Journal of Positive School Psychology http://journalppw.com
2022, Vol. 6, No. 2, 5390 – 5398

Performance Analysis of Commonly Used Simulators and Emulators in MANET

[1]Veepin Kumar, [2]Sanjay Singla

[1]Research Scholar, Department of Computer Science & Engineering, Dept. of CSE, IK Gujral Punjab Technical University, Jalandhar, Punjab, India, kumarvipi@gmail.com
[2]Professor, GGS College of Modern Technology, Kharar(Mohali), Punjab, India, dr.ssinglacs@gmail.com

Abstract

A Mobile Ad-Hoc Network is a collection of wireless mobile hosts that construct a temporary network without the use of a stand-alone infrastructure or centralizedcontrol. Because the nodes in the network are mobile, they self-organize and configure themselves. As real-world tests are expensive and time-consuming, simulation is an important part of studying MANET. Most of the mobile ad hoc network community uses simulation as their research approach.When working with MANET, it's critical to select a simulation environment that meets the researcher's needs and allows them to conduct studies in a specific area.The primary objective of this work is to offer researchers with a performance analysis of several simulators and emulators so that they may choose the best tool for simulating their network and analyzing their proposed research. This research work also discusses the simulation tools along with their characteristics, advantages and disadvantages.

Keywords: MANET, Simulation and Simulation Tools.

1. INTRODUCTION

T In the past decade, MANET has attracted a lot of attention and has supplied solutions for large number of applications.MANETis a wireless network that connects mobile devices[1]. It is a self-organizing network with no underlying structure. Every device in MANET can move in any direction to communicate information with other devices or network nodes. MANET has no administrator node in control of managing other nodes; instead, each MANET node acts as its own router and host, forming its own network. Routing in MANET is managed via a variety of routing protocols [2]. Routing the packets in an environment where the topology changes regularlyjust a few of the challenges. Simulation is the most popular approach for researching solutions to these complex MANET problems.

1.1 FUNDAMENTAL CONCEPT OF SIMULATION

It is costly, time-consuming, and difficult to build and maintain a MANET testbed. Furthermore, several factors have an impact on the trial results at the same time. Simulation is an important tool for studying MANET because of the difficulty of evaluation and the high expense of producing practical tests. MANET protocols are frequently built and tested using a simulator, especially in the early phases of development. Simulation is the procedure of designing a model of a real system and running experiments on it to better understand or compare various operating techniques. Simulating hundreds of nodes in a network is less expensive, and it may be accomplished rapidly[3]. MANET can be simulated using both generic and specialized simulators. It requires a proper model based on perfect assumptions and an acceptable framework to make implementation easier. Researchers can use

network simulators to model any computer network by describing the behavior of network nodes and communication channels. The following are the key features of a good Simulator[4]:-

- Simulator must be reusable and accessible to users 24*7.
- Simulator must be scale up and scale down according to the user requirement.
- For defining experiments and processing results, rich-semantics scripting languages are supported.
- Graphical, debugging, and tracing support all are available.
- Analysis of Failure is easy

The structure of paper is as follows: Section 2 presents the types of simulation Section 3 includes a survey and a description of the various simulators along with their merits and demerits.Section 4deals with when to utilize which simulator and criteria for the comparative study, as well as a table of comparison and a discussion of our findings. In Section 5 present a summary of our findings.

2. TYPES OF SIMULATION

Trace Driven Simulation (TDS) and Discrete-Event Simulation(DES) are the simulations that are extensively employed in MANET[5].

Discrete-Event Simulation

This simulation can simulate many jobs operating on different nodes;therefore,it is commonly utilised in MANET[5].DES also has dynamic memory management, which allows the programmer to add new entities to the model while removing old ones. In discrete-event simulation, researcher can use breakpoints in the debugger to inspect the code gradually without interrupting the program's operation.When state variables interact with one another and change on a continuous basis, discrete event simulation is not appropriate[6].

Trace-Driven Simulation

This type of simulation is frequently employed in real-world systems, and the results have a higher level of credibility[7]. It offers more precise workload information, allowing users to explore the simulation model in greater depth. However, there are some flaws in this simulation. As a result, the simulation becomes more complex, workloads may alter, and thus the simulation's representativeness must be discussed[8].

