent simulators for federated network simulation, which strikes a new balance. Our co-simulation approach performs event aggregation over a small time step to reduce the cost of frequent inter-federate interactions. We also perform a runtime evaluation to estimate the overhead introduced by our co-simulation approach.**

I. Introduction

The importance of mobile communication is still increasing rapidly with more and more diverse applications [1, 2]. Existing independent simulators are often not suitable for evaluating many advanced use cases. This is because one simulator would need to unify the details of simulating the complex behaviors of, e.g., cars, ships, drones, and satellites, as well as pedestrians, and wireless communication, each of which often requires full-fledged simulation tools of their own.

In response, there exist many tightly coupled simulation frameworks, such as Veins [3]. However, a loose coupling framework using middleware approaches increases the re-usability of the models [4, 5], as this eases the integration of different simulators or switching between related simulators.

In this paper, we propose a new middleware-based interface for coupling independent simulators, designed explicitly for simulating advanced vehicular mobile communication scenarios. In vehicular network simulation, a common approach is to couple different mobility or traffic simulators with usually one or, in some cases, more network simulators. At a high level, the overall simulation framework interaction involves exchanging information on mobility and data on various layers. Having a flexible middleware-based interface would facilitate researchers in performing more flexible heterogeneous mobile network simulations. However, a downside of using a more generic middleware-based approach is that it will likely require some manual tailoring before it can be effectively used for coupling simulators for mobile communication. On the other hand, a middleware-based approach, along with increased usability and flexibility, might also help open the door to research directions such as splitting larger scenarios into smaller sub-scenarios and utilizing multiple simulator instances to achieve faster simulation results. Coupling simulators using middleware likely enhances the usability and flexibility of the simulation framework, as it enables switching between simulators based on specific needs and use cases.

Neither coupling simulators nor coupling simulators using a middleware-based approach is a new concept. There exists an IEEE standard, the High-Level Architecture (HLA) [6], for loosely coupling simulators – and existing co-simulation frameworks, such as MOSAIC [7], which were developed based on the motivation of HLA, strive for generality.

Our co-simulation approach presents two novelties. First, it lies between the approaches of tightly coupled network simulation frameworks, such as Veins, and generic simulation frameworks, like MOSAIC, combining the advantages of both approaches for specific applications. Second, it performs event aggregation within small time steps to avoid frequent interaction between the middleware and simulators. It is reasonable to expect this to be significantly faster than generic frameworks in some instances.

We assume that a simulation is divided into equal time steps. This follows the logic of the most common mobility simulators, which are not event-based but rather step-based, which is a source of substantial but unavoidable error. MOSAIC provides the facility to simulate application logic using application federates, which may send or receive wireless messages at nanosecond intervals. As a result, a discrete event network simulator like OMNeT++ requires event-wise permission to advance the simulation time or must incur the full overhead of optimistic synchronization. Thus, the simulation is expected to be expensive, as the application logic performs frequent beaconing. Our co-simulation approach continues to simulate the application in the network simulator and aggregate events within each synchronization interval, enabling different simulators to process a batch of events separately during a single time interval. This solution may result in a loss of accuracy, which can be mitigated by adjusting the step interval length.

On the other hand, the generic gRPC-based interface of our co-simulation approach enables coupling with various simulators, allowing for the selection of coupled simulators by simply configuring a tailored configuration file for each application. This even allows replacing one simulator with another (e.g., OMNeT++ with NS-3), like in MOSAIC. Tightly coupled interfaces, such as Veins [3], do not provide this flexibility because they lack an abstract interface that hides the logic of the concrete simulators. The drawback of this approach is a slight increase in execution overhead, which is also expected due to multi-layer interfaces.

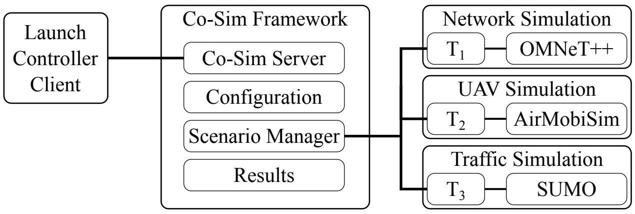


Figure 1. Architecture of our co-simulation approach. T_1, T_2, and T_3 are protocol translators for different simulators.

