Demo: VecSim, a Vehicular Edge Computing Simulator for Real-Time Applications

Chuanchao Gao, Arvind Easwaran
College of Computing and Data Science
Energy Research Institute @ NTU, Interdisciplinary Graduate Programme
Nanyang Technological University, Singapore
gaoc0008@e.ntu.edu.sg, arvinde@ntu.edu.sg

***Abstract*—This demo presents VecSim, a Vehicular Edge Computing (VEC) simulator designed to evaluate task offloading and resource allocation strategies for real-time applications in VEC. VecSim offers detailed and practical simulation of vehicle mobility, wireless communication in 5G networks, real-time service subscription architecture, and dynamic offloading control, bringing the gap between high-level algorithm design and practical deployment in VEC environments.**

***Index Terms*—Vehicular Edge Computing, Simulator**

I. Introduction

Vehicular Edge Computing (VEC) has emerged as a promising paradigm to enhance computational efficiency and service quality in intelligent transportation systems. By leveraging advanced wireless technologies such as the ultra-reliable low-latency communication (URLLC) capabilities of 5G, VEC enables vehicles to offload computation-intensive and time-critical tasks—such as intelligent traffic management and online navigation—to nearby Roadside Units (RSUs). These RSUs offer both wireless bandwidth for task offloading and computational resources for task execution. By processing tasks on nearby RSUs, VEC reduces reliance on centralized cloud infrastructure and lowers communication latency between vehicles and servers, making it well-suited for latency-sensitive vehicular tasks. However, the limited bandwidth and computational resources available at RSUs necessitate efficient strategies for *task offloading* (determining the RSU for task service deployment) and *resource allocation* (optimizing the allocation of bandwidth and computational resources), while satisfying the system resource and task deadline constraints.

Given VEC's potential, there has been growing research interest in developing effective task offloading and resource allocation strategies. However, deploying and evaluating these strategies on real testbeds is often cost-prohibitive and technically complex, making simulators essential tools for experimental evaluation. Popular edge computing simulators, such as iFogSim [1] and EdgeCloudSim [2], lack support for simulating 5G networks and are therefore unsuitable for modeling online offloading control that depends on real-time wireless channel quality feedback. Alternatively, network-focused simulators such as Simu5G [3] and Fogbed [4] offer fine-grained network modeling, yet they do not support integrated offloading frameworks that jointly optimize bandwidth and computational resources to meet task deadlines.

This work was supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

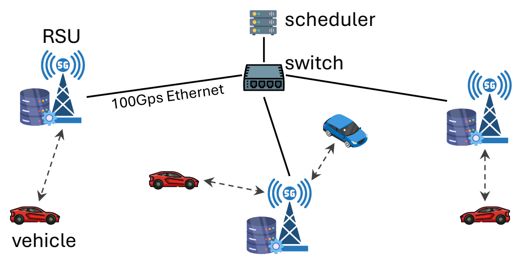

Fig. 1. VEC Simulator Environment

To bridge this gap, we develop VecSim, a novel VEC simulator that provides a comprehensive framework for real-time vehicular service subscription and resource scheduling. By integrating real-world data (e.g., traffic trace, task profiling) and 5G-based V2X network modeling, VecSim enables users to evaluate their scheduling strategies in a realistic VEC environment and assess various online offloading control mechanisms based on real-time channel quality feedback.

II. System Design and Architecture

A VEC (Fig. 1) comprises vehicles, RSUs, and a centralized scheduler. Vehicles communicate with RSUs over 5G networks, while RSUs are connected via wired Ethernet to a network switch, which is in turn connected to the scheduler. Each RSU provides wireless bandwidth for task offloading and computational resources for hosting vehicular services. The scheduler continuously monitors system status, including RSU resource utilization and the wireless channel quality between vehicles and RSUs, and periodically determines task offloading and resource allocation for vehicular requests. Given that many vehicular tasks, such as object detection, operate periodically to process continuous sensor data streams, our simulator primarily focuses on periodic vehicular tasks.