3. SIMULATION TOOLS

Nurul I. Sarkar et al.surveyed all papers published in the IEEE Transactions on Networking, IEEE/ACM Transactions on Networking, and proceedings of IEEE GLOBECOM, INFOCOM, and ICCto gain an understanding of the simulation tools used in the research. According to the survey results, 42.8 % of the 8370 articles polled stated that they utilizeNS-2 for network modelling. MATLAB was used in 36.8% of the total publications assessed, whereas OPNET was used in 7.6% publication.QualNet, GlomoSim, and OMNet++ were utilised in the remaining publications surveyed[9].

Based onabove survey, this section illustrates some of MANET's most often used simulation tools along with their characteristics, Advantages and Disadvantages.

3.1 NS-2

The acronym NS-2 stands for Network Simulator Version Two, which was initiallyinvented in 1989 as the REAL network simulator. This simulator is supported by Defense Advanced Research Projects Agency and the National Science Foundation[3]. This simulator is used in both wired and wireless networks, free to use and includes an online manual.

Advantages

- Support a wide number of protocols.
- The open-source architecture reduces simulation costs
- Online manuals make it simple for users to change and improve the codes.
- Presence of a visualization tool.

Disadvantages

- The programming language and modelling techniques must be familiar to the user.
- Using NS-2 to represent a desired work can be more complicated and time-consuming than using other simulators.
- It has a scalability problem because it can't simulate more than 200 nodes. Trace files will become too large to maintain as the size of network increases.
- It is difficult to add additional protocols to NS-2 because of its intrinsically constructed design.[10].

3.2 NS-3

This simulator is open-source and designed for research and education, written in C++ and Python. It was created for the first time in 2006[11]. After its initial creation, many versions have been delivered. On December 23, 2011, the latest version, NS-3.13, was released.With Cygwin, NS-3 supports variety of OS, including Linux and Windows. In NS-3, Each event has an execution time associated with it, and the simulation progresses by executing events in the order of simulation time. When an event is processed, it can result in the formation of zero, one, or several events. Events are consumed as a simulation runs, although new events may (or may not) be generated. When there are no more events in the event queue, or when a special "Terminate" event is identified, the simulation will automatically stop [12].

Advantages

- The architecture has been modularized to enable the usage of external routing by the node.
- It is a separate simulator, not a successor to NS-2[9].

Disadvantage

- Cygwin does not maintain Python bindings and it supports only IPv4 protocol.

3.3 TOSSIM

TOSSIM was developed in 2003 by the TinyOS project team at UC Berkeley.This simulator is a bit-level DES written in Python and C++[13]. With the help of Cygwin, TOSSIM can be run on Linux or Windows. TOSSIM also provides online free sources and documentation.TinyOS is not an OS in the conventional sense, it is a programming framework for embedded devices and a set of components that allow each application to have its own OS.Scalability, Completeness, Fidelity, and Bridging are the four elements for a TinyOS simulator [14].

Advantages

- The cost of emulation is reduced because to the open-source model free online document.
- Contains a GUI that displays visuals instead of written commands.
- Simulate tens of thousands of nodes.
- Simulate radio models and code executions in addition to networks.

Disadvantages

- TOSSIM can't accurately simulate energy consumption in network therefore researcher has use PowerTOSSIM which is another TinyOS simulator that helps to determine the power consumption of each node.
- Capable of simulating homogeneous applications only.
- Because it was designed specifically for WSN simulation, it can only simulate motes-like nodes.

3.4 EmStar

EmStar is an emulator written in C that was first invented at the University of California, Los Angeles.This simulator that includes a lot of the features of other simulators. Its component-based architecture provides for reasonable scalability. Many protocols are currently available in this simulator to use because it targets a specific platform.Em* provide tools for deployment, simulation, emulation&visualization and services for networking, sensing and time synchronization.This simulator has built-in support for a variety of devices, such as sensors and radio equipment [15].

Advantages

- Users can operate each module individually due to the modular programming methodology.
- Sensors are reduced to a minimum fault.
- Debugging and analyzing are significantly easier with the multiple modes.
- Between sensors, users can simply choose between deployment and simulation.
- People can use a graphical user interface to control electrical devices.
- Provides online materials to aid in the widespread use of this emulator

Disadvantages

- Impossible to simulate many sensors.
- Only real-time simulation is possible.
- Applies to MICA2 motes and iPAQ-class sensor nodes only.

3.5 OMNET++

Both a noncommercial and a commercial licence are available for OMNeT++, which can be utilized by academic institutions or non-profit research groups. The modular programming model is supported by this simulator. OMNeT++ is a well-known general-purpose network simulator that may be used in both wired and wireless networks[16]. Majority of OMNeT++'s frameworks and simulation models are free source. Parallel simulation execution is also supported by OMNeT++ [17].

