Journal of Applied Research and Technology

www.jart.ccadet.unam.mx
Journal of Applied Research and Technology 16 (2018) 437-445

Original

Simulation tools for mobile ad hoc networks: a survey

Irin Dorathy[*,a], M. Chandrasekaran[b]

[a]*Assistant Professor, Department of ECE, Government College of Engineering, Bargur, Krishnagiri District, Tamilnadu, India.*

[b]*Principal ,Government College of Engineering, Bargur, Krishnagiri District, Tamilnadu, India,*

Received dd mm aaaa; accepted dd mm aaaa
Available online dd mm aaaa

Abstract: Simulation tools are essential for validating any research idea before it is being implemented. This is very essential for all the researchers irrespective of the field of research. Simulators provide a way for analysing the design.There are many simulation tools available for the network researchers. Network Simulators are used to test the performance of existing or new algorithms and protocols.Some of the simulation tools in networking include OPNET (OPtimized Network Engineering Tool), NS-2 (Network Simulator version 2), OMNET++ (Objective Modular NETwork Testbed in C++), GloMoSim (Global Mobile Information System Simulator), QualNet, NetSim (Network Simulator), JiST/SWANS (Java in Simulation Time/ Scalable Wireless Network Simulator), J-Sim (Java-based simulation) and NS-3 (Network Simulator Version 3). Simulation tools may be open source or commercial. In this paper, both the open source and commercial simulators are discussed. This survey reveals that for simulating large networks, simulators like OPNET, GloMoSim, QualNet, NetSim, JiST/SWANs and NS-3 can be used. For small network simulation NS-2 and J-Sim can be preferred. Open source simulators are poor in documentation. Commercial simulators have good documentation. Based on the requirements of the research, a proper simulation tool can be selected.

Keywords: Simulator, OPNET, OMNET++, GloMoSim, QualNet

1. INTRODUCTION

Simulators play an important role in research. Network Simulators are used to design and evaluate the network algorithms and routing protocols under varying network conditions (Breslauet al., 2000). Mobile Ad hoc Network (MANET) is a collection of mobile nodes forming temporary network without any predefined infrastructure. These networks are used in emergency situations such as in battlefields and emergency medical rescue. For routing in Mobile Ad hoc Networks, routing protocol must be deployed. Before implementing the routing protocol its behaviour must be analysed by simulation. So network simulators are used for simulating these routing protocols or other algorithms in wired and wireless networks.

The organization of the rest of the paper is as follows: Next section discuss about the related work in simulator survey. Section 3 deals with different network simulation tools that are available for network research. It also describes the merits and demerits of the simulators. Section 4 presents the comparison of about nine network simulators based on the parameters like license, operating system and language used, year of release, user interface, protocol and mobility support, scalability, extendibility,

[*] Corresponding author.
E-mail address: callirin@gmail.com (P.E.Irin Dorathy).

Peer Review under the responsibility of Universidad Nacional Autónoma de México.

http://

documentation, network and emulation support etc. Last section concludes with the consolidation of merits and demerits of the simulators and suggests the design of new simulator that satisfies all the requirements of the researcher as future enhancement.

