JOURNAL LA MULTIAPP

VOL. 02, ISSUE 06 (012-020), 2021
DOI: 10.37899/journallamultiapp.v2i6.519

Review of Custom Grids for Updated Vehicles on VANET Simulators

Intisar Mohsin Saadoon[1], Qabas Abdal Zarhaa Jabbar[1], Dalia Shihab Ahmad[1]

[1]*Department of Computer Science, Mustansiriyah University-Iraq, Baghdad, Iraq*

**Corresponding Author: Intisar Mohsin Saadoon*
Email: Dr.intisar_muhson@uomustansiriyah.edu.iq

Article Info

Article history:
Received 27 August2020
Received in revised form 06 September 2020
Accepted 19 September 2020

Keywords:
VANet
V2V
V2I
RSU
OBU
V2R
NetSim
Veins
Eclipse MOSAIC
EstiNet
ezCar2X
VENTOS
VANETsim

Abstract

VANET deployment and testing is time-consuming and costly. Simulation is a handy and less expensive alternative to real implementation as a workaround. It is required to develop accurate models in order to receive excellent results from a VANET simulation, which difficult operation owes to the complexity of the VANET infrastructure (for example, simulators have to model the navigation models and communication protocols). The network and navigation components, which are the building blocks of contemporary VANET simulators, are described in this section. Simulators are a useful tool for testing VANETs at a minimal cost and without endangering users. However, in order to be helpful and convey trustworthy findings, simulators must be able to simulate new technologies that enter the VANET and enable safety and security procedures. To put it another way, if simulation is a good tool for VANET development it should be enhanced. VANET simulators have been the subject of research since early 2010 [1-4]. They analyze the correctness of VANET's numerous tools like a navigation simulator and network simulator, as well as how these building blocks are connected. The introduction of new network technologies such as 5G, SDN, edge computing, and VANET research as a result of investments in autonomous cars is forcing VANET simulators to re-evaluate their support for these new capabilities. We present an updated evaluation of VANET simulators in this post, highlighting their key features and current support for emerging technologies.

Introduction

The DSRC allows vehicles to interact directly with each other in one or more ways. V2V (vehicle to vehicle), V2I (vehicle to infrastructure), or hybrid (Sommer et al., 2019) communication modes are available, as shown in figure (1). These vehicle communication arrangements rely largely on high-resolution data collecting as well as high-speed data transmission. -terminate the kinetic data of both cars and the environment utilizing sophisticated positioning systems and wireless communication protocols.

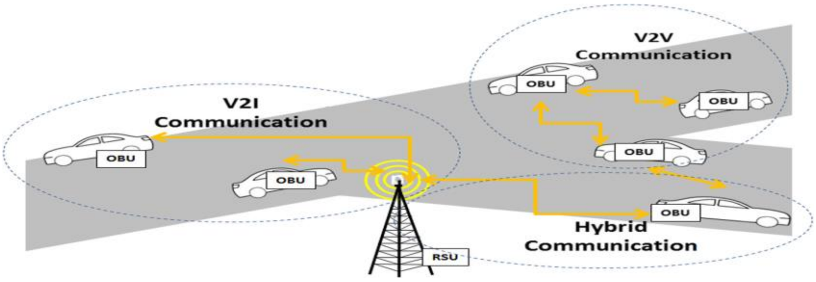

Figure 1. Communication types in VANET

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Network and mobility simulations are combined in VANET simulators (Mussa et al., 2015). Network simulators are in charge of simulating communication and message exchange protocols, whereas mobility simulators are in charge of controlling the movement of each node. The primary VANET simulators discovered in the literature are described in this section, with an emphasis on their design and capabilities. We examined papers that offered a simulation or comparative analysis of VANET simulators using common search databases and search engines (IEEE Xplorer, ACM Digital Library, Science Direct, Google Scholar, and others). We've also gone over the quotes again and again. Throughout this process, we looked at approximately 100 articles. We also looked on Google for private VANET simulators that were not necessarily utilized by colleges. The simulators we identified are listed in table (1). We were able to uncover several unique simulators, despite the fact that the majority are free source. We focus our analysis on simulators that offer a version after 2017, and we assume that older tools are no longer supported, so there's a good chance that the latest developments in VANET research, which is one of our analysis criteria in this document, won't be supported. A full examination of ancient simulators may be found in (Al-Sultan et al., 2014; Mussa et al., 2015).

