Weber *et al. Journal of the Brazilian Computer Society* (2021) 27:8
https://doi.org/10.1186/s13173-021-00113-x

Journal of the
Brazilian Computer Society

RESEARCH Open Access

VANET simulators: an updated review

Julia Silva Weber* †, Miguel Neves† and Tiago Ferreto†

*Correspondence:
julia.weber@acad.pucrs.br
†Julia Silva Weber, Miguel Neves and Tiago Ferreto contributed equally to this work.
School of Technology, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Ipiranga, Porto Alegre, BR

Abstract

Research on VANETs (vehicular ad hoc networks) date back to the beginning of the 2000s. The possibility of enabling communication between vehicles through a wireless network stimulated the creation of new protocols, devices, and diverse utilization scenarios. Due to the intrinsic difficulties of using a real testbed to evaluate these research contributions, several simulators were developed at the time. Recently, with the advent of autonomous vehicles and the emergence of novel technologies (e.g., 5G and edge computing), new research challenges on VANETs are coming into sight. Therefore, revisiting VANET simulators is required to identify if they are still capable of evaluating these new scenarios. This paper presents an updated review of VANET simulators, showing their current state and capabilities to assess novel scenarios in VANET research. Based on this analysis, we identify open research challenges that should be addressed in current and future VANET simulators.

Keywords: Vehicular ad hoc network, Simulation, SDN, Edge computing, 5G, Security

Introduction

Intelligent vehicles are a developing technology with promising future. However, to guarantee such technology to be safe, vehicles need to be able to communicate with each other and exchange information in real time. VANETs (vehicular ad hoc networks) were created to fulfill this necessity. VANETs are a special class of MANET (mobile ad hoc network) with predefined routes [1]. It allows vehicles to share information such as location, telemetry data, and safety warnings. VANET aims to ensure safe driving by improving the traffic flow and therefore significantly reducing car accidents. This is possible by providing appropriate information to the driver or to the vehicle.

Due to the easiness of embedding computers in vehicles, it is not far fetched to imagine in the forthcoming years that most of the vehicles will be equipped with an on-board wireless device (OBU), GPS (Global Positioning System), EDR (event data recorder), and a multitude of sensors. However, it is important to notice that, since VANET aims on ensuring the safety of its users on the road, any delayed communication or defective level of implementation may affect people lives. Therefore, any feature provided by a VANET protocol must be properly tested and validated.

Simulators present a valuable tool for testing VANETs at low cost and without risking the users. However, simulators must be able to model the novel technologies penetrating the VANET space (e.g., SDN, edge computing,and 5G) and provide support for safety and

security mechanisms in order to be useful and convey credible results. In other words, while simulators are a great tool for VANETs, they ought to improve to better support its evolution. Surveys on VANET simulators date back to the early 2010s [2–5]. They present an analysis over diverse VANET tools evaluating their accuracy as a mobility simulator, network simulator, and how these building blocks mend together. The emergence of new network technologies, such as 5G, SDN, and edge computing, and the reappearance of VANET research due to the investments on autonomous vehicles urge a reassessment of VANET simulators and their support for these new features.

In this paper, we provide an updated review on VANET simulators, presenting their main features and current support for novel technologies. In addition, we also analyze their footings for modeling important safety and security issues (and associated countermeasures) that have motivated extensive research in VANETs over the last years. To the best of our knowledge, this is the first study providing a detailed view on the aforementioned aspects in VANET simulators, and we hope it can motivate the community to develop further tools that assist VANETs towards widespread adoption. To take the first steps in that direction, we also list and discuss many research challenges we found during our investigations, which range from performance issues to lack of support for current standards. Overall, our main findings are as follows:

- Although many tools for simulating VANETs are available, most of them are outdated and not maintained anymore.
- Among the currently maintained tools (open-source and commercial), Veins [6] is currently the simulator with best support for modeling novel technologies and safety/security issues in VANETs. For example, the simulator contains pre-built extensions for modeling 5G networks, signal interference/attenuation, and privacy solutions.
- Despite the recent advances, current VANET simulators still lack support for more realistic models. For instance, none of them provide a full-stack implementation of the main security standards for VANETs (e.g., IEEE 1609.2) or offer any mechanism for systematically modeling faulty nodes (e.g., an unreliable RSU).

The remainder of this paper is organized as follows: The "VANET overview" section presents an overview of VANETs. The "Simulators" section describes the main building blocks of VANET simulators while the "VANET simulators" section shows an in-depth study of current tools. The "Support for novel technologies" section analyzes the support of selected simulators to novel technologies while the "Safety, security, and privacy functionalities" section does a similar analysis for safety and security issues. The "Research directions" section discusses some open research challenges. Finally, the "Conclusion" section concludes the paper.

VANET overview

VANET characteristics

As a collection of interconnected vehicles, VANETs present some unique characteristics not seen in other types of MANETs (e.g., smartphone-based ad hoc networks). First, deploying a VANET is usually expensive due to the fact that each VANET node (i.e., a vehicle) must contain a rich set of sensors (e.g., LIDAR and proximity sensors) as well as computation and communication resources (e.g., processors, memory, and

communication antennas) to analyze and exchange information [4]. Moreover, VANETs tend to primarily use short-distance communication (i.e., messages are typically sent when vehicles are close to each other), relegating long-range signals to some special scenarios (e.g., when vehicles need to communicate with road-side units in less populated areas). The life span of a VANET link is short as it is highly affected by the movement of vehicles. As a consequence, the network topology tends to change often and thus impose strict latency and bandwidth requirements for applications [7]. VANETs also have predictable mobility patterns as node movements are constrained by the road topology, and node locations must be very precise as any vaguely estimated vehicle location can put human lives in danger (e.g., by causing two vehicles to collide). Finally, VANETs have no issues with respect to power constraints as vehicles have the ability to provide a continuous source of power via long life batteries [8]. These characteristics enable VANETs to be used in a wide range of applications, including safe driving, improving passenger comfort and enhancing traffic efficiency [9].

VANET architecture

Vehicles participating in a VANET are equipped with a set of wireless sensors and On Board Units (OBUs). Those units allow wireless communication between the vehicles and their environment. These devices make each vehicle to act as packet sender, receiver, and router. It enables the vehicles to send and receive messages to other vehicles or Road Side Units (RSUs) within their reach via wireless medium [10]. The RSU, normally fixed along the road side, is equipped with one network device for DSRC (Dedicated Short Range Communication) based on IEEE 802.11p radio technology [11] and can also be equipped with other network devices for the purpose of communicating within the network infrastructure [4]. All vehicles move freely on the road network and mainly communicate within each other or with RSUs, as can be seen in Fig. 1.

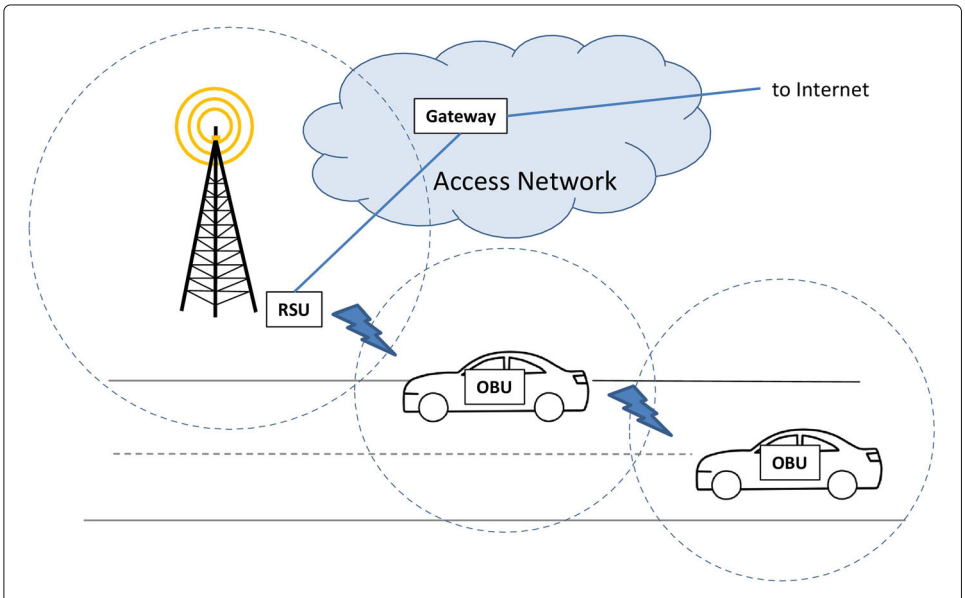

Fig. 1 Example of OBU and RSU at work. RSU work as information source and provides internet connectivity to the OBUs

Vehicles can communicate directly with each other using DSRC in a single or multi-hop way. The communication mode is either V2V (vehicle-to-vehicle), V2I (vehicle-to-infrastructure), or hybrid [12], as can be seen in Fig. 2. These vehicular communication configurations rely heavily on the acquisition of accurate and up-to-date kinematic data from both the vehicles and the surrounding environment with the aid of positioning systems and intelligent wireless communication protocols.

Simulators

Deploying and testing VANETs involve high cost and intensive labor. As an alternative solution, simulation is a useful and less expensive substitute prior to actual implementation. In order to achieve good results from VANET simulation, it is essential to generate accurate models, which is a non-trivial task given the complexities of the VANET infrastructure (e.g., simulators need to model both mobility patterns and communication protocols). In this section, we describe the main building blocks of current VANET simulators, namely their mobility and network components.

Mobility simulators. A critical aspect in a simulation study of VANETs is the need for a mobility model that closely reflects the real behavior of vehicles in traffic. Mobility simulators are mainly used to generate the movement of vehicles pattern under a certain trace. When dealing with vehicular mobility modeling, we distinguish between macro-mobility and micro-mobility descriptions [13]. For macro-mobility, simulators need to consider all the macroscopic aspects that influence vehicular traffic: the road topology, car movement constraints, speed limits, number of lanes, safety rules, and traffic signs governing the crossing rules at intersections.

Micro-mobility, on the other hand, refers to the drivers' individual behavior, when interacting with other drivers or with the road infrastructure: traveling speed in different traffic conditions, acceleration, deceleration and overtaking criteria, behavior in the presence of road intersections and traffic signs, general driving attitude related to driver's age, gender, or mood. An ideal VANET simulation should consider both macro- and

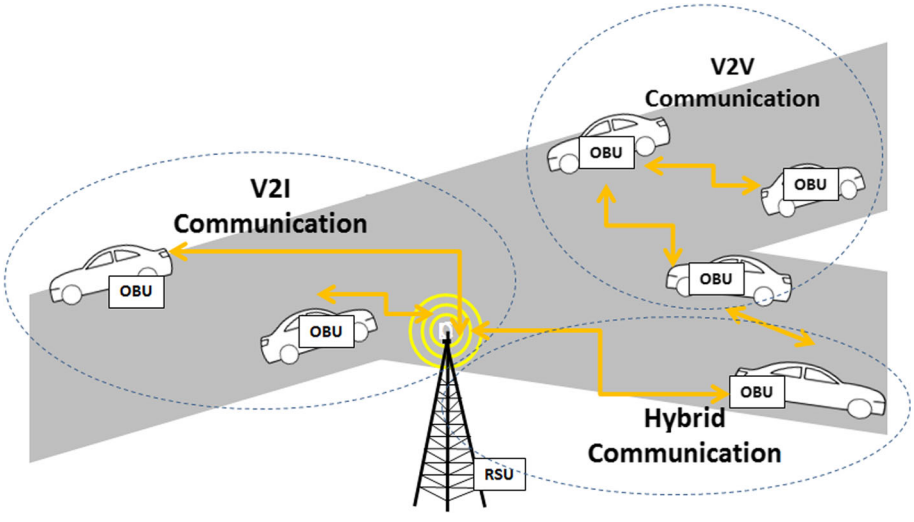

Fig. 2 Communication types in VANET

micro-mobility descriptions to be trustworthy. Examples of mobility simulators include SUMO [14], VISSIM [15], SimMobility [16], PARAMICS [17], and CORSIM [18].

Network simulators. A network simulator is used to simulate the exchange of messages among connected nodes. In the case of a VANET, this usually includes vehicles and RSUs and mostly involves wireless communications. Ideally, all components of the communication system (e.g., the whole protocol stack) must be modeled, and eventually, the simulation also includes other relevant metrics (e.g., signal to noise ratio, packet error rates) [19]. The network model describes both the network components and events. Nodes, routers, switches, and links are examples of components. Events, on their turn, can include data transmissions and packet errors.

For a given simulation scenario, the output from a network simulator usually includes network level metrics, link metrics, and device metrics. Trace files also use to be available. Such files record each event that occurred in the simulation and can be processed for further analyses. Most network simulators available are based on discrete-event simulation [5]. In this approach, a list of "pending events" is stored, and then processed in order at each simulation step. Some events may trigger new ones. For example, the arrival of a packet at a node may trigger the sending of a new packet. Examples of network simulators available (some of them widely used in VANETs) include OMNeT++ [20], OPNET [21], JiST/SWANS [22], NS3 [23], and NS2 [24].

