An experimentation environment for SDN-based autonomous vehicles in smart cities

Athanasios Papadakis, Tryfon Theodorou, Lefteris Mamatas, Sophia Petridou
University of Macedonia, Egnatia 156, 546 36, Thessaloniki Greece
{apapadaki, theodorou, emamatas, spetrido}@uom.edu.gr

***Abstract*—The emerging Internet of Things (IoT) in conjunction with Autonomous Vehicles (AVs) have increased the complexity of network administration and control. In this context, recent proposals bring together Software-Defined Networks (SDNs) with AVs practices, i.e., introducing innovative network control strategies bespoke for smart city infrastructures. However, an experimentation environment that combines the realism of an AVs ecosystem with the SDN network paradigm is a necessity, facilitating the investigation of particular research issues, including efficient data management and routing, as well as security. In this demo, we introduce a relevant facility that builds-up on novel open environments for hands-on experimentation, namely: (i) CARLA, an open realistic urban-driving simulator; (ii) Cooja, a state-of-the-art emulator for Wireless Sensor Networks (WSNs); and (iii) SD-MIoT, a novel SDN solution for mobile IoT. Our experimentation exercise currently focuses on maintaining connectivity of AVs to the fixed infrastructure, e.g., Road Side Units (RSUs), with SDN strategies. We demonstrate the capabilities of the proposed experimentation environment with proof-of-concept results on packet delivery ratio and control overhead, quantifying the efficiency of the considered SDN approach to maintain AV connectivity, as well as with a video visualizing the outcome of our experiments.**

I. Introduction

The AV ecosystem constitutes a rapidly emerging domain of advanced research, entrepreneurship and innovation. The modern AV is a complex system of various electronic components, multiple sensors and control units. Reliable communication, synchronization and performance of these components are essential to AV infrastructures [1].

To address such issues many recent proposals look at applying SDN principles to Vehicular Ad Hoc Networks (VANETs), paving the way for a new networking paradigm called Software-Defined Vehicular Networks (SDVNs) [2]. SDN offers centralized control, flexibility and programmability in VANETs that facilitate network management and enable new Vehicle-to-vehicle (V2V) and Vehicle-to-Infrastructure (V2I) services, including for example, vehicle and road safety services, data management and mobile vehicular cloud services, e.g., infotainment [3].

However, to investigate research challenges and corresponding mechanisms an open experimentation environment for SDN-based autonomous vehicles is a necessity. In this demo we propose an environment that combines CARLA simulator [4], Cooja emulator [5] and our SD-MIoT solution [6] for mobile IoT. CARLA is a well-known open urban-driving simulator that offers realism in our experiments, e.g., it consider moving AVs within a realistic smart city infrastructure. Cooja emulator excels in WSN-based experimentation and SD-MIoT solution [6] brings novel SDN strategies tailored for mobile IoT deployments, while providing holistic management.

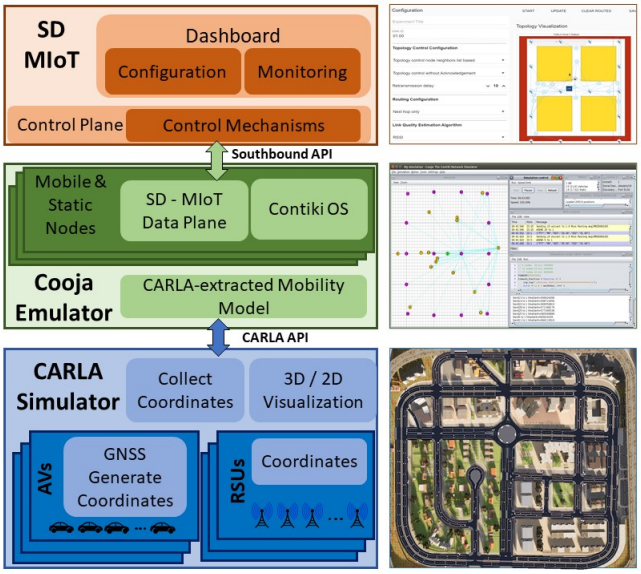

Fig. 1: An abstract view of demo architecture

The feasibility of our approach is evaluated in controlled scenarios, where we focus on V2I communication and maintain the connectivity of AVs to RSUs, i.e., exploiting SDN strategies provided by SD-MIoT solution [6]. Thus, for an increasing number of AVs that move around a smart city, we configure the SDN protocol and measure its performance in terms of *packet delivery ratio (pdr)* and *control overhead*. Our preliminary results demonstrate that our experimentation environment is able to investigate AVs network connectivity maintenance and and can also be utilized for further experimentation in the SDVN context.

II. Demo architecture

Fig. 1 presents an abstract view of our demo architecture, consisting of three layers. At the bottom, we assume a realistic smart city environment containing all relative items, such as roads, traffic lights and vehicles along with their mobility models. We utilize the CARLA simulator that offers, among others, all the aforementioned features facilitating our demo experiment; some of them are depicted on the corresponding

978-3-903176-36-2 ©2021 IFIP

screenshot. In the middle layer we handle the CARLA-extracted information to apply realistic simulations in Cooja, utilizing an SDN protocol for mobile IoT network environments [7]. The top layer accommodates the SDN controller and its mechanisms, as well as a dashboard functionality allowing protocol's configuration and network's monitoring.

