

Survey on Network Simulators

Ronit L. Patel
Student, B.Tech (IT)
CSPIT, CHARUSAT
Anand, Gujarat, India

Maharshi J. Pathak
Student, B.Tech (IT)
CSPIT, CHARUSAT
Anand, Gujarat, India

Amit J. Nayak
Assistant Professor (IT)
CSPIT, CHARUSAT
Anand, Gujarat, India

ABSTRACT

Simulation software plays a vital role in real-world implementation. Practically hardware setup of network topologies is very costly and strenuous to modify often. Hence various simulators act as the prototype of the real system. These days a wide range of network simulators are accessible, some of them are committed to the wired network, some of them to the wireless network or some dedicated to both type of networks. There are many simulators available such as Ns-2, Ns-3, OMNeT++, NetSim, J-SIM, REAL, OPNET, OMNEST, QualNet, GloMoSim, TOSSIM, DRMSim, TraNS, NTCUns etc. Out of these varieties of simulators, it is difficult for one to select network simulator based on the requirement for a specific job. The following paper provides comparison based analysis of various network simulators. Simulators taken into consideration are Ns-2, Ns-3, OMNeT++, NetSim, REAL, OPNET and QualNet. Information about simulators such as features, advantages, disadvantages, backend environment, the supporting operating system, and minimum hardware requirement is discussed in the following paper. So based on above-mentioned information and analysis, one can make choice of an appropriate network simulator based on the requirements.

Keywords

Simulation, Network Simulator, Simulators comparison, OMNeT++, Ns-2, Ns-3, REAL, NetSim, OPNET, QualNet, Event driven, GUI support.

1. INTRODUCTION

In computer networking, network simulation is a technique in which software program analyzes the behavior of a network by computing the association between different network devices/entity such as the hub, switch, router, access points, links etc. Network simulator provides visualization, animation of flow of packets, the performance of the network and cost-effective technique to design network. There are two types of simulators available, commercial and open source. The following paper presents both type of simulators and some comparative study between them. The following paper discusses various simulators such as Ns-2, Ns-3, OMNeT++, NetSim, REAL, OPNET and QualNet. Out of these varieties of simulators, it is difficult for one to select network simulator based on the requirement for a specific job. For each simulators mention above, the brief introduction, advantages, disadvantages, backend environment, the supporting operating system, minimum hardware requirement and comparison with other simulator is discussed. So based on above-mentioned information and analysis, one can make choice of an appropriate network simulator based on the requirements.

2. SIMULATORS

2.1 Ns-2

Ns-2 is a network simulator, which is object-oriented and it was essentially created at the University of California-

Berkeley. Ns-2 is a various case simulator intend on art network research, exceedingly concentrate on modeling protocols alike wired, wireless, satellite, TCP, UDP, multicast, unicast, telnet, FTP and web. Ns-2 is naturally an event driven simulation software advantageous in studying communication networks. Ns-2 gives the distinctive method for deciding system protocol and simulating their analogous conduct. The burgeoning maturity of the tool is marked ever since certain revolutions and improvements have been implemented. Now, Ns-2 has developed into the most extensive network simulator, and one of the most universally utilized open source network simulator [1].

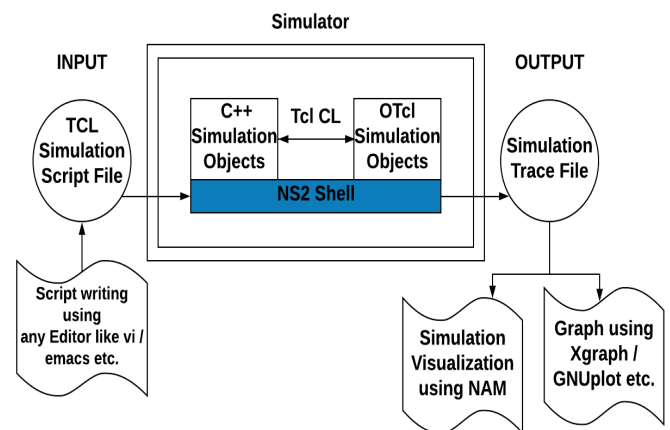


Fig 1: Ns-2 Architecture [1]

Developer: Defense Advanced Research Projects (DARPA) through the Virtual Internetwork Testbed (VINT) project at Lawrence Berkeley National Laboratory (LBL), Xerox PARC, UCB, and USC/ISI (Information Science Institute).

