An Exhaustive Consideration Of Wired and Wireless Network Simulators

Jaganath. M, Vasanth. R, Malarselvi. G

***Abstract:** In the present world, the network simulators are most important platform which is high price and great value to make different communications frequently all over the world. In the computer network, a different types of network simulators are available. During the past years, bunch of network simulation tools has been developed have impact on the growth of communication field. Network simulators are having the support for the most of general approval technologies and networks in use today such as Wireless LAN's and Wireless Sensor Networks, the Internet of things (IOT) and so on. In this Paper, we are going to discuss about group of current open-source and commercial network simulators with some features. Main features focused here are, the mobility support, power consumption, OSI layer support and wireless specifications. Under the brief discussion of every simulators, the properties were summarized in the following table. We hope that the end of this paper, comparisons of wired and radio network simulators with these features which is used to select the proper network simulators for future research needs.*

***Keywords:**NS-2,NS-3,Cooja,Castilia,DRMsim,EMstart,GloMosim,Groovenet,Jsim,Netsim,OMNET++,OPNET,PDNS,Qualnet,REAL,SENS,SHAWN,TRANS,TOSSIM,NEST.*

I. INTRODUCTION

Computer network, which is used to sending or sharing the data through the nodes. The node is communicating over the cable media such as wired or optic cables or wireless media such as Wireless Fidelity (Wi-Fi). In the computer network, network simulation is a method whereby simulator tools show the behavior of a network by calculating the communication between the various network entities (routers, nodes, switches, access points and links). The network simulators are providing the various measures needed for researchers in all over the world. Whereas in general, very difficult to find out the proper simulator for simulation work.

The most important strategy concentrated in simulators are on the basis of availability, easiness, scalability and other important properties (user interface, data manipulation, graphical display etc.) of them. for example: NS2, NS3, Cooja, Castalia, DRM Sim, EMstar, GloMoSim, GrooveNet, J-SIM, NetSim, OMNeT++, OPNET, PDNS, QualNet, REAL, SENS, SHAWN, NEST, Trans, TOSSIM etc. Finally, we have concluded that every simulator and its properties are shown in the following Table. At end of the result shown here, it easy for researcher to decide and choose the necessary simulators for their implementations easily.

Revised Manuscript Received on July 05, 2019.

Jaganath. M, Department of Software Engineering, SRM Institute Of Science and Technology, Kattankulathur, Tamilnadu.

Vasanth. R, Department of Computer Science and Engineering, M. Kumarasamy College Of Engineering, Karur, Tamilnadu.

Malarselvi. G, Department of Computer Science and Engineering, SRM Institute Of Science and Technology, Kattankulathur, Tamilnadu.

II. OUTLINE FOR COMPUTER NETWORKS

A. Medium

Electromagnetic waves and radio waves are used to send the data into wireless communications. In the simulation, the electromagnetic waves have too much fading, particularly in the large frequencies and also it need the huge transmission energy and big antenna for communications. Radio signal, which is used to reach the goal of data rate communications in the electromagnetic field, but electromagnetic waves are quickly moving to the different directions and listening in the networks. There is a different in the two waves are following: electromagnetic waves used to short distance interaction and radio signals are used to long distance communications [1]. This is the reason medium of transmission used for communication networks.

B. Environment

Wired and wireless communication have a different type of parameters. Physical medium implementation technique is used a variety of environment in the network. The applications are done in the channel modelling and it has a various environment technique such as: distance, component, modularity, efficiency and closeness.

C. Node Mobility

Wired and wireless communication network have some significant purpose network nodes with the medium mobility. Depends on the efficient parameters, it required some protocols to be implemented in the protocol stack. Most of the host computer has numerous physical network nodes in the internet or intranet. It is monitored by the different IP address and physical node.

D. Physical Layer

In the network channel modeling is given high importance, as electromagnetic or radio waves are the best match for physical layer communication. It has only quality channel modeling supports the top layer protocols like Medium access control and Routing protocols. Parameters like signal fading, receiver energy, segment loss, propagation delay, transmission loss, background noise, node depth, density, mobility should all be grant for active channel modeling.

E. MAC Protocol

In the wireless communication, wake-up systems have used to decrease the power consumption and development of MAC protocols for networks shall contain the utilization of modem features.MAC protocols have two prime requirements are as follows: localization and synchronization.