More specifically, CARLA simulator provides a realistic smart city environment with the option to select among a plethora of smart city alike maps, where a number of AVs are deployed. It supports 2D/3D visualization of each map along with the selected items. The navigation of AVs is controlled by the Traffic Manager (TM). In order to collect accurate and usable data from CARLA, we attach a Global Navigation Satellite System (GNSS) sensor to each one of the autonomous vehicles and pinpoint the exact location of each RSU throughout the map. We implement the CARLA-API to operate as a mediator between the bottom and middle layer. We convert the realistic AVs moving behavior to mobility data used by the Cooja, i.e., to simulate a VANET IoT scenario.

For the communication between AVs and RSUs, we are using SD-MIoT, our own open-source SDN solution for mobile IoT environments [6]. SD-MIoT consists of an OpenFlow-like data-plane protocol [8] for IoT motes that use Contiki OS and a modular SDN controller that incorporates novel mobility-aware topology discovery mechanisms, routing policies and flow-rule establishment methods, all of them balancing control overhead with routing robustness. We use a flexible, web-based, and user-friendly GUI Dashboard for the overall network monitoring and system management, providing advanced system visualization and configuration options.

To demonstrate the research potential of our experimentation environment, we used CARLA's pre-build image alongside with Cooja emulator and our SD-MIoT controller. CARLA simulator is based on Unreal Engine and uses the OpenDRIVE standard for the definition of roads and urban settings. The demonstration starts with the deployment of vehicles - set on autonomous mode - within CARLA's map. Then, Cooja emulator receives the corresponding coordinates of each vehicle and RSUs via *CARLA-API*. Cooja and Contiki are concentrating on network behavior and simulate the wireless node network of CARLA's actors. The final component of our experiment's sequence is the SD-MIoT controller, where we apply the topology control mechanisms, construct the network connectivity graph and begin the network packet generation process. SD-MIoT's dashboard illustrates the data, and control packets in real-time, while indicating the finalization of our experiment. A relative short video is available online[1].

III. Proof-of-concept results

In this section, we present our simulation setups, configurations, and proof-of-concept results. Our main goal is to realize the benefits of the application of SDN protocols in VANET environments towards robust communication performance. We measure the *pdr* as well as the communication demand in terms of network control overhead. We calculate the *pdr* as the ratio of the *received* data packets R_x over *sent* data packets S_x; high values represent reliable data transmission, whereas lower ones reveal deficiencies in network connectivity. We compute the *control overhead* as the ratio of the control packets C_x (i.e., topology discovery packets) propagated by each AV through RSUs over the total number of packets received by the destination $T_x = R_x + C_x$. We investigate network management and control by carrying out several simulation-runs that exploit the flexibility of centralized SDN solutions in altering network protocol parameters dynamically, in our case network topology discovery interval time. The latter is a critical parameter in mobile network conditions as its proper adjustment affects the network's connectivity perception, which in turn improves network *pdr*.

TABLE I: Simulation parameters and configuration setups

Parameters		**Configurations**
Simulation Scenario	Mobility model	CARLA Traffic Manager
	Map	Town 3
	Map dimensions	400×500 m
	Border routers	1 node
	RSUs	16 nodes
	AVs	15 nodes
	AVs speed	$[0 - 40]$ km/h
Data	Packet size	128 $Bytes$
	Transmission Rate	90 data-packets/h
Network	Transport	UDP
	Network	SD-MIoT
	Physical/MAC	IEEE 802.15.4
Hardware	Radio Interface	2.4 GHz
	Radio Range	110 m
	Mote	Cooja mote
Simulation	OS / Simulator	Contiki / Cooja
	Duration	30 min

We carried out our simulations considering a use-case scenario of 15 AVs operating in a smart city environment where 16 RSUs positioned to offer full radio coverage. The setup parameters are shown in Table I. Since we focus our study on routing processes under mobility, we assume that data loss is only related to distance and no other signal issues of the radio environment. This configuration provides specific judgments related to the SDN network layer mechanisms only and establishes a deterministic environment. As such, there is no need to validate the statistical accuracy of our results.

In Fig. 2, we depict our results using a bar chart that exhibits at the end of the simulation (i.e., 30 min) the PDR for both AVs and RSUs (blue bar), that of RSUs only (red bar) and that of AVs only (green bar). Our results present three simulated scenarios where the configuration parameter Topology Refresh Time interval (TRTi) varies, i.e., $20s$, $30s$, and $40s$. As an initial observation, we underline that frequent topology refresh processes improve the network's *pdr* performance. In detail, we observe that the *pdr* of the RSUs (i.e., static network nodes), is close to 100%. This is expected since the RSUs represents fixed network nodes which maintain stable data routing paths.