VANET simulators

VANET simulators are the combination of network and mobility simulators [5]. Network simulators are responsible for modeling communication protocols and the exchange of messages, while mobility simulators are in control of the movement of each node, i.e., its mobility. In this section, we describe the main VANET simulators found in the literature, focusing on both their architecture and functionalities. We based our search on popular research databases and search engines (IEEE Xplorer, ACM Digital Library, Science Direct, Google Scholar, among others) and considered papers that propose a simulator or a comparative study of VANET simulators. In addition, we also carefully checked their citations. In total, we reviewed more than 100 papers during this process. We also used Google Search Engine to find proprietary VANET simulators that are not necessarily used by the academia.

Table 1 summarizes the simulators we found. Although most are open-source, we were also able to find some proprietary simulators. We focus our analysis on simulators that present a release after 2015 (highlighted with a gray background in Table 1). We consider that outdated tools are probably discontinued and thus have a high probability of not supporting the latest advances in VANET research, which is one of our analysis criteria in this paper. We refer to [4, 5] for a detailed analysis of older simulators.

NetSim

NetSim is a commercial discrete event simulator covering a broad range of wired, wireless, mobile, and sensor networks. The simulator offers three types of licences: pro, standard, and academic, where only the first two provide support for VANET simulations. To simulate VANETs, Netsim interfaces with SUMO. The former handles the WAVE standard for wireless communication between vehicles, while the latter takes care of modelling road traffic conditions. NetSim provides a set of network performance metrics, link and

Table 1 List of VANET simulators. The ones in gray (last release after 2015) are covered in detail in this work

Simulator	Last release	License	Network simulator	Mobility simulator
NetSim	2021	proprietary	own	SUMO
Veins	2020	open-source	OMNeT++	SUMO
Eclipse MOSAIC[1]	2020	open-source	NS-3, OMNeT++, SND and Eclipse MOSAIC Cell	SUMO and VISSIM
EstiNet	2020	proprietary	own	own
ezCar2X	2020	proprietary	NS-3	SUMO
VENTOS	2018	open-source	OMNeT++	SUMO
VANETsim	2017	open-source	own	own
GrooveNet	2013	open-source	NS-2	own
VNS	2012	open-source	NS-3, OMNet++	own
iTETRIS	2010	open-source	NS-3	SUMO
NCTUns	2010	proprietary	NS-2	own
CityMob	2009	open-source	NS-2	own
TraNS	2009	open-source	NS-2	SUMO
FreeSim	2008	open-source	NS-3	own
STRAW	2007	open-source	JiST/SWAN	own
VanetMobiSim	2007	open-source	NS-2	CanuMobiSim

application throughput plots. Metrics will vary depending upon the type of network simulated. Using packet trace and event trace, users can log details of each packet, as it flows in the network. Figure 3 presents a simplified version of the NetSim architecture. Each type of network corresponds to a component of the simulator with its respective communication protocols. Netsim provides a reasonable number of components and the possibility to connect real hardware running live applications to the simulation.

Within Netsim VANET modules, one of them worth taking note is RF Propagation Models. This module includes Path Loss, Shadowing Model, and Fading Model, which are essential to predict the signal loss or signal encounters inside a building or densely

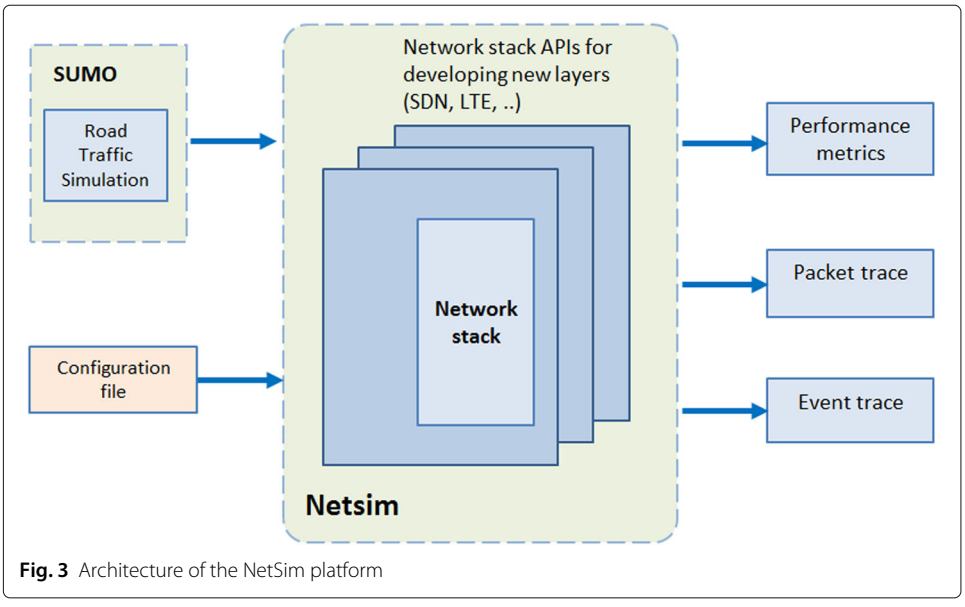

Fig. 3 Architecture of the NetSim platform

populated areas over distance. This module assists in making the simulation more realistic, as in a real scenario VANET will most likely encounter many obstacles (e.g., buildings, heavy traffic) for it signal communication. According to the architecture of Netsim, it allows adding new components to fit the user's needs and the appearance of new technologies. Technologies such as W-LAN, cognitive-radio, LTE, MANET, Military-radio, IoT, VANET, Software-Defined Networking (SDN), and satellite communications have been implemented on the simulator over the years. The SDN module, in particular, supports various networking commands to control simulation, routing, access control, etc., which can be executed on the controller command line during the simulation.

Veins

Veins [6] is an open source framework for running vehicular network simulations. It is based on OMNeT++ and SUMO. Figure 4 shows the different modules that form the Veins architecture. Overall, the simulator instantiates an OMNeT++ node for each vehicle present in the simulation and then pairs node movements with movements of vehicles in the road traffic simulator (i.e., SUMO). In this case, both the network and mobility simulations can run in parallel. This is possible due to a bidirectional coupling achieved by a standardized connection protocol, the Traffic Control Interface (TraCI) [26]. TraCI enables OMNeT++ and SUMO to exchange messages (e.g., containing mobility traces) while the simulation runs, as part of TCP connections [27].

The simulator includes many extensions (currently, more than 17 [28]) that allow modeling different protocol stacks (e.g., IEEE 802.11p [29], ETSI ITS-G5 [30]) as well as applications (e.g., car platooning [31]). In summary, Veins is designed to serve as an execution environment for user-written programs, which facilitates modeling new environments and applications. As a disadvantage, it needs both SUMO and OMNeT++ to run correctly in order to obtain precise results. Any bug in one of those can cause Veins to give unreliable results. Veins can run on Linux, Windows, and Mac OS.

Eclipse MOSAIC

Eclipse MOSAIC, formerly known as V2X Simulation Runtime Infrastructure (VSimRTI) [32], is an open-source, multi-scale and multi-domain simulation framework for the assessment of new solutions for connected and automated mobility. Eclipse MOSAIC

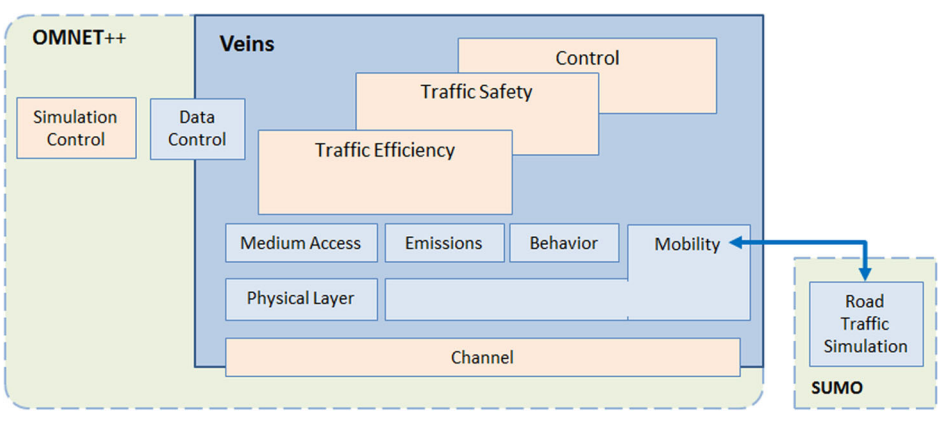

Fig. 4 Architecture of the Veins platform. Adapted from [25]

main objective is to provide users with the flexibility to perform various V2X simulations with their own choice of simulators. To guarantee that, Eclipse MOSAIC couples different simulators for a more realistic presentation of vehicle traffic, emissions, and wireless communication. Examples of simulators currently supported by Eclipse MOSAIC are SUMO and PHABMACS for traffic simulation; NS3, OMNET++, and SNS for communication simulation; and Eclipse MOSAIC Application for application simulation. Other simulators and analysis tools (especially those from third-parties) can also be easily integrated, as can be seen in Fig. 5.

To integrate the simulators to Eclipse MOSAIC, there are three core elements needed in the runtime infrastructure. The Federation Management is responsible to connect each participating simulator with the runtime infrastructure. A federate consists of an original simulator and two connectors, one to receive data from the runtime infrastructure and the other one to send data to it. The Time Management is necessary for coordinating the simulation and synchronizing participating federates. It assures that each federate processes its events in correct order. The Interaction Management enables the exchange of data among federates through a publish-subscribe paradigm.

According to the architecture of Eclipse MOSAIC, one of its unique features is the possibility to visualize data in several ways. The same scenario can be evaluated in different visualization tools that can be connected to a running simulation. Some of these tools are WebSocket Visualizer, Integrated Test and Evaluation Framework (ITEF), and PHABMap (for 3D visualization).

EstiNet

EstiNet [34] is a commercial network simulator and emulator with high time fidelity. EstiNet uses an innovative methodology, called kernel re-entering [35], to combine the advantages of both the simulation and emulation approaches. Basically, the kernel re-entering methodology uses tunnel network interfaces to automatically intercept the packets exchanged by two real applications and redirect them into the EstiNet simulation engine, as shown in Fig. 6.

VANET is an optional module add-on to EstiNet. To simulate vehicular traffic, EstiNet supports a road-building function, in which a road network can be built from scratch or

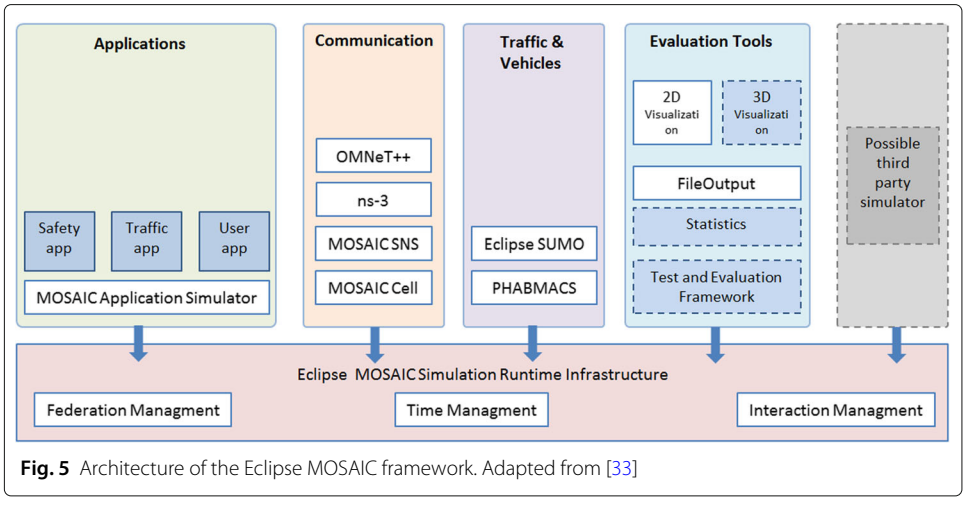

Fig. 5 Architecture of the Eclipse MOSAIC framework. Adapted from [33]

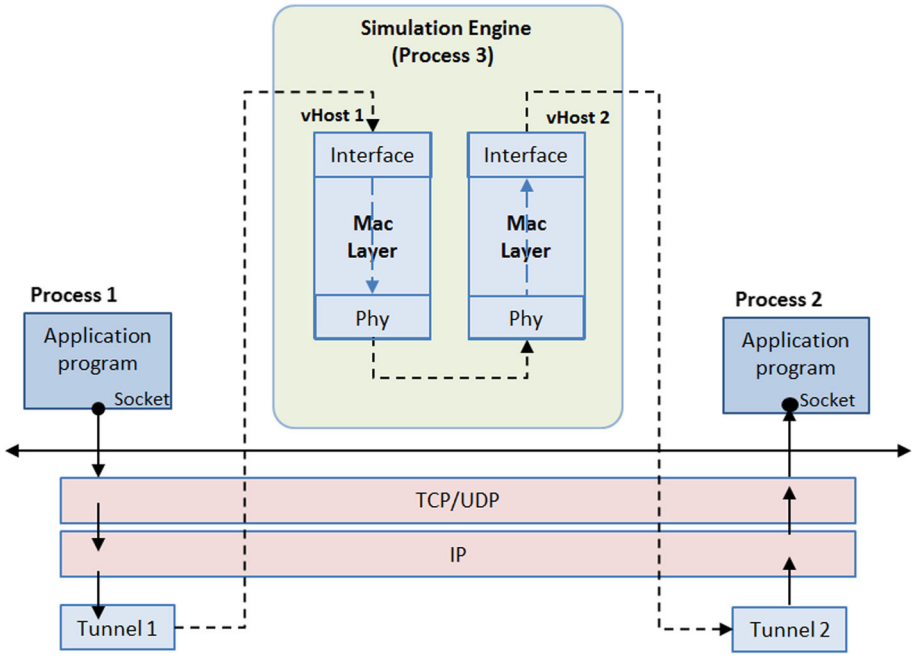

Fig. 6 Simulation architecture of EstiNet, host-to-host case. Adapted from [34]

by importing a roadmap file. EstiNet possess their own mobility simulator that allows for basic vehicle and human driving behavior, such as car following, lane changing, overtaking, and compliance with traffic light signals. As for protocols, IEEE 802.11p, IEEE 1609.3, and IEEE 1609.4 are supported by EstiNet.