Table 1. List of VANET simulators.

Simulator	Network simulator	Mobility simulator	License	Last release
NetSim	own	SUMO	proprietary	2021
Veins	OMNeT++	SUMO	open-source	2020
Eclipse MOSAIC	NS-3, OMNeT++, SND and Eclipse MOSAIC Cell	SUMO and VISSIM	open-source	2020
EstiNet	own	own	proprietary	2020
ezCar2X	NS-3	SUMO	proprietary	2020
VENTOS	OMNeT++	SUMO	open-source	2018
VANETsim	own	own	open-source	2017

NetSim

NetSim is a standalone business simulation that supports a wide range of wired, wireless, and mobile networks as well as sensors. Professional, Standard, and University licenses are available, with only the first two allowing VANET emulation. Netsim works with SUMO to simulate VANETs. The first is about the WAVE standard for wireless vehicle communication, while the second is about simulating road traffic conditions. For VANET simulators, NetSim provides a set of network, link, and table (1) performance data. The productivity graphs of this business application detail the gray sections (latest version after 2017). Depending on the sort of network being emulated, the metrics will differ. Users can record the information of each packet as it passes over the network using packet monitoring and event tracking. The NetSim architecture is depicted in figure (2) in a simplified form. Each network corresponds to a simulator component with its own set of communication protocols. Netsim has a sufficient number of components as well as the ability to link real devices to execute simulated live applications. It's important to keep in mind the radio frequency propagation models in the Netsim VANET modules. This module contains the route loss, shadow model, and fading model, all of which are required to anticipate signal loss or signal encounters in a building or at a distance in heavily crowded locations. Because VANET is likely to meet various impediments for its signal connections in a real situation, this module aids in making the simulation more realistic. Netsim's architecture allows new components to be added in response to user requests and the introduction of new technologies. W-LAN, cognitive radio, LTE, MANET, military radio, IoT, VANET, Software Defined Networking (SDN), and satellite communications have all been used in the simulator. The SDN module, in particular, enables a variety of network commands for simulation control,

ISSN: 2716-3865 (Print), 2721-1290 (Online)

routing, access control, and other functions, all of which may be run from the console command line during simulation.

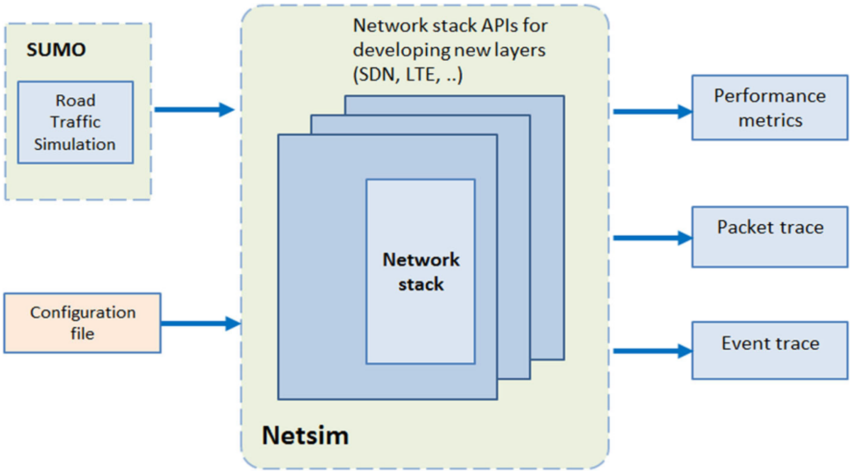

Figure 2. Architecture of the NetSim platform

Veins

According to Sommer et al. (2019) Veins is a free and open source framework for simulating vehicle networks. It's built on top of OMNeT ++ and SUMO. The different components that make up the Veins architecture are shown in figure (3). The simulation, in general, constructs an OMNeT ++ node for each vehicle in the simulation, then connects the node's motions with the movements of the cars in the road traffic simulation (for example SUMO). Network simulation and navigation can run in parallel in this situation. This is made feasible by the bidirectional connection achieved through the Traffic Control Interface (TraCI) (Wegener et al. 2008), a standard communication protocol. Within the scope of TCP connections, TraCI allows OMNeT ++ and SUMO to exchange messages (for example, navigation traces) throughout the simulation execution (Sommer et al., 2010). Many extensions (now more than 17 (Sommer, 2006) are included in the simulator, allowing packet modeling of various protocols (e.g. IEEE 802.11p (Eckhoff et al. 2012), ETSI ITS-G5 (Riebl et al. (2015) as well as applications (for example, Car family (Segata et al., 2014). In a nutshell, Veins is a runtime environment for user-written programs that makes it simpler to simulate new settings and applications. Its disadvantage is that it requires the appropriate operation of both SUMO and OMNeT ++ in order to get reliable findings. Veins can arise due to any fault in any of them, with inconsistent outcomes. Veins is compatible with Linux, Windows, and Mac OS.

