

Towards a Novel Vehicular Ad Hoc Networks Based on SDN

Houda Guesmi(✉), Anwar Kalghoum, Ramzi Guesmi, and Leïla Azouz Saïdane

CRISTAL LAB, National School of Computer Science, Manouba, Tunisia
{houda.guesmi,anwar.kalghoum,leila.saidane}@ensi-uma.tn,
ramzi.guesmi@gmail.com

Abstract. Driven by long traffic jams and numerous road accidents, vehicle networks (Vehicular Ad hoc NETwork, VANET) have emerged to make the journey more pleasant, the road safer and the transport system more efficient. Today's vehicle network architectures suffer from scalability issues as it is challenging to deploy services on a large scale. These architectures are rigid, difficult to manage and suffer from a lack of flexibility and adaptability due to vehicular technologies' heterogeneity.

Over the past few years, the emerging paradigm of Software-Defined Networking (SDN) network architecture has become one of the most important technologies for managing large-scale networks such as vehicle networks. By the first vision and under the SDN paradigm umbrella, we propose a new VANET network architecture based on the SDN paradigm named "SDN-based vehicular ad hoc networks" (SDN-VANET). Through our simulation, we show that in addition to the flexibility and fine programmability brought by the SDN paradigm, the latter opens the way to the development of efficient network control functions.

Keywords: VANET · Software Defined Networking · SDN · openFlow · Distributed controller

1 Introduction

Nowadays, we spend more and more time in transport, whether in personal vehicles or public transport. In addition, our uses of the means of communication have become more and more nomadic, especially with the significant advances in information and communication technologies (ICT). Therefore, the concept of ad hoc vehicle networks emerged as a result of this development to offer a wide variety of services, ranging from improving road safety to optimizing traffic, including entertainment. Driver and passengers. Indeed, this type of network is mainly characterized by the nodes' high mobility, the incredibly dynamic topology, and the network's enormous scale. Vehicle networks represent a projection of Intelligent Transportation Systems (ITS), in which vehicles can communicate with each other and with infrastructure along roads. In this chapter, we introduce the basic concepts of vehicle networks before describing the state of the art of the software-defined networking paradigm applied to vehicle networks.

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12949, pp. 663–675, 2021.
https://doi.org/10.1007/978-3-030-86653-2_48

Despite the development and rapid emergence of vehicle networks with futuristic capabilities, their architectures suffer from several shortcomings. In addition to the problem of the heterogeneity of network equipment, which causes great difficulties in management and integration [1], we can cite, for example, (i) the lack of scalability in the deployment of services on a large scale in such a dense and dynamic topology such as that of vehicle networks [1,2], (ii) lack of intelligence, mainly due to the closed aspect of vehicular equipment and their inherent characteristics such as the absence of programmability and their development dependence on suppliers. This implies severe and challenging to manage architectures, (iii) the lack of flexibility and adaptability, induced by the great diversity of deployment environments and the vast heterogeneity of wireless communication technologies (4G/5G, WiFi, etc.).

Vehicular Ad-hoc Networks (VANETs) [1,3,4] architectures today suffer from scalability issues as it is tough to deploy solutions on a large scale. These architectures are rigid, difficult to manage, and suffer from a lack of flexibility and adaptability in control [1]. Therefore, it isn't easy to choose the right solution, given the current context, due to the diversity of deployment environments and the wide variety of solutions. These constraints limit the functionality of the system and often lead to the under-exploitation of network resources. Thus, the need for new, more flexible, and scalable architectures becomes an absolute requirement to face the new needs of the next generations of vehicle networks.

In recent years, the Software-Defined Networking (SDN) paradigm [5] has been proposed as an innovative solution to manage large-scale networks such as VANET. Indeed, several works [1] have shown that by separating the control plane from that of the data and that by adopting a centralized control mode, SDN can provide flexibility, scalability, and programmability to the architectures of VANET. SDN also makes more efficient use of network resources and introduces new services.