2. RELATED WORK

Many researchers have presented the survey on various network simulation tools. Malhotra (2014) explained the MANET Simulators such as NS-2, NS-3, OMNET++, OPNET, GloMoSim, QualNet and SWANS in their work. Some features of these simulators such as the languages involved and the network supported by them were discussed and their limitations were presented. In addition to this, the issues related to documentation, installation were also discussed. But this survey does not deal with the protocols and the mobility models supported by the simulator. Gupta, Ghonge, Thakare, and Jawandhiya (2013) discussed the open source network simulators such as NS-2, NS-3, OMNET++ and J-Sim. The main features of the open source simulators were presented along with their advantages and disadvantages. The architecture of the four simulators was also explained. This survey may not be considered as a complete survey of network simulation tools because they have considered only the open source simulators. Commercial simulators were not taken into account by the authors. They suggested NS-2 as thebest simulator, for GUI interested research OMNET++ can be preferred for simulating large networks, NS-3, OMNET++ and J-Sim were better than NS-2. Jambli, Lenando, Zen, Suhaili, and Tully (2012) made the survey of existing simulation tools for Mobile Adhoc Sensor Networks. Nearly fifteen simulation tools were discussed and a comparative study of selected simulators were made based on certain evaluation metrics like energy model, mobility model, license, bridging of code, scalability, protocol support, technical support and GUI support. Many features of the simulator were presented but it lacks in the types of protocols supported by the simulator in different layers and the mobility models. In this survey, the details of the individual feature of the simulators were not clearly mentioned. According to the authors, integration of different simulators may be useful for getting better results. Mallapur, (2012) described four network simulators such as NS-2,OMNET++, NCTUns (National Chiao Tung University network simulator) and GloMoSim. They discussed the features, advantages and disadvantages of the above mentioned four simulators. Then comparison of these simulators was done based on certain mainstreams like emulation support, license, GUI, interface, available modules, documentation and user support, scalability, extendable and simulation technique. Several network simulators are available but only four simulators were described by the authors in this survey. They conclude that NS-2 and OMNET++ were the best simulators for ad hoc network because NS-2 supports wide range of protocols in all layers and OMNET++ provides powerful GUI. They added that GloMoSim can be used for large network simulation. In the survey on telecommunication network simulators (Sarkar & Halim, 2011), a comparison of network simulators was done based on the type, deployment mode, network impairments and protocol support. Only limited features of simulators were highlighted in this survey. In their survey they found that most of the papers published in IEEE conference and proceedings use NS-2 for network modelling and simulation task. Three specific recommendations were provided by the authors for the network researchers such as to choose a credible simulator, to build credible simulation models and to use statistical methods to improve the credibility of results.

Our survey contributes a lot to network researchers in identifying the simulator for their research work. In this survey, all the important features of the simulators are considered which helps the researchers in choosing the proper simulator for their work based on the requirement. This survey differs from the existing surveys in two factors (i) The protocols supported in different layers such as application layer, transport layer and network layer are listed clearly (ii) The latest version of the simulator with released date (updated till 2015) is mentioned. These two factors are not included in the existing surveys. Many simulators are available nowadays which makes simulator selection process as a time consuming one for the researchers. This survey is necessary for the network researcher which helps them by saving their time in simulator selection.

3. SIMULATION TOOLS:

Simulation tools may be open source or commercial. Open source simulation tools can be freely downloaded from their official website. Commercial simulators need license and it must be purchased for high rate.

In this paper both commercial and open source simulators are reviewed. The simulators considered for this survey includes: OPNET, REAL, NS-2, OMNET++, GloMoSim, QualNet, NetSim, JiST/SWANS, J-Sim and NS-3. The details of these simulators along with their merits and demerits are discussed in this section. They are collected from the official website, user manual and existing surveys.

3.1 OPNET

OPNET refers to OPtimized Network Engineering Tool. It is a commercial Object Oriented Discrete event general purpose simulator. OPNET is high level and event based network level simulation tool. OPNET is very powerful and large software with many features. The development work of OPNET was started in 1986 by MIL3 Inc. Now it is called as OPNET Technologies Inc. It was developed for military purpose, but latter it rises to be one of the leading commercial network simulation tools. The latest version is OPNET-17.5 released in 2014. OPNET is very expensive for commercial purpose but there are free licenses for educational uses. It is used for modelling, simulating, analysing and designing protocols, devices and networks. Actually OPNET was built for simulating fixed networks. It contains a large library of accurate models for commercially available fixed network protocols and hardwares. It has various tools like OPNET modeler, model editors, model library, analysis tools and animation viewer. Modeling is done mainly in three levels: Network Domain, Node Domain and Process Domain.