License: Open source.

Language: OTCL (Object-Oriented Tool Command Language) and C++.

Operating system supported by Ns-2: Windows (Windows XP, Vista, 7, 10), FreeBSD, GNU/Linux and Mac OS X.

Minimum hardware requirement: 5 GB disk space and 256 MB RAM.

2.1.1 Features

- Ns-2 isolate control path implementation from the data path implementation.
- To chop down packet and event processing time essential network component objects in the data path are composed and incorporated utilizing C++.

- OTCL accomplishes the feature that are absent in C++.
- The merger of OTCL and C++ languages justify being active. To schedule, the event OTCL is used and to implement detailed protocol C++ is used.
- Ns-2 provides radio propagation, mobility tools, energy models, visualization tool, topology generation tool and extensibility.

2.1.2 Advantages

- Availability of analysis tool and network visualization tool.
- Interaction with the real system.
- It bolsters an extensive scope of protocols in all layers.
- Large numbers of models are available.
- Testing of complex scenarios is easy [2].

2.1.3 Disadvantages

- It does not support for creating methodologically sound simulations.
- Tracing system is difficult to use.
- Simulation process may slow down, if large numbers of nodes are taken into consideration.
- Difficult to understand and analyze the code.

2.1.4 Comparison with GloMoSim

- Execution of GloMoSim is faster compared to Ns-2.
- GloMoSim is more stable compared to Ns-2.
- The primary output file generated by GloMoSim is statistic file (can also generate trace file), while for Ns-2 primary output file is trace file.
- GloMoSim supports C, while Ns-2 supports OTCL (Object-Oriented Tool Command Language) and C++.

2.2 Ns-3

A network simulator is software which characterizes the conduct of a computer network. Besides, it additionally enables the client to show the system topology to determine the hubs and the connections between those hubs [1]. It also provides animation of the packet flow. It is executed in C++ with a discretionary Python scripting API. It is a free, open source programming venture sorted out around investigates group improvement and maintenance.

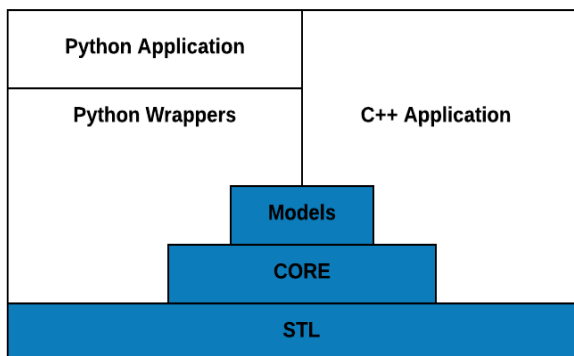


Fig 2: Ns-3 Architecture [4]

Developer: Community-based (Ns-3 Consortium).

License: Open source.

Language: C++ and optional Python bindings.

Operating system supported by Ns-3: Windows (Windows XP, Vista, 7, 10), FreeBSD, Win-Vista, Linux/GNU and Mac OS X.

Minimum hardware requirement: 5 GB disk space and 256 MB RAM.

2.2.1 Features

- Ns-3 acts both as simulator as well as emulator so that we can create a live network (i.e. capture data and perform analysis from the LAN card).
- Attention to authenticity: Protocol substances are intended to be nearer to genuine PCs.
- Software coordination: As it is open source networking software, it decreases the need to rework models for simulation [3].
- Support for virtualization: It supports lightweight virtual machines.
- Tracing design: As NS3 is building up a following and insights gathering structure attempting to empower custom-worked of the output without working up the simulation core [3]

2.2.2 Advantages

- It has high modularity as compared to Ns-2 [2].
- It has special feature called emulation mode, which allows network administrators for integration with real networks.
- It is flexible than other simulators.
- It has an extensive variety of utilization in both development and improvement of the current systems.
- Maintaining responsive mailing list is one of the key features of Ns-3.
- It provides realistic environment with well-organized source code along with lower base level of abstraction [5].