Advantages

- Easy error tracing and debugging.
- MAC protocols as well as some localised protocols are supported.

Disadvantages

- The number of protocols provided is insufficient
- Compatibility issue at the time of combination of models.

3.6 JSIM

This simulator includes a graphical user interface (GUI) library that allows users to develop J-Sim modelsusing Mathematical Modeling Language, a "text-based language". This simulator is often utilized in the fields of physiology and biomedicine, but it can also be used to simulate WSNs/ MANET. This simulator allows researchers to simulate real-time processes.JSIM is made up of three Java packages. First is known as the JSIM foundation packages which has little to do with web-based simulation but is required for any simulation system. Second is known as JSIM engine packages and third is JSIM environment packages which makes easier to create simulation models, connect to databases, run models in a controlled manner, and perform query-driven simulation[18].

Advantages

- Interchangeable and reused model.
- To assist users in tracing and debugging programmes it provides data diffusions, routings, and localisation simulations.
- Simulate many nodes while conserving memory.

Disadvantage

- Execution time is larger than NS-2.
- For wireless networks, 802.11 is the only MAC protocol available.

3.7 COOJA

It's a wireless network emulator that comes included with the Contiki OS[19]. Cooja also included in Instant Contiki which is a virtual computer that contains a fully functional Contiki development environment[4]. Cooja can simulate large and small networks of Contiki motes. COOJA can also simulate non-Contiki nodes, such as Java-based nodes or nodes running a different OS. There are advantages and drawbacks to each technique[20]. Motes can be imitated at a lower level, which is faster but allows for larger network simulations, or at a higher level, which is slower but allows for more precise observation of system behaviour.

Advantage

- Flexible andExtensible

- Platforms for sensor nodes, OS software, radio transceivers, and radio propagation models are all available.
- It uses the Java Native Interface to connect with the built Contiki code.

Disadvantage

- It's adaptable, but it's also inefficient.

3.8 GLOMOSIM

This is a widely used simulator for large and wireline communication networks, created by a similar computing laboratory at the University of California, Los Angeles. It is a scalable network modelling environment that works with both wireless and wired networks. It differs from most other sensor network simulators because of its parallel discrete-event design. Although GloMoSim is a broad network simulator, it currently only supports protocols for wireless networks. This simulator is being created using a layered architecture similar to the OSI architecture, as is the case with most network systems. Between multiple simulation layers, simple APIs are created. This enables for the quick integration of models created by various people at different stages [21].

Advantage

- Scalable
- GloMoSim framework is designed and developed using a comprehensive protocol stack to achieve scalability and feasibility.
- Supports both unicast and multicast wireless protocols.

Disadvantage

There are no specialized protocols for routing, transport layer energy usage models, or IP address support.

3.9QUALNET

This simulator is used by Scalable Network Technology for security purpose, first published it in 2000at the University of California, Los Angeles[10]. It is a GloMoSim commercial version, which is an academic research version. It has GUI, animation, and analysis approach used for huge heterogeneous networks and distributed applications.Researcher can utilize QualNet to create new protocol models, optimize current models and create huge wired and wireless networks, analyze network performance to improve it [22].

Advantages

- Multiprocessor systems are supported.
- On each layer, there are built-in measures.
- Modular and layered stack design
- Standard API are available for working with protocols at several layers
- Scalable
- System/protocol modelling is done with graphical user interface tools.1xm

Disadvantages

- This simulator is a commercialization of Glomo's work.
- QualNet is tough to set up on Linux.
- Java based user interface provided by this simulator is slow.

There are many simulators that are used to simulate MANETs, but only the most widely used simulator tools are described in this research study. Table 1 lists some of the features of the simulation tools listed above.

Table 1: *Characteristics of Different Simulators*

Simulator	**Programming Language**	**Documentation Available**	**Platform**	**GUI**	**License Type: Open Source or Commercial**	**Simulator Type (Discrete-Event/Trace-Driven)**
NS2	C++/ OTCL	Yes	Linux, Windows	No	Open Source	Discrete- Event