F. Network Protocol

Wireless sensor networks are not applicable to network protocols for land-based sensor networks. The protocol suite is very important to every research and computer networks has some special features.

G. Application Layer

Application layer should give their help to the upcoming features in the communication networks are reusability, performance, scalability, availability and support for high end scripting languages to find experiments and operation results.

01. Ns-02

Ns2 is most popular network simulators and it is provided as open source. Ns -2 is written by two programming languages are C++ and tool command language. It has the option to connect through wireless and wired networks. It runs on the platforms such as Linux, Free BSD, Mac OS and Windows operating system.NS-2 simulator provide the support for network and data link layer. It also gives mobility support. NS2 provides support for both wired and radio simulation of group events and approved designs such as TCP, UDP and so on. NS-2 was undergone growth from the ideology of true simulator. It is a discrete event network simulator and the model of performance in the different simulation nodes and OTcl language scripts that is specifying in the network topology. NS-2 is one of the general approval simulators, because of its capability to make ready adjustments and to break the units into parts of behavior [2]. Network animator is used in the Ns-2 simulator to give the graphical view of the network. The simulation process is considered only a virtual, not a real time simulation.

02. Ns -03

The Ns3 network simulator marked primarily to the purpose of observations and primarily used for educational purpose. Users has been given the permission to use the software under GNU license. Several users are using NS3 to design non-Internet-based systems. The simulators are entirely written in C++. NS3 provides a strong library which is useful for the user to do their work by getting ready ns-3 it-self. In the Ns-3, the sockets are giving the support to both simulation and emulation process. In the wired communication the NS-3 simulator gives equipment model of a basic network and Ethernet is utilizing the CSMA/CD as a protocol scheme with exponentially increasing back-off to contend for split transmission medium. It is communicating with actual systems, also provide the support for a real-time scheduler that facilitates a number of simulation in the loop [3]. The Ns-3 simulation provides the support to research on both IP and non-IP network.

02. Cooja

In this simulator is an open source software and run with low energy consumption. Contiki sensor node operating system has the same features like COOJA simulator. Cooja simulator has the support of Network layer from the OSI layer support system. It is working on the specific wireless connection of IEEE 802.15.4.Windows, Linux, Contiki operating systems are the support operation system to the COOJA simulator. A virtual machine is hold on the Contiki development environment and the simulator must choose the propagation model while beginning of the simulation [4]. The time line are available in this simulator to see the results from the user end. MPSim used to make the emulation at a possibility of bit level and got the instruction from the machine code.

04. Castalia

Castalia network simulator is an open source wireless network simulator. Also it is an application-level network simulator for Wireless Sensor Network depends on OMNeT++ simulator. It should be modeled with the 802.15.6 wireless specification for transferring the message through the network. It can run on the following operating systems: Linux, Microsoft Windows. And it has the mobility support and with the low power consumption. It can be used to perform on several platform features for particular applications, since it is huge parametric and simulate a wide range of platforms [5].

Limitations: Castalia can be announced that it doesn’t have the sensor-platform particularly. Castalia simulator is providing a generic reliable and realistic infrastructure for the beginning order validation of an algorithm, before shifting to apply on a special sensor platform.

05. DRM Sim

DRMSim (Dynamic Routing Model Simulator) is a Discrete-Event Simulation software that is developed with the purpose of evaluating the routing models at a big scale. It is the open source tool. Its development has begun in the context of a study focusing on dynamic compact routing. DRMSim is written Java programming language and the simulation software in which follows the design philosophy made to compromise on the quality of the code written so that extensibility and reusability are maximized [6]. DRMSim which is running on the following operating systems: UNIX, Linux, Mac OS. DRMsim is the wireless simulator with the mobility support. Also provide the support on the Network and Data Link Layer from the OSI model. The following Table 1 will explain the wired and wireless network simulator comparisons based on the general information.