2.2.3 Disadvantages

- It has a major issue with the lack of reliability.
- Limited support for Python in scripting and visualization.
- Maintainers are required to react to the client inquiries with the bug reports and furthermore to approve and test the framework [2].
- Scalability limits.

2.2.4 Comparison with Ns-3

Table 1. Difference between Ns-2 and Ns-3

Ns-2	Ns-3
It supports OTcl and C++ scripting.	It supports C++ and optional Python scripting.
Network Animator (NAM) is used for visualization.	NetAnim tool is used for visualization.
In scalability, it has sequential simulation.	In scalability, it has distributed simulation.
Debugging is complex as it involves C++/OTcl.	Debugging is not complex as it involves C++/Python.
It is simulator.	It works as simulator as well as emulator.

2.3 Omnet++

OMNeT++ remains for objective modular network testbed in C++. Another name of OMNeT++ is Optical Micro-Network Plus Plus or Optical modeling network using C++. It is open source and free of cost only for the academic and non-profit purpose. OMNeT++ is extensible, component-based C++ and modular discrete event-driven network simulation library and framework. OMNeT++ is similar to Ns-2 and Ns-3 in the network simulation of both wired and wireless [2]. It is widely used in wireless networks of the sensor [6]. It supports parallel distribution simulation and also provides infrastructure and tools for writing simulations. Aside from network simulation, it can be utilized for modeling of multiprocessors and dispersed equipment frameworks. OMNeT++ provides modeling concept. The model structure uses high-level dialect NED (Network Description).

It can be utilized in various areas such as:

- Wireless and wired communication network modeling.
- Queuing network modeling.
- Protocol modeling.
- Evaluation of the performance aspects of the software system which are complex.

Components of simulation system:

- Simulation kernel consists of code which manages simulation and simulation class library
- The User interface for debugging, demonstration (graphical animating user interface) or batch files execution (command-line user interface).

Significant network simulation model structures for OMNeT++:

- MiXim and Castalia framework.
- The Mobility and INET structure.

2.3.1 Modeling concept

OMNeT++ model consists of modules (simple and compound module) that communicate with message passing. These modules are written in C++. Active modules are named as simple modules. As shown in figure 3 by combining simple modules compound module can be formed. The arrow connecting small boxes indicates connections and gates. Entryways of the module are the input and output interface.

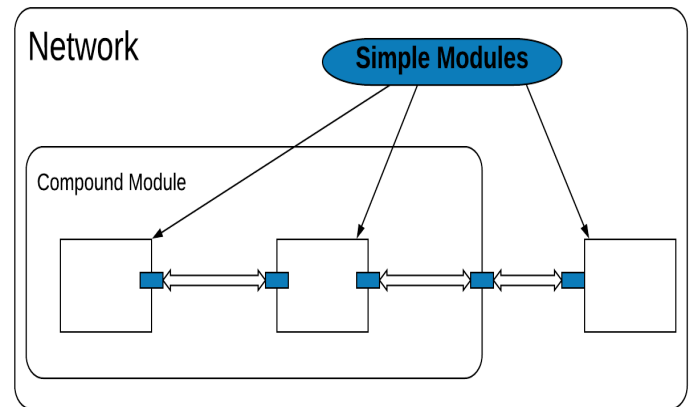


Fig 3: Simple and Compound Module Structure [6]

Developer: OpenSim Ltd.

License: Academic public license.

Language: Written in C++. It also supports JAVA in its integrated development environment (IDE) and C#.

Operating system supported by OMNeT++: Linux (Fedora Core 25, OpenSUSE 42, Ubuntu 16.04 LTS and Red Hat Enterprise Linux Desktop Workstation 7.x), Mac OS, Windows (Window 7, 10 - 64 bits) and Other Unix based framework.

Minimum hardware requirement: 400 MB disk space and 512 MB RAM.

2.3.2 Advantages

- Using OMNeT++ simulation can be performed or executed under graphical user interface [7].
- Debugging and Tracing is easy in OMNeT++ compared to other simulators [2].
- OMNeT++ is not only limited to network protocol simulation.
- Supports MAC protocols as well as some localized protocols in WSN.
- The simulation model for internet, IPV6.
- Highly modular and very well structured.