NS3	C++/ Python	Yes	Linux, Windows	No	Open Source	Discrete- Event
TOSSIM	C++/ Python	Yes	Linux	No	Open Source	Discrete- Event
EmStar	C	No	Linux	Yes	Open Source	Trace-Driven
OMNET++	C++	Yes	Linux, Windows	Yes	Open Source; Commercial	Discrete- Event
JSIM	Java	Yes	Linux, Windows	Yes	Open Source	Discrete- Event
COOJA	Java	Yes	Linux	Yes	Open Source	Discrete- Event
GloMosim	C and Parsec	No	Linux, Windows	Yes	Open Source	DES
Qualnet	C++ and Parsec	Yes	Windows	Yes	Commercial with Separate Licenses for Industry and Academic	Discrete-Event

4. WHEN TO USE WHICH SIMULATOR

Simulators for MANETs provide a variety of features and models. The requirements should help in the selection of a simulator [13]. It's crucial to figure out how much information researcher will need. If high-precision PHY layers are required, ns-2 [14] is without a doubt the best option. Recent simulators present high-level abstractions and sophisticated object-oriented designs that will be more adaptable if wireless technology has no impact on the desired protocol.

The simulation tool selected is also determined by the number of nodes targeted. More than 1,000 nodes should not be anticipated from sequential simulators. Parallel simulators are a good alternative if bigger ranges are required. Researcher might also look into integrating stage simulation with highly optimized simulators like NS-2.

Most of non-commercial simulators lack adequate documentation and support. In the presence of disturbances, using a commercial one may be beneficial. Furthermore, commercial simulators typically have a huge list of supported protocols, whereas open-source solutions provide complete control.

4.1 COMPARISON OF SIMULATOR TOOLS AND DISCUSSION

Here weanalyze the performance of some of the above-mentioned simulator tools based on their ability to model the Network.The following are the comparison criteria [2]: -

Consumption of energy by a node

Because the node consumes a significant amount of energy during data transmission and reception, energy conservation of node is a vital research topic in MANET. in this criterion, we evaluate the simulator's capacity to model the energy consumption of the nodes and network.

Node Mobility

Because MANET nodes migrate from one location to another due to its dynamic topology,

it is critical to assess the simulator's capabilities to simulate node mobility in the network.

Scalability of the Network

This criterion evaluates the simulator's scalability i.e., its capacity to support the number of nodes in the network.

Extensibility

This criterion assesses the capacity of the simulators to integrate with other modules and to adapt new protocols and algorithm.

The result of our comparative studybased on a set of criteria mentioned above is mentioned in table given below.

It has directed us to the conclusion that NS2, NS-3, EmStar, JSim, OMNET++, Cooja can consume energy whereas TOSSIM, Qualnet, and GlomoSim cannot.

In terms of node mobility, NS2, NS-3, OMNET++, Cooja, GloMoSim, Qualnet, and JSim support it, although TOSSIM does not and there is no information available about EmStar.

Table 2: *Comparison of Simulator Tools*

Name of Simulator	**EnergyConsumption**	**Mobility of Node**	**Scalability**	**Extensibility**
NS-2	Yes	Yes	200	It is possible, but hard to achieve.
NS-3	Yes	Yes	20000	Yes
TOSSIM	No	No	1000	Yes
EmStar	Yes	-	100	Yes
OMNET++	Yes	Yes	1000	Yes
JSIM	Yes	Yes	500	Yes
COOJA	Yes	Yes	-	Yes
GLOMOSIM	No	Yes	1000	Yes
QUALNET	No	Yes	1000	Yes

Regarding the number of nodes supported by each simulator, we deduced that Emstar, J-Sim, NS-2 and can simulate a small network of less than or equal to 500 nodes. For a network composed of 1000 node, we can use the simulator like GloMoSim, Qualnet, OMNET++, TOSSIM and for a large network of 20000 node, NS-3 can be used and there is no information available about Cooja.

Each of the above-mentioned simulators integrate with other modules and able to adapt new protocols and algorithms, making them extendable in nature.

5. CONCLUSION

Due to the difficulty of analysis and setting up of genuine tests, simulation is a vital tool for studying Wireless Sensor Networks. This research work provides detailed analysis of the most utilized simulator and emulatorsalong with their characteristics, benefits and drawbacks and gives suggestions to researchers for choosing an effective tool for their research. Knowing the features, advantages and disadvantages of numerous simulators is useful to choose a simulator to test real-world scenarios or to create new modules and incorporate them into the simulator.

Reference

[1] A. O. Bang and P. L. Ramteke, "MANET : History , Challenges And Applications," Int. J. Appl. or Innov. Eng. Manag., vol. 2, no. 9, pp. 249–251, 2019.

[2] J. Kaur and A. Singh, "A Review Study on the Use of MANET for Wireless Devices," SSRN Electron. J., pp. 2–5, 2020, doi: 10.2139/ssrn.3572817.

