IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A BRIEF GLANCE TOWARDS THE SIMULATION IN THEVEHICULAR ENVIRONMENT

[1] Ms. Poonam M. Mahajan,[2] Dr. Gouri M. Patil

[1] Asst. Professor,[2] Asst. Professor

[1]Dept. of Computer Science and IT,

[1] Bhusawal Arts, Science and P. O. Nahata Commerce College, Bhusawal, India.

Abstract: The number of vehicles on the road has increased dramatically in recent years, resulting in an increase in the risk of accidents. To address this issue and make driving more fun and comfortable, a new technology based on Mobile Ad-hoc Wireless Network Infrastructure (MANET) has emerged. This kind of technology is called Vehicular Ad-hoc Networks (VANETs). It enables vehicles to interact with one another or with infrastructure in order to share various types of information and warning signals. Prior to actual implementation, simulation is one of the most effective alternatives for testing an approach on a machine to avoid potential dangers, realise the experiment results and behaviour in the real world. For a smooth simulation of a VANET, the simulation environment included both a network and traffic simulator.This study tries to focus at the simulation tools that are available for the VANET environment. This research uncovers the advantages and disadvantages of simulators, as well as their popularity. This study will assist researchers in selecting appropriate simulating tools based on their needs.

***Index Terms* - Network simulators, Routing Protocols, Simulation, VANET, Vehicular Mobility Generators.**

I. INTRODUCTION

Over the last two decades, wireless and mobile networks had a significant impact on our lives. Wireless networks have benefited people all around the world, from Bluetooth and wireless LANs to satellite networks. Vehicular Adhoc Networks are evolving at a breakneck pace as part of the future Intelligent Transportation System (ITS). Vehicle to vehicle (V2V) communication, vehicle to infrastructure (V2I) communication, or a mix of the two is referred to as vehicular networking. Mobile adhoc networks(MANETs) and VANETs have a lot of similarities. Both are adhoc networks of mobile nodes that can communicate wirelessly. As VANET nodes are vehicles rather than handheld devices, VANET simulation has less issues of energy consumption, storage capacity, and compute power.VANETs also have the added problem of a rapidly changing topology because vehicles move considerably quicker than humans. In order to evaluate VANET, simulation is required so that any changes can be made in simulation rather than in deployment, which would be prohibitively expensive[1].

Assuring security and privacy in the VANET is also a difficult task. As VANET applications are primarily concerned with ensuring safety and traffic management, simulation must account for the unique peculiarities of the automotive environment. To test and install the car network scenario, a huge sum of money, a large workforce, and a lot of labour are required. Computer simulation is used to evaluate apps and innovations before they are deployed. It is very feasible,risk-free, requires less money, time, and effort, and produces reports on many aspects of the programme. VANET simulation examines various aspects such as road topology, traffic signals, traffic congestion, and changing the vehicle's speed and mobility. This distinguishes VANET simulation from MANET simulation.

VANET vs. MANET

MANET (Mobile Ad-hoc Network) and VANET (Vehicular Ad-hoc Network) are the focal points of the networking area due to their usefulness .VANETs (vehicle ad hoc networks) are a subset of MANETs (Mobile Ad Hoc Networks). In VANET, where the patterns of motion are constrained by the topology of the road, the node's movement is predicted as the vehicle moves on the road in a random course. VANET is a potential topic of research that scientists, automakers, and mobile internet users are all interested in. A comparison of VANET and MANET is shown in Table 1[2].

Table 1.1 Comparisons between VANET and MANET

Properties	VANET	MANET
Bandwidth	1000 kps	100 kps
Transmission Range	500 m	100 m
Cost of production	Expensive	Cheap compared to VANET
Nodes Lifetime	Depends on vehicle lifetime	Depends on power sources
Acquisition of Position	Using GPS, Radar	Using Ultrasonic.

Vehicular networks, on the other hand, have distinct characteristics:

- Predictable mobility: Unlike MANETs, where node mobility is difficult to forecast, vehicular movements are normally restricted to roadways, making them relatively predictable. Vehicles are aware of their location and surroundings incorporation with computerised maps and location sensors such as the Global Positioning System (GPS). A vehicle's future position can be predicted based on its current location, speed, and route trajectory.
- Power and computational capabilities are limitless: Ascars' batteries are often recharged, they can offer continuous power to computer, sensing, and communication equipment, removing power limits. However, with the arrival of electric vehicles, this may become a problem because they should squander as little energy as possible.
- Large scale - Unlike most ad hoc network studies, which assume a small network size, vehicular networks can encompass an entire metropolitan region, including all of its vehicles and roadside equipment.
- Extremely dynamic - Vehicular networks are extremely dynamic, with various configurations depending on the situation. While vehicle density may be high in cities where vehicles move at low speeds, vehicle density may be low on highways where vehicles travel at high speeds and so have a low density.
- Partitioned network: Large vehicle gaps may develop in sparsely populated environments due to the dynamic nature of vehicular networks, resulting in multiple isolated groups of vehicles.
- Network topology and connectivity - Vehicles are continually moving and changing their positions, and they are only in communication range for a limited period of time. Links are formed and terminated at a rapid rate, resulting in rapid changes in network topology [3].

ROUTING PROTOCOL IN VANET

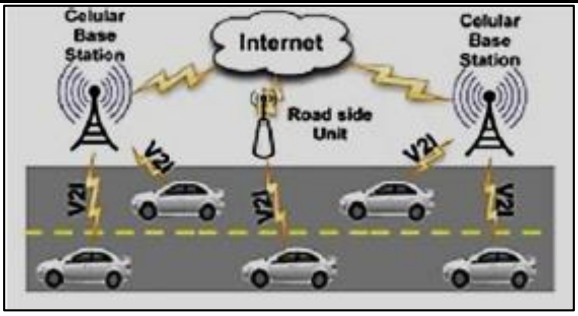
One of the key difficulties in the construction of a vehicular ad-hoc network is the development of a dynamic routing protocol that may help distribute information from one node (car) to another. On VANET, just a few protocols designed for the MANET environment have been tested.

