Demo: Making It Real – Virtual Edge Computing in a 3D Driving Simulator

Gurjashan Singh Pannu*, Jorge Luis López Altet*, Takamasa Higuchi†, Seyhan Ucar†, Onur Altintas† and Falko Dressler*

*Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany

†InfoTech Labs, Toyota Motor North America R&D, CA, U.S.A.

{pannu, jorge-luis.lopez-altet, dressler}@ccs-labs.org,
{takamasa.higuchi, seyhan.ucar, onur.altintas}@toyota.com

***Abstract*—We present a prototype for virtual edge computing based on embedded Linux PCs running the OpenC2X Vehicle to Everything (V2X) communications platform. The road traffic mobility of the cars is simulated by SUMO but all cars are represented by software instances running on the embedded system. This allows to move on to real-world testing by connecting the embedded systems to GPS and the car's onboard bus instead of the simulated mobility. Our virtual edge system supports setting up so-called vehicular micro clouds, i.e., groups of cars, which in turn can provide virtual edge computing services to other cars or nearby users. For this demo, we connected one of the simulated cars to a driving simulator so that it can be controlled, i.e., driven, by a user. The interaction with the system is completed by a 3D engine that visualizes the cars and the micro cloud information.**

I. Introduction

In recent years, many successful field tests have been conducted to test-drive self-driving cars on a road. For remote monitoring and cooperative decision making among self-driving cars, they demand continuous high-speed internet connectivity for getting quick response from cloud services. In addition, coordination among such self-driving cars will become essential, increasing the network load.

From a communications perspective, several standards exist supporting Vehicle to Everything (V2X) services, e.g., Cellular V2X (C-V2X) or the IEEE 802.11p-based DSRC/WAVE and ETSI ITS-G5 stacks. Both communication technologies support direct V2X communication as well as providing connectivity to the cloud either directly using the cellular network or via some road side unit (RSU). As bandwidth is limited and the distance to the back-end cloud is no longer negligible, the communication is expected to suffer from longer delays.

To overcome such delay problems, the concept of Multi-access Edge Computing (MEC) has been introduced [1]. The idea is to install infrastructure, i.e., computing and storage resources, in close proximity to the end users so that frequently requested data can be cached close to the users and data requests can be served with lower delays. Furthermore, computational tasks can be offloaded to the edge server rather than to the back-end cloud. Since the edge servers cover only small regions, the delay can be reduced significantly. On the downside, such MEC infrastructure needs to be ubiquitous which might make it infeasible to deploy.

At the same time, we see that cars are getting equipped with powerful computing, storage, and networking capabilities. Originating from the concept of vehicular clouds [2], the concept of vehicular micro clouds [3], [4] evolved, realizing virtual edge computing supported by cars on the road. The idea is to form groups of cars that, in turn, offer computational and storage resources and act as virtual edge servers.

Research has been done on the formation and maintenance of micro clouds, keeping data available in micro clouds, and assigning of the computational tasks – all substantiated by means of network simulations. In this demo paper, we take one step ahead and introduce a first virtual edge computing prototype making use of the Ego Vehicle Interface (EVI) [5], which integrates a real-time coordinated ego vehicle with a large-scale vehicular networking simulation. We further allow to operate multiple semi-real cars running the OpenC2X [6] framework to perform all vehicular networking tasks on a real IEEE 802.11p link provided by embedded Linux boxes representing On-Board Units (OBUs).

II. Virtual Edge Computing Prototype

Conceptually, our virtual edge computing prototype is based on the vehicular micro cloud architecture presented in [3]. The system automatically establishes micro clouds, which can later be used to act as a distributed cache or compute server. A high level diagram of designed system modules is shown in Figure 1. We implemented virtual edge computing prototype based on OpenC2X [6], which is an open source ETSI ITS-G5-based protocol stack for quick prototyping. The developed prototype can be used for field tests with simple embedded PC systems such as PC Engines Alix 3d3 system boards, equipped with Atheros AR9220 WLAN cards. The prototype consists of the following modules:

a) Micro Cloud Manager: This module is responsible for formation and maintenance of vehicular micro clouds. For this demo, we consider formation of micro clouds at road intersections, which can provide services to cars and users at the locations with high vehicle density. We implemented a geographic map-based micro cloud formation algorithm [7]. In brief, cars periodically broadcast their position and speed to the nearest RSU and the RSU computes a list of cars which can participate in the micro cloud at regular intervals.

978-1-7281-4571-6/19/$31.00 ©2019 IEEE

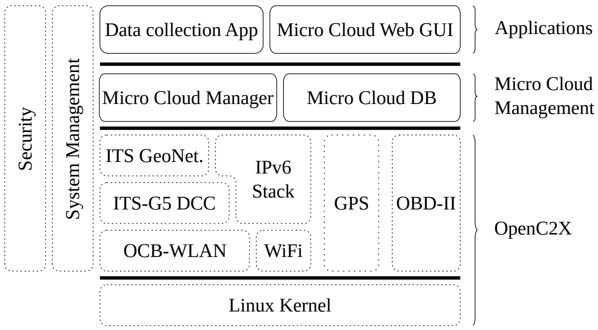

Figure 1. High level architecture of the virtual edge computing prototype based on OpenC2X. The modules represented by solid lines are implemented in the prototype, and the modules represented by dotted lines belong to OpenC2X.