EstiNet provides OBU and RSU abstractions in VANET simulation. To simulate OBUs, EstiNet provides different OBU communication interfaces. Each communication interface represents a specific communication behavior, such as agent-based vehicles (IEEE 802.11p/1609) and module-based vehicles (IEEE 802.11p) [36]. This feature in EstiNet allows for better vehicle driving intelligence implementation, as the user has more freedom to implement OBUs communication and behavior according to their needs. In the case of RSU, EstiNet provides two specific communications protocol stacks: IEEE 802.11p/1609 and IEEE 802.3.

ezCar2X

ezCar2X [37] is a modular software framework for rapid prototyping of cooperative ITS applications and novel communication protocols. Currently, ezCar2X simulator is proprietary, but planned to be made open-source in the near future [38]. It was created for Car2X communication [39] with the objective of facilitating vehicle manufacturers, suppliers, and road infrastructure operators to implement new applications and evaluate them in a simulation environment. ezCar2X is implemented in C++ with specific optimizations for efficient use of system resources. It also includes SUMO, which can be coupled with other simulators using its TraCI API [26].

ezCar2X architecture is based on the European Telecommunication Standards Institute (ETSI) architecture for Intelligent Transport Systems (ITS) stations [40], which defines

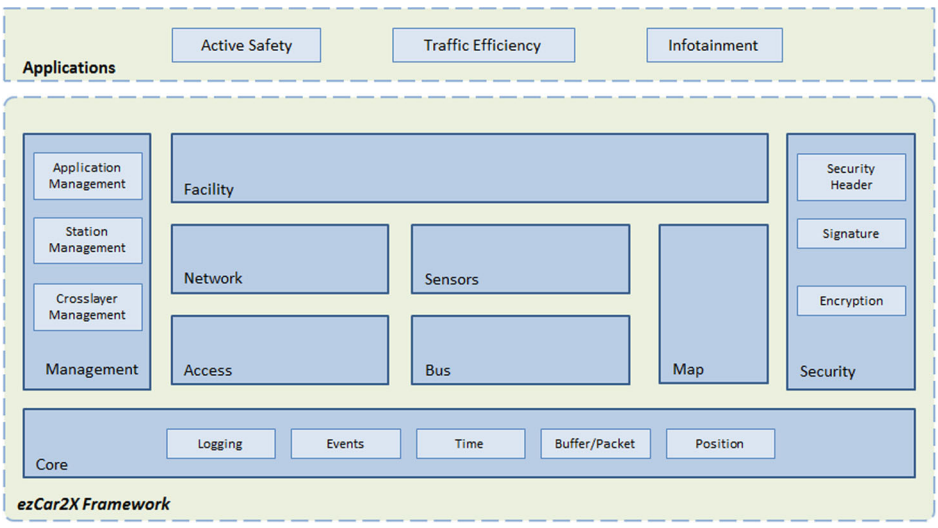

Fig. 7 Architecture of the ezCar2X framework. Adapted from [37]

access, network, facility, management, and security layers, as can be seen in Fig. 7. Regarding ezCar2X modules, its Core module provides a logging system as well as an event scheduler for asynchronous tasks and simple timeout realization. It also has modules to Access and Network that support ITS-G5 and 3G/4G, GeoNetworking, and the Basic Transport Protocol (BTP) according to ETSI standards. A security module provides an implementation of a network security entity to sign and encrypt transmitted messages as well as validate and decrypt received ones.

ezCar2X simulator can be executed on both Linux and Windows.

VENTOS

VENTOS [41] is an open-source simulator designed for analyzing vehicular network applications (e.g., collaborative driving, automated cruise control, and platooning). Similar to Veins, it also uses SUMO and OMNET++ for mobility and network modeling, respectively. However, unlike its counterparts, VENTOS has many prebuilt modules that facilitate simulating complex application scenarios. For example, the simulator offers implementations of traffic signal control (TSC) algorithms, as well as platoon management operations (e.g., merge, split, entry and leave).

VENTOS has two special modules that simplify the process of generating traffic demands at a microscopic level: *addNode* and *trafficControl*. The former allows users to easily add both fixed and mobile nodes to the simulation, while the latter enables controlling vehicular traffic by changing its speed or specifying platooning maneuvers. Figure 8 shows VENTOS architecture. The simulator can be expanded to interact with real OBUs and RSUs in a hardware-in-the loop (HIL) scenario [41]. In this case, each physical device connected to the computer has a corresponding virtual node in the simulation, and any action performed on the physical device is reflected on its alias and vice-versa.

The machine running VENTOS can be connected to the hardware device via an Ethernet port and communicate with it through SSH connections. The simulator requires support for a small control program (responsible for managing data and control instructions) to be run on the device though, which may hinder its integration with some

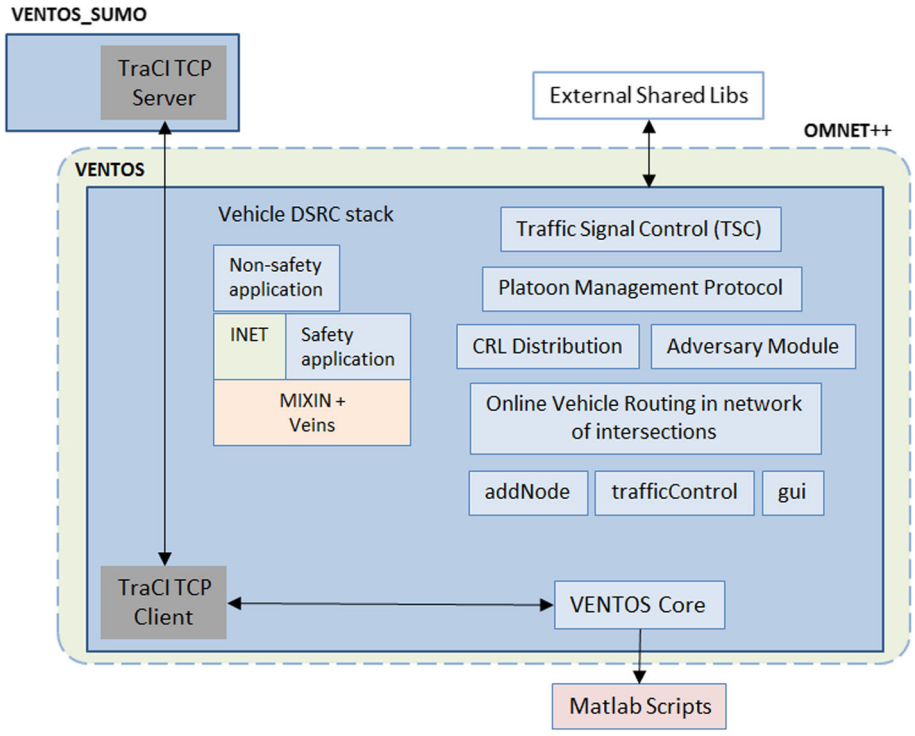

Fig. 8 Architecture of the VENTOS platform. Adapted from [42]

boards. Similar to Veins, VENTOS also depends on both SUMO and OMNeT++ working smoothly to get correct results. The simulator can run on Linux, Windows, and Mac OS based platforms.

VANETsim

VANETsim [43] is an event-driven simulator particularly designed to investigate security and privacy issues in vehicular communications. It allows analyzing attacks and countermeasures from an application perspective, e.g., by creating an attack and measuring its impact on different types of vehicles [44]. VANETsim architecture contains four main components: a Graphical User Interface (GUI), the Scenario Creator, the Simulation Core, and the Post Processing Engine, as can be seen in Fig. 9.

The GUI provides a graphical map editor that allows users to create and manipulate road maps. Maps can be either created from scratch or imported from OpenStreetMap [45]. It is possible to change imported maps and store them as XML files, which facilitates interoperability with other tools (e.g., TraNS [46], VanetMobiSim [47]). VANETsim interface aims to be accessible for users, with features to interact with it on-the-fly (i.e., while the simulator is running). It can also indicate the transmission range of vehicles, a functionality that can be activated on demand.

The Scenario Creator offers to users the ability to design a set of experiments and store their configuration in XML files. This feature facilitates the reproducibility of experiments as configurations can be shared online. The Simulation Core carries out the actual simulation. It coordinates the traffic map, network infrastructure, and all security and privacy modules specified. Finally, the Post Processing Engine processes generated logs

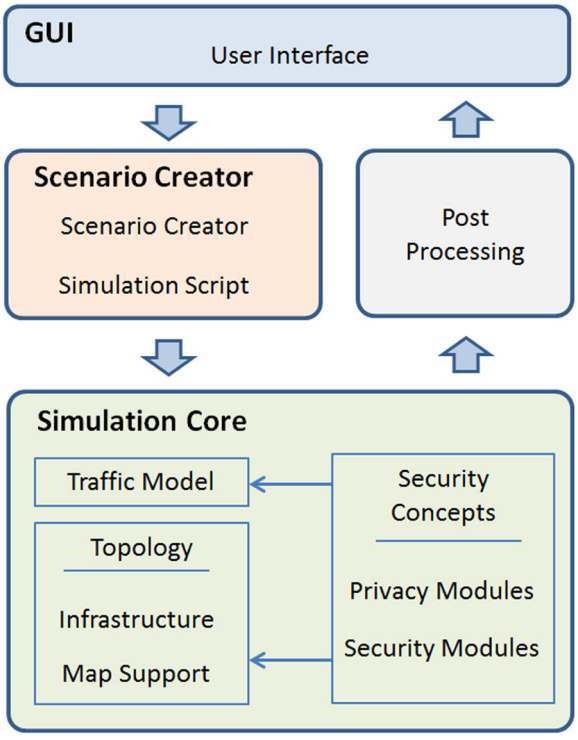

Fig. 9 Architecture of the VANETsim platform. Adapted from [43]

for exhibiting relevant events/metrics (e.g., showing compromised pseudonyms after an attack).

VANETsim comes with a few predefined security and privacy modules, which implement concepts such as silent periods [48] and Mix Zones [49]. Vehicles navigate to individually determined destinations routed by the A* algorithm [50], and communication among vehicles can contain two types of messages: *beacons*, which broadcast regular information (e.g., position, speed), and *special-purpose messages*, transmitted whenever a relevant event (e.g., an emergency vehicle approaching) occurs. VANETsim project was closed in April of 2017. However, despite the tool not being updated anymore, the simulator site [51] still features its documentation, downloadable content, and a very easy to follow guide about how to use the simulator. VANETsim is available for the Windows operating system.

Support for novel technologies

The number of new technologies penetrating the VANET space has been growing fast over the last years [52–54]. From autonomous vehicles (a.k.a. self-driving cars) to 5G, they play a crucial role to improve bandwidth, latency, and reliability of VANET applications, which enables their adoption in production environments. In this section, we analyze the current support of VANET simulators for various technologies.

Table 2 summarizes our results.

Software-Defined Networking (SDN).

SDN is a suitable solution for dealing with dynamic network environments, especially those with a large number of connected devices and heterogeneous applications [55].

Table 2 Support for novel technologies in current VANET simulators

Technology	NetsSim	Veins	Eclipse MOSAIC	EstiNet	exCar2X	VENTOS	VANETsim
SDN	X	X		X			
Edge computing		X			X	X	
5G	X	X	X	X		X	
Self-driving cars		X	X		X	X	
Unmanned aerial vehicles (UAVs)	X	X					

Interestingly, this is exactly the case for vehicular networks. SDNs are characterized by a separation between control and data planes, and the existence of a logically centralized network controller that coordinates the whole network operation [56]. Recently, researchers have focused on the integration of SDN and VANETs in the so called Software-Defined Vehicular Networks, which have spawn many initiatives in the area and made support for SDN in VANET simulators a must [57].