MOSAIC utilizes an abstract interface mechanism for coupling simulators, and this interface may require further extension depending on the nature and type of simulators, for example, OMNeT++, a discrete-event network simulator. In our co-simulation approach, for integrating a new simulator, a user only needs to implement the protocol services defined in the Protobuf file, which is equivalent to the correct instantiation of the Protobuf messages for interactions. This approach likely makes the integration of additional simulators more straightforward.

This paper presents the first steps towards an alternative, generic approach for coupling distributed simulators for mobile communication simulation.

II. Architecture

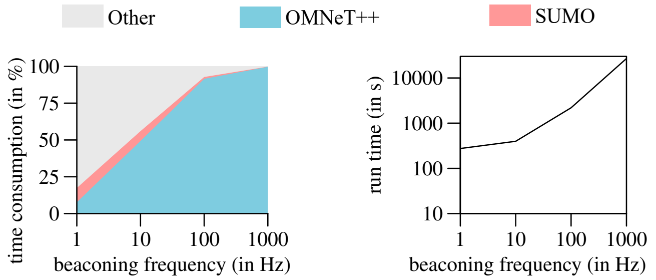
Figure 1 shows the architecture of our co-simulation approach. The proposed architecture features two distinct types of interfaces: a launch protocol that connects the launch client to the Co-Sim server, and a generic federation protocol that couples simulators and is independent of the specific simulator to be coupled. The launch protocol, which connects the client to the Co-Sim server, enables an implementation of our co-simulation approach to be hosted on a remote server, allowing an external agent, either a human or an AI tool, to launch the simulation and collect results remotely.

The interaction management among simulators is performed by exchanging interactions during the fixed-step synchronization phase. Once all simulators' local clocks are aligned to a common point, all simulators are asked to submit any interactions they wish to perform.

III. Evaluation

To evaluate the performance of our co-simulation approach, we conduct a runtime evaluation of an implementation developed using the Python programming language. We analyze the overhead of our co-simulation approach as the simulation complexity increases in an ad-hoc vehicular network simulation. We coupled OMNeT++ and SUMO using our co-simulation approach and ran the simulation for 400 seconds of simulation time.

Figure 2 presents the time analysis of our co-simulation approach for various beaconing frequencies, that is, network load. Figure 2a shows the impact on the percentage of CPU busy time for simulation components. It is apparent from the results that as the simulation complexity increases, the overhead



(a) CPU busy time as a percentage of the real simulation completion time against beaconing frequency.

(b) Real simulation completion time against beaconing frequency.

Figure 2. Execution time analysis of our co-simulation approach.

due to our co-simulation approach decreases, and the majority of the time is spent in the actual simulators (OMNeT++ and SUMO). This trend of decreasing overhead is due to our co-simulation approach's ability to aggregate event handling at synchronization points. For our evaluation, we selected a synchronization interval of 1 second.

IV. Conclusion and Future Work

In this paper, we propose a new middleware-based approach for coupling heterogeneous components of a combined mobile communication simulation, striking a new balance between generality and specificity. It also leverages the nature of fixed-time step mobility simulators to enhance simulation efficiency.

In the future, we plan to extend our implementation to couple not just new network but also new mobility simulators, such as for UAVs and LEO satellites.

Acknowledgment

The 6G-TWIN project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation program under Grant Agreement No 101136314.

References

[1] F. Liu, Y. Cui, C. Masouros, J. Xu, T. X. Han, Y. C. Eldar, and S. Buzzi, "Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 6, pp. 1728–1767, Jun. 2022.

[2] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G Networks: Use Cases and Technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, Mar. 2020.

[3] 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.

[4] N. Mustafee, K. Katsaliaki, and S. J. E. Taylor, "Distributed Approaches to Supply Chain Simulation: A Review," *ACM Transactions on Modeling and Computer Simulation*, vol. 31, no. 4, Aug. 2021.

[5] A. Gazis and E. Katsiri, "Middleware 101: What to know now and for the future," *ACM Queue*, vol. 20, no. 1, pp. 10–23, Mar. 2022.

[6] J. S. Dahmann, R. M. Fujimoto, and R. M. Weatherly, "The Department of Defense High Level Architecture," in *29th Conference on Winter Simulation*, Atlanta, Georgia, USA: ACM, 1997, pp. 142–149.

[7] K. Schrab, M. Neubauer, R. Protzmann, I. Radusch, S. Manganiaris, P. Lytrivis, and A. J. Amditis, "Modeling an ITS Management Solution for Mixed Highway Traffic With Eclipse MOSAIC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6575–6585, 2023.