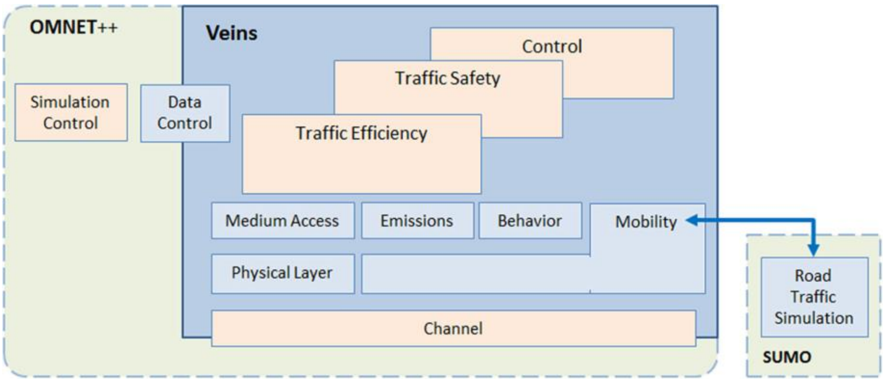

Figure 3. *Architecture of the Veins platform. Adapted from (Sommer et al., 2019)*

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Eclipse MOSAIC

According to Schünemann, (2011) Eclipse MOSAIC, originally V2X Simulation Runtime Infrastructure (VSim-RTI), is a multi-domain and multi-domain simulation platform for assessing innovative linked and automated mobility solutions. The primary purpose of Eclipse MOSAIC is to allow users to execute many V2X simulations with their preferred simulation. EclipseMOSAIC integrates many simulations to create a more realistic representation of vehicle traffic, pollution, and radio communications.

SUMO and PHABMACS for traffic simulation; NS3, OMNET ++, and SNS for communications simulation; and the Eclipse MOSAIC application simulation are some of the simulations presently supported by Eclipse MOSAIC. As shown in figure (4), simulations and other analytic tools (particularly those from third parties) may be simply incorporated.

Three essential aspects of the runtime architecture are necessary to incorporate simulators into Eclipse MOSAIC. Each participating simulator must be connected to the runtime infrastructure by the consortium management. A native simulator and two connectors, one to accept data from the runtime infrastructure and the other to provide data to it, make up the union. To coordinate the simulation and synchronize the participating associations, time management is required. It guarantees that each union's events are processed in the proper sequence. Interaction management is a publishing subscription approach that allows unions to communicate data.

One of the distinctive aspects of the EclipseMOSAIC architecture is the ability to view data in numerous ways. Different visualization tools that may be connected to a running simulation can be used to examine the same scenario. WebSocket Visualizer, Integrated Testing and Evaluation Framework (ITEF), and PHABMap are a few of these tools (for 3D visualization).

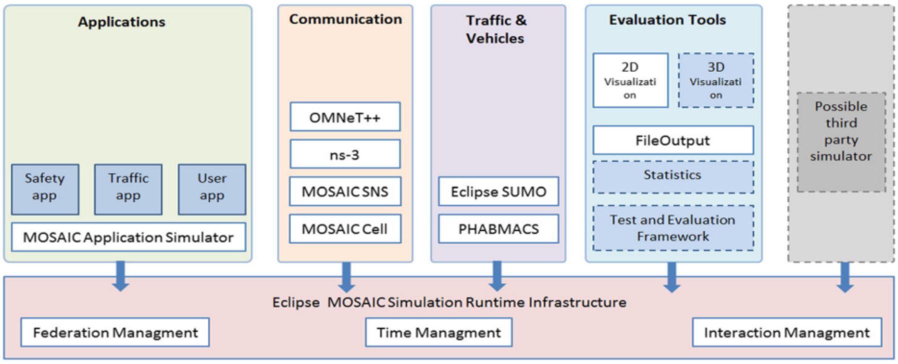

Figure 4. Architecture of the Eclipse MOSAIC framework. Adapted from (DCAITI, 2006)