2.3.3 Disadvantages

- The number of the protocols provided by OMNeT++ is less.
- Framework dependent.

2.3.4 Comparison with OMNEST

OMNEST is network simulator, which is the commercial version of OMNeT++. OMNEST and OMNeT++ are identical, models of simulation written for OMNeT++ are guaranteed to compile and run with OMNEST, and vice versa.

- License for OMNeT++ is academic public while OMNEST is for commercial purpose.
- Both offer Eclipse-based Simulation IDE.

- OMNeT++ does not support Microsoft Visual C++ while OMNEST supports.
- There exist Pre-compiled (and tested) simulation libraries for Windows for OMNEST, but not for OMNeT++.
- Both support GCC compiler.
- No SVG image export in OMNeT++ while it is there for OMNEST.
- OMNEST provides systemC integration while OMNeT++ does not.

2.4 Opnet

OPNET stands for optimized network engineering tool. The primary programming dialect in OPNET is C. The introductory design (topology setup, parameter setting) is normally accomplished utilizing graphical user interface (GUI), an arrangement of XML records or through C library calls. OPNET is helpful in the study of communication networks, devices, protocols, and its applications [4]. It also has the powerful impact of visual or graphical support for the users.

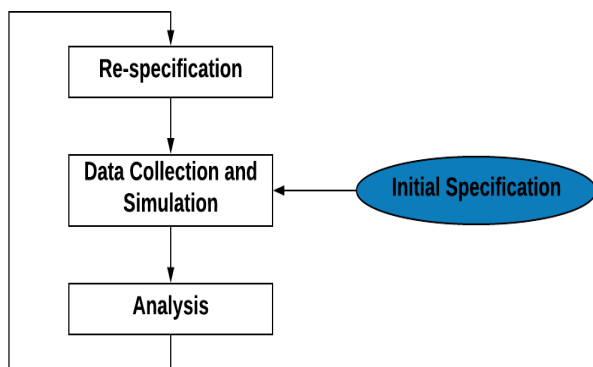


Fig 4: OPNET Architecture

Developer: OPNET Technologies Inc.

License: Commercial.

Language: C and C++.

Operating system supported by OPNET: Windows (Windows XP, Vista, 7, 10) and Windows NT 4.0.

Minimum hardware requirement: 200 MB disk space and 256 MB RAM.

2.4.1 Features

- It can work with 32-bit and 64-bit GUI.
- Fast discrete event-driven simulation engine [8].
- Availability of segment library.
- Scalable remote simulations bolster.
- The discrete event, hybrid, and analytical simulation [8].
- Integrated, GUI-based debugging and analysis.

2.4.2 Advantages

- Fast discrete event simulation engine [8].
- A large number of project scenarios that are offered information on OPNET Network Simulator.
- Stratified modeling environment.
- Custom-make wireless modeling.

2.4.3 Disadvantages

- Memory consuming models.
- GUI operations are complex.
- Costly.
- Insufficient tutorial.
- Buggy, old and hard to modify ready-made models.
- Less usability.

2.4.4 Comparison with Ns-2

Table 2. Difference between Ns-2 and OPNET

Simulator	Open source	Documen tation	Pre-implemen ted algorithm s and protocols	Preferable
OPNET	No	A lot	Many	Network operators
Ns-2	Yes	Less	Less	Network researchers

2.5 Netsim

NetSim was developed by Tetcos in 1997 in combination with Indian Institute of Science. NetSim is used for network lab experimentation and exploration [4]. It has been displayed in 5TH edition of computer networks and internets. NetSim is finest network simulation software for protocol modeling and simulation, granting us to inspect computer networks with incomparable bottom, power, and resilience [9]. Netsim lands with very good development environment. This environment will work as admix among developer's code and Netsim's libraries and simulation kernel. It arranges network work metrics at various absorption altitudes such as Network, sub-network, Node and a detailed packet trace. NetSim appears with two versions, Standard or Academic versions. In short, NetSim is truly an exotic product that is not only accomplished but also robust and provides those features that are hard to come with any simulators.