However, few simulators provide explicit support for SDN or are used in researches involving SDN and VANETs. NetSim version 11 provides a module that supports SDN and is OpenFlow compatible, thus making the use of such technology easy to be implemented with different types of network such as Internetworks, IoT, MANETs, VANETs, and LTE. In the case of VANETs, the RU is equipped with an SDN controller. In [58], the objective is to study the internal structure of network equipment models of the OmNET ++ modeling system, as well as create alternative models that take into account all the features of various software-defined equipment implementations. In this paper, NetSim is one of the simulators used to improve the simulation accuracy in terms of packet processing delay parameters.

When considering the Veins simulator, there are more papers related to SDN, such as [59–61]. In [59], SDN is used to provide a secure platform for VANET communication by creating a framework minimizing storage load, communication load, and response time. Veins is configured with OpenFlow [62] to support SDN controllers which interact with RSUs to control the entire network. Differently, in [60], SDN is used to offer an energy-efficient multicast routing protocol. In this case, Veins incorporate POX [63] to simulate SDN controllers. The SDN controller assists in sending data from the source to the set of target nodes in different locations. For example, when a vehicle exits the range from a formed VANET and enters another one. In [61], Veins is once again combined with OpenFlow to obtain reliable and fast emergency message dissemination in low RSU density areas.

Eclipse MOSAIC is a very powerful simulator used to model and assess new solutions for Cooperative ITS Systems. It can integrate several simulators which are individually used to model vehicular environment, communication environment, and social application environment. However, we could not find any significant research on SDN and VANETs using this simulator.

EstiNet provides an OpenFlow module as an original part of the simulator [64]. With this SDN module, a simulated OpenFlow-enabled Ethernet switch can support in-band control plane or out-of-band control plane through which is controlled by a single or multiple controllers. With EstiNet version 9, a VANET module was incorporated on the

simulator. EstiNet provides a good potential to make realistic scenarios using SDN, as presented in [65].

ezCar2x provides the key components needed to rapidly create prototype applications for networking vehicles. Due to a lack of access to this commercial tool, it is more difficult to identify if ezCar2x supports SDN technology. We also could not find papers using ezCar2x to test SDN scenarios.

VENTOS has many practical applications, ranging from studying platoon management [66] to studying security vulnerabilities of VANET-assisted cooperative driving; however, there are no researches related to SDN that use the simulator. Nevertheless, VENTOS should also be capable to use OpenFlow as OMNet++ supports [67].

VANETsim documentation does not present any information regarding SDN support. Besides, it carries all its components inbuilt, which makes it harder to extend the simulator to support novel technologies [43].

Edge computing. Edge computing is one of the approaches for controlling the huge volume of data being exchanged in VANETs. Existing solutions, such as cellular networks, RSUs, and mobile cloud computing, are far from perfect because these are highly dependent on centralized architecture and bear the cost of additional infrastructure deployment. Edge computing platforms, on the other hand, show the potential to replace RSUs as they support services and applications using extensively distributed deployments [68]. Edge computing is a topic that is being well explored in VANET research.

NetSim provides an IoT module, which has been used to simulate fog computing applications [69], and a cellular network module [70]. Both modules could be used in a edge computing scenario. It would enable the simulation of multi-access edge computing (MEC), which applies cloud computing technology to provide an application deployment platform closer to the end users. However, there are no public projects or researches that use Netsim to further evaluate edge computing. On the same note, Eclipse MOSAIC and EstiNet do not present significant contributions in this field.

Veins displays a number of researches on the topic [71–73]. Veins includes two-ray path models [74], which are applied in [71] to test a vehicular edge computing architecture. This paper uses vehicles as support infrastructure to form edge nodes to efficiently alleviate the bandwidth congestion by using both vehicles and road side units. In [72], edge services are co-located with RSUs in order to augment contextual information in real-time. In other words, in order to improve the offloading efficiency, the authors propose a new vehicle-to everything communication by adding a microcell in an RSU as an edge server that communicates with a macrocell server before reaching a cloud data center. In [73], the authors consider the task offloading among vehicles and propose a solution that enables vehicles to learn the offloading delay performance of their neighboring vehicles while offloading computation tasks. Veins uses an autonomous vehicular edge (AVE) [75] framework to enable V2V and V2I offloading.

ezCar2X is used by a project called Car2MEC [76]. The project aims to improve connectivity, especially for delay-sensitive traffic safety applications, by using local message distribution and processing based on MEC to improve communication latency for short-range information exchange via cellular communication. In [77], a MEC-enabled cooperative Collision AVoidance (CAV) is created to anticipate the detection and localization of road hazards by extending vehicles perception range beyond the capabilities of their own sensors. The CAV service is a software application that runs on MEC servers

allocated at the roadside and at mobile network infrastructures. The CAV service receives ETSI ITS-G5 standard-compliant messages transmitted by vehicles: periodic cooperative awareness messages, which include the position, velocity, and direction of the vehicle, and event-triggered environmental notification messages, which include the position of detected road hazards.

VENTOS main features focus on car-following models and dynamic traffic routing. As such, edge can be applied to secure traffic monitoring [78]. In this case, while VENTOS is used to create diverse road conditions, the centralized server (cloud) and the edge server are simulated in separate workstations. Basically, the edge server is associated to a certain region and is responsible to analyze traffic events (e.g., accident, congestion). With that in mind, it is possible to use the edge to assist in platooning, as the edge layer collects information about the platoon through beacon messages and observations from other connected entities. This is the case in [79], where the edge server is connected to RSUs and cellular Base Stations (BS) over broadband connections.

VANETsim focus on the implementation of security and privacy concepts, being mostly oriented towards the performance study of communication protocols on vehicular networks. This characteristic makes it harder to simulate newer technologies, due to the lack of detailed edge/network infrastructure implementation for application life cycle management, or detailed mobility models for realistic urban scenarios [80].

5G. 5G is meant to deliver high bandwidth and ultra-low latency network connectivity by using millimeter waves [81]. Comparing with 4G, 5G offers greater coverage, accessibility, and higher network density. Moreover, not only 5G presents a new access technology, but it also aims to provide a unifying platform that leverages the existing techniques offering a diverse set of services to customers. One of the services that would benefit from 5G is vehicles communication. With the current vehicular communication standards (IEEE802.11p) [82], there is no guarantee of service delivery in large-scale network deployments. With 5G, existing investments can be leveraged and its capabilities extended to guarantee a better performance for vehicular communications. For example, currently, 4G LTE [83] provides support for the integration of Wi-Fi and the unlicensed spectrum. With 5G, the capability will be extended to include 3G, 4G, Wi-Fi, ZigBee, and Bluetooth. This feature will enable vehicles and passengers to connect with the most suitable network to support the specific requirements of safety, non-safety, and infotainment applications [81].

NetSim provides a whole library to simulate the 5G NR standard [84]. The library is based on the 3GPP38 (3rd Generation Partnership Project) series [85] and enables the simulation of different devices (e.g., user equipment and next generation base stations) as well as protocol specifications (e.g., Packet Data Convergence Protocol, Radio Link Control). NetSim 5G library can also interface with the proprietary TCP/IP stack deployed in the simulator and provide simulation capabilities for 5G across all layers of the network stack. As an example on using NetSim to couple 5G and VANET simulations, the authors in [86] combined both capabilities to analyze different routing protocols for cooperative collision warning in underground mining scenarios.

Veins supports 5G through the INET framework [87]. The framework, which is an open-source library for OMNeT++, provides models for the TCP/IP stack as well as the 3GPP standard via Simu5G (an extension of SimuLTE) [88]. The works of Chekired et al. [89], Zhang et al. [90], and Huang et al. [91] provide good examples on how to couple

5G, SDN, and edge computing simulations in Veins. Eclipse MOSAIC can also support 5G simulation through Simu5G, as the tool can include OMNeT++. As an example on coupling Eclipse MOSAIC and 5G, the authors in [92] use VsimRTI (the predecessor of Eclipse MOSAIC) to test their proposed routing algorithm in a 5G-VMesh network (i.e., a hybrid network combining the features of VANET, mesh and 5G infrastructures).

EstiNet supports 5G network simulation by integrating the 5G core network stack deployed by the free5GC alliance [93]. The stack includes abstract modules to simulate simplified Radio Access Network (RAN) attributes and User Equipment (UE) behaviors [94]. The work of [95] claims that if VANET is made capable of relaying 5G, each car could act as a mobile cell tower in a downtown area. EstiNet is then used to compare their performance in urban dense regions that have better conditions like slow-moving and densely packed vehicle traffic. ezCar2X, according to the official documentation [37], only allows for 3G/4G connections with its access module.

5G technology can be implemented in VENTOS, however not to the same depth as in Veins. The study [96] uses VENTOS to simulate 5G communication between the platooning to implement platoon maneuvers (e.g., merge, split, and lane change). 5G can be used in VENTOS by applying Simu5G [97].

Self-driving cars. Autonomous vehicles (AVs) are defined as vehicles that have the capability of sensing and navigating their environment without or with minimum human input. However there are different levels of automation which classify autonomous vehicles. Currently, there are 6 levels of automation, with level 0 being no automation and level 5 being full automation. What lies in between those levels are different driving modes that increasingly aim for full automation [98]. For example, level 1 consists of minor driver assistance with a high degree of human input, while level 4, high automation, the driving system does most of the driving tasks with little human input. Regardless of the level (excepting level 0), AVs can benefit VANET by improving network capacity, keeping traffic flow steady, and taking into account faster responses and more tightly spaced vehicles [99]. In regard to adding AVs to VANET simulation, no significant change would have to be done to emulate a self-driving car. All simulators are already capable of supporting AVs to some capacity. The changes more relevant to properly accommodate AVS are related to mobility simulators. In this case, the different levels of automation need to be taken into consideration for the simulation vehicles to behave accordingly to each level. This can be accomplished by applying different percentage of both conventional cars and full automation AVs. Basically, the parameters of each vehicle need to considered different longitudinal movement, acceleration, deceleration, and gap acceptance. The mobility simulator, however, needs to consider both macro and micro models to properly simulate AVs behavior, something that can be accomplished in SUMO [99]. To be able to simulate AVs, SUMO is modified to permit USARsim [100] to be integrated with its architecture. For simplification, SUMO simulates the traffic while USARsim controls the robotic simulator that is responsible for the autonomous driver agent. For further details, all modifications and methodology are explained in [101]. With SUMO being able to accommodate AVs in their mobility simulator, Veins, VENTOS Eclipse MOSAIC, and ezcar2X are able to support AVs experiments, as it can be seen in the works of [102–104] for Veins, [102, 104] for VENTOS, [105] for Eclipse MOSAIC, and finally [106] for ezcar2X.

Unmanned aerial vehicles (UAVs). Unmanned aerial vehicles (UAVs), due to their ability to move in three dimensional space and reach high altitudes, present great flexibility

in creating a networked environment. Due to their mobility, UAVs can be deployed as mobile infrastructure elements to provide service to vehicles. For example, in vehicular communication scenarios, there can be some cases where direct multi-hop car-to-car communications are not reliable at ground level, such as a rural area with lots of hills. The possible solution for cases, like the one mentioned, would be to deploy the UAVs to forward the information related to car-to-car communication, acting as information relays [107]. As UAVs can adjust their position dynamically if they want to offer the best signal coverage to ground vehicles, this technology could greatly benefit VANETs.

Simulators usually deal with a 2D approach [2]. Yet, with technologies such as UAVs, it is important to consider the effects of a three-dimensional scenario and how it could improve VANET simulation. However, in order to achieve so, the simulation framework has to be extended. Generating 3D road networks and providing an extension to 3D network simulation would be the basic requirements to simulate three-dimensional vehicular networks [108]. Veins, while not presenting any current module for 3D support, is able to create 3D scenarios with the inclusion of Digital Elevation Models (DEMs) [109], three-dimensional antenna patterns , and environmental diffraction [108]. In addition, [107] presents a 3D mobility model algorithm as a module for Veins, specifically allowing UAV communications in a 3D environment. Fundamentally, UAV movement is computed in OMNeT++, while the signal strength is affected by the elevation obtained from DEM. As for signal blockage, Veins utilize a path loss model from the Two-Ray Interference module.

NetSim can be used to simulate unmanned aerial vehicle (UAV) communication [110]. It allows co-simulation of UAV flight dynamics and UAV-BS network communication. Basically, it involves interfacing NetSim and UAV toolbox of MATLAB [111]. For each drone/UE in NetSim, an UAV is instantiated in MATLAB. MATLAB then calculates the flight path and passes the mobility information to NetSim. However, while VANET is not directly correlated to UAV communication, NetSim provides the source-code for user modification, which could be used to future experimentation in VANET cases.

Both Netsim and Eclipse MOSAIC, despite having useful elements that seem promising for combining UAV communication with VANET, present a lack of research in this area. Although Netsim is capable of simulating VANET and co-simulating UAV, no research combining both elements using this simulator was found. Eclipse MOSAIC, in contrast, has a 3D visualization tool. This tool is based on the PHABMACS vehicle simulator and uses the same 3D engine and models to visualize vehicle movements and various events which occur during the simulation. Due to the main characteristic of Eclipse MOSAIC that enables combining simulators, future works for visualization of 3D UAVs with Veins and Eclipse MOSAIC are a possibility. However, Eclipse MOSAIC does not possess any module to test UAVs experiments, and no further studies about the use of UAVs in Eclipse MOSAIC were found.