Table 1: Wired and Wireless Network Simulator

S.No	Simulators Name	OSI Layer support	Open Source / Commercial	Wired / Wireless Specification	Supported OS	Mobility Support	Power Consumption (good/ave/low)	Wired/ wireless support
1.	NS-2	Network and Data Link Layer	Open source	IEEE 802.15.6	Linux, FreeBSD, macOS, Windows	Yes	NA	Wired and wireless support
2.	NS-3	Network(TCP/IP)	Open source	IEEE 802.11& 802.11a	Linux, FreeBSD, macOS, Cygwin(windows)	Yes	NA	Wired and wireless support
3.	COOJA(Contiki)	Network(TCP/IP)	Open Source	IEEE 802.15.4	Windows, Linux, Contiki Operating System		Low	Wireless
4.	Castalia	Data Link and Network Layer	Open Source	IEEE 802.15.6	Linux, Windows	Yes	Low	Wireless
5.	DRMSim	Network and Data Link Layer	Open Source	NA	UNIX,Linux, Mac OS	Yes	NA	Wireless
6.	EMStar	Network Layer	Open Source	NA	Linux, windows	NA	NA	Wireless
7.	GLoMoSim	Physical, Data link layer, Network, Transport, Application Layer	Open Source	802.11	Windows, FreeBSD	Yes	NA	Wireless
8.	Groovenet	Data Link and Transport Layer	Open Source	IEEE 802.11a/b/g	Linux OS with Kernel version 2.6*	Yes	Low	Wireless
9.	Jsim	Network and Data Link Layer	Open Source	Mac 802.11	Linux, Windows & MacOS	Yes	Low	Wired
10	NetSim	Network and Data Link and physical Layer	Commercial	WLAN- IEEE 802.11 a/b/g/e/n	Windows 10, Windows 8, Windows 7,Windows Vista	Yes	Low	Wired and wireless support
11	OMNET++	Network and Datalink Layer Support	Open Source; Commercial	IEEE 802.15.6	Linux, Unix-like systems, Windows(XP,Win2K)	Yes	Low	Wireless
12	OPNET	Network Layer Support	Commercial	IEEE 802.11	Windows	Yes	NA	Wired and wireless support
13	PDNS	Network Layer Support	Open Source	802.11	Intel Linux, Intel Solaris,Mac OS X, Microsoft Windows		NA	Wired and wireless support
14	Qual Net	Physical Layer and Network & Application Layer	Commercial	IEEE 802.11s	Intel Linux, Intel Solaris, Mac OS X, Microsoft Windows	Yes	Low	Wired and wireless support
15	REAL	Network Layer Support	Open Source		Unix,BSD, Solaris, SunOS,Windows	NA	NA	NA
16	SENS	Physical,Network& Application Layer	Open Source	IEEE 802.11	Linux,Mac OS X, Windows XP with Cygwin	NA	Low	Wired and wireless support

Comparisons Based On The General Information

S.No	Simulators Name	OSI Layer support	Open Source / Commercial	Wired / Wireless Specification	Supported OS	Mobility Support	Power Consumption (good/ave/ low)	Wired/ wireless support
17	SHAWN	Physical , Data link and Network Layer Support	Open source	NA	NA	Yes	High	Wireless
18.	NEST	Network Layer Support	Open source and Under GNU	NA	Unix , Windows and Linux	Yes	NA	NA
19.	TRANS	Physical, Network and Data link Layer Support	Open Source	NA	Linux, Windows(trace-generation mode)	Yes	NA	Wireless
20.	TOSSIM	Physical and Network Layer Support	Open Source	NA	Linux, Cygwin(windows)	NA	High	wireless support

06. EMstar

EmStar is an emulator particularly designed for Wireless sensor network developed in a programming language of C, and it is a first software built by University of California, Los Angeles. EmStar is a trace-driven emulator and it is running in real-time simulation. EMstar which is running on the following operating systems: Linux, Microsoft windows. EMstar is the simulator is wireless simulator which is moving may be the mobility model support [7]. Also provide the support on the Network Layer from the OSI model. Here the execution is done using the same configuration files and same code.

Limitations: It doesn't support wide range of sensors simulation and decrease the reality of simulation with the limited scalability.

07. GloMosim

GloMosim is an emulator specially designed for Wireless sensor network developed in C++, it is a simulator which is running on the following operating systems: FreeBSD, Microsoft windows. GloMosim is a wireless simulator which has the mobility model support. Glomosim is well known as Global Mobile Information System simulator. It is a public domain simulator and it developed by UCLA. It has scalable environment for wide range of wireless and wired communication networks. GlomoSim is going to be used in parallel discrete-event simulation. This capability is getting from the Parsec. Efficient parallel simulators model decomposition or portioning to reach load balance [8].