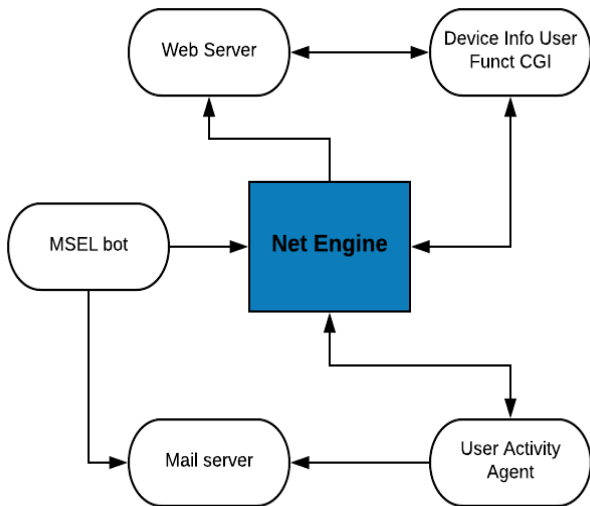


Fig 5: NetSim Architecture [4]

Developer: Tetcos Ltd.

License: Commercial/Proprietary.

Language: C and Java.

Operating system supported by NetSim: Windows (Windows XP, Vista, 7, 8, 10).

Minimum hardware requirement:

- Processor: 1 GHz Pentium processor or proportionate (Minimum), 3 GHz Pentium processor or comparable (Suggested).
- RAM: 512 MB (Minimum), 2 GB (Suggested).
- Hard Disk: 200 MB of accessible space.
- Display: 1024 x 768, 256 hues (Minimum), 1024 x 768 high shading, 32-bit (Suggested).
- Web availability.

2.5.1 Features

- It simulates a wide range of routers.
- Protocol library is open source. This can be altered by the user.
- Modeling and simulation backing TCP, UDP, IPv4 and IPv6 routing and more protocols and technology [2].
- NetSim has one advantage over other market rivals, it has very good boson virtual packet technology engine.
- It has a very good debugging ability. It provides breakpoints and checkpoints during simulation with the help of this user easily check observation [9].
- NetSim has feature for simulation and study of voice and data communication for different situations for high-frequency global communication systems.

2.5.2 Advantages

- It has very good GUI support. This will help to design good network and also help in the analysis of the network.
- It has very good analysis framework. This

framework is very useful for comparisons between intra and inter-protocol. NetSim has good graphical options to display this kind of comparisons [2].

- NetSim has built-in packet animator. This feature will be very useful for analysis of the network.

2.5.3 Disadvantages

- It is very costly.
- NetSim is a single process discrete event simulator. A single event queue is used for the simulation which at any given time contains one entry for each station on the network [2].

2.5.4 Comparison with GloMoSim

- GloMoSim is open source while NetSim is proprietary.
- GloMoSim has very limited GUI support then NetSim.
- GloMoSim is very hard to use then NetSim.
- GloMoSim is not updated regularly.
- GloMoSim has discrete-type simulation but NetSim has stochastic discrete-event.

2.6 Real

REAL is used because, as in today's fast-growing world performance analysis of computer networks is greatly increasing which gains more importance [10]. As the clients are continuously increasing so is the network traffic which can quickly overwhelm the network from topology-oblivious broadcasts or a central coordinator for centralized consensus protocols [10]. It is utilized for watching the dynamic behavior of flow and congestion control schemes in packet switch information networks. It is implemented by using C as the programming language.

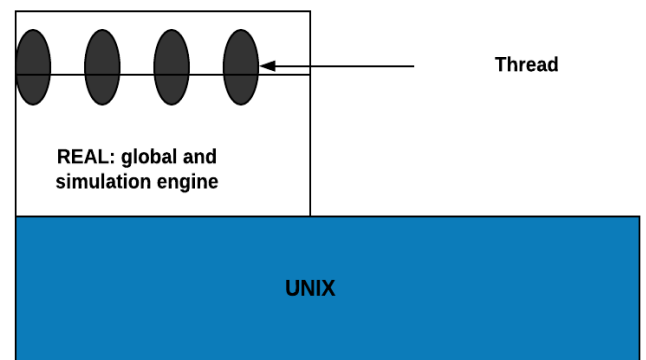


Fig 6: REAL Architecture [4]

Developer: The test system has been developed by Warren Toomey and Elan Amir.

License: Open source.