We could not find any module or extendable framework to support 3D environments in VANETsim, VENTOS, EtiNet, and ezCar2X.

Safety, security, and privacy functionalities

As important as modeling novel technologies, VANET simulators need to provide support for testing safety, security, and privacy issues (and associated countermeasures) in vehicular networks. In this section, we analyze which functionalities they provide for reaching each of these goals.

Safety

One of the main motivations behind VANET development is to improve road safety. As a result, significant effort has been made to create high-performance applications that utilize connected vehicles to exchange safety messages (e.g., collision avoidance and hazardous spot detection) [112, 113]. At the same time, VANETs themselves have to meet stringent reliability requirements to properly deliver their critical services, which demands fault tolerance techniques (e.g., heartbeats, information redundancy) to be applied [114]. In this section, we start by analyzing the support of current VANET simulators for testing road safety applications as well as VANET safety techniques. We base our discussion on the reviews provided by [115–117], although their focus is on VANET safety and reliability issues in general, not on the simulator themselves.

Road traffic safety. VANETs can increase road safety by sharing information about both vehicles (e.g., position, speed and direction) and traffic conditions (e.g., accidents, jams, aquaplaning) as *beacons*. Safety messages, in this case, are the key information to avoid accidents. However, collisions can occur when safety messages and transmission of packets are improperly broadcast from multiple vehicles. The study of [118] deals with the exact issue as it proposes a Novel Segment based on safety message broadcasting in Cluster (NSSC). Basically, the VANET simulator incorporates three successive processes that are cluster formation, collision avoidance, and safety message broadcasting. The latter focus on mitigating broadcast storms, so safety messages are not lost. On a different note about safety, [119] considered that context awareness in vehicular re-routing is essential, since drivers can have different preferences when choosing their routes. Safety is one of the parameters that is considered (e.g., route passes through a good neighborhood have the less street bump). In this work, the simulator is used to create a realistic traffic mobility and to implement a non-deterministic multi-objective re-routing approach to reduce public safety risks. Lastly, the study [120] proposes an effective delay-aware packet forwarding (DAPF) for safe and efficient driving in vehicular networks. In this work, the VANET simulator is used to provide a more realistic environment compared to the mathematical analysis.

VANET safety. VANETs themselves are susceptible to failures which is particularly dangerous for critical applications. As a consequence, ensuring VANETs can operate safely is of utmost importance for their widespread adoption. According to Dharmaraja et al. [116], the reliability of a VANET depends on how reliable are both its end nodes (OBUs, RSUs, etc.) and communication channels. Nodes can fail due to a broad range of reasons, both at hardware and software levels. Hardware faults can include power outages, damaged sensors, and/or malfunctioning antennas, while software faults typically include protocol, firmware, and operating system bugs [121].

Usually, node failures manifest as silent entities, either permanently or in transient/intermittent intervals depending on the fault type. For example, a system crash may stop the node from working permanently, while a malformed packet will simply forbid it from properly communicating the associated message. VANET nodes are also subject to Byzantine faults, in which a node forwards correct but misleading (e.g., false) information to a peer [122].

Similar to nodes, links are also subject to different types of faults in VANETs. According to Albano et al. [117], two of the most common types of link faults are interference and signal attenuation. The former frequently results from congestion and/or ground

reflection [123], while the latter is typically a consequence of buildings close to the road or vehicles moving away (e.g., exiting a freeway) [124].

Table 3 summarizes the current support of VANET simulators to different types of faults. As can be seen, VANETsim does not provide any feature for simulating faults, though some functionalities (e.g., silent periods) could be adapted to model limited types of node and link failures [43]. Unfortunately, that may require generating a large amount of code, as the simulator was not designed to support generic applications. NetSim possesses an RF Propagation module which includes path loss, shadowing, and fading models [69]. Together, these models enable the simulation of a rich set of interference and attenuation link faults.

Veins is able to simulate Byzantine faults through its Framework for Misbehavior Detection (F2MD) [125]. This framework provides a solution for simulating malfunctioning nodes that produce erroneous information (e.g., inaccurate position, velocity, and acceleration for vehicles), as well as misbehavior detection algorithms (e.g., based on local plausibility checks) [126]. The simulator also carries two modules, Obstacle Shadowing [127], and Vehicle Obstacle Shadowing [128], which can capture the effect of buildings and vehicles, respectively, on the quality of data transmissions. Together with the Two-Ray Interference module [129], which simulates signal interference due to ground reflection, these modules can be used to induce link failures in a simulation.

There is no current support for fault injection in Eclipse MOSAIC. The closest it gets, to test similar fault tolerance scenarios, is through the application simulator. The application simulator in Eclipse MOSAIC provides the capability to model the application logic for different simulation units (e.g., vehicles, Road Side Units (RSUs), traffic lights, and others), as well as possible interaction attempts between the units via different communication links [33].

EstiNet does not have any apparent support for fault injection or any modules that can be used to facilitate testing of fault tolerance. It is worth noting that EstiNet does allow for different settings and definition of vehicle driving behavior [64]. The user could indirectly cause a car accident to test the safety mechanisms of VANET. This is possible in EstiNet, since the user can implement a car profile method, which enables each vehicle to possess an unique driving behavior.

ezCar2X has a Bus module that supports basic serial interfaces (RS232, USB) and controller area networks [37]. It is mostly used to integrate sensors and actuators within a vehicle or along the road. However, an additional abstraction layer is provided by the Sensor module to enable reuse of algorithms based on sensor input across different equipment types. The library provides several basic types, such as position, speed, acceleration, and object detection (radar or laser scanner). Furthermore, self-description of a specific sensor adaptation provides properties, like accuracy, that vary among different devices of the same category. However, there is no publication indicating this feature can be used to inject faults in OBUs.

VENTOS enables the detection of bad behavior in small and specific scenarios by implementing extensions such as another programming language [79] or software environment [130]. In [131], time series analysis and misbehavior detection schemes are implemented in Python [131] and integrated with VENTOS. [130] uses VENTOS to collect the speed value observations (e.g., exchanged beacon messages), while those observations are analyzed with R [132] separately. By analyzing the data, it is possible to detect abnormal

Table 3 Current support for fault injection in VANET simulators

VANET component	Fault type	Example	NetSim	Veins	Eclipse MOSAIC	EstiNet	ezCar2X	VENTOS	VANETsim
Node (e.g., OBU, RSU)	Software	Malformed packet, OS bug							
	Hardware	Power outage, malfunctioning antenna							
	Byzantine	False messages		X				X	
Link	Interference	Congestion, ground reflection	X	X				X	
	Attenuation	Building shadowing, vehicle moving away	X	X				X	

behavior. In both cases, the simulation scenario is based on platooning and the misbehavior detection is based on the input of vehicle's speed acceleration. Lastly, VENTOS, although not possessing any modules for transmission interference, does possess the ability to change the Transmission Power and Data Rate of each V2V or V2I communication [25]. Basically, while having more limited options, such as not being able to create obstacles to interfere with the transmission, it is still possible to variate the transmission range to create faulty links communication.

Fault tolerance. Although many fault tolerance mechanisms have been proposed for VANETs, almost none of them were implemented on top of the simulators studied in this paper (an exception is the work at [133]). Moreover, we were not able to find enough information about the support for fault tolerance on the simulators' documentation. For this reason, we avoid making a detailed assessment on the support of each simulator for fault tolerance techniques, and leave that as a future work. We refer the interested reader to the work of Almeida et al. [115] for more information about fault tolerance in VANETs and how these mechanisms are evaluated.

Security and privacy

VANET security greatly depends on the security of the exchanged messages (i.e., the delivery of messages should be secure and fast). In this sense, the exchanged messages must be not modified or captured by any malicious party. Although the use of simulators alone cannot solve any of these issues, they could be used to further explore security concerns in order to find possible solutions without putting any human life at risk.

Table 4 shows diverse security mechanism and their relation to the studied VANET simulators. The table is organized as follows: it establishes what security service these mechanisms belong to, and whether VANET simulators have the means to include them in their simulation. We base our discussion on the works in [12, 134, 135], which report general security and privacy issues in VANETs.

Confidentiality. Responsible for preventing data from being accessed by unauthorized nodes, confidentiality is essential to VANETs as it determines a set of rules or a promise that limits access to classified information [12]. Support for confidentiality services vary according to the simulator. NetSim, being a commercial simulator, does not disclose the details of its cryptographic support. However, it allows Key Management Scheme based on symmetric and asymmetric encryption to be used on its simulations, as seen in the work [136]. Veins, VENTOS, and Eclipse MOSAIC do not provide confidentiality mechanisms by default, but allow their usage by means of importing cryptographic libraries (e.g., Crypto++ [137]) to their network simulator, OMNeT++. ezCar2X has a security module [37] that provides implementation of a network security entity for signing and encrypting messages to transmit, as well as validating and decrypting received messages. Although VANETsim aims at providing easy support for testing novel security and privacy mechanisms, no real encryption of messages is performed (or supported). Instead, each message contains a boolean attribute that indicates whether it is "encrypted" or not. VANETsim assumes that adversaries cannot break the "employed" cryptographic primitives [43].

Integrity. Ensures that the data exchanged in a VANET is not altered by unauthorized third parties. This service is usually associated to mechanisms such as message authentication codes and digital signatures, which prevent attacks such as message tampering, forgery or replay. Similarly to confidentiality, Veins, VENTOS, and Eclipse MOSAIC do

Table 4 Current support of VANET simulators to different security services

Attribute	Example mechanism	NetSim	Veins	Eclipse MOSAIC	EstiNet	ezCar2X	VENTOS	VANETsim
Confidentiality	Symmetric and asymmetric cryptography	X	X	X		X	X	
Integrity	Message Authentication code, Digital signature		X	X		X	X	
Availability	Watchdog, Redundancy		X	X	X		X	
Authentication	Position verification, Analyze signal strength, Dual Authentication	X	X	X	X	X	X	X
Non-repudiation	ID-Based cryptosystem		X					
Privacy	Silent periods, Mix Zones, Periodical pseudonym change		X					X

not have native support for integrity services, but allow them to be included as third-party libraries. ezCar2X security module also allows the management of available certificates and regular pseudonym updates. VANETsim, on its turn, adopts the concept of detecting false data in VANETs by identifying its associated malicious actor (e.g., an inside attacker) through an intrusion detection system (IDS). In this case, the IDS monitors application layer data (e.g., position and time) to perform a context verification. VANETsim offers a general-purpose IDS module that can be configured with different rule sets depending on the particular application [43].

Availability. Availability in VANETS should be assured both in the communication channel and in participating nodes. Although being an important security attribute, it is one of the lessen researched topics [138], as the solutions for it often falls into the fault tolerance field. It is possible to implement some security mechanism to test, to some extent, availability issues on VANET. In the case of implementing a watchdog on the ad hoc network, it is made possible for Veins, VENTOS ,and Eclipse MOSAIC by incorporating this mechanism on the Network simulator [139, 140]. Basically, this watchdog defines one set of nodes as monitor nodes. Those nodes then check they neighborhood nodes. Redundancy is also a mechanism that can be implemented in different ways, such as equipment redundancy , for instance by adding more sensors to OBUs. Redundancy can also be used in other security mechanisms, as it can reduce attacker influence, since fewer redundant copies of attacker-influenced information will be received [141]. EstiNet allows the users to modify their OBUs when creating their communication behavior [64], which also consists of changing the number of sensors in each OBU. As another example, redundancy can be used in implementing new protocols for VANET, such as the work of Achour [142] where a novel density based dissemination protocol called "Redundancy Based Protocol - RBP" is created. Considering how redundancy would affect the simulations, Veins, VENTOS, and Eclipse MOSAIC would be ideal to test protocols that use redundancy, as it just needs to be implemented in the network simulator.

Authentication. In addition to message/data authentication (which is closely related to integrity), VANETs also require authentication of origin to be provided. Authentication mechanisms must be designed to protect VANET nodes from impersonation attacks. ID-Based cryptography is one of the main mechanisms to solve authentication issues [135]. The main idea is to use any known information which represent the identity of the user for the purpose of verifying the digital signature. This public information could be email address, network address, user name, or any combination of these identities. When considering simulators, Veins is the only one that supports such mechanism [143]. The main advantage of using the ID-Based cryptosystem is that it uses pseudonym generation, which can be changed as required for security purposes. ID-based techniques could be an efficient replacement for the PKI technique in VANET environments, since it is not required to store, fetch, and verify the public key certificates of some road safety scenarios by a trusted third party.

Non-repudiation. The main goal of non-repudiation is to forbid an entity (e.g., a car) from being able to deny an action. Non-repudiation of Origin (NRO) and Non-repudiation of Receipt (NRR) are the most common examples in computer networks, yet both services are different by nature and so are their implementing mechanisms in VANETs. While NRR has not been extensively explored in VANETs, NRO is traditionally implemented in VANETs using digital signatures [144]. In the case of each simulator, the

network simulation, OMNET++, VANETsim, NetSim, EstiNet, and ezCar2X inbuilt network simulators [37, 43, 64, 69], respectively, would be the ones responsible with ensuring digital signature. In the case of OMNET++, it is possible to deal with digital signatures and even modify them to better satisfy VANET characteristics [145]. Non-repudiation is also needed for the sender in V2V warnings and beacons. In this way, if a vehicle sends some malicious data, there will be a proof that could be employed for liability purposes [144]. In this case, it is confirmed that Veins [146], VENTOS [147], and VANETsim [148] at least guarantee non-repudiation of origin, so wrong warning messages can be undoubtedly linked to the sending node.