However, lowering the time spent transporting data from one node to the next remains a difficulty. The creation of real-time VANET applications can be aided by overcoming certain hurdles in MANET protocols [3].

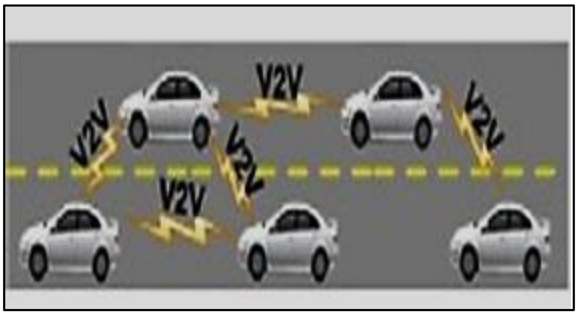
Routing protocols are divided into two categories based on (a) the network architecture and (b) the mode of operation.

(a) The Network Architecture

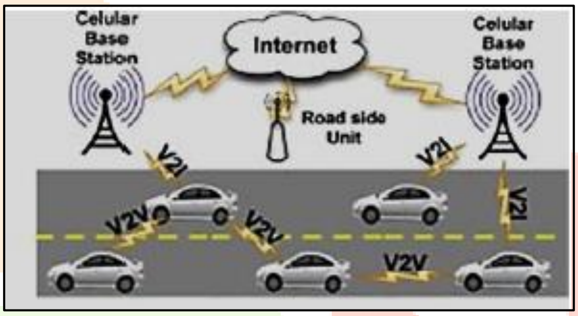
The network topology must be considered when performing the routing protocol, according to the type of architecture. As a result, as indicated in Figure 1, such architecture can be classified as ad hoc, infrastructure, or hybrid. Figure 1(a) depicts the infrastructure mode, also known as Vehicle-to-Infrastructure (V2I). In this mode, VANET simply employs access points to allow vehicles to connect to the Internet and traffic while also obtaining routing data information. Only the vehicles and the designated Access points communicate. This architecture allows access to mobile nodes, but it may be impracticable when infrastructure and edification costs are factored in. Figure 1(b) shows the ad hoc mode, commonly known as the Vehicle-to-Vehicle (V2V) mode. There is no requirement for a central access point because mobile networks can share information directly with each other. The nodes must be able to connect with one another and act as routers, sending messages from the origin to the destination. For example, such design could be used to prevent accidents at intersections without traffic signals, as well as to notify traffic jams, detect accidents, and detect traffic slowing. VANETs have also used hybrid architectures like the one shown in Figure 1(c), which combines infrastructure and ad hoc systems. This type of architecture is one of many that can be used to allow vehicles to communicate with one another, and it can also be connected to the Internet via fixed access points.



(a)



(b)



(c)

Figure 1.1 Network architectures for VANETs. (a) Infra-structured. (b) Ad hoc (c) Hybrid

(b) The Mode of Operation

The type of routing is included in the mode of operation. It is organised into two categories: topology and geography. Figure 2 shows an example of VANET routing protocol classification. Topology-based routing technologies can operate in either unicast or multicast mode and are divided into two categories: proactive and reactive. Proactive protocols can keep network topology information and continuously update network topology information in routing tables in nodes, regardless of the usage of the routes maintained. One example of proactive protocols is the Destination-Sequenced Distance Vector (DSDV) [13]. Reactive protocols do not save routing information and only check for routes when a source node wishes to send data packets to a destination node. One of the reactive protocols is Ad hoc On-Demand Distance Vector (AODV) routing.

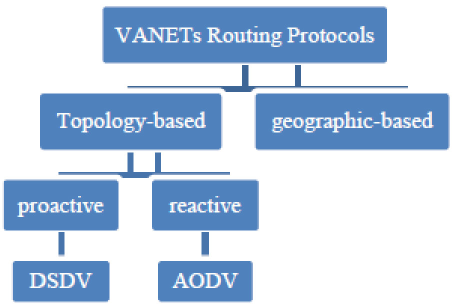


Figure 1.2 Routing protocols in VANETs

Challenges Faced by VANET Simulation

1. **Roadblocks -** Roadblocks obstruct communication and vehicular mobility. Signals may be disrupted when a wireless signal goes through barriers such as buildings, mountains, and other structures.
2. **Trip motion -** A trip is a path that connects a source and a destination. From source to destination, there are several options. It is up to the motorist to decide which path he will choose. As a result, driver interest must be taken into account in VANET simulation.
3. **Path motion -** A path is a collection of road segments taken by a vehicle on its journey from point A to point B. Drivers choose their route based on a variety of factors such as traffic congestion, distance, time of day, and so on.
4. **Traffic patterns -** The density of traffic varies depending on the time of day. The difference between rush hour and normal hour must be taken into account when simulating. This period has a significant impact on a vehicle's path decision.
5. **External Influence -** External impacts such as traffic accidents and temporary road closures must be factored into the simulation.
6. **Human driving patterns -** Drivers take into account both static and dynamic impediments such as other cars and pedestrians. As a result, choosing a good and efficient simulation environment is a critical task since it enables the creation of various communication environments such as city, highway, and town scenarios with ease and with less hazard and safety concerns. This encourages us to continue our investigation of current VANET simulators. The following section provides an overview of existing VANET simulators.

Simulation in VANET

Various VANET simulators used by the scientific community are mentioned in this section. Only freely available and open source software is presented in the study, allowing free access to the simulator source code. Three separate simulators are utilised to simulate the VANET in three phases. VANET simulators are a) what they're called. b) Network simulators; c) Generators of vehicular motion. The taxonomy of VANET simulation software is shown in Figure 3[4].