Limitations: GloMoSim is providing the support to the protocols in wireless network. The developers are thinking of adding the functionality to simulate a wired as well as a hybrid network with both wired and wireless capabilities in the feature.

08. Groovenet

GrooveNet is an open source simulator and written by two key languages are C++ and Matlab has provide the GUI for structures and graph. It is the simulator which is running on the following operating systems: Linux OS with Kernel version 2.6*.GrooveNet is a hybrid simulator for geographic routing that address the need of very strong and easy-to-use in the realistic network and the traffic simulation. It designed to be an opportunistic broadcast protocol with minimal hand-shaking between posting and getting parties with few or no shared state information among neighboring vehicles [9].

Limitations: As GrooveNet is an Open Source, every online document and online support is not available on each and every time.

09. J-sim

J-sim is an open source simulator and written by the key language of Java. It is a wired simulator which is running with the mobility model support and low power consumption. Also provide the support on the Network and data link Layer from the OSI model. It has the support for the network with specification of MAC 802.11.This simulator runs on Window Mac and Linux platforms.it is composed of wide range of inner components. Components communicate via their ports, and connection types of one-to- one, one-to-many and many-to-many are supported. The port is used to connect between the components in the network [10].

Limitations: The component behavior is described with a contract. For the simulation of wireless networks, J-Sim has a wireless extension

10. NetSim

NetSim is a commercial simulator and written by the key language Java. It will connect to the network through wired and wireless communication. It is the simulator which is running with the mobility model support and low power consumption [11]. Also provide the support on the Network and data link Layer from the OSI model. It has the support for the network with specification of 802.11 a/b/g/e/n. This simulator runs on the windows platform.

Advantages: NetSim has a GUI which features drag and drop functionality for devices, links etc. i.e. Modeling in NetSim is simple and user friendly.

11. OMNET++

OMNET++ is an open source for study and research sector. Also it is available commercially to industry people and it is written in a key language of C++. It is commonly used in research and educational purposes and in the world scientific community. It is providing the offers an Eclipse-based IDE, a graphical real-time environment and a host of different tools. It is a common-purpose simulator that capable of simulating to any kind of system composed of devices interacting with each other. It is communicating the network through wireless communication. The support for wire and wireless simulation is fairly incomplete [12]. It is one of the simulator which supports running with the mobility model support and low power consumption. Also provide the support on the Network and data link Layer from the OSI model. It has the specification of network is 802.15.6.

Limitations: It has incomplete mobility extension and poor analysis. It is allow only a few protocols into the implementation process.

12. OPNET

OPNET simulator is one of the best among simulators which provides the Graphical User Interface (GUI) support. It can be used to provide performance management for computer networks and applications. It has written by the programming language of C and C++, source code blocks from a tool library of OPNET simulator separate functions. It doesn’t support the following operating systems are LINUX, UNIX. It can be run only on the windows machines with the proper product keys. Because it is not an open source simulator, it is a commercial product. All kind of protocol models can be created in the OPNET simulator [13]. It has the mobility support with the wireless specification of IEEE 802.11. It has the possibilities to maintain the large amount of devices and difficult networks with huge number of devices and traffic flows.

Limitations: OPNET does not give the permission to enter the more number of nodes into the separate interface system. It is one of the speed discrete event simulation machine.

13. PDNS

Parallel and Distributed network simulator is one of the most familiar and easily accessible simulator. Previously the extension library was developed by the PADS research group which is supported for applying parallel and distributed simulations. It was written in the ns TCL syntax for communicating the simulations. It has the support to run the Linux, Solaris, SunOS and Windows operating systems and it could be running with mobility support [14]. It also provide the 802.11 model that attempt to bring the accurate OSI support layers.

Limitations: PDNS provides the benefit of research in telecommunications networks. Also it depends on the device memory and CPU time essentials.

14. QualNet

Qualnet is a commercial network simulator and it is written by C++ programming language. They provide the single user licenses to the academicians and others. The layered architecture of Qualnet is combined with the Application, Transport, MAC and physical Layers. It has the mobility support with low power consumptions. Qualnet can simulate in the wireless specification of 802.11s. Qualnet simulator is giving the support to run on the Linux, Mac OS x, Microsoft Windows operating systems [15]. It need the model of large network with the high fidelity to run on cluster and variety of processor systems, multi-core systems. It can take advantages for making the model of thousands of nodes.