Privacy. Privacy is one of the most important requirements for VANET security [135]. Malicious attacks to security do not only aim to potentially cause harm to the traffic, but also to obtain personal information about the user. It is responsible for hiding the identity of the user against unauthorized nodes using temporary and anonymous keys. Other than personal data, privacy should also guarantee location privacy, so no attacker can track the trajectory of any node.

Veins, on the other hand, has a privacy extension (PREXT [149]) which comes equipped with a diverse set of predefined privacy schemes (e.g., periodical pseudonym change, PeriodicalPC, cooperative pseudonym change based on the number of neighbors, CPN [150], and context aware privacy scheme - CAPS [151]). PREXT supports simulating an adversary who aims at tracking vehicles by eavesdropping beacon messages. Although PREXT does not represent a significant overhead for simulations involving low vehicle densities, simulations can be up to 30% slower when the module is active depending on the adopted privacy scheme. VANETsim implements a number of privacy concepts for VANETs, including MixZones [49], SilentPeriods [48, 152], SLOW [153], and ProMix [154], which provide unlinkable pseudonym switchover via radio silence and/or encryption. It can also be used to simulate attacks against these concepts through the addition of new adversary modules [43]. The remaining studied simulators, VENTOS, NetSim, Eclipse MOSAIC, EstiNet, and ezCar2X, do not currently support any privacy service [41] or due to their commercial nature, makes it difficult to investigate further.

Research directions

Although current VANET simulators have a great number of functionalities, we were able to identify important issues with respect to their support for novel technologies, as well as safety and security mechanisms. Solving these issues turns out to be interesting research directions that we believe could be explored by the community to develop better VANET simulation tools. In this section, we describe some of these issues.

Implementation of security standards. Over the last years, there has been a considerable amount of effort to develop security standards for intelligent transportation systems, and in particular VANETs. As a result, we currently have two major standards: IEEE 1609.2 [155] (USA) and ETSI ITS Security Standards [156–161] (Europe). Although Veins and VENTOS support implementing parts of the proposed standards (e.g., the recommended cryptographic algorithms) using library extensions, none of the VANET simulators we found is currently compliant to them. Ultimately, this forbids researchers and practitioners from comparing their novel security proposals with the *status quo*. We believe that extending the current simulators to support the proposed security standards

is not a trivial task and will require certain systems and programming research to produce correct and efficient artefacts.

Systematic fault injection. Fault injection consists in the observation of the system behavior in response of deliberately introduced faults [162]. It is adopted to perform either robustness testing or dependability evaluation. It allows testing the fault coverage of fault tolerance strategies implemented in a system. It also can determine the possible ways a system fails in presence of rare or unexpected faults (e.g., transient and permanent faults, arbitrary faults, hardware faults, and data faults) [163]. It is generally applied during the development of a system. However, each system relies in its own ad hoc injection solution and no general tools for fault injection on VANET has been proposed to date. An appropriate fault injection tool could be used to verify and validate VANET operation in the presence of possible faults without having to wait for those faults to occur in a real scenario. This could lead to more reliable systems and also to dependability benchmarks that could be used to compare the safety of different solutions [164].

Real-time simulations. The coupling of real-time systems with non-real-time event-based simulation in the so called Hardware-in-the-loop (HIL) scenario brings new challenges. In particular, current simulators cannot meet performance constraints of hardware prototypes when simulating a whole network with many vehicles, mainly due to resource limitations [165]. Some workarounds, such as the Ego Vehicle Interface (EVI) [166], have been proposed in order to reduce the complexity of the simulation and thus making it run faster. However, these workarounds usually do not take into account the extra performance overhead resulting from the simulation of security mechanisms such as cryptographic procedures [167, 168], which can negatively impact the behavior of real VANET components. Investigating the coupling of VANET simulators and hardware devices in the presence of security primitives is an interesting research direction.

Model inaccuracies. The quality of a VANET simulation strongly depends on the accuracy of the underlying models. Noticeably, the degree of realism has increased over the last years, and some simulators now include modules that incorporate signal attenuation, different antenna patterns, and environmental diffraction. Nevertheless, the advent of novel (computation and communication) technologies, and their increasing adoption in vehicular networks, poses a constant challenge to produce accurate simulations. For example, 5G networks are more susceptible to weather conditions than its predecessors, edge computing requires more processing capacity on end hosts, and unmanned aerial vehicles heavily rely on 3D scenarios for moving and interacting with other nodes. In this sense, extending current VANET simulators to encompass these new conditions could be an important feature for current and future simulators.

Automated testing. As Table 1 shows, the number of simulators currently maintained (or recently proposed) is small compared to the overall number of VANET simulators, and this is actually a trend we observed in our investigations (i.e., new functionalities tend to be added as extensions to current open-source simulators rather than triggering the creation of a new one). That makes sense from the perspective in which current simulators (e.g., Veins) offer a plethora of basic functionalities (e.g., different communication standards and routing protocols) that can be easily used off-the-shelf by new modules, but at the same time brings new challenges in terms of program performance and engineering. For example, it becomes harder to cope with development bugs such as the ones reported in [169] and [170], when one needs to run and test a larger code base. In this sense,

developing automated context-driven testing tools for VANET simulators is an interesting research direction that can help developers to debug and deploy new functionalities faster.

Conclusion

The increasing popularity and attention to VANETs has prompted researchers to develop accurate and realistic simulation tools. In this work, we extensively studied the current state of VANET simulators, specially from the perspective of their support for novel technologies as well as safety and security mechanisms. When comparing the simulators, Veins seems to be the one with best support for these features at the moment of writing this paper. Finally, we also identified a number of challenges that should be addressed in order to improve the quality of VANET simulations.

Acknowledgements
This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001

Authors' contributions
All authors contributed to the writing of this article and read and approved the final manuscript.

Funding
This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 10 July 2020 Accepted: 20 April 2021
Published online: 06 May 2021

References

1. Qu F, Wu Z, Wang F-Y, Cho W (2015) A security and privacy review of VANETs. IEEE Trans Intell Transp Syst 16(6):2985–2996
2. Martinez FJ, Toh CK, Cano J-C, Calafate CT, Manzoni P (2011) A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). Wirel Commun Mob Comput 11(7):813–828
3. Spaho E, Barolli L, Mino G, Xhafa F, Kolici V (2011) Vanet simulators: a survey on mobility and routing protocols. In: 2011 International Conference on Broadband and Wireless Computing, Communication and Applications. IEEE. pp 1–10
4. Al-Sultan S, Al-Doori MM, Al-Bayatti AH, Zedan H (2014) A comprehensive survey on vehicular ad hoc network. J Netw Comput Appl 37:380–392
5. Mussa SAB, Manaf M, Ghafoor KZ, Doukha Z (2015) Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives. In: 2015 International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE. pp 1–8
6. Sommer C, Eckhoff D, Brummer A, Buse DS, Hagenauer F, Joerer S, Segata M (2019) Veins: The open source vehicular network simulation framework. In: Recent Advances in Network Simulation. Springer. pp 215–252
7. Lee KC, Lee U, Gerla M (2010) Survey of routing protocols in vehicular ad hoc networks. In: Advances in Vehicular Ad-hoc Networks: Developments and Challenges. IGI Global. pp 149–170
8. Jakubiak J, Koucheryavy Y (2008) State of the art and research challenges for VANETs. In: 2008 5th IEEE Consumer Communications and Networking Conference. IEEE. pp 912–916
9. Luckshetty A, Dontal S, Tangade S, Manvi SS (2016) A survey: comparative study of applications, attacks, security and privacy in VANETs. In: 2016 International Conference on Communication and Signal Processing (ICCSP). IEEE. pp 1594–1598
10. Eze EC, Zhang S, Liu E (2014) Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward. In: 2014 20th International Conference on Automation and Computing. IEEE. pp 176–181
11. Jiang D, Delgrossi L (2008) IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments. In: VTC Spring 2008-IEEE Vehicular Technology Conference. IEEE. pp 2036–2040
12. Hasrouny H, Samhat AE, Bassil C, Laouiti A (2017) Vanet security challenges and solutions: a survey. Veh Commun 7:7–20

13. Fiore M, Harri J, Filali F, Bonnet C (2007) Vehicular mobility simulation for VANETs. In: 40th Annual Simulation Symposium (ANSS'07). IEEE. pp 301–309
14. Lim KG, Lee CH, Chin RKY, Yeo KB, Teo KTK (2017) SUMO enhancement for vehicular ad hoc network (VANET) simulation. In: 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS). IEEE. pp 86–91
15. Fellendorf M, Vortisch P (2010) Microscopic traffic flow simulator VISSIM. In: Fundamentals of Traffic Simulation. Springer. pp 63–93
16. Azevedo CL, Deshmukh NM, Marimuthu B, Oh S, Marczuk K, Soh H, Basak K, Toledo T, Peh L-S, Ben-Akiva ME (2017) Simmobility short-term: An integrated microscopic mobility simulator. Transp Res Rec 2622(1):13–23
17. Cameron GD, Duncan GI (1996) Paramics–parallel microscopic simulation of road traffic. J Supercomput 10(1):25–53
18. Halati A, Lieu H, Walker S (1997) CORSIM-corridor traffic simulation model. In: Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and OpportunitiesUrban Transportation Division, ASCE; Highway Division, ASCE; Federal Highway Administration, USDOT; and National Highway Traffic Safety Administration, USDOT
19. Korkalainen M, Sallinen M, Kärkkäinen N, Tukeva P (2009) Survey of wireless sensor networks simulation tools for demanding applications. In: 2009 Fifth International Conference on Networking and Services. IEEE. pp 102–106
20. Varga A (2010) Omnet++. In: Modeling and Tools for Network Simulation. Springer. pp 35–59
21. OPNET Projects team (2015) OPNET - Optimin Network Performance. https://opnetprojects.com/. Accessed 5 Feb 2021
22. Barr R, Haas ZJ, Van Renesse R (2005) Jist/swans. Wireless networks laboratory, Cornell University. http://jist.ece.905cornell.edu. Accessed 6 Aug 2020
23. Carneiro G (2010) NS-3: Network simulator 3. In: UTM Lab Meeting April Vol. 20. pp 4–5
24. Issariyakul T, Hossain E (2009) Introduction to network simulator 2 (NS2). In: Introduction to Network Simulator NS2. Springer. pp 1–18
25. Amoozadeh,M.,b.VENTOS Manual. https://veins.car2x.org/documentation/
26. Wegener A, Piórkowski M, Raya M, Hellbrück H, Fischer S, Hubaux J-P (2008) TraCI: an interface for coupling road traffic and network simulators. In: Proceedings of the 11th Communications and Networking Simulation Symposium. pp 155–163
27. Sommer C, German R, Dressler F (2010) Bidirectionally coupled network and road traffic simulation for improved IVC analysis. IEEE Trans Mob Comput 10(1):3–15
28. Sommer C (2006) Veins - Vehicles Network Simulation. https://veins.car2x.org/. Accessed 6 Aug 2020
29. Eckhoff D, Sommer C, Dressler F (2012) On the necessity of accurate IEEE 802.11 p models for IVC protocol simulation. In: 2012 IEEE 75th Vehicular Technology Conference (VTC Spring). IEEE. pp 1–5
30. Riebl R, Günther H-J, Facchi C, Wolf L (2015) Artery: Extending veins for VANET applications. In: 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE. pp 450–456
31. Segata M, Joerer S, Bloessl B, Sommer C, Dressler F, Cigno RL (2014) Plexe: A platooning extension for veins. In: 2014 IEEE Vehicular Networking Conference (VNC). IEE. pp 53–60
32. Schünemann B (2011) V2X simulation runtime infrastructure VSimRTI: an assessment tool to design smart traffic management systems. Comput Netw 55(14):3189–3198
33. DCAITI (2006) Eclipse MOSAIC - Smart Mobility Simulation. https://www.dcaiti.tu-berlin.de/research/simulation/. Accessed 5 Feb 2021
34. Wang S-Y, Chou C-L, Yang C-M (2013) EstiNet openflow network simulator and emulator. IEEE Commun Mag 51(9):110–117
35. Wang SY, Kung H (1999) A simple methodology for constructing extensible and high-fidelity TCP/IP network simulators. In: IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future Is Now (Cat. No. 99CH36320). IEEE Vol. 3. pp 1134–1143
36. Ullah K (2016) On the use of opportunistic vehicular communication for roadside services advertisement and discovery. PhD thesis. Universidade de São Paulo, São Paulo
37. Roscher K, Bittl S, Gonzalez A, Myrtus M, Jiru J (2014) ezCar2X: rapid-prototyping of communication technologies and cooperative ITS applications on real targets and inside simulation environments. In: 11th Conference Wireless Communication and Information. pp 51–62
38. Jiru J (2021) Fraunhofer-Instituts für Kognitive Systeme IKS. https://www.ezcar2x.fraunhofer.de/en.html. Accessed 5 Mar 2021
39. Schweppe H, Roudier Y, Weyl B, Apvrille L, Scheuermann D (2011) Car2x communication: securing the last meter-a cost-effective approach for ensuring trust in car2x applications using in-vehicle symmetric cryptography. In: 2011 IEEE Vehicular Technology Conference (VTC Fall). IEEE. pp 1–5
40. Festag A, Hess S (2009) ETSI technical committee ITS: news from european standardization for intelligent transport systems (ITS)-[global communications newsletter]. IEEE Commun Mag 47(6):1–4
41. Amoozadeh M, Ching B, Chuah C-N, Ghosal D, Zhang HM (2019) VENTOS: Vehicular network open simulator with hardware-in-the-loop support. Procedia Comput Sci 151:61–68
42. Sommer C (2017) Veins User Manual documentation. https://goo.gl/rLdn2v. Accessed 6 Aug 2020
43. Tomandl A, Herrmann D, Fuchs K-P, Federrath H, Scheuer F (2014) VANETsim: an open source simulator for security and privacy concepts in VANETs. In: 2014 International Conference on High Performance Computing & Simulation (HPCS). IEEE. pp 543–550
44. Sliman A, Madi K, Khadour A, Maala B, Ahmad AS (2017) Fabrication attack effect on medical applications based on Q4 VANETs. Int J Comput Sci Trends Technol (IJCST) 5(2):1–4
45. Haklay M, Weber P (2008) Openstreetmap: User-generated street maps. IEEE Pervasive Comput 7(4):12–18
46. Piorkowski M, Raya M, Lugo AL, Papadimitratos P, Grossglauser M, Hubaux J-P (2008) TraNS: realistic joint traffic and network simulator for VANETs. ACM SIGMOBILE Mob Comput Commun Rev 12(1):31–33
47. Härri J, Filali F, Bonnet C, Fiore M (2006) VanetMobiSim: generating realistic mobility patterns for VANETs. In: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks. ACM. pp 96–97