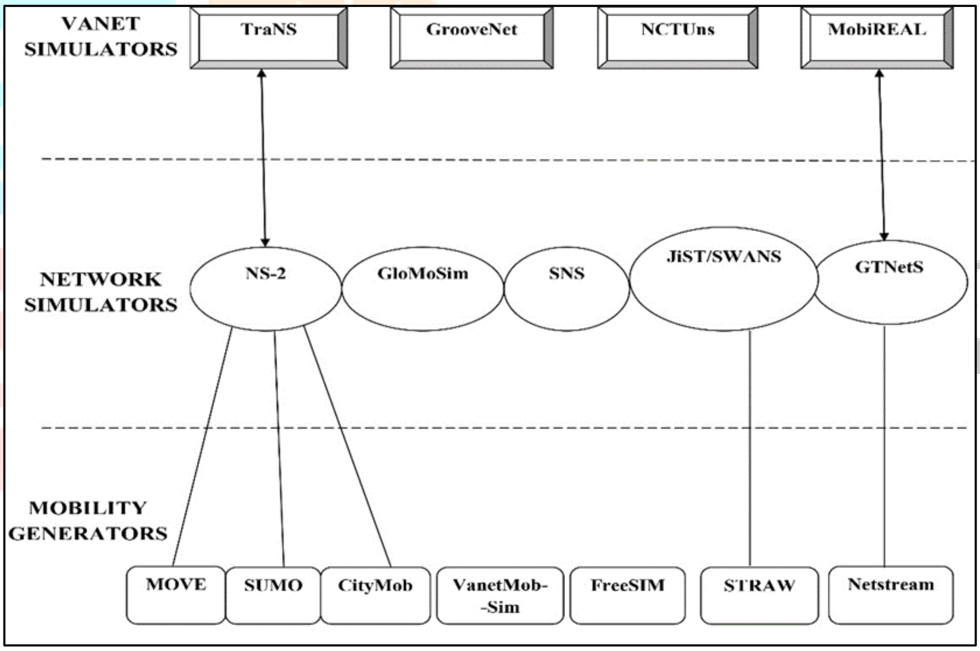


Figure 1.3 Taxonomy of VANET simulation software

In VANET simulations, Vehicular Mobility Generators are used to add realistic nature. The mobility generator produces traces of realistic vehicular mobility, which the network simulator uses as an input. The road model, scenario characteristics such as vehicle arrival and departure rates, maximum vehicle speed, and so on all are inputs to the mobility generator. It is easy to identify the location of each vehicle at each time instant for the duration of the simulation and their movement profiles using the trace information. SUMO, MOVE, Free SIM, STRAW, Netstream, and the others are examples. The next phase is network simulation, which comes after traffic simulation. Network simulators mimic computer network configurations and simulate source, destination, background load, route, links, and channels at the packet level. We can differentiate alternative network topologies, discover constraints, and overcome them using a network simulator before deploying and undertaking costly experiments in the real world. VANET extensions are required by MANET network simulators in order to be used in VANET simulation. NS-2, NS-3, SNS, J-SIM, OMNET++, GloMoSim, and more are examples. Finally, VANET simulators allow network simulation as well as traffic simulation.

II. VANET SIMULATORS

Network and mobility simulations are combined in VANET simulators. Network simulators are in charge of simulating communication protocols and message exchange, whereas mobility simulators are in charge of each node's movement, or mobility. Table 2 lists the simulators we discovered. We were able to discover some proprietary simulators among the open-source simulators. We concentrate our research on simulators that have a release date after 2015. We believe that obsolete tools are likely to be retired, and hence have a high likelihood of not supporting the most recent breakthroughs in VANET research, which is one of the factors we used in this paper's analysis.

Table 2.1 Comparison of VANET simulators

Simulator	Last release	License Type	Network simulator	Mobility generator
NetSim	2021	proprietary	own	SUMO
Veins	2020	open-source	OMNeT++	SUMO
Eclipse MOSAIC	2021	open-source	NS-3, OMNeT++, SND and Eclipse MOSAIC Cell	SUMO and VISSIM
VENTOS	2018	open-source	OMNeT++	SUMO
GrooveNet	2013	open-source	NS-2	Own
NCTUns	2010	proprietary	NS-2	Own
CityMob	2009	open-source	NS-2	Own
TraNS	2009	open-source	NS-2	SUMO

A. NetSim

NetSim is a discrete event simulator that can replicate a variety of networks, including wired, wireless, mobile, and sensor. There are three sorts of licences available for the simulator: pro, standard, and academic, the first two of which include support for VANET simulations. NetSim works with SUMO to simulate VANETs. The former is in charge of the WAVE wireless vehicle communication standard, while the latter is in charge of road traffic modelling. NetSim comes with a collection of network performance indicators, as well as graphs of link and application throughput. The metrics will differ based on the sort of network that is being simulated. Users can log details of each packet as it goes via the network using packet trace and event trace. Each network type corresponds to a simulator component with its own set of communication protocols. NetSim has a sufficient number of components and can connect real hardware to the simulation that is running live applications.

RF Propagation Models is one of the NetSim VANET modules worth mentioning. Path Loss, Shadowing Model, and Fading Model are all included in this module, and they're all necessary for predicting signal loss or signal encounters inside a building or in highly crowded locations over time. This module aids in making the simulation more realistic, as in a real-world scenario, the VANET will most likely encounter numerous impediments (e.g., buildings, heavy traffic) when signal communication is required. NetSim's architecture enables for the addition of additional components to meet the needs of the user and the appearance of new technologies. W-LAN, cognitive radio, LTE, MANET, Military radio, Internet of Things, VANET, Software-Defined Networking (SDN), and satellite communications have all been implemented on the simulator over the years. The SDN module, in particular, enables a variety of networking commands for controlling simulation, routing, access control, and so on, all of which may be run on the controller command line during the simulation.

B. Veins

Veins [5] is a free and open-source framework for simulating vehicle networks. It's built on top of OMNeT++ and SUMO. In general, the simulator creates an OMNeT++ node for each vehicle in the simulation and then matches node movements to vehicle movements in the road traffic simulator (i.e., SUMO). Both the network and mobility simulations can run in parallel in this situation. This is made possible via a bidirectional coupling achieved by the Traffic Control Interface (TraCI) [6], a defined connection protocol. TraCI allows OMNeT++ and SUMO to exchange messages (e.g., mobility traces) through TCP connections while the simulation is running.