In line with the first vision and under the SDN paradigm umbrella, we provide a new network architecture that allows joint control of different vehicle access network technologies. We also explore the possibilities offered by such an architecture combining both the advantages of hybridization of various technologies and the SDN paradigm properties. We show through a few use cases that in addition to the flexibility and fine programmability brought by the SDN paradigm, the latter paves the way for efficient network control functions.

This article will combine the VANET paradigm and the SDN architecture and propose a VANET architecture based on the SDN paradigm. Our proposal called SDN-based vehicular ad hoc networks strategy (SDN-VANET), relies on SDN controllers. The experimental results show that SDN-VANET significantly reduces the network load while improving the VANET network's performance.

This article is organized as follows. Section 2 presents some preliminary concepts such as Software-defined networking and vehicle networks. We present a synthesis of existing work in the scientific literature in Sect. 3. Section 4 details the proposed approach. Section 5 describes the simulation results of a use case. Finally, the last Sect. 6 concludes this article.

2 Software-Defined Networking

Initially designed for wired networks and, in particular, data centers, SDN has had tremendous success in industry and academia. In 2012, Google engineers announced that they had switched to using SDN to connect their WANs (Wide Area Networks) to data centers [6]. There are currently even network devices compatible with SDN and available on the market, such as the OpenStack [7] or OpenDaylight [6] controller. Since then, researchers have explored all the possibilities to take advantage of SDN's advantages to improve performance and facilitate the management of today's vehicle network architectures.

2.1 Principle and Characteristic of SDN

The concept of software-defined network, widely known as Software-Defined Networking (SDN) [6], is an emerging new paradigm of network architecture primarily based on (i) a physical separation between the control plane (i.e., the functionalities which ensure the management of the network) and the data plane (i.e., the functionalities which assure the transfer of the data), and (ii) control and a logically centralized intelligence in one or more software controllers. In SDN, controllers have a holistic view of the entire network state and manage other network data plane equipment. These become simple transmitters/receivers of data with minimal intelligence. SDN promises to bring flexibility, scalability, and programmability to vehicle network architectures today. They also facilitate network management and introduce new services [6].

Architecture. SDN is based on a hierarchical three-layer architecture, see Fig. 1:

- Data plane layer: The data plane represents all of the equipment on the network, often called broadcast equipment, which only sends and receives data with minimal intelligence. The broadcast equipment also performs the actions of the controller.
- Control plane layer: The control plane represents all network equipment, often called SDN controllers, which centralize network intelligence and manage other data plane broadcast equipment.
- Application layer: it groups all the services and applications of the systems installed on the SDN controller.

2.2 Interface de Communication

To allow communication between the three planes, SDN defines several unified communication interfaces [6], see Fig. 1:

- North-bound API: it allows communication and data exchange between the SDN controller on the control plane and network applications. The type of information exchanged and its forms and frequencies, depending on each network application. There is no standardization for this interface.

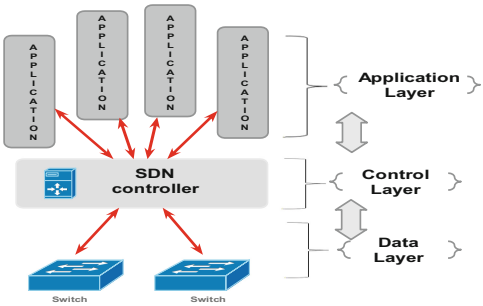

Fig. 1. Software-defined network (SDN) architecture

– South-bound API: it refers to the different APIs that allows communication between control plane equipment and data plane equipment. OpenFlow [6] is the most widely used standard for this interface.