EstiNet

According to Wang et al. (2013) EstiNet is a high-precision networking simulator for commercial use. EstiNet combines the benefits of simulation and simulation techniques using a unique process termed kernel reintroduction (Wang & Kung, 1999). As shown in figure (5), the kernel reentry approach intercepts packets exchanged by two actual apps and passes them to the EstiNet simulation engine via tunnel network interfaces. VANET is an optional EstiNet add-on. EstiNet has a road construction option that allows you to create a road network from scratch or import a road map file to simulate automobile traffic. EstiNet offers its own navigation simulator, which allows vehicles and humans to perform fundamental driving behaviors such as vehicle tracking, lane change, overtaking, and traffic signal compliance. EstiNet supports IEEE 802.11p, IEEE 1609.3, and IEEE 1609.4 protocols. In VANET simulations, EstiNet provides OBU and RSU abstractions. EstiNet provides a variety of OBU communication interfaces to emulate OBUs. Each communication interface, such as agent-based vehicles (IEEE 802.11p / 1609) and unit-based vehicles (IEEE 802.11p) (Ullah, 2016),

ISSN: 2716-3865 (Print), 2721-1290 (Online)

defines a distinct communication behavior. Because the user has more freedom to construct OBU connections and behavior according to their needs, this feature of EstiNet allows for better implementation of vehicle driving intelligence. EstiNet offers two sets of unique communication protocols for RSU: IEEE 802.11p / 1609 and IEEE 802.3.

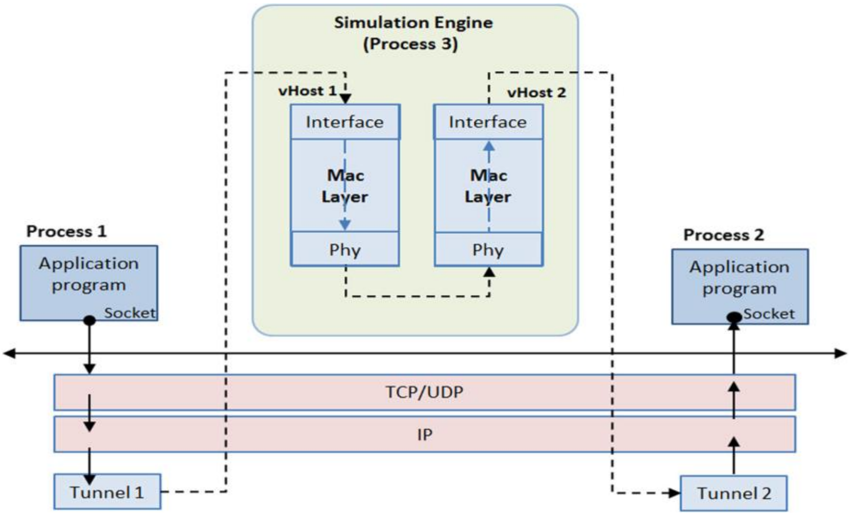

Figure 5. Simulation architecture of EstiNet, host-to-host case. Adapted from (Wang et al., 2013)

ezCar2X

According to Roscher et al., (2014) ezCar2X is a software framework for developing collaborative ITS applications and new communication protocols quickly. The ezCar2X simulator is currently proprietary, but plans are in the works to make it open source in the near future (Jiru, 2021). It was built for Car2X Communications (Schweppe et al. 2011) with the goal of allowing car manufacturers, suppliers, and road infrastructure operators to test novel applications in a virtual environment. ezCar2X is written in C++ and includes particular optimizations to make the most use of system resources. It also has SUMO, which can be used in conjunction with other simulators via the TraCI API (Wegener et al., 2008). The ETSI design for intelligent transport system (ITS) terminals (Festag & Hess, (2009) provides the foundation for the ezCar2X architecture, which specifies the access, network, utility, management, and security layers, as shown in figure (6). Its module offers a Logging System Plus Event Scheduler for asynchronous processes and easy timeout accomplishment for ezCar2X modules. It also features ETSI-compliant access and network modules that enable ITS-G5, 3G / 4G, geo-network, and Basic Transport Protocol (BTP). The security module implements the network security concept by signing and encrypting messages sent as well as validating and decrypting messages received. On both Linux and Windows systems, the ezCar2X simulator may be used.