Merits

- Provides user friendly graphical interface
- Provides grid computing for distributed simulation
- Supports high fidelity modelling, simulation, analysis and designing of wide range of communication networks
- Allows simulation of entire heterogeneous networks with various protocols
- Provides additional modules and tools for easing the usage of the software

Demerits

- Expensive for commercial purpose
- Provides limited wireless mobility
- Supports limited protocols support
- Lack of energy model

3.2 NS-2

Network Simulator (NS) is based on REAL Simulator. REAL (Keshav, 1997) refers to REalistic And Large. REAL Network Simulator was developed by University of California and Cornell University in the year 1988. It was built on NEST (Network Simulation Test bed) Package from Columbia University. This work was supported by Xerox Corporation, Palo Alto Research Center and DARPA. It was monitored by Naval Electronics Systems Command. The last version was REAL-5.0 released in 1997. REAL can simulate packet switched, store and forward networks that is similar to wide area networks (Keshav, 1988) REAL has two parts namely simulation server and display client. It was implemented as a simulation tool for analysing the dynamic behaviour of flow control and congestion control techniques in packet switched networks. The main limitation of REAL Simulator is that there are no updates after 1997.

NS-2 (The Network Simulator-NS2, 2015) refers to Network Simulator version 2. It is an Opensource Object Oriented, discrete event simulator. It is the second version of Network Simulator. NS-2 is one of the most commonly used simulators for network research. This work was the joint effort of people at Lawrence Berkeley National Laboratory (LBL), University of California at Berkeley (UC), University of Southern California's Information Sciences Institute (USC/ISI) and Xerox Palo Alto Research Center(PARC). This work was started in the year 1989 and the first version was released in 1995.NS-2 was maintained by USC.It was developed under VINT (Virtual Internetwork Testbed) Project. It was sponsored by Defence Advanced Research Projects Agency (DARPA) and National Science Foundation (NSF). The latest version of NS-2 is NS-2.35 released on Nov 4, 2011. It uses dual languages: C++ and OTcl. C++ is very efficient for implementing a design but it is not easy to be visual. OTcl is a Tcl script language with Object Oriented extension developed at MIT.In NS-2 the control path implementation is separated from data path implementation. For reducing packet and event processing time, the event scheduler and basic network component objects in the data path are written and compiled using C++. C++ is normally used to implement the detailed protocol and OTcl is used to control the simulation scenario and schedule the events.NS-2is fully equipped with protocols, models, algorithms and accessory tools.

Merits:

- Includes energy model
- Provides rich collection of models than other simulators
- Allows users to generate node movement patterns and traffic patterns.
- Supports simulation of TCP, routing protocols and multicast protocols
- Allows simulation of wired and wireless network functions and protocols

Demerits:

- NS-2 is not user friendly
- Poor GUI support
- Poor debugging support
- Recompilation is needed each time when user code is changed
- Difficult to use the tracing system
- Limited tools are available.
- Limited and poor out-of- date documentation
- Cannot simulate large networks
- Computational overhead is more
- Memory usage is more

3.3 OMNET++

OMNET++ (What is OMNET?, 2015) refers to Objective Modular NETwork Testbed in C++. OMNET++ is an object oriented discrete event network simulator. It was developed by Andras Varga from Technical University of Budapest with some intermittent contributors. It was licensed under own Academic Public License. The development work was started in the year 1998. Latest version OMNET++ 5.0b1 was released in March 11, 2015. OMNEST is the commercial version of OMNET++. OMNET++ is free for academic purpose. OMNEST licenses can be obtained from Simulcraft Inc for commercial purposes. OMNET++ has a generic architecture. It can be used in several problem domains like modeling of communication networks, queuing networks, multiprocessors and other distributed hardware systems, protocol modeling, hardware architectures validation and performance evaluation of complex software system. OMNET++ simulation runs under several user interfaces. Graphical animating user interfaces are useful for demonstration and also for debugging purposes. Command-line user interfaces are good for batch execution. Simulator, user interfaces and tools are highly portable. OMNeT++ provides component architecture for models. Components are programmed in C++ and it is then assembled into larger components using a high-level language named as NEtwork Description Language (NED). OMNET++ has extensive GUI support. The modular architecture of OMNET++ makes it easy to embed the simulation kernel into real time applications.