2.3 OpenFlow

OpenFlow is a communication standard for SDN managed by the ONF (Open Networking Foundation) [6], a foundation of industrialists whose primary mission is to design and promote network equipment compatible with to facilitate its marketing. OpenFlow defines two types of network equipment: OpenFlow controllers, software that centralizes all network control functions, and OpenFlow vStwichs, which are virtual switches that only perform packet data transfer functions. Each vSwitch has a flow table containing controller flow entries. Indeed, the controller manages the vStwichs by installing flow rules in the flow table.

3 Related Work

We mainly focus on work involving the adoption of SDN as an architecture for vehicular networks. This paradigm mainly aims to separate the data plane from the control plane.

Based on the architecture proposed in [8,9] take advantage of the global view provided by the SDN Controller to calculate routing paths in a centralized manner. The simulation compares these approaches to traditional VANET routing protocols. The performance results show that these approaches outperform traditional approaches in terms of reliability and latency. This further demonstrates the value of integrating the SDN paradigm into VANET networks and seeing vehicles as nodes programmable via SDN.

To respond to scaling issues, especially in very dense environments, the authors [10] propose a decentralized SD-VANET architecture in which the control plane is distributed. As expected, the results show that the distribution of the control plane improves the scalability of the network (measured in the

number of requests of the data plane processed according to the density of the vehicles) while ensuring an acceptable delay of data delivery.

We cite the architecture proposed by Xiaohu Ge et al. [11] in which they consider a heterogeneous data plane following a hierarchical control plane. They propose using a fog cell based on multi-hop links where a vehicle functions as a gateway to minimize the frequent changes of RSU (handover) attachment points.

Jianqi Liu et al. [12] propose a heterogeneous vehicular network architecture based on control via SDN assisted by MEC (Mobile Edge Computing) to support the most demanding services in terms of latency. Two safety and non-safety use cases are simulated and the performance shows that the proposed architecture supports the expected prerequisites in terms of latency, reliability, and throughput.

Data to guide the control of the vehicular network has also been mentioned recently in the literature. We cite the positioning work [13] listing all the opportunities offered by the use of data for network control. They notably advocate the use of machine learning-based approaches to learn the dynamics of the vehicular network and perform effective network control.

In the same vein and the context of a vehicular network controlled via SDN, the authors focus on developing a specific network control function [14]. It is about the allocation of resources. They propose an approach that takes advantage of the global vision of the network. Their mechanism is based on machine learning techniques.

4 SDN-VANET: SDN-Based Vehicular Ad Hoc Networks

In line with the vision conveyed by our work, namely the proposal for an architecture of vehicular networks, we propose the adoption of the SDN paradigm as the main base of our architecture. We are taking advantage of the contributions of the SDN paradigm to facilitate management and improve VANET control. This paves the way for the development of new network control algorithms that take advantage of 1) insight into the state of various communication networks; 2) the ability to jointly control these networks dynamically and with scaling up (Hierarchical control); and 3) knowledge of the environment in which vehicles operate from the data that orbit this ITS system. This data can come from the various players in the system, for example, ITS operators, road authorities, etc. (Data-driven control).

Figure 2 illustrates the overall view of the proposed architecture. Indeed, integrating the SDN paradigm into the architecture consists of separating the control plane and the data plane. As shown in Fig. 2, SDN controllers hold the network's intelligence and enforce various network policies via specific protocols (e.g. an extension of the OpenFlow standard). The data plane nodes ensure the routing of data according to the instructions provided by the controllers.

We present in what follows the three fundamental design principles of our architecture, namely: 1) Heterogeneous data plane (vehicle, BS, sensor, Object IoT); 2) Hierarchical control plane and 3) data-driven plane.