On the contrary, AVs mobility imposes changes to the network connectivity graph that results in reduced *pdr* mainly

[1]Short demo video

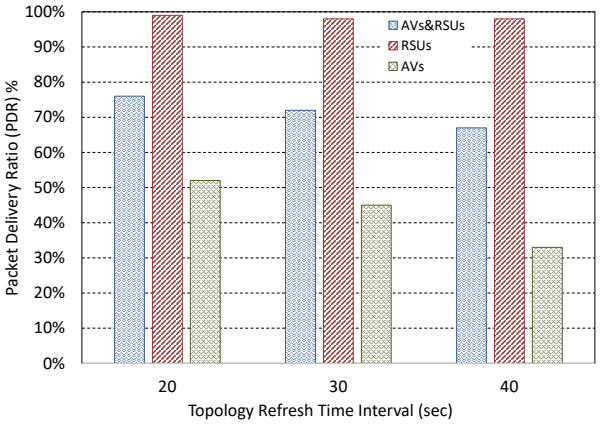

Fig. 2: Packet delivery ratio of autonomous vehicles (AVs) and road side units (RSUs) in line with the topology refresh time interval (sec)

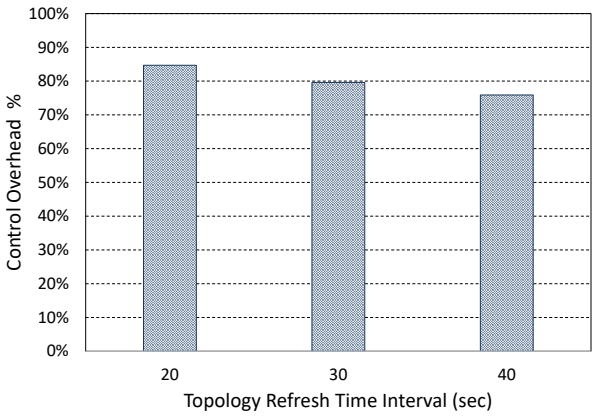

Fig. 3: Control overhead in line with the topology refresh time interval time (sec)

because the network protocol cannot detect these changes in a timely manner. In our scenario, we observe that the effect of mobility reduces the *pdr* of AVs to 50% and more compared to RSUs. The percentage of the *pdr* reduction is strongly related to the mobility behaviour of the AVs, i.e., speed and trajectory. However, we observe a notable improvement of *pdr* from 33% to 52% when we adjust the TRTi time from $40s$ to $20s$.

Although frequent topology discovery attempts improve *pdr*, it results in increased *control overhead*. In Fig. 3 we observe a 10% increase from $40s$ to $20s$ TRTi. This drawback reduces network efficiency in terms of energy consumption and data traffic. However, such encumbrance can be considered acceptable depending on the network application requirements, i.e., emergency communication scenarios. Wind up we highlight that the golden rule of such scenarios lies with the network application needs, and as a result, flexible network management that dynamically configures network parameters efficiently can produce acceptable results per use-case scenario. We argue that such flexibility is embedded in SDN paradigm and even though in our current framework these changes are done by a network administrator using the SD-MIoT Controller Configuration Dashboard. We envisage as a future extension an intelligent SDN controller module that adapts network's operation based on network application parameters, e.g., on the AV's speed, aiming to enhance network's QoS.

IV. Conclusions

In this work, we provide insights about the value of an experimentation environment for SDN-based autonomous vehicles in smart cities. We briefly discussed the benefits and challenges within the SDVN domain and continued by presenting our demo architecture and simulation environment. Furthermore, we deliver information regarding our simulation setup, configuration and proof-of-concept results. Our preliminary results demonstrate that the SDN protocols will be an asset when applied in VANET environments.This environment can provide future directions to researchers and motivation to further study data management and communication in the SDVN context.

References

[1] J. Bhatia, Y. Modi, S. Tanwar, and M. Bhavsar, "Software defined vehicular networks: A comprehensive review," *International Journal of Communication Systems*, vol. 32, no. 12, p. e4005, 2019.

[2] I. Yaqoob, I. Ahmad, E. Ahmed, A. Gani, M. Imran, and N. Guizani, "Overcoming the key challenges to establishing vehicular communication: Is sdn the answer?" *IEEE Communications Magazine*, vol. 55, no. 7, pp. 128–134, 2017.

[3] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, "Towards software-defined vanet: Architecture and services," in *2014 13th annual Mediterranean ad hoc networking workshop (MED-HOC-NET)*. IEEE, 2014, pp. 103–110.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[5] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *1st IEEE Int. Workshop on Practical Issues in Building Sensor Netw. Appl. (SenseApp)*, 2006.

[6] T. Theodorou and L. Mamatas, "SD-MIoT: A software-defined networking solution for mobile Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4604–4617, 2020.

[7] T. Theodorou and L. Mamatas, "A Versatile Out-of-Band Software-Defined networking solution for the Internet of Things," *IEEE Access*, vol. 8, pp. 103 710–103 733, Jun 2020.

[8] T. Theodorou and L. Mamatas, "CORAL-SDN: A software-defined networking solution for the Internet of Things," in *2017 IEEE conference on network function virtualization and software defined networks (NFV-SDN)*. IEEE, 2017, pp. 1–2.