48. Huang L, Matsuura K, Yamane H, Sezaki K (2005) Enhancing wireless location privacy using silent period. In: IEEE Wireless Communications and Networking Conference, 2005. IEEE Vol. 2. pp 1187–1192
49. Beresford AR, Stajano F (2004) Mix zones: User privacy in location-aware services. In: IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second. IEEE. pp 127–131
50. Stentz A (1997) Optimal and efficient path planning for partially known environments. In: Intelligent Unmanned Ground Vehicles. Springer. pp 203–220
51. Tomandl A, Scheuer F, Gruber B, Federrath H (2013) VANETsim- VANET simulator. https://svs.informatik.uni-hamburg.de/vanet/. Accessed 3 Aug 2020
52. NEC (2020) Smart Transportation Systems Will Be the Central Pillar of the Smart City. https://www.nec.com/en/global/insights/article/2020022504/index.html. Acessed 3 Jan 2021
53. Ge X, Li Z, Li S (2017) 5G software defined vehicular networks. IEEE Commun Mag 55(7):87–93
54. Hussein A, Elhajj IH, Chehab A, Kayssi A (2017) SDN VANETs in 5G: An architecture for resilient security services. In: 2017 Fourth International Conference on Software Defined Systems (SDS). IEEE. pp 67–74
55. Wang W, Chen Y, Zhang Q, Jiang T (2016) A software-defined wireless networking enabled spectrum management architecture. IEEE Commun Mag 54(1):33–39
56. Farhady H, Lee H, Nakao A (2015) Software-defined networking: a survey. Comput Netw 81:79–95
57. Jaballah WB, Conti M, Lal C (2019) A survey on software-defined VANETs: benefits, challenges, and future directions. arXiv preprint arXiv:1904.04577:1–17
58. Ushakova M, Ushakov Y, Bolodurina I, Parfenov D, Legashev L, Shukhman A (2020) Research of productivity of software configurable infrastructure in vanet networks on the basis of models of hybrid data transmission devices. In: 2020 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC). IEEE. pp 1–9
59. Arif M, Wang G, Balas VE, Geman O, Castiglione A, Chen J (2020) Sdn based communications privacy-preserving architecture for vanets using fog computing. Veh Commun 26:100265
60. Kadhim AJ, Seno SAH (2019) Energy-efficient multicast routing protocol based on SDN and fog computing for vehicular networks. Ad Hoc Netw 84:68–81
61. Zhu W, Gao D, Zhao W, Zhang H, Chiang H-P (2018) SDN-enabled hybrid emergency message transmission architecture in internet-of-vehicles. Enterp Inf Syst 12(4):471–491
62. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J (2008) Openflow: enabling innovation in campus networks. ACM SIGCOMM Comput Commun Rev 38(2):69–74
63. Murphy (2012) POX- networking software platform. https://github.com/noxrepo/pox. Accessed 19 Oct 2020
64. EstiNet Technologies Inc (2015) USER INTERFACE (GUI) MANUAL GRAPHIC VANET MODULE OF ESTINET SIMULATOR 9.0. http://www.estinet.com/ns/wp-content/uploads/2015/12/EstiNet_9.0_VANET_GUI_Manual_20150303.00.pdf. Accessed 15 Jan 2021
65. Manzanares-Lopez P, Malgosa-Sanahuja J, Muñoz-Gea JP (2018) A Software-Defined Networking Framework to Provide Dynamic QoS Management in IEEE 802.11 Networks. Sensors 18(7):2247
66. Amoozadeh M, Deng H, Chuah C-N, Zhang HM, Ghosal D (2015) Platoon management with cooperative adaptive cruise control enabled by VANET. Veh Commun 2(2):110–123
67. OMNeT++ Projects (2020) SOFTWARE DEFINED NETWORKING PROJECTS USING OMNET++ SIMULATOR. https://omnet-manual.com/omnetsdn-projects/. Accessed 23 Jan 2021
68. Garg S, Singh A, Kaur K, Aujla GS, Batra S, Kumar N, Obaidat MS (2019) Edge computing-based security framework for big data analytics in VANETs. IEEE Network 33(2):72–81
69. Tayal A (2018) Fog computing in IoT. https://www.tetcos.com/file-exchange.html. Acessed 22 Feb 2021
70. TETCOS (2019) Cellular Network User Manual. https://www.tetcos.com/downloads/v12.2/NetSim_User_Manual.pdf. Accessed 22 Feb 2021
71. Boukerche A, Soto V (2020) An efficient mobility-oriented retrieval protocol for computation offloading in vehicular edge multi-access network. IEEE Trans Intell Transp Syst 21(6):2675–2688
72. Zhou P, Braud T, Zavodovski A, Liu Z, Chen X, Hui P, Kangasharju J (2020) Edge-facilitated augmented vision in vehicle-to-everything networks. IEEE Trans Veh Technol 69(10):12187–12201
73. Sun Y, Guo X, Song J, Zhou S, Jiang Z, Liu X, Niu Z (2019) Adaptive learning-based task offloading for vehicular edge computing systems. IEEE Trans Veh Technol 68(4):3061–3074
74. Sommer C, Joerer S, Dressler F (2012) On the applicability of two-ray path loss models for vehicular network simulation. In: 2012 IEEE Vehicular Networking Conference (VNC). IEEE. pp 64–69
75. Feng J, Liu Z, Wu C, Ji Y (2017) AVE: Autonomous vehicular edge computing framework with ACO-based scheduling. IEEE Trans Veh Technol 66(12):10660–10675
76. Guderitz A (2020) Enhanced Traffic Safety with LTE and Mobile. Edge Computing. https://www.iks.fraunhofer.de/en/projects/car2mec.html. Accessed 22 Feb 2021
77. Olmos AG, Vazquez-Gallego F, Sedar R, Samoladas V, Mira F, Alonso-Zarate J (2019) An automotive cooperative collision avoidance service based on mobile edge computing. In: International Conference on Ad-Hoc Networks and Wireless. Springer. pp 601–607
78. Roy A, Madria S (2019) Secured traffic monitoring in VANET. arXiv preprint arXiv:1909.1005:1–12
79. Kan X, Ganlath A, Ucar S, Han K, Tiwari P, Karydis K (2019) Edge assisted misbehavior detection for platoons. In: 2019 IEEE Vehicular Networking Conference (VNC). IEEE. pp 1–4
80. Gilly K, Alcaraz S, Aknin N, Filiposka S, Mishev A (2020) Modelling edge computing in urban mobility simulation scenarios. In: 2020 IFIP Networking Conference (Networking). IEEE. pp 539–543
81. Shah SAA, Ahmed E, Imran M, Zeadally S (2018) 5G for vehicular communications. IEEE Commun Mag 56(1):111–117
82. Uzcátegui RA, De Sucre AJ, Acosta-Marum G (2009) Wave: A tutorial. IEEE Commun Mag 47(5):126–133
83. Dahlman E, Parkvall S, Skold J (2013) 4G: LTE/LTE-advanced for Mobile Broadband. Academic press
84. TETCOS (2013) NetSim 5G NR Technologies. https://www.tetcos.com/5g.html. Accessed 14 Feb 2021
85. 3rd Generation Partnership Project (1998) Mobile Broadband Standard. https://www.3gpp.org/about-3gpp. Accessed 6 Mar 2021

86. Chehri A, Chehri H, Hakim N, Saadane R (2020) Realistic 5.9 GHz DSRC vehicle-to-vehicle wireless communication protocols for cooperative collision warning in underground mining. In: Smart Transportation Systems 2020. Springer. pp 133–141
87. Veins_INET Subproject. https://veins.car2x.org/documentation/modules/#veins_inet. Accessed on: 15 August 2020
88. Virdis A, Nardini G (2020) 5G New Radio User Plane Simulation Model for INET and OMNeT++. http://simu5g.org/. Accessed 20 Aug 2020
89. Chekired DA, Togou MA, Khoukhi L, Ksentini A (2019) 5G-slicing-enabled scalable SDN core network: Toward an ultra-low latency of autonomous driving service. IEEE J Sel Areas Commun 37(8):1769–1782
90. Zhang J, Zhong H, Cui J, Tian M, Xu Y, Liu L (2020) Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks. IEEE Trans Veh Technol 69(7):7940–7954
91. Huang J, Qian Y, Hu RQ (2020) Secure and efficient privacy-preserving authentication scheme for 5G software defined vehicular networks. IEEE Trans Veh Technol 69(8):8542–8554
92. Dharanyadevi P, Venkatalakshmi K (2016) Proficient routing by adroit algorithm in 5G-Cloud-VMesh network. EURASIP J Wirel Commun Netw 2016(1):1–11
93. Chen H (2019) free5GC. https://www.free5gc.org/. Accessed 11 Mar 2021
94. EstiNet Technologies Inc (2013) EstiNet 11. https://www.estinet.com/ns/?page_id=21140. Accessed 6 Mar 2021
95. Puttagunta H, Agrawal DP (2020) Performance of 802.11 P in VANET at 5G Frequencies for Different Channel Models. In: 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE. pp 0468–0472
96. Ibrahim A, Math CB, Goswami D, Basten T, Li H (2018) Co-simulation framework for control, communication and traffic for vehicle platoons. In: 2018 21st Euromicro Conference on Digital System Design (DSD). IEEE. pp 352–356
97. 5G New Radio User Plane Simulation Model for INET and OMNeT++. http://simu5g.org/
98. Shadrin SS, Ivanova AA (2019) Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. In: Avtomobil'. Doroga. Infrastruktura Vol 3 Issue 21. p 10
99. Lu Q, Tettamanti T, Hörcher D, Varga I (2020) The impact of autonomous vehicles on urban traffic network capacity: an experimental analysis by microscopic traffic simulation. Transp Lett 12(8):540–549
100. Carpin S, Lewis M, Wang J, Balakirsky S, Scrapper C (2007) USARSim: a robot simulator for research and education. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE. pp 1400–1405
101. Pereira JL, Rossetti RJ (2012) An integrated architecture for autonomous vehicles simulation. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp 286–292
102. Alotibi F, Abdelhakim M (2020) Anomaly Detection for Cooperative Adaptive Cruise Control in Autonomous Vehicles Using Statistical Learning and Kinematic Model. In: IEEE Transactions on Intelligent Transportation Systems. pp 1–11. https://doi.org/10.1109/TITS.2020.2983392
103. Li Y, Liu Q (2020) Intersection management for autonomous vehicles with vehicle-to-infrastructure communication. PLoS ONE 15(7):0235644
104. Teixeira M, d'Orey PM, Kokkinogenis Z (2020) Simulating collective decision-making for autonomous vehicles coordination enabled by vehicular networks: A computational social choice perspective. Simul Model Pract Theory 98:101983
105. Zehe D, Nair S, Knoll A, Eckhoff D (2017) Towards citymos: a coupled city-scale mobility simulation framework. 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication 2017:03
106. Roscher K, Maierbacher G (2016) Reliable message forwarding in VANETs for delay-sensitive applications. In: 2016 International Symposium on Wireless Communication Systems (ISWCS). IEEE. pp 199–203
107. Hadiwardoyo SA, Dricot J-M, Calafate CT, Cano J-C, Hernández-Orallo E, Manzoni P (2020) UAV Mobility model for dynamic UAV-to-car communications in 3D environments. Ad Hoc Netw 107:102193
108. Brummer A, German R, Djanatliev A (2018) On the necessity of three-dimensional considerations in vehicular network simulation. In: 2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS). IEEE. pp 75–82
109. Montgomery DR, Foufoula-Georgiou E (1993) Channel network source representation using digital elevation models. Water Resour Res 29(12):3925–3934
110. NetSim Applications Unmanned Aerial Vehicle (UAV) Communication. https://www.tetcos.com/uav-drone-communication.html. Accessed on: 18 February 2021
111. Higham DJ, Higham NJ (2016) MATLAB Guide. SIAM - Society for Industrial and Applied Mathematics, Philadelphia. pp 1–502
112. Oliveira R, Montez C, Boukerche A, Wangham MS (2017) Reliable data dissemination protocol for VANET traffic safety applications. Ad Hoc Netw 63:30–44
113. Bello Tambawal A, Md Noor R, Salleh R, Chembe C, Oche M (2019) Enhanced weight-based clustering algorithm to provide reliable delivery for VANET safety applications. PLoS ONE 14(4):0214664
114. Sutagundar AV, Kalyani T (2017) Fault tolerance in VANET's. In: 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon). IEEE. pp 1039–1043
115. Almeida J, Rufino J, Alam M, Ferreira J (2019) A survey on fault tolerance techniques for wireless vehicular networks. Electronics 8(11):1358
116. Dharmaraja S, Vinayak R, Trivedi KS (2016) Reliability and survivability of vehicular ad hoc networks: An analytical approach. Reliab Eng Syst Saf 153:28–38
117. Albano WA, Nogueira M, de Souza JN (2015) A taxonomy for resilience in vehicular ad hoc networks. IEEE Lat Am Trans 13(1):228–234
118. Alkhalifa IS, Almogren AS (2020) NSSC: Novel segment based safety message broadcasting in cluster-based vehicular sensor network. IEEE Access 8:34299–34312
119. de Souza AM, Braun T, Botega LC, Villas LA, Loureiro AA (2019) Safe and sound: Driver safety-aware vehicle re-routing based on spatiotemporal information. IEEE Trans Intell Transp Syst 21(9):3973–3989
120. Shahwani H, Mugabarigira BA, Shen Y, Jeong JP, Shin J (2020) DAPF: Delay-aware packet forwarding for driving safety and efficiency in vehicular networks. IET Communications 14(9):1404–1411