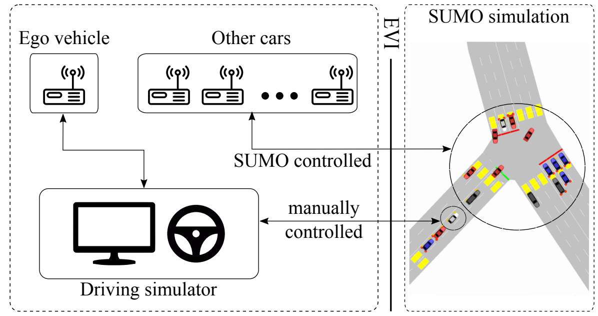

Figure 2. Our edge computing prototype runs on a real hardware, e.g., PC Engines Alix 3d3 system. The position and speed of ego vehicle is controlled manually, while others are controlled by SUMO simulation.

b) Micro Cloud DB: This database module manages all relevant information maintained by the micro cloud manager. This includes (but is not limited to) information about received beacon messages, locally stored data, data requests, and others.

c) Data Collection Application: We consider a data collection application as the micro cloud application. The application collects and aggregates data from nearby users and micro cloud members, followed by uploading to the nearest RSU. Although the aggregated data can also be uploaded via a cellular uplink, we rely upon IEEE 802.11p-based communication to RSU.

d) Micro Cloud Web GUI: The system runs a web server for remote monitoring. We can connect to the server from a remote desktop to view the status of a car, its role in the micro cloud, and information about the micro cloud, e.g., micro cloud members, the amount of data collected or uploaded.

III. Integration with SUMO and Unity 3D

In the demo setup, the prototype runs on a real system, but the position and speed of cars is simulated by the road traffic simulator SUMO. In addition, there is also an ego vehicle, which can be controlled manually by a driving simulator. The user can see information relevant to micro clouds in real time through the Unity 3D-based visualizer. This is done by integrating the developed prototype with the EVI [5]. The EVI helps to receive GPS and speed information from SUMO. In addition, it supports controlling one vehicle in SUMO, called the *ego vehicle*, manually using a gaming console. We replaced the vehicular network simulations in EVI with our implemented prototype to run all system components in real time using a real wireless channel. The demo setup is shown in Figures 2 and 3.

Figure 3. The laptop screen on the left shows Micro Cloud Web GUI which is connected to an Alix 3d3 system remotely. It shows a map, neighboring cars, and information about the micro cloud. On the central screen, a 3D driving simulator is running and the user can control the ego vehicle using a console. On the right screen, the SUMO simulation is running.

We allow a user take control of an ego vehicle, whereas the mobility of all other cars is simulated by SUMO. All cars, however, communicate in real time and over a real wireless connection via On-Board Units (OBUs) running OpenC2X.

IV. Conclusion

We have presented our novel virtual edge computing prototype based upon an open source vehicular networking platform OpenC2X. We implemented our vehicular micro cloud approach, which maintains small groups of cars that offer storage and computational resources distributed to all micro cloud members. The vehicular micro cloud thus acts as a virtual edge server. For the demo, we also integrate the prototype with the Ego Vehicle Interface (EVI) and the Unity 3D visualization.

References

[1] "Mobile Edge Computing (MEC), Framework and Reference Architecture," ETSI, GS MEC 003 V1.1.1, Mar. 2016.

[2] M. Gerla, "Vehicular Cloud Computing," in *IFIP/IEEE Med-Hoc-Net 2012*, Ayia Napa, Cyprus: IEEE, Jun. 2012, pp. 152–155.

[3] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Vehicular Micro Clouds as Virtual Edge Servers for Efficient Data Collection," in *ACM MobiCom 2017, CarSys Workshop*, Snowbird, UT: ACM, Oct. 2017, pp. 31–35.

[4] F. Dressler, G. S. Pannu, F. Hagenauer, M. Gerla, T. Higuchi, and O. Altintas, "Virtual Edge Computing Using Vehicular Micro Clouds," in *IEEE ICNC 2019*, Honolulu, HI: IEEE, Feb. 2019.

[5] D. S. Buse, C. Sommer, and F. Dressler, "Integrating a Driving Simulator with City-Scale VANET Simulation for the Development of Next Generation ADAS Systems," in *IEEE INFOCOM 2018, Demo Session*, Honolulu, HI: IEEE, Apr. 2018.

[6] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer, and F. Dressler, "OpenC2X - An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5," in *IEEE VNC 2016, Demo Session*, Columbus, OH: IEEE, Dec. 2016, pp. 152–153.

[7] F. Hagenauer, T. Higuchi, O. Altintas, and F. Dressler, "Efficient Data Handling in Vehicular Micro Clouds," *Elsevier Ad Hoc Networks*, vol. 91, p. 101 871, Aug. 2019.