The simulator comes with a number of extensions (currently over 17) that let you emulate different protocol stacks (e.g., IEEE 802.11p, ETSI ITS-G5 [7]) and applications (e.g., car platooning [8]). In conclusion, Veins is intended to be used as an execution environment for user-written programmes, making it easier to simulate new ecosystems and applications. As a drawback, it requires the proper operation of both SUMO and OMNeT++ in order to produce precise findings. Veins can offer unreliable results if one of those has a defect. Veins is compatible with Linux, Windows, and Mac OS.

C. Eclipse MOSAIC

Eclipse MOSAIC [9], formerly known as V2X Simulation Runtime Infrastructure (VSim- RTI), is an open-source, multi-scale, multi-domain simulation platform for evaluating novel linked and automated mobility solutions. The major goal of Eclipse MOSAIC is to give users the freedom to execute multiple V2X simulations with the simulators of their choosing. Eclipse MOSAIC accomplishes this by combining various simulations to provide a more realistic representation of car traffic, pollution, and wireless connectivity.

SUMO and PHABMACS for traffic simulation; NS3, OMNET++, and SNS for communication simulation; and Eclipse MOSAIC Application for application simulation are some of the simulators currently supported by Eclipse MOSAIC. Other simulators and analytic tools (particularly those from third companies) can be simply integrated as well.

Three essential pieces of the runtime architecture are required to integrate the simulators with Eclipse MOSAIC. It is the responsibility of the Federation Management to link each participating simulator to the runtime infrastructure. A federate is made up of two connectors: one for receiving data from the runtime infrastructure and the other for transmitting data to it. The simulation must be coordinated and the participating federates must be synchronised, hence Time Management is required. It ensures that each federate's events are processed in the correct order. The Interaction Management uses publish-subscribe paradigm to enable data communication amongst federates.

D. VENTOS

VENTOS [9] is a free and open-source simulator for evaluating vehicular networks (e.g., collaborative driving, automated cruise control, and platooning). It leverages SUMO and OMNET++ for mobility and network modelling, respectively, in the same way as Veins does. VENTOS, on the other hand, contains a number of prebuilt modules that make simulating complex application scenarios easier. The simulator, for example, includes algorithms for traffic signal control (TSC) and platoon management operations (e.g., merge, split, entry and leave).

VENTOS contains two specific modules: addNode and trafficControl, which make the process of generating traffic demands at a microscopic level easier. Users may simply add both stationary and mobile nodes to the simulation with the former, while the latter allows them to manipulate vehicle traffic by modifying its speed or specifying platooning procedures. The architecture of VENTOS is shown in Figure 8. In a hardware-in-the-loop (HIL) situation, the simulator can be enhanced to communicate with real OBUs and RSUs [9]. In this situation, each physical device linked to the computer has a virtual node in the simulation, and any action taken on the actual device is mirrored on its alias, and vice versa.

The hardware device can be linked to the VENTOS machine through an Ethernet port and communicated with via SSH sessions. The simulator does, however, demand that a tiny control programme (responsible for data management and control instructions) be run on the device, which may make it difficult to integrate with some boards. VENTOS, like Veins, relies on the seamless operation of both SUMO and OMNeT++ to produce accurate findings. The simulator is compatible with Linux, Windows, and Mac OS.

E. GrooveNet

GrooveNet[10] is an integrated simulator that allows large-scale simulations in street maps of any US metropolis using numerous models that characterise communication, transport, and traffic control. One-way streets and the altitude of the street are not currently indicated in the map database. GrooveNet is written in C++ and runs on Linux using the Qt graphics cross-platform toolkit. GrooveNet is based on the TIGER/Line 2000+ database format used by the US Census Bureau and can dynamically load counties at runtime. GrooveNet receives text files from map databases and translates topological data into a binary encoded file with a graph structure when it starts up.

III. NETWORK SIMULATORS

The exchange of messages among connected nodes is simulated using a network simulator. In the case of a VANET, vehicles and RSUs are typically involved, and wireless communications are the most common mode of communication. Ideally, all communication system components (e.g., the entire protocol stack) should be emulated, and the simulation should eventually include other relevant metrics (e.g., signal to noise ratio, packet error rates) [6]. Both network components and events are described in the network model. Components can include nodes, routers, switches, and connections. Data transfers and packet errors, for example, are two events that can occur.
A network simulator's output typically contains network level data, link metrics, and device metrics for a specified simulation scenario. Trace files used to be provided as well. These files provide a record of each simulation event that can be processed for additional analysis. Discrete-event simulation is used by the majority of network simulators [11]. A list of "pending events" is maintained in this method, which is then handled in sequence at each simulation step. Some events can lead to the emergence of new ones. The arrival of a packet at a node, for example, may cause a new packet to be sent. OMNeT++ [12], OPNET [13], NS2 [14], and NS3[15] and other network simulators are examples of available network simulators (some of which are frequently used in VANETs).

A. NS-2

The discrete event network simulator NS-2 [16] is one of the most widely used open source simulators. It's an object-oriented, discrete event driven network simulator created at the University of California, Berkeley. While the basic modules, such as protocols, channels, and agents, are written in C++, the simulation models are written in Tool Command Language (TCL) scripts. Users interact with NS-2 by writing TCL scripts that should include all of the instructions needed to set up the network topology, specify wireless parameters, and keep track of statistics.Several mobility simulators can generate vehicular trace files suitable for use in NS-2, from which an event trace output file and an animation trace file are generated. The NS-2 Network Animator (NAM) programme may use the animation file to offer simulation visualisation.

NS-2 supports both wired and wireless networks, with numerous MANET routing protocols and an IEEE 802.11 MAC layer implementation. NS-2 was utilised to investigate the impact of mobility on IEEE 802.11p MAC performance in [17]. NS-2, despite its widespread use in vehicular network research, does not scale well to large topologies (more than 300 nodes) and takes a lot of memory per simulated node [18]. It also lacks support for hierarchical models, making it extremely difficult to extend.