121. Offor P (2012) Vehicle Ad Hoc Network (VANET): Safety Benefits and Security Challenges. SSRN Electron J. https://doi.org/10.2139/ssrn.2206077. http://ssrn.com/abstract=2206077. Accessed 8 Mar 2021
122. Liu H, Lin C-W, Kang E, Shiraishi S, Blough DM (2019) A byzantine-tolerant distributed consensus algorithm for connected vehicles using proof-of-eligibility. In: Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. pp 225–234
123. Sommer C, Dressler F (2011) Using the right two-ray model? A measurement based evaluation of PHY models in VANETs. In: Proc. ACM MobiCom. pp 1–3
124. Sommer C, Eckhoff D, Dressler F (2013) IVC in cities: Signal attenuation by buildings and how parked cars can improve the situation. IEEE Trans Mob Comput 13(8):1733–1745
125. SystemXIRT (2019) F2MD: Framework For Misbehavior Detection. https://www.irt-systemx.fr/. Accessed 3 Mar 2021
126. Kamel J, Ansari MR, Petit J, Kaiser A, Jemaa IB, Urien P (2020) Simulation framework for misbehavior detection in vehicular networks. IEEE Trans Veh Technol 69(6):6631–6643. https://doi.org/10.1109/TVT.2020.2984878
127. Sommer C, Eckhoff D, German R, Dressler F (2011) A computationally inexpensive empirical model of IEEE 802.11 p radio shadowing in urban environments. In: 2011 Eighth International Conference on Wireless On-demand Network Systems and Services. IEEE. pp 84–90
128. Sommer C, Joerer S, Segata M, Tonguz OK, Cigno RL, Dressler F (2014) How shadowing hurts vehicular communications and how dynamic beaconing can help. IEEE Trans Mob Comput 14(7):1411–1421
129. Sommer C, Joerer S, Dressler F (2012) On the applicability of two-ray path loss models for vehicular network simulation. In: 2012 IEEE Vehicular Networking Conference (VNC). pp 64–69. https://doi.org/10.1109/VNC.2012.6407446
130. Ucar S, Ergen SC, Ozkasap O (2017) Data-driven abnormal behavior detection for autonomous platoon. In: 2017 IEEE Vehicular Networking Conference (VNC). IEEE. pp 69–72
131. Oliphant TE (2007) Python for scientific computing. Comput Sci Eng 9(3):10–20
132. R Core Team (2013) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. https://www.Rproject.org/. Accessed 3 Mar 2021
133. Hasan S (2020) Fail-operational and fail-safe vehicle platooning in the presence of transient communication errors. PhD thesis, Mälardalen University, Västerås and Eskilstuna
134. Tanwar S, Vora J, Tyagi S, Kumar N, Obaidat MS (2018) A systematic review on security issues in vehicular ad hoc network. Secur Priv 1(5):39
135. Bariah L, Shehada D, Salahat E, Yeun CY (2015) Recent advances in VANET security: a survey. In: 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall). IEEE. pp 1–7
136. Alaya B, SELLAMI L (2021) Clustering method and symmetric/asymmetric cryptography scheme adapted to securing urban VANET networks. J Inf Secur Appl 58:102779
137. Walton J (2012) Data Security with Crypto++. O'Reilly Media, Inc, Newton
138. Singh V, Mahajan K (2016) Vanet and its security issues-a review. Int J Comput Sci Eng 4(10):59–64
139. Bakar KAA, Irvine J (2010) A scheme for detecting selfish nodes in MANETs using OMNET++. In: 2010 6th International Conference on Wireless and Mobile Communications. IEEE. pp 410–414
140. Hortelano J, Ruiz JC, Manzoni P (2010) Evaluating the usefulness of watchdogs for intrusion detection in VANETs. In: 2010 IEEE International Conference on Communications Workshops. IEEE. pp 1–5
141. Dietzel S, Gürtler J, Kargl F (2016) A resilient in-network aggregation mechanism for VANETs based on dissemination redundancy. Ad Hoc Netw 37:101–109
142. Achour I, Bejaoui T, Busson A, Tabbane S (2015) A redundancy-based protocol for safety message dissemination in vehicular ad hoc networks. In: 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall). IEEE. pp 1–6
143. Khacheba I, Yagoubi MB, Lagraa N, Lakas A (2018) CLPS: context-based location privacy scheme for VANETs. Int J Ad Hoc Ubiquit Comput 29(1-2):141–159
144. De Fuentes JM, González-Tablas AI, Ribagorda A (2011) Overview of security issues in vehicular ad-hoc networks. In: Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts. IGI global. pp 894–911
145. Soliman JN, Mageed TA, El-Hennawy HM (2017) Digital signature and authentication mechanisms using new customized hash function for cognitive radio networks. In: 2017 12th International Conference on Computer Engineering and Systems (ICCES). IEEE. pp 175–181
146. Liu Y, Wang Y, Chang G (2017) Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm. IEEE Trans Intell Transp Syst 18(10):2740–2749
147. Singh PK, Tabjul GS, Imran M, Nandi SK, Nandi S (2018) Impact of security attacks on cooperative driving use case: CACC platooning. In: TENCON 2018-2018 IEEE Region 10 Conference. IEEE. pp 138–143
148. Mohan AP, Elshakankiri M (2019) Enhanced priority-based routing protocol (EPRP) for inter-vehicular communication. In: International Conference on Computing. Springer. pp 325–337
149. Emara K (2016) Poster: Prext: Privacy extension for veins vanet simulator. In: 2016 IEEE Vehicular Networking Conference (VNC). IEEE. pp 1–2
150. Pan Y, Li J (2013) Cooperative pseudonym change scheme based on the number of neighbors in vanets. J Netw Comput Appl 36(6):1599–1609
151. Emara K, Woerndl W, Schlichter J (2015) CAPS: Context-aware privacy scheme for VANET safety applications. In: Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks. pp 1–12
152. Huang L, Yamane H, Matsuura K, Sezaki K (2006) Silent cascade: enhancing location privacy without communication qos degradation. In: International Conference on Security in Pervasive Computing. Springer. pp 165–180
153. Buttyán L, Holczer T, Weimerskirch A, Whyte W (2009) Slow: A practical pseudonym changing scheme for location privacy in vanets. In: 2009 IEEE Vehicular Networking Conference (VNC). IEEE. pp 1–8
154. Scheuer F, Fuchs K-P, Federrath H (2011) A safety-preserving mix zone for vanets. In: International Conference on Trust, Privacy and Security in Digital Business. Springer. pp 37–48

155. IEEE (2016) IEEE standard for wireless access in vehicular environments-security services for applications and management messages. Std 1609.2-2016 (Revision of IEEE Std 1609.2-2013):1–240. https://doi.org/10.1109/IEEESTD.2016.7426684. Accessed 8 Mar 2021
156. ETSI TS (2010) 102 731 v1.1.1 - Intelligent Transport Systems (ITS); Security; Security Services and Architecture. Standard, TC C-ITS. https://www.etsi.org/deliver/etsi_ts/102700_102799/102731/01.01.01_60/ts_102731v010101p.pdf. Accessed 8 Mar 2021
157. ETSI TS (2012) ETSI TS 102 941 v1.1.1 - Intelligent Transport Systems (ITS); Security; Trust and Privacy Management, Standard, TC C-ITS. https://www.etsi.org/deliver/etsi_ts/102900_102999/102941/01.01.01_60/ts_102941v010101p.pdf. Accessed 8 Mar 2021
158. ETSI TS (2019) 102 941 v1.3.1 - Intelligent Transport Systems (ITS); Security; Trust and Privacy Management, Standard, TC C-ITS. https://standards.iteh.ai/catalog/standards/etsi/9ba459d3-be83-4b0d-82ca-28644f10d731/etsi-ts-102-941-v1-3-1-2019-02. Accessed 8 Mar 2021
159. ETSI TS (2015) 103 097 v1.2.1 - Intelligent Transport Systems (ITS); Security; Security Header and Certificate Formats. Standard, TC C-ITS. https://www.etsi.org/deliver/etsi_ts/103000_103099/103097/01.03.01_60/ts_103097v010301p.pdf. Accessed 8 Mar 2021
160. ETSI TS (2016) 102 940 v1.2.1 - Intelligent Transport Systems (ITS); Security; ITS Communications Security Architecture and Security Management. Standard, TC CITS. https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/01.02.01_60/ts_102940v010201p.pdf. Accessed 8 Mar 2021
161. ETSI TS (2017) 102 893 v1.2.1 Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA). Standard, TC C-ITS. https://www.etsi.org/deliver/etsi_tr/102800_102899/102893/01.02.01_60/tr_102893v010201p.pdf. Accessed 8 Mar 2021
162. Cinque M, Cotroneo D, Di Martino C, Russo S, Testa A (2009) Avr-inject: A tool for injecting faults in wireless sensor nodes. In: 2009 IEEE International Symposium on Parallel & Distributed Processing. IEEE. pp 1–8
163. Sailhan F, Delot T, Pathak A, Puech A, Roy M (2010) Fault injection and monitoring for dependability analysis of wireless sensor-actuators networks. In: 4th Workshop Gestion des Données dans les Systèmes d'Information Pervasifs (GEDSIP) in Cunjunction with INFORSID. Citeseer
164. Raposo D, Rodrigues A, Silva JS, Boavida F (2017) A taxonomy of faults for wireless sensor networks. J Netw Syst Manag 25(3):591–611
165. Buse DS, Dressler F (2019) Towards real-time interactive V2X simulation. In: 2019 IEEE Vehicular Networking Conference (VNC). IEEE. pp 1–8
166. Buse DS, Schettler M, Kothe N, Reinold P, Sommer C, Dressler F (2018) Bridging worlds: Integrating hardware-in-the-loop testing with large-scale VANET simulation. In: 2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS). IEEE. pp 33–36
167. Riebl R, Monz M, Varga S, Janicke H, Maglaras L, Al-Bayatti AH, Facchi C (2016) Improved security performance for vanet simulations. In: 4th IFAC Symposium on Telematics Applications
168. Baee MAR, Simpson L, Foo E, Pieprzyk J (2019) Broadcast Authentication in Latency-Critical Applications: On the Efficiency of IEEE 1609.2. IEEE Trans Veh Technol 68(12):11577–11587
169. Sommer C (2020) Veins 5.1 Changelog. https://veins.car2x.org/download/#changelog. Accessed 8 Mar 2020
170. DCAITI (2020) Eclipse MOSAIC Download Area. https://www.dcaiti.tu-berlin.de/research/simulation/download/. Accessed 8 Mar 2020

Publisher's Note