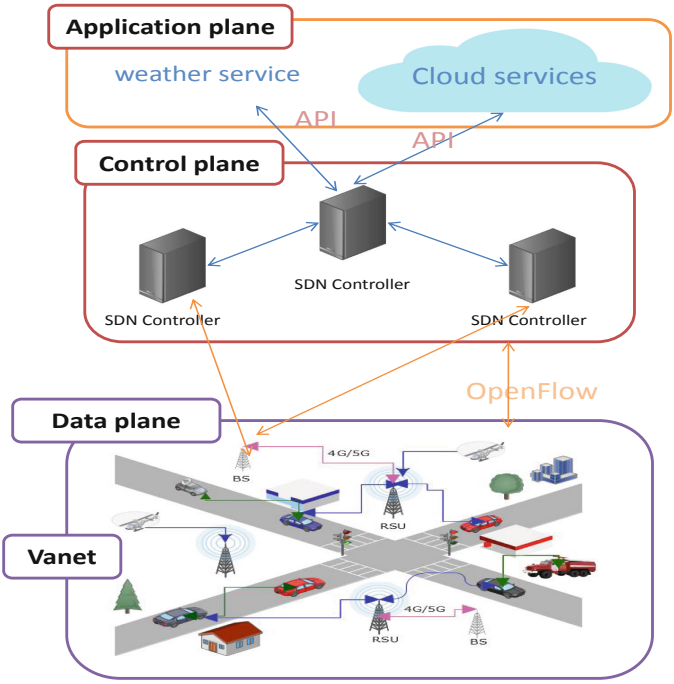

Fig. 2. SDN-VANET architecture

4.1 Key Principles of the Proposed Architecture

Heterogeneous Data Plane. As shown in Fig. 2, the data plane is made up of vehicles, RSUs, base stations (BS), temperature sensors, and IoT objects, all of which can be programmed via SDN.

However, we have chosen to consider the vehicles as programmable nodes via SDN. This choice is initially motivated by the first studies applying the SDN paradigm to VANET networks [2,3]. These studies show that routing is more efficient when calculated centrally in SDN controllers, compared to be distributed computing using conventional routing protocols (OLSR, AODV). We are also convinced that the global vision of the network offered by SDN reduces the interference and risk of collisions through topology control (e.g. transmission parameters, etc.). Therefore, we assume that the interface implemented by the nodes allows controllers to perform control beyond routing, for example, power control, choice of wireless channels, etc.

The data plane nodes are controlled in a transparent manner following a unified model such as OpenFlow (or a southern interface specific to the mobile wireless context), regardless of the technology or the manufacturer of the equipment. In fact, from an SDN controller point of view, the difference between the nodes of the data plane lies in their characteristics and the functionalities supported by each node.

The Data Plane Layer comprises all the network equipment that is solely responsible for collecting and transmitting information, RSUs, and all network equipment located between the cellular base station (BS) and the SDN controller. In our architecture, the data plane layer is divided according to the mobility of its components into two sublayers:

- Fixed data plane: it is said to be fixed because it is made up of static RSUs and all the fixed components of the network (sensors and IoT objects), ensuring the transmission of data between the base station and the SDN controller.
- Mobile data plan: it is said to be mobile because it comprises mobile vehicles, mobile sensors, mobile IoTs objects and Drones.

4.2 Hierarchical Control Plan

SDN controllers are the central part of the architecture. They host all the network control functions to define the various rules to be communicated to the data plane's nodes. They are connected to the various nodes via wired or wireless links, depending on their placement and the type of nodes (Vehicle, RSU, BS).

We consider two main types of controllers:

- Global SDN Controller: The Global SDN Controller has compelling storage and compute capacity and has a holistic view of the entire network topology. It only intervenes to process requests that require a global view of the entire network and/or very significant resources in terms of computation and storage. The global SDN controller is considered the master of all network SDN controllers.
- Core SDN Controllers: They have robust storage and compute capabilities. Central SDN Controllers handle specific requests requiring a central view of the network's state; large compute and storage resources, operations that are not time-sensitive, or local, lower level SDN controllers cannot serve that. A central SDN controller acts in terms of control as a slave to the overall SDN controller. The choice of organizing the control plane hierarchically is motivated by the vision conveyed by our work, namely joint control of these networks. In effect, the central controller builds a global view of the communication infrastructure using each network's information controller. It defines and sends to each controller the global rules that describe the network's general behavior and define the specific rules to be implemented by each node of the network.