B. NS-3

NS-3 is a discrete-event network simulator widely used in research, succeeding the popular NS-2 but with a completely revamped simulation architecture. NS-3 aims to address the performance issues that plagued NS-2 while also improving realism. The simulation models in NS-3, like those in its predecessor, are implemented in C++. It does not, however, use TCL scripts to control simulations, as they are now implemented entirely in C++ (python bindings are also provided). The goal was to construct an open-source network simulator that was more scalable and easier to use, and that was consistent with present research needs.NS-3 incorporates the architectural concepts and code from GTNetS[15], a scalability-optimized simulator. Aside from performance enhancements, NS-3 was built to be extensible, allowing for the creation of additional models. The most significant benefit is the ongoing maintenance and rapid expansion provided by a broad developer community.

C. OMNET++

The Optical Micro-NetworksPlusPlus (OMNET++) [19] simulation system is based on discrete events. Because it gives both commercial and non-commercial licences, OMNeT++ is currently quite popular in academia and industry. It is used at academic institutions and non-profit organisations. It's a C++ simulation library and framework that's extendable and component-based. The components are then put together to form larger parts and models. A number of simple modules (cSimpleModule) make up the simulation, each of which represents the behaviour of a certain protocol. A compound module is made up of multiple simple modules that have been coupled together.

A host node, for example, can be a composite module made up of numerous simple modules that provide Internet protocol stack models. These modules are written in C++ and adhere to its object-oriented design principles. The Network Description Language (NED) language is used to combine these simple modules into compound modules, as well as to set up simulations and network topologies. NED also allows for the design of network module parameters as well as the specification of simulation parameters.When the simulation is compiled, the NED code is transparently transformed to C++ code. To create and control simulations, OMNeT++ offers both a graphical and command-line interface. It also includes a C++ kernel simulation library that may be integrated into other projects.

D. OPNET

Riverbed Technologies' Optimized Network Engineering Tool (OPNET) is a licenced network simulator that can be used to simulate both wireless and wired networks. For researchers and educational groups, an academic edition of OPNET called OPNET IT Guru is offered with a free licencing version.

Users can develop and test network communication devices, protocols, applications, and routing protocols using OPNET simulators. Different wireless technologies, such as IEEE 802.11, IEEE 802.15.1, IEEE 802.20, and satellite networks, are supported by OPNET simulators [20]. The ease of use, friendly graphical user interface (GUI), and high quality documentation and technical support are the key benefits of OPNET simulator.

With the help of the survey approach, we examined the popularity of the most commonly used network simulators, as shown in Table 3.

Table 3.1 Popularity Measurement of Network Simulators

	NS2	NS3	OMNeT++	OPNET
Average Number of citations found on Google Scholar	25	36	17	16
Average Number of citations found on IEEE Xplore	10	17	7	6
Popularity Index	35	53	24	22

For this study, we looked at 40 of the most frequently cited articles from the last five years that employed the specified network simulator for VANET simulation implementation, which were found on both Google scholar and IEEE Xplore. The average popularity index of NS2, NS3, OMNeT++, and OPNET was determined. By comparing these indices, we can see that NS3 has been the most used network simulator during the previous five years, surpassing NS2, OMNeT++, and OPNET.

IV. VEHICULAR MOBILITY GENERATORS

Vehicle mobility generators give the VANET simulation more realism. Road models and scenario characteristics are used as inputs, and the result is a trace of detailed traffic movement. The simulator must be compatible with the network simulator because it cannot be used alone for simulation and must be used in conjunction with it.

There are two types of mobility models: macroscopic and microscopic. Microscopic models treat each vehicle as a separate entity, focusing on its movement and behaviour in relation to others, whereas macroscopic models take into account quantities of interest such as traffic density, automobile velocity, traffic lights, highways, and streets.

A. SUMO

SUMO is a free, downloadable, microscopic, and continuous traffic simulation programme for managing huge road networks [21]. It allows for the modelling of intermodal traffic networks that include road cars, public transportation, and pedestrians. Finding routes, visualising, importing networks, and estimating emissions are just a few of the activities that SUMO can help with. OpenStreetMap, NavTeqMATsim, VISSIM, VISUM, and OpenDRIVE are among the import formats supported by SUMO.SUMO is written in C++ and Python, and it exclusively makes use of portable libraries. SUMO can also be used on a variety of platforms, including (Windows, Linux, or macOS), and can be used in conjunction with a network simulator such as (OMNeT++, NS-2, or NS-3).

B. MOVE

MOVE (MObility model generator for VEhicular networks) [22] creates realistic VANET mobility models quickly. SUMO is the foundation for MOVE. MOVE generates a mobility trace file that provides information on realistic vehicle movements and can be utilised right away by network simulation tools like ns-2 or GloMoSim. MOVE also has a graphical user interface (GUI) that allows users to quickly create realistic simulation situations without having to write simulation scripts or learn about the simulator's internal workings.

C. CityMob

The CityMob mobility generator is an open source simulator that allows users to construct urban mobility scenarios, including the ability to simulate accidents and broadcasting events depending on flooding. It generates vehicle mobility traces that are compatible with the NS-2 network simulator and is written in C under the GNU licence.

Three different mobility models are proposed by CityMob[23]:

(1)Simple model (SM): Without semaphores, vertical and horizontal mobility patterns are available.

(2)Manhattan model (MM): the city is modelled after the Manhattan grid, with all streets being two-way and car movements restricted by lanes. It allows semaphores to be placed at arbitrary locations. When a vehicle is blocked by a semaphore, it remains still until the semaphore turns green.

(3)Downtown model (DM): To make the traces more realistic, traffic density is added to the Manhattan model. In CityMob, DM is the optimum mobility model.

D. STRAW

STreetRAndom Waypoint provides accurate simulation results based on mobility in a car in a real-world US city model with real-world traffic. The existing STRAW[24] implementation was designed for the JiST / SWANS discrete event simulator, hence traces of its mobility are unreliable. Other network simulators, such as NS-2, were also included. The C3 (Car-to-Car Cooperation) programme includes STRAW. For proper network modelling in this scenario, an extra practical mobility model with a suitable level of detail for vehicle networks is required.The problem of this strategy is that, like in the case of North American countries, the model will only recognise node motions to streets predetermined by map information. Individual vehicle congestion will be restricted, and traffic control procedures will be simplified.