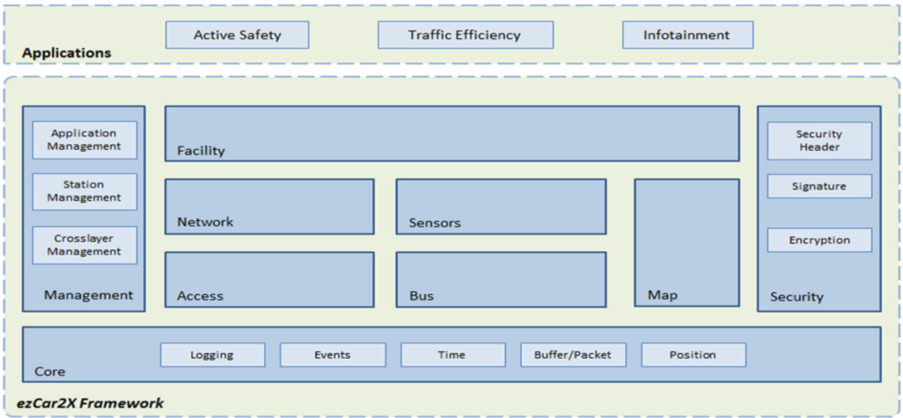

Figure 6. Architecture of the ezCar2X framework. Adapted from (Roscher et al., 2014)

ISSN: 2716-3865 (Print), 2721-1290 (Online)

VENTOS

According to Amoozadeh et al. (2019) VENTOS is a free and open-source simulator for analyzing vehicle-network applications. It employs SUMO and OMNET ++ for navigation and network modeling, similar to Veins. VENTOS, on the other hand, contains a large number of pre-built modules that make it simple to simulate complicated application situations. The simulator, for example, has applications for traffic light control algorithms (TSC) and platoon management operations. VENTOS offers two unique modules, addNode and trafficControl, that facilitate the production of traffic requests on a microscopic level. Users may simply add fixed and mobile nodes to the simulation using the former, while the latter allows them to regulate vehicle traffic by adjusting its speed and setting platoon movements. VENTOS' architecture is seen in figure (7). In a hardware-in-the-loop (HIL) situation, the simulator may be expanded to communicate with actual OBUs and RSUs. Each physical device linked to the computer has a virtual node in the simulation, and every action taken on the actual device is mirrored in its alias, and vice versa. An Ethernet port may be used to connect to a device running VENTOS, and SSH connections can be used to communicate with the device. The simulator needs that control applets , manages data and controls commands, operate on the device, which may prevent it from working with some cards. VENTOS, like Veins, uses SUMO and OMNeT ++ to get the job done. Linux, Windows, and Mac OS are all supported by the simulator.

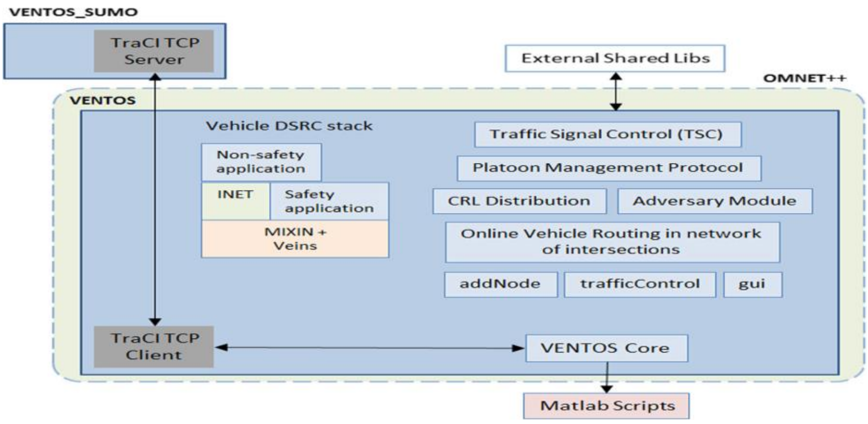

Figure 7. Architecture of the VENTOS platform. Adapted from Sommer (2017)

VANETsim

Tomandl et al. (2014) VANETsim is an event simulator created specifically for researching security and secrecy in vehicle communications. It enables for application-level study of assaults and responses, such as designing an attack and analyzing its impact on various types of vehicles (Sliman et al., 2017). Figure (8) shows the four key components of the VANETsim architecture: a graphical user interface (GUI), a scenario generator, a simulation simulator, and a post-processing engine. A map editor is included in the graphical interface, which allows users to build and change roadmaps. OpenStreetMap Haklay & Weber, (2008) may be used to generate maps from scratch or to import them. Imported maps may be altered and saved as XML files, making them more interoperable with other programs (Piorkowski et al., 2008; Härri et al., 2006).