4.3 External Data-Driven Control

SDN controllers can call on external data to perform more efficient network control. This data can come from external actors (e.g. weather service, road manager, etc.) and can be used to enrich the overall view of the network and derive a potential view of the state of the network. This allows for proactive/anticipatory network control. Figure 3 shows an example of data exchange between a few actors in the system and their use for network control and the design of ITS services.

The data plane nodes regularly send information to the control plane concerning their states and the characteristics of the links that connect them. This information can be coupled with external data to enrich the vision built by SDN controllers. We refer to the example of predicting potential network quality to perform anticipatory network monitoring.

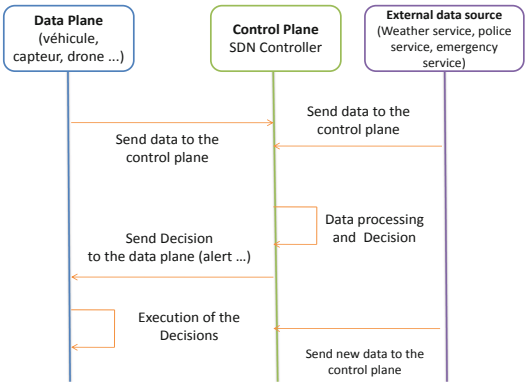

Fig. 3. External data-driven control

5 Simulation

The experiment aims to show how the global view of the network established at the controller level, combined and enriched with the external actors' data, allows a more informed and efficient network control to support ITS services effectively.

We show through evaluations how the SDN Controller can take advantage of its holistic view of current and potential network loads to guide the node in selecting the point of network attachment with the best expected performance.

We first present the simulation tools supporting the SDN-VANET architecture simulation prerequisites. We describe the tools chosen as part of our studies. Finally, we present the simulation results obtained.

5.1 Simulation Tools

To study, evaluate, compare our approach with the VANET without SDN, the simulation environment must combine both:

- Support for SDN programmable networks (e.g. via OpenFlow),
- A wireless communication medium (e.g. 802.11p/LTE vehicular communication standard),
- A vehicle mobility support.

To perform simulations requiring the use of a real SDN controller and/or the OpenFlow protocol, we opt for the choice of MiniNet-WiFi [15,16]. This tool supports the main prerequisite for simulating a vehicle network programmable via SDN. This is the implementation of the OpenFlow protocol and its integration with various nodes (RSU, vehicles, etc.), as well as the possibility of including real SDN controllers.

5.2 Description de la Simulation

The simulation is based on a simple scenario of a Vanet network. Indeed, we are simulating an SDN-VANET network with a topology deployed over an area of $12{,}000 \times 12{,}000$ m, which contains an area of the road not covered by the fixed infrastructure. A cellular base station (BS) is placed at the side of the road 1000 m from vehicles. The route is divided into virtual segments of size equal to 200 m each. The node density is 220 vehicles. Each vehicle has two wireless communication interfaces: a DSRC (IEEE 802.11p) interface with a transmission range of up to 300 m and a 4G cellular interface. The vehicles travel at speeds between 15 and 25 m/s. The simulation time is 300 s and the packet generation rate is 10 packets/s.

The performance metrics we used are:

- Flow rule installation delay: represents the time elapsed since a vehicle requests a new flow rule from the SDN controller and the moment when the flow rule is installed in the vehicle flow table.
- Average central processing unit (CPU) usage: Measured at the level of each vehicle.

5.3 Flow Rule Installation Delay

In this experiment, we study the impact of the distance to the SDN controller, which represents the physical distance between the SDN controller and vehicles, on the installation time of flow rules. So, we simulate a scenario where a set of vehicles send 120 packets to request new flow rules from an SDN controller, and we vary the distance between the vehicle and the SDN controller. We repeat this experiment for different package sizes and different numbers of vehicles.