Merits:

- Supports parallel distributed simulation
- Used for queuing network simulations
- Provides rich class of libraries for implementation of various modules
- Graphical tools are available for analysing the simulated output
- Provides both event based and process based programming style

Demerits:

- Poor documentation
- Limited protocol support
- Analysis of performance measures is poor
- Poor run time performance

3.4 GloMoSim

GloMoSim refers to Global Mobile Information System Simulator. It is a library based parallel Simulator. It was developed at Parallel Computing Laboratory at University of California, Los Angeles (UCLA). It provides a scalable simulation environment for wireless networks. It is designed using PARSEC which is a C based parallel simulation language. PARSEC stands for PARallel Simulation Environment for Complex Systems. It is the combination of library based sequential and parallel discrete event simulator. It comes with rich set of models and this simulation tool can scale up to thousands of nodes. The highly modular design of this tool makes it easy to extend and modify the basic functionalities. The ability to scale up and definition of standard APIs for modular protocol stack opens the possibility to flexibly investigate the effect of multiple models at different levels of detail. Analysis and visualization tools are basic but it is sufficient for general studies. GloMoSim is not under active development.

Merits:

- Supports large scale networks with thousands of nodes
- Initially supports wired and wireless networks but

now it supports only wireless networks

- Provides parallel simulation environment
- Supports simulation of large network
- PARSEC allows execution of asynchronous protocols
- Good propagation models are available

Demerits:

- Needs separate installation of PARSEC
- Does not support devices like sensor and actuators
- In-depth documentation is not available
- Simulator is not up-to-date
- No recent release after 2000

3.5 QualNet

QualNet (QualNet Network Simulator Software, 2015) provides comprehensive environment for designing protocols, creating and animating network scenarios, and analyzing their performance. It is the commercial version of GloMoSim. Mainly it is mainly used for large heterogeneous networks. It is a scalable simulator with high fidelity network models. It is a commercial product from Scalable Network Technologies (SNT) which is derived from GloMoSim. The first version was released in the year 2000. QualNet has many additional features compared to GloMoSim. The latest release was QualNet 7 in Oct, 2013. QualNet has large number of models and protocols for both wired and wireless networks. It has extensive documentation and technical support. QualNet is a complete network simulator in terms of available protocols, models and tools. It offers parallel developer for managing and running parallel executions. It is a planning, testing and training tool that imitates the behaviour of a real communications network. The tools available in QualNet include QualNet architect, analyzer, packet tracer, file editor and command line interface.

Merits

- Allows simulation of large network with tens of thousands of nodes
- Provides animation tools and graphical tools for analysing the output
- Able to predict the performance of wired, wireless and mixed networks and devices
- Able to analyse and explore the design of devices and code in the early stage in closed networks at real time speed
- Allows users to set up, develop and run custom network models
- Provides good debugging support
- Provides fast simulation results
- Provides powerful Graphical User Interface for code development
- Offers unmatched platform portability and interface flexibility

Demerits:

- Expensive commercial Simulator
- Interface is slow
- Hides the source code of some files
- Difficult to install in LINUX

3.6 NetSim

NetSim (Tetcos, 2015) is used for network design, planning and network R & D. NetSim covers many technologies like TCP, IP, Wi-Max, WLAN, Wireless Sensor Networks and Mobile Ad hoc Networks. NetSim is also a discrete event simulator. It was developed by TETCOS in alliance with Indian Institute of Science. First version was released in June 2002. NetSim has an in-built development environment, that serves as the interface between NetSim's protocol libraries, user's code and simulation kernel. Protocol libraries can be modified by the user easily because it is available as open C code. Recently the latest version NetSim 8.3 is released in 2015. The advanced feature of NetSim is that during simulation the custom code can be debugged. This can be done at various levels including at per-packet interval. This avoids time consuming process of programming, customizing and configuring commercial simulators to meet customer specific requirements.

Merits:

- Simple and User friendly
- Programmability is excellent
- Has inbuilt analysis framework
- Has inbuilt packet animator

Demerits

- Limited to Academic use

3.7 JiST/SWANS

JiST (Barr, 2015) refers to Java in Simulation Time.SWANS stands for Scalable Wireless Network Simulator. JiST/SWANS represent SWANS built on JiST platform. It is a java based discrete simulator.

It was developed at Cornell University in 2004. In 2005, the last version JiST/SWANS v1.0.6 was released. It is used for simulating large networks. SWANS was created mainly because existing wireless network simulators are not sufficient for the current research needs. SWANS also serves as a validation for the virtual machine-based approach to simulator construction.