V. CONCLUSION

Researchers have been working to produce accurate and realistic simulation tools as VANETs have gained in popularity and attention. We investigated the present state of VANET simulators in depth in this paper, particularly in terms of their support for novel technologies as well as safety and security methods. When evaluating the VANET simulators, Veins appears to support these characteristics the best at the time of writing this study.

SUMO, MOVE, CityMob, and STRAW were among the mobility generators investigated. Software features and traffic model support are all good in SUMO, MOVE, and STRAW. In terms of software functionality and traffic model support, CityMob is excellent. SUMO has good software qualities, but it is lacking in other areas.

NS-2, NS-3, OMNET++, and OPNET are among the network simulators that have good software support. However, NS-2 has limited scalability, and OMNET++ is more difficult to use than the rest. In fact, none of the network simulators specifically address VANET scenarios and requirements, such as 802.11p considerations, obstructions, vehicular traffic flow, and so on.

Finally, in order to improve the quality of VANET simulations, we identified a number of difficulties that must be solved. As a result of our research, we discovered that NS3 is the most often used network simulator due to its performance and ease of use.

REFERENCES

[1] B. Lakshmipraba and V. Sivakumar, "Vehicular Network Simulators for VANET:Review," *International Journal of Innovative Research in Science, Engineering and Technology,* vol. 6, no. 5, pp. 9645-9648, 2017.

[2] I. A. Aljabry and G. A. Al-Suhail, "A Survey on Network Simulators for Vehicular Ad-hoc Networks (VANETS)," *International Journal of Computer Applications(0975 – 8887),* vol. 174, no. 11, pp. 1-9, 2021.

[3] P. M. Mahajan and G. M. Patil, "Routing Protocols in VANET: A Comprehensive Study," *International Journal of Computer Research and Technology (IJCRT),* vol. 7, no. 1, pp. 15-18, 2021.

[4] R. S. Batth and K. S. Mann, "A Comparative Analysis of VANETS Simulator," *International Journal of Emerging Technologies in Engineering Research (IJETER),* vol. 4, no. 7, pp. 222-229, 2016.

[5] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer and M. Segata, "Veins: The Open Source Vehicular Network Simulation Framework," *Recent Advances in Network Simulation (Springer),* pp. 215-252, 21 May 2019.

[6] J. S. Weber, M. Neves and T. Ferreto, "VANET simulators: an updated review," *Journal of the Brazilian Computer Society,* vol. 27, no. 8, p. 31, 2021.

[7] R. Riebl, H.-J. Günther, C. Facchi and L. Wolf, "Artery: Extending Veins for VANET applications," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Budapest, Hungary, 3-5 June 2015.

[8] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler and R. L. Cigno, "Plexe: A platooning extension for Veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, Paderborn, Germany, 3-5 Dec. 2014.

[9] M. Amoozadeh, B. Ching, C.-N. Chuah, D. Ghosal and H. M. Zhang, "VENTOS: Vehicular Network Open Simulator with Hardware-in-the-Loop Support," *Procedia Computer Science,* vol. 151, pp. 61-68, 2019.

[10] R. Mangharam, D. Weller, R. Rajkumar, P. Mudalige and F. Bai, "GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle

Networks," in *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, San Jose, CA, USA, 17-21 July 2006.

[11] R. J. Fernandes, Large-scale Simulation of Vehicular Ad Hoc Networks, University of Porto, Minho and Aveiro: MAP-Tele Doctoral Programme in Telecommunications, 2014.

[12] K. Wehrle , M. Güneş and J. Gross, "OMNeT++," in *Modeling and Tools for Network Simulation*, Berlin, Springer, 2010, pp. 35-59.

[13] S. Yang, R. He, Y. Wang, S. Li and B. Lin, "OPNET-based modeling and simulations on routing protocols in VANETs with IEEE 802.11p," in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, Shanghai, China, 2014.

[14] A. K. Ali, I. Phillips and H. Yang, "Evaluating VANET routing in urban environments," in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, Vienna, Austria, 2016.

[15] R. Fernandes and M. Ferreira, "Scalable VANET Simulations with NS-3," in *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, Yokohama, Japan, 2012.

[16] J. Li, Y. Zhang, J. Zhao, Y. Wang, X. Ma and W. Wu, "NS-2 Simulation of VANET for Safety Applications: Issues and Solutions," in *ACM The 8th International Conference on Computer Modeling and Simulation (ICCMS 2017)*, Canberra, Austrlia, 2017.

[17] W. Alasmary and W. Zhuang, "The Mobility Impact in IEEE 802.11p Infrastructureless Vehicular Networks," in *2010 IEEE 72nd Vehicular Technology Conference - Fall*, Ottawa, ON, Canada, 2010.

[18] E. Weingartner, H. vom Lehn and K. Wehrle, "A Performance Comparison of Recent Network Simulators," in *2009 IEEE International Conference on Communications*, Dresden, Germany, 2009.

[19] A. Virdis, G. Stea and G. Nardini, "SimuLTE - A modular system-level simulator for LTE/LTE-A networks based on OMNeT++," in *4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*, Vienna, Austria, 2014.

[20] M. Chen, Y. Miao and I. Humar, "Introduction to OPNET Network Simulation," *OPNET IoT Simulation*, pp. 77-153, 2019.

[21] K. G. Lim, C. H. Lee, R. K. Y. Chin, K. B. Yeo and K. T. K. Teo, "SUMO enhancement for vehicular ad hoc network (VANET) simulation," in *IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Kota Kinabalu, Malaysia, 2017.

[22] K.-C. Lan, "MOVE: A Practical Simulator for Mobility Model in VANET," in *Telematics Communication Technologies and Vehicular Networks: Wireless Architectures and Applications*, IGI Global, 2010, pp. 355-368.

[23] F. J. Martinez, J. C. Cano, C. T. Calafate and P. Manzoni, "Citymob: a mobility model pattern generator for VANETs," in *ICC Workshops-2008 IEEE International Conference on Communications Workshops-IEEE*, Beijing, China, 2008.