The VANETsim interface attempts to be user-friendly, with capabilities that can be used on the move while the simulator is running. It can also refer to a vehicle's transmission range, which is a characteristic that can be turned on and off at will. Scenario Creator allows users to create a series of tests and save their settings in XML files. Because settings may be

ISSN: 2716-3865 (Print), 2721-1290 (Online)

exchanged over the Internet, this feature makes it easy to clone studies. The simulation is carried out in the simulation facility. The traffic map, network architecture, and any required security and privacy modules are all coordinated by it. Finally, the post-processing engine examines the generated logs in order to present pertinent events and metrics .

VANETsim has a few security and privacy modules that implement notions like periods of quiet Huang et al. (2005) and mixed zones (Beresford & Stajano, 2004). The A * algorithm (Stentz, 997). guides cars to individually specified destinations, and vehicle communication can include two sorts of messages: signals, which broadcast routine information (e.g., position, speed), and special messages, which are sent whenever a relevant event occurs for example, an emergency vehicle is approaching. In April 2017, the VANETsim project came to an end. Despite the fact that the program is no longer maintained, the simulator website Tomandl et al., (2013) still has documentation, downloadable material, and a user-friendly tutorial to using the simulator. VANETsim is a Windows-based simulation program.

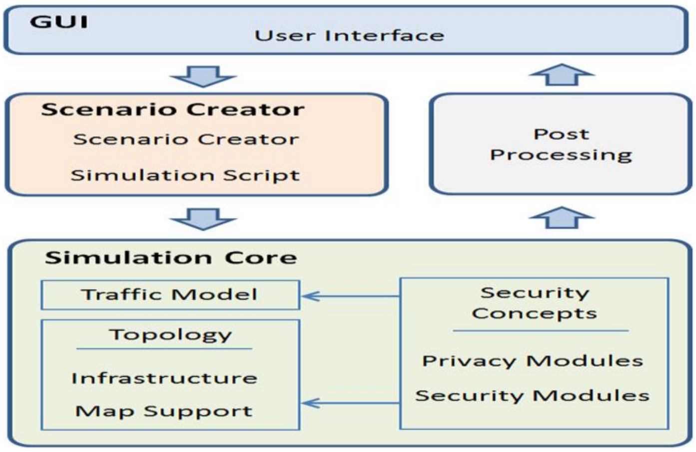

Figure 8. Architecture of the VANETsim platform. Adapted from Tomandl et al. (2014)

Conclusion

Researchers have developed accurate and realistic simulation tools in response to the increased popularity and interest in VANETs. We investigated the present status of VANET simulators in thoroughly in this paper, particularly in terms of their support for new technologies as well as safety and security methods. When comparing simulators, Veins appears to offer these characteristics the best at the time of writing. Finally, in order to increase the quality of VANET simulation, we identified a number of difficulties that must be solved.

References

Martinez, F. J., Toh, C. K., Cano, J. C., Calafate, C. T., & Manzoni, P. (2011). A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wireless Communications and Mobile Computing*, *11*(7), 813-828.

Spaho, E., Barolli, L., Mino, G., Xhafa, F., & Kolici, V. (2011, October). Vanet simulators: A survey on mobility and routing protocols. In *2011 International Conference on Broadband and Wireless Computing, Communication and Applications* (pp. 1-10). IEEE.

Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., & Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, *37*, 380-392.

Mussa, S. A. B., Manaf, M., Ghafoor, K. Z., & Doukha, Z. (2015, October). Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives. In *2015*

ISSN: 2716-3865 (Print), 2721-1290 (Online)

International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-8). IEEE.

Sommer, C., Eckhoff, D., Brummer, A., Buse, D. S., Hagenauer, F., Joerer, S., & Segata, M. (2019). Veins: The open source vehicular network simulation framework. In *Recent Advances in Network Simulation* (pp. 215-252). Springer, Cham.

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).

Sommer, C., German, R., & Dressler, F. (2010). Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Transactions on mobile computing*, *10*(1), 3-15.