Merits:

- Large networks can be simulated
- Java makes this a powerful simulator
- Computational overhead is less (Weingärtner, Vom Lehn, & Wehrle, 2009)
- Requires less memory

Demerits

- No development after 2005

3.8 J-Sim

J-Sim (JSim, 2015) refers to Java-based simulation system. It is a discrete event, extensible and reusable platform independent simulator for building quantitative numeric models and then analyzing them with respect to some experimental reference data. J-Sim was developed by Distributed Real time Computing Laboratory (DRCL). This work was sponsored by National Science Foundation (NSF), DARPA's Information Technology office (DARPA/IPTO), Air Force Office of Scientific Research‘s Multidisciplinary University Research University (AFOSR/MURIA), Ohio State University and University of Illinois, Urbana Champaign. It is an application development environment based on the component based software architecture. It is implemented on top of component-based software architecture called Autonomous Component Architecture (ACA) (Sobeih Viswanathan, Marinov, & Hou, 2007). Components are the basic entities in ACA. They communicate with one another by sending or receiving data at the ports. On the top of ACA a generalized packet-switched InterNETworking framework (INET) was laid based on the common features extracted from various layers in protocol stack. Both the ACA and INET are implemented in Java. The resulting code, scripting framework and GUI interfaces together called as J-Sim. J-Sim model calculations are specified in J-Sim's own MML. MML refers to Mathematical Modeling Language. It is an easy-to-read text-based language. MML models are often expressed in mathematical equations. J-Sim supports multiple independent variables such as time, space etc. MML variables may vary over any combination of declared domains.

Merits:

- Models can be easily reusable and interchangeable
- Offers flexibility
- Provides Graphical User interface library

Demerits:

- Execution time is worse

3.9 NS-3

NS-3 (NS3 Network Simulator, 2015) refers to Network Simulator Version 3. NS-3 is a discrete-event network simulator. In NS-3 the simulation models are implemented in C++. First version of NS3 was released in the year 2008. It was funded by University of Washington, INRIA, Sophia Antipolis, Georgia Tech University, Atlanta. The latest version is NS-3.23 released on May, 2015. NS-3 is built as a library which is statically or dynamically linked to a main C++ program. This program defines the simulation topology and it also starts the simulator. NS-3 also exports its API to Python. It allows Python programs to import an NS-3 module in the same way as NS-3 library linked by executables in C++. NS-3 is not the updated version of NS-2. NS-3 is designed mainly for Internet systems, primarily targeted for research and educational purpose. NS-3 is free software and licensed under GNU GPLv2 license. It is available publicly for research and development. NS-3 software encourages the development of simulation models that are sufficiently realistic to allow NS-3 simulator to be used as real time network emulator interconnected with real world environment. It allows several existing real world protocol implementations to be reused within NS-3.

Merits

- Provides Scalability
- Very flexible
- Provides real world integration
- Fastest simulator
- Memory efficient simulator
- Performance is better that NS2 in terms of memory management
- Supports light weight Virtual machines
- Supports both IP and non-IP based networks research.
- Supports a real-time scheduler for interacting with real systems

- Efficient in terms of computation time

Demerits

- Limited models available
- No GUI to build topology
- Limited visualization support
- Not backward compatible with NS2

4. COMPARISON & DISCUSSION

The comparison of various parameters of the simulators are summarised in Table 1. Simulators such asNS-2, NS-3, OMNET++, JiST/SWANS, J-Sim are open source network simulators while OPNET,GloMoSim and QualNet are commercial simulators. Open source simulators are available freely and can be downloaded from the official website at free of cost. Many researchers use open source simulators for their work. Latest and updated versions are released periodically or randomly for almost all the simulators except for GloMoSim and JiST/SWANS. OPNET, QualNet and NetSim provide excellent Graphical User Interface. Table 2 show the list of protocols and the mobility models supported by the network simulators. Almost all the network simulators support many protocols in the different layers of protocol stack. QualNet supports more number of mobility models than other simulators. Table 3 summarises the important features of the network simulators. Scalable simulators such as OPNET, GloMoSim, QualNet, NetSim, JiST/SWANS and NS-3 can be used for simulating large networks. For small network simulation other simulators can be used. When compared to open source simulators, documentation is good for commercial simulators. All the simulators have the feature of extendibility. GloMoSim and JiST/SWANS have no support for emulating real time networks while other simulators provide limited or good emulation support.