[24] B. Ahmed, A. W. Malik, T. Hafeez and N. Ahmed, "Services and simulation frameworks for vehicular cloud computing: a contemporary survey," *EURASIP Journal on Wireless Communications and Networking*, 2019.

[25] S. Malik and P. K. Sahu, "A comparative study on routing protocols for VANETs," *Elesevier ScienceDirect Heliyon 5*, 2019.

[26] M. A. Iqbal, M. Aleem, M. Ibrahim, S. Anwar and M. A. Islam, "Amazon cloud computing platform EC2 and VANET simulations," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 30, no. 3, pp. 127-136, 2019.

[27] B. Samatha, K. Raja Kumar and N. Karyemsetty, "Design and Simulation of Vehicular Adhoc Network using SUMO and NS2," *Advances in Wireless and Mobile Communications*, vol. 10, no. 5, pp. 1207-1219, 2017.

[28] C. H. Lee, K. G. Lim, M. K. Tan, R. Y. Chin and K. T. Teo, "Hybrid simulation network for vehicular ad hoc network (VANET)," *ICTACT Journal on Communication Technology*, vol. 9, no. 1, pp. 1686-1695, 2018.

[29] S. Sridharan and S. Ramprakash, "Simulation of Vehicle Adhoc NET Work (VANET) using Network Simulator (NS-2) Tools," *International Journal of Engineering Research & Technology (IJERT)*, no. Special Issue-2016, pp. 1-4, 2016.

[30] S. A. B. Mussa, M. Manaf, K. Z. Ghafoor and Z. Doukha, "Simulation Tools for Vehicular Ad Hoc Networks: A Comparison Study and Future Perspectives," in *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2015.

[31] F. Kaisser , C. Gransart and M. Berbinea, "Simulations of VANET Scenarios with OPNET and SUMO," in *International Workshop on Communication Technologies for Vehicles*, Berlin, Heidelberg, 2012.

[32] C. Tripp-Barba, A. Zaldívar-Colado and L. Urquiza-Aguiar, "Survey on Routing Protocols for Vehicular Ad Hoc Networks Based on Multimetrics," *Electronics*, vol. 8, no. 10, 2019.

[33] M. J. Silva, G. I. Silva, F. A. Teixeira and R. A. Oliveira, "Temporal Evolution of Vehicular Network Simulators: Challenges and Perspectives," in *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, 2018.

[34] W. Angeles, V. P. Borin, A. Munaretto and M. Fonseca, "The Impact of Propagation Models in the Performance of Ad Hoc Routing Protocols for Urban VANET," in *IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Montreal, QC, Canada, 2016.

[35] G. K. Jain, S. Kumar, N. Gupta and D. Bhardwaj, "The Simulation Tool of Vehicular Ad-hoc Networks," *IJLTEMAS*, vol. II, no. VI, pp. 76-80, 2013.

[36] A. Hassan, "VANET Simulation," School of Information Science, Computer and Electrical Engineering Halmstad

University, 2009.

[37] E. Spaho, L. Barolli, G. Mino, F. Xhafa and V. Kolici, "VANET Simulators: A Survey on Mobility and Routing Protocols," in *2011 International Conference on Broadband and Wireless Computing, Communication and Applications*, Barcelona, Spain, 2011.

[38] R. Hajlaoui, T. QMoulahi and H. Guyennet, "Vehicular ad hoc networks: From simulations to real-life scenarios," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1.1, p. 169 – 174, 2019.

[39] O. T. T. Kim, V. Nguyen and C. S. Hong, "Which network simulation tool is better for simulating Vehicular Ad-hoc network?," in *Proceedings of the Korean Society of Information Sciences*, 2014.

[40] P. Karunakar, J. Matta, R. P. Singh and O. R. Kumar, "Analysis of Position Based Routing VANET Protocols using Ns2 Simulator," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 5, pp. 1105-1109, 2020.

[41] D. Rathi and R. Welekar, "Performance evaluation of AODV routing protocol in VANET with NS2," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. Special Issue on Advances and Applications in the Internet of Things and Cloud Computing, pp. 23-27, 2017.

[42] D. Zhang, T. Zhang and X. Liu, "Novel self-adaptive routing service algorithm for application in VANET," *Applied Intelligence*, vol. 49, p. 1866–1879, 2018.

[43] P. Manickam, K. Shankar, E. Perumal, M. Ilayaraja and K. Sathesh Kumar, "SecureData Transmission Through Reliable Vehicles in VANET Using Optimal Lightweight Cryptography," in *Cybersecurity and Secure Information Systems*, Springer, 2019, pp. 193-204.

[44] E. Moridi and H. Barati , "RMRPTS: a reliable multi-level routing protocol with tabu search in VANET," *Telecommunication Systems*, vol. 65, no. 1, p. 127–137 , 2017.

[45] B. Cherkaoui, A. Beni-Hssane, M. E. Fissaoui and M. Erritali, "Road traffic congestion detection in VANET networks," *Procedia Computer Science*, vol. 151, pp. 1158-1163, 2019.

[46] X. Ji, H. Yu, G. Fan and W. Fu, "SDGR: An SDN-Based Geographic Routing Protocol for VANET," in *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Chengdu, China, 2016.

[47] S. Dahmane and P. Lorenz, "Weighted Probabilistic Next-Hop Forwarder Decision-Making in VANET Environments," in *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, 2016.

[48] F. L. Lugayizi, B. M. Esiefarienrhe and A. Warren, "Comparative evaluation of QoS routing in VANET," in *IEEE International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Durban, South Africa, 2016.

[49] A. Setiabudi, A. A. Pratiwi, D. Perdana and R. F. Sari, "Performance comparison of GPSR and ZRP routing protocols in VANET environment," in *IEEE region 10 symposium (TENSYMP)*, 2016.

[50] A. Bengag and E. Mohamed, "Performance evaluation of VANETs routing protocols using SUMO and NS3," in *IEEE 5th International Congress on Information Science and Technology (CiSt)*, 2018 .

