Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

SIMULATION OF COOPERATIVE DOWNLOADING IN PYTHON

Michael Niebisch, Reinhard German, Anatoli Djanatliev

Friedrich-Alexander-Universität Erlangen-Nürnberg
Computer Networks and Communication Systems
Martensstr. 3
Erlangen, 91058, GERMANY

ABSTRACT

With the deployment of more software in vehicles, the need for efficient and cost-effective updates arises. One method to face this need is cooperative downloading, which enables vehicles to exchange parts of an update with each other, to collect a full set. Thereby, simulation of applicable strategies is crucial, as these can influence the efficiency of such systems, however, state of the art network simulators are not build for large scale test scenarios with long simulated time-spans. We introduce Cooperative Downloading in Python (CoDiPy), a framework for efficient analysis of cooperative downloading, while improving execution time. With CoDiPy, we have investigated relevant topics like encoding, communication strategies and cost optimization.

1 INTRODUCTION

The use of communication technologies in the vehicular world is growing. As more software features are deployed in the vehicles, more data exchange is needed. Besides interactive communication demands, like web searches, the need to update the vehicles' software over the air is increasing. However, only relying on cellular communication to exchange data between the vehicle and server is impacting the costs for communication. Additional technologies in a heterogeneous vehicular network, like WLAN and direct communication, offer a cheaper and opportunistic access to data. By using a cooperative downloading scheme, a fleet of vehicles exchanges data chunks with each other, in order to collect a full software update. This can reduce the reliance on cellular data, and reduce the overall costs. However, the setup and equipment of a large fleet of vehicles, to test different strategies for their effectiveness, is expensive and time consuming. Simulation offers a quick and easy way to determine the viability of different communication schemes. Here, network simulators specialize on exact replication of transmission processes, to determine the reception of every single message. This method slows down the overall simulation. However, in cooperative downloading, the interest is in the overall performance, not the reception of specific messages. Therefore, we have implemented a light-weight simulation framework for simulation of cooperative downloading, that enables us to quickly determine the effectiveness of a communication strategy.

2 COOPERATIVE DOWNLOADING IN PYTHON

For the simulation of cooperative downloading, we have developed a Python based framework called Cooperative Downloading in Python (CoDiPy). We use SUMO [1] for mobility and simulate the communication behavior per vehicle for cooperative downloading. This setup can be seen in Fig. 1. A *Fleet Manager* coordinates the deployment of vehicles, while a *Backend* coordinates the initial distribution of the

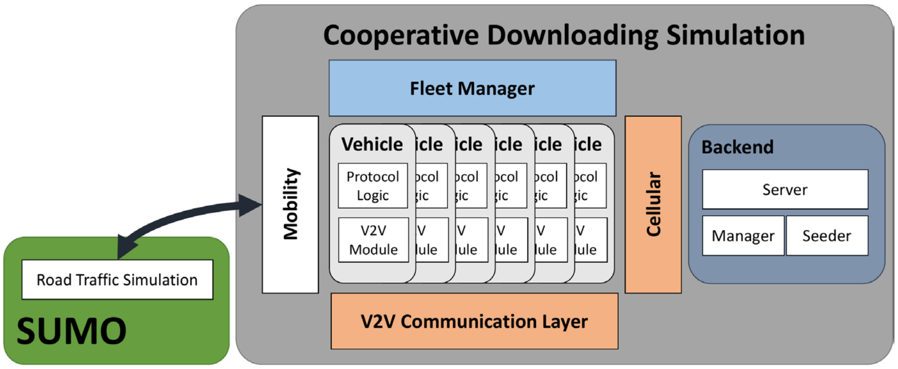

Figure 1: Cooperative Downloading in Python (CoDiPy) [2]

update. Each vehicle contains a *Protocol Logic* and a *V2V Module*, enabling the exchange of messages with other vehicles. This exchange is orchestrated in the *V2V Communication Layer*, which represents the transmission medium. In future steps, this framework will contain a WLAN-based communication layer, enabling the exchange of data between the vehicles and the backend via roadside WLAN access points.

In the course of this research project, multiple papers have been published. We have analyzed the effect of cooperative downloading on monetary costs in [3]. Here we could show, that an optimal initial distribution of the update can reduce costs by up to 80% compared to cellular communication. In [2], we have investigated different strategies for cooperative downloading and introduced a novel encoding scheme. In both [2, 3], our cooperative downloading framework was used. We further validated our framework with a network simulator in [4]. To enable the future modeling of WLAN access point communication, we conducted a measurement study in [5]. We introduced a novel method for approximating the effective communication duration between a vehicle and a stationary access point, as well as determining the availability of roadside communication.

3 CONCLUSION

In the context of cooperative downloading, the use of simulation is the most effective tool for evaluation. However, current network simulators focus on exact replication of message transmission, which slows down the simulation progress significantly. As we are only interested in the macroscopic behavior of the vehicle fleet, our cooperative downloading framework simplifies the communication process, while enabling fast paced simulations. Therefore, we have developed a framework in our past work, which focuses on direct communication, while our future work will introduce WLAN-based communication, to efficiently model backend communication.

REFERENCES

[1] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, “Micro-scopic traffic simulation using sumo,” in The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE, November 2018, pp. 2575–2582.

[2] M. Niebisch, D. Pfaller, R. German and A. Djanatliev, "Performance Improvements in Cooperative Downloading: Encoding and Strategies for Heterogeneous Vehicular Networks," 2022 IEEE 47th Conference on Local Computer Networks (LCN) (LCN 2022), 2022, pp. 1-9.

[3] M. Niebisch, D. Pfaller and A. Djanatliev, "Cooperative Downloading in Heterogeneous Vehicular Networks: A Cost Analysis," 2022 International Conference on Electronics, Information, and Communication (ICEIC), 2022, pp. 1-4, doi: 10.1109/ICEIC54506.2022.9748425.

[4] M. Niebisch, D. Pfaller, and A. Djanatliev. "CoDiPy: Performance Evaluation of Vehicular Cooperative Downloading in Python." Proceedings of the 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2022), Thessaloniki 2022, in-press.

[5] M. Niebisch, R. German and A. Djanatliev, "GRETEL: Graph-based Street Coverage Calculation for Vehicle-to-Infrastructure Communication," 2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS), 2021, pp. 1-7, doi: 10.23919/WONS51326.2021.9415594.