Table 1. Comparison of various parameters of the simulators.

Simulators	Licence	Operating System	Language	Released Year	Release of Latest version	Latest version	User Interface
OPNET	Commercial	Windows, Linux	C++	1986	2014	OPNET 17.5	Graphical User Interface/ Excellent
NS-2	Open source	Window, Linux	C++/ OTcl	1995	2011	NS-2.35	Command Line interface/ limited
OMNET++	Open source	Windows, Unix Based Mac OS	C++/NED	1997	2015	OMNET++ 5.0b1	Graphical User Interface/ Good
GloMoSim	Open source	Windows, Linux	C/PARSEC	1998	No	No updates	Graphical User Interface/ limited
QualNet	Commercial	Windows, Linux Mac OS	PARSEC	2000	2013	Qualnet 7	Graphical User Interface/ Excellent
NetSim	Commercial	Windows	Java/C	2002	2015	Netsim 8.3	Graphical User Interface/ Excellent
JiST/ SWANS	Open source	Platform independent	Java/Tcl	2004	2005	Jist/SWANS v1.0.6	Graphical User Interface/ limited
J-Sim	Open source	Platform independent	Java	2004	2014	J-Sim 2.15	Graphical User Interface/ limited
NS-3	Open source	Windows, Linux, MacOS	C++/ Python	2008	2015	NS-3.23	Command Line interface/ limited

Table 2. Comparison of list of protocols and mobility models.

Simulators	Application layer	Transport layer	Network layer	Mobility Models
OPNET	CBR,FTP, VBR	TCP,UDP	IPv4,IPv6,ICMP, ICMPv6,OSPF, RIPv1,RIPv2	Random Way Point, Vector
NS-2	FTP, Telnet, Web, CBR, VBR	TCP,UDP	DSR,AODV, DSDV	Random Walk, Random Way Point
OMNET++	Telnet	TCP, UDP, SCTP	IPv4, IPv6, OSPF, BGP	Random Walk, Turtle
GloMoSim	CBR, FTP, Telnet,Web	TCP, UDP, DBS	Distributed Bellman Ford, Flooding, Fisheye, DSR,DSDV, WRP, LAR, NS-DSDV, DREAM, MMWN	Random Way Point, Random drunken
QualNet	CBR,FTP, VBR	TCP,UDP,RTP, RSVP, MPLS, DiffServ	Distributed Bellman-Ford, OSPFv2, RIPv2, BGP, Flooding, Fisheye, DSR, DSDV, WRP, LAR, AODV	Group, Pedestrian, Random Way Point, File-based Dead Reckoning
NetSim	Voice. Video, FTP, Database, HTTP, Email	TCP, UDP	OSPF,RIP,AODV, DSR	Random Way Point, Random Walk
JiST/ SWANS	CBR,FTP, VBR	TCP, UDP	ZRP,AODV,DSR	Random Way Point Random Walk, Teleport
J-Sim	CBR,FTP, VBR	TCP, UDP	Shortest path, Spanning tree OSPF,DVR,DVMRP DSR,AODV	Random Way Point, SDF trace-based, Model
NS-3	API,On/off Application	TCP, UDP	IPv4,IPv6,OLSR ,AODV,DSR, DSDV	Random Direction, Random Walk, Random Way Point, 3D GMM

Table 3. Comparison of the important features of the network simulators.

Simulators	Scalability	Extendibility	Documentation & user support	Network Support	Emulation support
OPNET	Enterprise	Yes	Good	Wireless Ad hoc, Cellular, WLAN,PAN,Satellite	Yes
NS-2	Small scale	Yes	Poor	Wired, Wireless,Ad hoc, satellite	Yes/ limited
OMNET++	Small scale	Yes	Medium	Wired,Wireless Ad hoc	Yes/ limited
GloMoSim	Large scale	Yes	Poor	Wireless,WSN,Ad hoc	No
QualNet	Enterprise	Yes	Good	MANET,Wired,Cellular, Satellite	Yes
NetSim	Large scale	Yes	Good	Wired,Wireless Ad hoc	Yes
JiST/ SWANS	Large scale	Yes	Medium	Wired,Wireless Ad hoc	No
J-Sim	Medium	Yes	Medium	Wired, Wireless, WSN	Yes
NS-3	large scale	Yes	Good	Wired, Wireless,Ad hoc, IP and Non IP based network,WSN	Yes