The results in Fig. 4 and Fig. 5 clearly show that the average installation time of flow rules increases with the number of vehicles. This, perhaps justified by the fact that as the number of vehicles increases, the interference increases, and consequently, the waiting time increases. Also, as the distance and the packets' size increase, it is clear that the time required to transmit the end-to-end packets increases. The simulation clearly shows that our solution's application makes it possible to reduce the flow rule installation delay, unlike VANET without SDN.

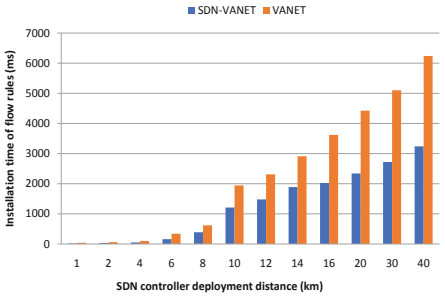

Fig. 4. Flow rule installation delay 80 vehicles

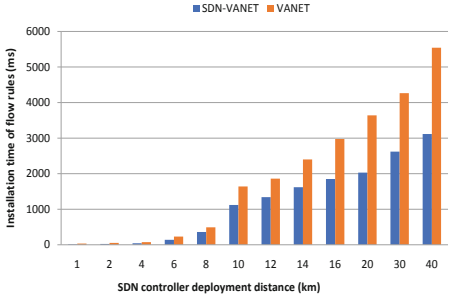

Fig. 5. Flow rule installation delay 120 vehicles

5.4 Average CPU Usage

Figure 6 shows the CPU usage, depending on the number of vehicles. We can see that CPU usage increases in proportion to the number of vehicles. The comparison shows that our method made it possible to considerably reduce CPU resource consumption compared to the VANET without SDN.

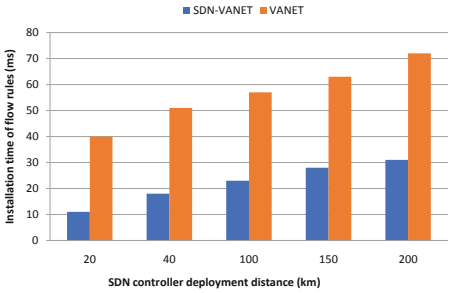

Fig. 6. Average CPU usage

5.5 SDN-VANET Contributions for Vehicle Networks

The main advantages of our SDN-VANET solution in-vehicle networks are:

- Facilitate efficient network management: thanks to the separation of control and data dissemination functionalities and the logically centralized control mode, the integration of SDN in-vehicle networks makes it possible to simplify the network architecture and offer more flexible management and faster configuration of the network. Indeed, instead of configuring and managing each vehicle individually, all network management is centralized in the SDN controller.
- More efficient use of network resources: With the real-time global view of the SDN controller over the entire network topology and the centralized control mode, it becomes easy to identify the current traffic state, which allocates all types of network resources more efficient (i.e., bandwidth, spectrum, transmission power, etc.).
- Facilitate scaling and reduce costs: SDN's logically centralized control mode makes it easy to deploy services and offer new features at scale with minimal cost. This, by only intervening at the SDN controller level instead of acting at each vehicle's level separately. For example, the SDN controller can easily handle a large-scale software update by simply installing entries in remote vehicle flow tables.

6 Conclusion

In this article, we have proposed a new VANET architecture based on SDN to benefit from the SDN approach and improve the VANET network's performance.

The simulations' results clearly showed the positive effect of using the SDN paradigm on the VANET network. This integration considerably reduces CPU consumption and Flow rule installation delay in the VANET network. Our solution represents an important step for the deployment of the VANET architecture in real Internet networks.

In future work, we will use the controller's controllability to enhance other features of VANET, such as intelligent control of network resources.