[51] A. Mukhopadhyay, S. Raghunath and M. Kruti, "Feasibility and performance evaluation of VANET techniques to enhance real-time emergency healthcare services," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016.

[52] D. Anadu, C. Mushagalusa, N. Alsbou and A. S. Abuabed, "Internet of Things: Vehicle collision detection and avoidance in a VANET environment," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2018.

[53] E. Limouchi and I. Mahgoub, "BEFLAB: Bandwidth efficient fuzzy logic-assisted broadcast for VANET," in *IEEE Symposium Series on Computational Intelligence (SSCI*, 2016.

[54] Y. Yang, H. Si-Yuan and C. Ha-Bin , "Comparison and evaluation of routing protocols based on a collaborative simulation using SUMO and NS3 with TraCI," in *International Conference on Information System and Artificial Intelligence (ISAI)*, 2016.

[55] W. Liu, X. Wang, W. Zhang, L. Yang and C. Peng, "Coordinative simulation with SUMO and NS3 for vehicular ad hoc networks," in *22nd Asia-Pacific Conference on Communications (APCC)*, 2016.

[56] A. M. Shaban, S. Kurnaz and A. M. Shantaf, "Evaluation DSDV, AODV and OLSR routing protocols in real live by using SUMO with NS3 simulation in VANET," in *International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020.

[57] "Performance analysis and comparison of the DSDV, AODV and OLSR routing protocols under VANETs," 2018.

[58] D. Su and A. Sanghyun , "Autonomous platoon formation for VANET-enabled vehicles," in *International Conference on Information and Communication Technology Convergence (ICTC)*, 2016.

[59] K. Xiao, K. Liu, J. Wang, Y. Yang, L. Feng, J. Cao and V. Lee, "A fog computing paradigm for efficient information services in VANET," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[60] C. Obermaier, R. Riebl and C. Facchi, "Fully reactive hardware-in-the-loop simulation for VANET devices," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[61] A. Korichi, A. Lakas and M. E. A. Fekair, "An efficient QoS-compliant routing scheme for VANET," in *5th International conference on electronic devices, systems and applications (ICEDSA)*, 2016.

[62] J. Cui, W. Lu , H. Zhong, J. Zhang, Y. Xu and L. Liu , "Edge computing in VANETs-an efficient and privacy-preserving cooperative downloading scheme," in *IEEE Journal on Selected Areas in Communications 38*, 2020.

[63] E. Karim, "Poster: Prext: Privacy extension for veins vanet simulator," in *IEEE Vehicular Networking Conference (VNC)*, 2016.

[64] S. Joanne and I. Mahgoub, "A geographical hybrid solution for inter-vehicular communication in VANET," in *nternational Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016.

[65] T. Petrov, M. Dado, P. Kortis and T. Kovacikova, "Evaluation of packet forwarding approaches for emergency vehicle warning application in VANETs," in *ELEKTRO*, 2018.

[66] P. Sharma, H. Liu, H. Wang and S. Zhang, "Securing wireless communications of connected vehicles with artificial intelligence," in *IEEE international symposium on technologies for homeland security (HST)*, 2017.

[67] F. Michaeler and C. Olaverri-Monreal, "3D driving simulator with VANET capabilities to assess cooperative systems: 3DSimVanet," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[68] N. Malik, P. Nanda, A. Arora, X. He and D. Puthal, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," in *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018.

[69] M. Syfullah and J. M.-Y. Lim, "Data broadcasting on Cloud-VANET for IEEE 802.11 p and LTE hybrid VANET architectures," in *3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 2017.

[70] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau and D. Gruyer, "Performance evaluation of AODV, DSR, GRP and OLSR for VANET with real-world trajectories," in *15th International Conference on ITS Telecommunications (ITST)*, 2017.

[71] A. Elbery and H. A. Rakha, "VANET communication impact on a dynamic eco-routing system performance: Preliminary results," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, , 2018.

[72] T. Saeed, Y. Mylonas, A. Pitsillides, V. Papadopoulou and M. Lestas, "Modeling probabilistic flooding in vanets for optimal rebroadcast probabilities," in *IEEE Transactions on Intelligent Transportation Systems 20*, 2018.

[73] M. S. Ahmed, M. A. Hoque and P. Pfeiffe, "Comparative study of connected vehicle simulators," in *SoutheastCon 2016*, 2016.

[74] M. Ushakova, Y. Ushakov, I. Bolodurina, D. Parfenov, L. Legashev and A. Shukhman, "Research of productivity of software configurable infrastructure in vanet networks on the basis of models of hybrid data transmission devices," in *International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC)*, 2020.

[75] M. Abdelgadir, R. Saeed and A. Babiker, "Vehicular Ad-hoc Networks (VANETs) dynamic performance estimation routing model for city scenarios," in *International Conference on Information Science and Communications Technologies (ICISCT)*, 2016.

[76] R. Brendha and V. S. J. Prakash, "A survey on routing protocols for vehicular Ad Hoc networks," in *4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2017.

[77] Y. Zhang, Y. Cao, Y. Wen, L. Liang and F. Zou, "Optimization of information interaction protocols in cooperative vehicle-infrastructure systems," in *Chinese Journal of Electronics 27*, 2018.

[78] S. A. Ahmad and M. Shcherbakov, "A survey on routing protocols in vehicular adhoc networks," in *9th international conference on information, intelligence, systems and applications (IISA)*, 2018.

[79] H. H. Taha and A. M. Alhassan, "Performance evaluation of AODV, OLSR and GRP routing protocols in vehicular highway environment," in *International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2018.

[80] R. Kumar and R. Rishi, "Routing and security analysis in vehicular ad-hoc networks (VANETs)," in *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, 2016.

[81] F. J. Martinez, J.-C. Cano, C. T. Calafate, P. Manzoni and C. K. Toh, "A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 813 - 828, 2009.

[82] B. Schünemann, "V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 55, no. 14, p. 3189–3198, 2011.

[83] M. M. A. Elgazzar and A. Alshareef, "VANET Simulator: Full Design Architecture," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 3, pp. 2024-2034, 2020.