When a vehicle intends to subscribe to a vehicular service (e.g., to reduce the computational load of onboard devices or improve task performance), it first sends a request to the scheduler via a nearby RSU, which is queued for processing when the next scheduling cycle starts. The scheduler period-

2325-1301/25/$31.00 ©2025 IEEE
DOI 10.1109/RTCSA66114.2025.00033

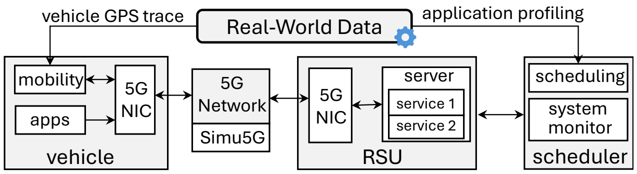

Fig. 2. VEC Simulator Architecture

ically invokes the scheduling algorithm to process queuing requests. Once a request is scheduled, a grant is sent to the designated RSU to initiate the vehicular service and enable the offloading of the corresponding task. Meanwhile, when a task is allowed to offload, its vehicle periodically transmits Sounding Reference Signals (SRS) to the RSU hosting its requested service for real-time wireless channel quality estimation. If the channel quality falls below a threshold, the RSU can instruct the vehicle to temporarily suspend task offloading, thereby avoiding potential deadline violations caused by increased transmission delays under poor channel conditions.

The architecture of VecSim is illustrated in Fig. 2. The vehicle module consists of three primary submodules: the application submodule, the mobility submodule, and the 5G Network Interface Card (NIC) submodule. Users can configure the number of tasks running on each vehicle. Each application submodule supports configurable parameters, such as task type, period, deadline, and data size, enabling users to use their own profiled application data. The mobility submodule governs vehicle movement, where both synthetic vehicle trajectories (e.g., using SUMO [5]) and real-world GPS traces are supported, enabling realistic traffic simulations.

The 5G NIC submodule plays a central role in modeling data transmission over 5G networks. VecSim integrates Simu5G to model 5G-based V2X communication, enabling realistic data offloading behavior and online offloading control based on real-time wireless channel quality estimation. In vehicles, the 5G NIC submodule handles task offloading and SRS feedback. At RSUs, it manages bandwidth allocation, estimates channel quality from received SRS reports, and forwards data to the RSU server submodule or the scheduler module. Extending the capabilities of the original Simu5G library, VecSim introduces an online offloading control mechanism that dynamically reallocates bandwidth based on varying channel quality or temporarily suspends offloading under degraded channel conditions to ensure task deadline compliance. The control logic is modular and can be customized by users to evaluate different adaptive offloading strategies.

In addition to the 5G NIC submodule, the RSU module also includes a server submodule responsible for managing computational resource allocation and hosting vehicular services assigned by the scheduler. The service submodule handles the initialization and termination of these services. Before each scheduling cycle begins, the RSU module collects the bandwidth usage and estimated channel quality from the 5G NIC submodule, as well as the current computational resource usage from the server submodule, and then forwards this information to the scheduler.

The scheduler module comprises a scheduling submodule and a system monitor submodule. The system monitors buffers service requests from vehicles, monitors bandwidth and computational resource usage at RSUs, and collects wireless channel quality metrics between vehicles and RSUs. The scheduling submodule is invoked periodically, allowing users to evaluate their task offloading and resource allocation strategies. To enhance the realism of VEC simulation, VecSim supports the use of profiling data obtained from real-world application execution. When a new scheduling cycle begins, the scheduling submodule retrieves system status data from the system monitor and application profiling data to decide the service deployment and resource allocation for pending requests in the current scheduling cycle.

III. Demonstration Description

In the demo, we construct a simulation environment using real-world taxi GPS trajectory data [6] to model vehicle mobility. The dataset was collected in Shanghai on April 1, 2018, by the Shanghai Qiangsheng Taxi Company. Besides, we evaluate four image classification applications (ResNet-152, VGG-16, VGG-19, Inception-v4) and two object detection applications (SSD-Mobilenet-v2, SSD-Inception-v2), profiling their execution on an NVIDIA Jetson Nano (representing onboard devices) and NVIDIA RTX 3090/4090/4500 GPUs (representing RSU servers). The demonstration includes:

- Customize simulation scenario via configuration files.
- Visualization of vehicle mobility and task offloading.
- Analysis of simulation results.

IV. Conclusion

This demo highlights the capabilities of VecSim in filling the gap between high-level algorithm design and practical deployment. By supporting customizable scheduling algorithms and online offloading control mechanisms, VecSim provides a flexible platform for evaluating task deployment and resource management strategies in practical VEC systems.

References

[1] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[2] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 39–44.

[3] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks," *IEEE Access*, vol. 8, pp. 181 176–181 191, 2020.

[4] A. Coutinho, F. Greve, C. Prazeres, and J. Cardoso, "Fogbed: A rapid-prototyping emulation environment for fog computing," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[5] 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," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[6] SODA, "Shanghai qiangsheng taxi gps data trace (2018-04-01)," 2018. [Online]. Available: https://github.com/hetianzhang/Edge-DataSet?tab=readme-ov-file#taxi-trajectory-data