[3] D. R. V. Mrs. C. Gayathri, "An Overview: Basic Concept of Network Simulation Tools," Int. J. Adv. Res. Comput. Commun. Eng., vol. 6, no. 1, pp. 19–22, 2017.

[4] S. R. S, R. Akshay, and A. A. A. M, "A Study and Analysis of Different Simulators in Computer Network," pp. 14188–14194, 2017, doi: 10.15680/IJIRSET.2017.0607213.

[5] Dr. V. Vasanthi, "Simulators and Emulators used for Wireless Sensor Network," Int. J. Adv. Res. Comput. Commun. Eng., vol. 6, no. 1, pp. 171–175, 2017.

[6] K. Dooley, "Simulation research methods," Companion to Organ. Joel Baum (ed.), London Blackwell, pp. 829–848, 2002.

[7] R. S. Anand Nayyar, "A Comprehensive Review of Simulation Tools for Wireless Sensor Networks (WSNs)," A Compr. Rev. Simul. Tools Wirel. Smoke Networl(WSNs), vol. 19–47, no. 1, pp. 19–47, 2015, doi: 10.5923/j.jwnc.20150501.03.

[8] F. Yu, "A Survey of Wireless Sensor Network Simulation Tools," http://www1.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html, 2011.

[9] M. P. Chhimwal, "Comparison between Different Wireless Sensor Simulation Tool," IOSR J. Electron. Commun. Eng., vol. 5, no. 2, pp. 54–60, 2013, doi: 10.9790/2834-0525460.

[10] E. Weingärtner, H. Vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," IEEE Int. Conf. Commun., 2009, doi: 10.1109/ICC.2009.5198657.

[11] M. Saidallah, A. El Fergougui, and A. E. Elalaoui, "A Survey and Comparative Study of Open- Source Wireless Sensor Network Simulators," vol. 7, no. January, pp. 1–7, 2017.

[12] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni, "Computer network simulation with ns-3: A systematic literature review," Electron., vol. 9, no. 2, pp. 1–25, 2020, doi: 10.3390/electronics9020272.

[13] I. Minakov, R. Passerone, A. Rizzardi, and S. Sicari, "A comparative study of recent wireless sensor network simulators," ACM Trans. Sens. Networks, vol. 12, no. 3, 2016, doi: 10.1145/2903144.

[14] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," SenSys'03 Proc. First Int. Conf. Embed. Networked Sens. Syst., pp. 126–137, 2003.

[15] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, "EmStar: A software environment for developing and deploying wireless sensor networks," Proc. Free. Track 2004 USENIX Annu. Tech. Conf., no. June 2014, 2004.

[16] L. Hogie, P. Bouvry, and F. Guinand, "An overview of MANETs simulation," Electron. Notes Theor. Comput. Sci., vol. 150, no. 1, pp. 81–101, 2006, doi: 10.1016/j.entcs.2005.12.025.

[17] V. András and H. Rudolf, "AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT," SIMUTools, Marseille, Fr., 2008.

[18] J. A. Miller, A. F. Seila, and X. Xiang, "JSIM web-based simulation environment," Futur. Gener. Comput. Syst., vol. 17, no. 2, pp. 119–133, 2000, doi: 10.1016/S0167-739X(99)00108-9.

[19] B. Musznicki and P. Zwierzykowski, "Survey of Simulators for Wireless Sensor Networks," Int. J. Grid Distrib. Comput., vol. 5, no. 3, pp. 23–50, 2012, [Online]. Available: http://www.researchgate.net/publication/234065225_Survey_of_Simulators_for_Wireless_Sensor_Networks/file/79e4150eeb702c9511.pdf.

[20] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," Proc. - Conf. Local Comput. Networks, LCN, pp. 641–648, 2006, doi: 10.1109/LCN.2006.322172.

[21] L. Bajaj, M. Takai, R. Ahuja, and K. Tang, "Glomosim: A Scalable Network Simulation Environment," Comp. A J. Comp. Educ., vol. 28, no. 1, pp. 154–161, 1999, [Online]. Available:

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.45.7167&rep=rep1&type=pdf.

[22] “Manish Sharma 1 and Gurpadam Singh, " Evaluation of proactive, Reactive and Hybrid Adhoc Routing Protocol for IEEE 802.11 MAC and 802.11 DCF in VANET using Qualnet 2 1,” vol. 11, pp. 209–220, 2011, doi: 10.5121/csit.2011.1220.