References

1. Bhatia, J., Modi, Y., Tanwar, S., Bhavsar, M.: Software defined vehicular networks: a comprehensive review. Int. J. Commun Syst **32**(12), e4005 (2019). https://doi.org/10.1002/dac.4005
2. Mishra, R., Singh, A., Kumar, R.: VANET security: issues, challenges and solutions. In: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). IEEE (2016). https://doi.org/10.1109/iceeot.2016.7754846
3. Al-Heety, O.S., Zakaria, Z., Ismail, M., Shakir, M.M., Alani, S., Alsariera, H.: A comprehensive survey: benefits, services, recent works, challenges, security, and use cases for SDN-VANET. IEEE Access **8**, 91028–91047 (2020). https://doi.org/10.1109/access.2020.2992580
4. Islam, M.M., Khan, M.T.R., Saad, M.M., Kim, D.: Software-defined vehicular network (SDVN): a survey on architecture and routing. J. Syst. Archit. 101961 (2020). https://doi.org/10.1016/j.sysarc.2020.101961
5. Zhang, Y., Cui, L., Wang, W., Zhang, Y.: A survey on software defined networking with multiple controllers. J. Netw. Comput. Appl. **103**, 101–118 (2018). https://doi.org/10.1016/j.jnca.2017.11.015
6. Singh, S., Jha, R.K.: A survey on software defined networking: architecture for next generation network. J. Netw. Syst. Manag. **25**(2), 321–374 (2016). https://doi.org/10.1007/s10922-016-9393-9
7. Kristiani, E., Yang, C.-T., Huang, C.-Y., Wang, Y.-T., Ko, P.-C.: The implementation of a cloud-edge computing architecture using openstack and kubernetes for air quality monitoring application. Mob. Netw. Appl. **26**(3), 1070–1092 (2020). https://doi.org/10.1007/s11036-020-01620-5
8. Ku, I., Lu, Y., Gerla, M., Gomes, R.L., Ongaro, F., Cerqueira, E.: Towards software-defined VANET: architecture and services. In: 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET). IEEE (2014). https://doi.org/10.1109/medhocnet.2014.6849111
9. Ji, X., Yu, H., Fan, G., Fu, W.: SDGR: an SDN-based geographic routing protocol for VANET. In: 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE (2016). https://doi.org/10.1109/ithings-greencom-cpscom-smartdata.2016.70
10. Kazmi, A., Khan, M.A., Akram, M.U.: DeVANET: decentralized software-defined VANET architecture. In: 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW). IEEE (2016). https://doi.org/10.1109/ic2ew.2016.12
11. Ge, X., Li, Z., Li, S.: 5G software defined vehicular networks. IEEE Commun. Mag. **55**(7), 87–93 (2017). https://doi.org/10.1109/mcom.2017.1601144
12. Liu, J., Wan, J., Zeng, B., Wang, Q., Song, H., Qiu, M.: A scalable and quick-response software defined vehicular network assisted by mobile edge computing. IEEE Commun. Mag. **55**(7), 94–100 (2017). https://doi.org/10.1109/mcom.2017.1601150
13. Liang, L., Ye, H., Li, G.Y.: Toward intelligent vehicular networks: a machine learning framework. IEEE Internet Things J. **6**(1), 124–135 (2019). https://doi.org/10.1109/jiot.2018.2872122
14. Khan, S., et al.: 5G vehicular network resource management for improving radio access through machine learning. IEEE Access **8**, 6792–6800 (2020). https://doi.org/10.1109/access.2020.2964697

15. dos Reis Fontes, R., Rothenberg, C.E.: Mininet-WiFi. In: Proceedings of the 2016 ACM SIGCOMM Conference. ACM (2016). https://doi.org/10.1145/2934872.2959070
16. Ghosh, S., et al.: SDN-sim: integrating a system-level simulator with a software defined network. IEEE Commun. Stand. Mag. **4**(1), 18–25 (2020). https://doi.org/10.1109/mcomstd.001.1900035