Limitations: the simulation software is very slow to run on the java based user interface and very difficult to install on the Linux platform.

15. REAL

REAL simulator is an open source network simulator. It was written in c Programming language. It has the support for thirty models approximately. It can simulate with the network layer support. The GUI lets users to rapidly make simulation order s of events with move and outline connections. It can be install into the UNIX, BSD, Solaris, SunOS, Windows operating systems. REAL network simulator is intended to do analysis for dynamic behavior of congestion control and flow of schemes in the packet switching networks. The graphical [16] user interface to give the permission to build simulation techniques with draw and drag interfaces.

Limitations: It can make an attempt to be like the several protocol control approved and takes as an input of event-ready space with topology, protocols and amount of work and control parameters.

16. SENS

A Sensor Environment and Network Simulator is supporting to the wired and wireless communication network. It is written by programming language of C++. It provides the support to run on the particular operating systems are Linux, Mac OS X, and Windows XP. It supports 802.11 wireless specifications. It has some extra specifications are sound, radio signal strength and power consumption range. It is a very familiar to wireless sensor network applications [17]. Also get the support from the Physical, Network and application Layer.

Limitations: It has a huge argument as for a simulator to supporting whether one or several phenomena extremely well.

17. SHAWN

SHAWN network simulator has been designed for huge scale network simulation and it is a custom-make sensor network simulator. It is an open source network simulator. It has the ability to bring the large opening level and it supports the network comparisons to other simulators such as ns-2, SENSE, Omnet++ and TOSSIM.SHWAN network simulator is written in Java. Also get the support from the Physical, Network and application layer. So far did not find the exact network specifications but its running on the wireless networks. There is a possibility to change the archive reusability, extensibility and flexibility models [18].

Benefits: It is decoupling and persistence of the simulation technique by establishing the new method as a Tags. It connects the both volatile data and persistence data to separate nods and to the communication world. That parts of a potentially complicated protocol can be restoring without changing the code, because the inner state is save in tags and not in a special node implementation technique.

18. NEST

Network Simulation Testbed (NEST) is a graphical environment and it is based on the distributed network protocols and systems. The routing protocol need to explain the mechanism of steady-state performance and features. It is supported with the UNIX operating systems. Its function is getting involved in the procedure of sending, receiving and broadcasting of a message and the modifications of code. A set of graphical tools used by the NEST generic monitoring tools for the functioning and development of the simulation model in the communication network [19]. Communication link behaviors such as node functions and routing functionalities are typically coded using C programming language.

19. Trans

Traffic and Network Simulation Environment (TraNS) is an open-source network simulator. Its goals is to ignore simulation results that vary significantly from those obtained by the current–world experiments of adhoc and mobile networks. Tran’s simulator has two important distinct mode of operation for addressing a particular need. Both of the mode can be used in the VANET communication protocols and applications to influence the real-time node mobility [20].

20. TOSSIM

TOSSIM simulator is supporting to run on the Linux, Windows operating System. It is a sensor network that gives various mechanisms to communicate with in the network; watching packet traffic in the simulation time, statically or dynamically injecting packet into the network and invocation of TinyOS call [21]. It has built-in Python and C++ programming language. It has the ability to compile the TinyOS code directly modifies the network link or post information to add, delete and a TCP socket application can connect to TOSSIM simulator. TinyOS is an extension of TOSSIM simulator that provide the clean and effective emulator for wireless sensor network (WSN).

Limitations: TOSSIM is not always the exact solution for simulation, it targets on making a lot assumption while modifying others.

I. CONCLUSION

Here the comparative study reveals that simulators are having more impact on how it works, necessary measures and efficient simulator for the specific area of a research work has become easy. The comparison showed here has depicted the properties of the simulators extensively and clearly. The general information compiled in the Table e.g. license type, supported OS & OSI layer format, document availability etc. It also useful along with the power consumption and mobility support. Some cells in the Tables are left blank because of lack of information has found in related web link and documents. Even though, we think our study will become useful to the researchers and industries in the computer communication fields. It has been extended for other properties such as number of node support, version availability and software size. Successful for their research work in wired and wireless sensor networks using the network simulators.