CONCLUSION AND FUTURE ENHANCEMENT

All the simulators have certain merits and demerits. They have different characteristics and features. The networks and the protocols supported by each simulator are different. So the selection of a particular simulator depends on the nature of the research.For large network simulation, use any one of the simulators like OPNET, GloMoSim, QualNet, NetSim, JiST/SWANS and NS-3. For simulating small networks one can use NS-2 or J-Sim. Open source simulators are poor in documentation. Commercial simulators have good documentation. So based on the requirements of the research, a particular simulation tools can be selected. But none of the simulators satisfy the entire requirement. In future a new simulator may be designed that satisfies all the requirements of the network researchers. The new simulator must support all types of networks, protocols and mobility models. It must be scalable, extendible and must provide good emulation support. The new simulator must be at reasonable cost with good documentation and product support.

CONFLICT OF INTEREST

The authors have no conflicts of interest to declare.

REFERENCES

Barr, R. (2015). Introduction to JiST. Retrieved July 15, 2015, from http://jist.ece.cornell.edu/

Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., ...& Yu, H. (2000). Advances in network simulation. *Computer*, *33*(5), 59-67.

Gupta, S. G., Ghonge, M. M., Thakare, P. D., & Jawandhiya, P. M. (2013). Open-Source Network Simulation Tools: An Overview. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *2*(4), 1629-1635. Retrieved from http://ijarcet.org/wpcontent /uploads/IJARCET-VOL-2-ISSUE-4-1629-1635.pdf

Jambli, M. N., Lenando, H., Zen, K., Suhaili, S. M., & Tully, A. (2012). Simulation Tools for Mobile Ad-hoc Sensor Networks: A State-of-the-Art Survey. *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)* (pp. 1-6). IEEE.

J-Sim. (2015). Retrieved June 17, 2015, from http://www.physiome.org/jsim/.

Keshav, S. (1997). REAL 5.0 Overview, Retrieved from June 17, 2015, http: //www.cs.cornell.edu/skeshav/real/overview.html

Keshav, S. (1988). *REAL: A network simulator.* Berkeley, Calif, USA: University of California. https://dl.acm.org/citation.cfm?id=894210

Malhotra, J. (2014, November). A survey on MANET simulation tools. In *Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH), 2014 Innovative Applications of* (pp. 495-498). IEEE.

Mallapur, S. V. (2012). Survey on Simulation Tools for Mobile Ad-Hoc Networks, *International Journal of Computer Networks and Wireless Commssssunications (IJCNWC)*, ISSN: 2250-3501 Vol.2, No.2, April. 241-248*2*(2), 241-248.

NS3 Network Simulator. (2015). Retrieved from July 23, 2015, https://www.nsnam.org/

QualNet Network Simulator Software. (2015). Retrieved from July 14, 2015, https://web.scalable-networks.com/qualnet-network-simulator-software

Sarkar, N. I., & Halim, S. A. (2011). A review of simulation of telecommunication networks: simulators, classification, comparison, methodologies, and recommendations.Journal of Selected Areas in Telecommunications (JSAT), Vol.2, March.10-17.

Sobeih, A., Viswanathan, M., Marinov, D., & Hou, J. C. (2007). J-Sim: An integrated environment for simulation and model checking of network protocols. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International* (pp. 1-6). IEEE.

Tetcos. (2015). NetSim-Network Simulator & Emulator | Home, Retrieved May 17, 2015, from http://www.tetcos.com.

The Network Simulator - NS2. (2015). Retrieved April 19, 2015, from http://www.isi.edu/nsnam/ns/

What is OMNET? (2015). Retrieved from April 22, 2015, https://omnetpp.org/intro/

Weingartner, E., Vom Lehn, H., & Wehrle, K. (2009, June). A performance comparison of recent network simulators. In *Communications, 2009. ICC'09. IEEE International Conference on* (pp. 1-5). IEEE.