Sommer C (2006) Veins - Vehicles Network Simulation. https://veins.car2x.org/. Accessed 6 Aug 2020

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)* (pp. 1-5). IEEE.

Riebl, R., Günther, H. J., Facchi, C., & Wolf, L. (2015, June). Artery: Extending veins for VANET applications. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)* (pp. 450-456). IEEE.

Segata, M., Joerer, S., Bloessl, B., Sommer, C., Dressler, F., & Cigno, R. L. (2014, December). Plexe: A platooning extension for Veins. In *2014 IEEE Vehicular Networking Conference (VNC)*(pp. 53-60). IEEE.

Schünemann, B. (2011). V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems. *Computer Networks*, *55*(14), 3189-3198.

DCAITI (2006) Eclipse MOSAIC - Smart Mobility Simulation. https://www.dcaiti.tu-berlin.de/research/simulation/. Accessed 5 Feb 2021

Wang, S. Y., Chou, C. L., & Yang, C. M. (2013). EstiNet openflow network simulator and emulator. *IEEE Communications Magazine*, *51*(9), 110-117.

Wang, S. Y., & Kung, H. T. (1999, March). 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)* (Vol. 3, pp. 1134-1143). IEEE.

Ullah, K. (2016). *On the use of opportunistic vehicular communication for roadside services advertisement and discovery* (Doctoral dissertation, Universidade de São Paulo).

Roscher, K., Bittl, S., Gonzalez, A. A., Myrtus, M., & Jiru, J. (2014, October). 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).

Jiru J. (2021). Fraunhofer-Instituts für Kognitive Systeme IKS. https:// www.ezcar2x.fraunhofer.de /en.html. Accessed 5 Mar 2021

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Schweppe, H., Roudier, Y., Weyl, B., Apvrille, L., & Scheuermann, D. (2011, September). 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)* (pp. 1-5). IEEE.

Festag, A., & Hess, S. (2009). ETSI technical committee ITS: news from european standardization for intelligent transport systems (ITS)-[global communications newsletter]. *IEEE Communications Magazine*, *47*(6), 1-4.

Amoozadeh, M., Ching, B., Chuah, C. N., Ghosal, D., & Zhang, H. M. (2019). VENTOS: Vehicular network open simulator with hardware-in-the-loop support. *Procedia Computer Science*, *151*, 61-68.

Sommer, C. (2017) *Veins User Manual documentation*. https://goo.gl/rLdn2v. Accessed 6 Aug 2020

Tomandl, A., Herrmann, D., Fuchs, K. P., Federrath, H., & Scheuer, F. (2014, July). VANETsim: An open source simulator for security and privacy concepts in VANETs. In *2014 International Conference on High Performance Computing & Simulation (HPCS)* (pp. 543-550). IEEE.

Sliman, A., Madi, K., Khadour, A., Maala, B., & Ahmad, A. S. (2017). Fabrication Attack Effect on Medical Applications based on VANETs.

Haklay, M., & Weber, P. (2008). Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, *7*(4), 12-18.

Piorkowski, M., Raya, M., Lugo, A. L., Papadimitratos, P., Grossglauser, M., & Hubaux, J. P. (2008). TraNS: realistic joint traffic and network simulator for VANETs. *ACM SIGMOBILE mobile computing and communications review*, *12*(1), 31-33.

Härri, J., Filali, F., Bonnet, C., & Fiore, M. (2006, September). VanetMobiSim: generating realistic mobility patterns for VANETs. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks* (pp. 96-97).

Huang, L., Matsuura, K., Yamane, H., & Sezaki, K. (2005, March). Enhancing wireless location privacy using silent period. In *IEEE Wireless Communications and Networking Conference, 2005* (Vol. 2, pp. 1187-1192). IEEE.

Beresford, A. R., & Stajano, F. (2004, March). Mix zones: User privacy in location-aware services. In *IEEE Annual conference on pervasive computing and communications workshops, 2004. Proceedings of the Second* (pp. 127-131). IEEE.

Stentz, A. (1997). Optimal and efficient path planning for partially known environments. In *Intelligent unmanned ground vehicles*(pp. 203-220). Springer, Boston, MA.

Tomandl. A, Scheuer. F, Gruber. B, Federrath. H. (2013) VANETsim- VANET simulator. https://svs.informatik.unihamburg.de/vanet/. Accessed 3 Aug 2020

ISSN: 2716-3865 (Print), 2721-1290 (Online)