REFERENCES:-

1. Raj, C., & Sukumaran, R. (2015). Modeling UWSN simulators–a taxonomy. *World Acad Sci Eng Technol Int J Comput Electr Autom Control Inf Eng*, *9*(2), 585-592.
2. Issariyakul, Teerawat, and Ekram Hossain. "Introduction to Network Simulator 2 (NS2)." Introduction to Network Simulator NS2. Springer, Boston, MA, 2009. 1-18.
3. Baldo, Nicola, et al. "Validation of the IEEE 802.11 MAC model in the ns3 simulator using the EXTREME testbed." *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.
4. Eriksson, Joakim, et al. "COOJA/MSPSim: interoperability testing for wireless sensor networks." *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.
5. Imran, M., Said, A. M., & Hasbullah, H. (2010, June). A survey of simulators, emulators and testbeds for wireless sensor networks. In *2010 International Symposium on Information Technology* (Vol. 2, pp. 897-902). IEEE.
6. Johnson, David B., David A. Maltz, and Josh Broch. "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks." *Ad hoc networking* 5 (2001): 139-172.
7. Girod, Lewis, et al. "Emstar: A software environment for developing and deploying heterogeneous sensor-actuator networks." *ACM Transactions on Sensor Networks (TOSN)*3.3 (2007): 13.
8. Zeng, Xiang, Rajive Bagrodia, and Mario Gerla. "GloMoSim: a library for parallel simulation of large-scale wireless networks." *Proceedings. Twelfth Workshop on Parallel and Distributed Simulation PADS'98 (Cat. No. 98TB100233)*. IEEE, 1998.
9. Mangharam, R., Weller, D., Rajkumar, R., Mudalige, P., & Bai, F. (2006, July). Groovenet: A hybrid simulator for vehicle-to-vehicle networks. In *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services* (pp. 1-8). IEEE.
10. Sobeih, A., Chen, W. P., Hou, J. C., Kung, L. C., Li, N., Lim, H., ... & Zhang, H. (2005, April). J-sim: A simulation environment for wireless sensor networks. In *38th Annual Simulation Symposium* (pp. 175-187). IEEE.
11. Plana, L. A., Bainbridge, J., Furber, S., Salisbury, S., Shi, Y., & Wu, J. (2008, April). An on-chip and inter-chip communications network for the spinnaker massively-parallel neural net simulator. In *Second ACM/IEEE International Symposium on Networks-on-Chip (nocs 2008)* (pp. 215-216). IEEE.
12. Varga, A., & Hornig, R. (2008, March). An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops* (p. 60). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
13. Chang, X. (1999). Network simulations with OPNET. In *WSC'99. 1999 Winter Simulation Conference Proceedings.'Simulation-A Bridge to the Future'(Cat. No. 99CH37038)* (Vol. 1, pp. 307-314). IEEE.
14. Lee, S., Leaney, J., O'Neill, T., & Hunter, M. (2005, June). Performance benchmark of a parallel and distributed network

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

simulator. In *Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation* (pp. 101-108). IEEE Computer Society.

15. Siraj, S., Gupta, A., & Badgujar, R. (2012). Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, *1*(4), 199-206.
16. Keshav, S. (1988). *REAL: A network simulator*. Berkeley, Calif, USA: University of California.
17. Sundresh, S., Kim, W., & Agha, G. (2004, April). SENS: A sensor, environment and network simulator. In *Proceedings of the 37th annual symposium on Simulation* (p. 221). IEEE Computer Society.
18. Fekete, S. P., Kroller, A., Fischer, S., & Pfisterer, D. (2007, June). Shawn: The fast, highly customizable sensor network simulator. In *2007 Fourth International Conference on Networked Sensing Systems* (pp. 299-299). IEEE.
19. Eppler, J. M., Helias, M., Muller, E., Diesmann, M., & Gewaltig, M. O. (2009). PyNEST: a convenient interface to the NEST simulator. *Frontiers in neuroinformatics*, *2*, 12.
20. 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.
21. Levis, P., Lee, N., Welsh, M., & Culler, D. (2003, November). TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 126-137). ACM.