Language: It is implemented and written in C.

Operating system supported by REAL:UMIPS framework, Ultrix, SunOS, digital UNIX, IRIX, BSD4.3 and Solaris.

Minimum hardware requirement:

- Processor: Above 500 MHz.
- RAM: 128 MB.
- Hard Disk: Greater than or equal to 10 GB.
- I/O Devices: Standard keyboard & mouse, high-resolution monitors.

2.6.1 Features

- It has indirect 30 modules (written in C) that precisely imitate the activities of a few surely understood stream control protocols, (for example, TCP), and research booking disciplines, (for example, Fair Queuing and Hierarchical Round Robin) [8].
- This type of design also allows users to add new modules (if required) very easily and with very little effort.
- The simulator takes as information a situation, which is a description of system topology, protocols and workload and control parameters. It produces as output statistics such as the quantity of packets sent by each source of information, the queuing delay at each queuing point and the quantity of dropped and retransmitted packets.
- This makes it unnecessary for clients to download, compile and build the simulator in order to run simulations. Rather, the situations, once made, are dispatched to the server for execution, and the outcomes are sent back to the GUI for the show.

2.6.2 Advantages

- It is very cheap (basically free) and usually fast to set up and run tests on.
- They are great for sanity-testing an app's user interface and catching the most common bugs.
- Flexibility due to dynamic configuration.
- Many different implementations for various use cases.

2.6.3 Disadvantages

- There are certainly many limitations available but with each subsequent release, it is tried to overcome.
- Compatibility problem (not portable).
- The GUI support is very slow.

2.7 Qualnet

The QualNet simulator is a commercial software platform used for defense purpose as well as used to analyze and design networks, networked systems, and distributed application behavior. It is used to simulate the behavior of network under a various user-defined operating scenarios and application traffic patterns and its implemented in C/C++. It is a high fidelity simulation tool which predicts mixed, wireless and wired platform network and performance of networking device and most commonly utilized for testing, research, network design, modeling and simulation. It allows programmers to create and analyze network scenarios, design

protocols and analyze their performance.

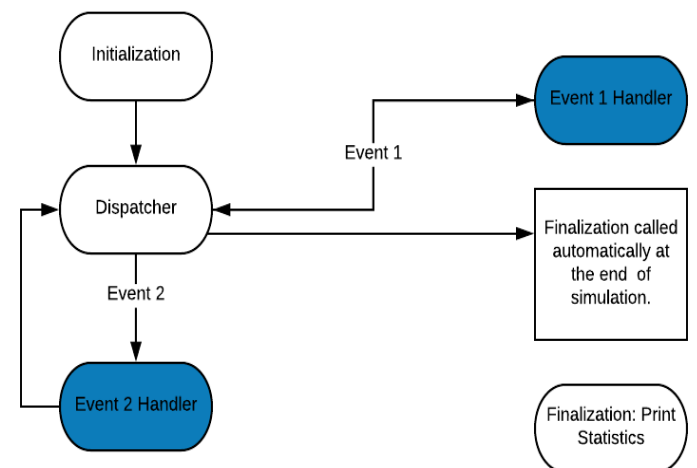


Fig 7: QualNet Architecture[4]

Developer: Scalable network technologies.

License: Commercial. But provides separate licenses for the academic and other purposes.

Language: C++

Operating system supported by QualNet: Linux 64-bit framework(x86) (Red Hat Enterprise Linux(RHEL) 6.6 and Ubuntu 14.04LTS) and Windows (Windows 7 Home Premium and Proficient 64-bit version, Windows 8 and Windows 8 Pro 64-bit version).

Minimum hardware requirement:

- Processor: 64-bit (x86-64 compatible).
- Memory: 2 GB for local area network size simulations with graphical user interface, 4 GB for large networks consisting of more than 1000 nodes.
- Disk space: 2 GB disk space.
- Video: 128 MB memory supporting hardware 3D acceleration for a discrete graphics card and 1024 x 768 screen resolution or better.

2.7.1 Features

- For better representation of the situation, it underpins 3D objects.
- Using optimized memory management and keen architecture, it offers quicker simulation speed and gullibility.
- Cyberlibrary provides distributed denial of service attack, radio jamming attacks, security against intrusion, third-party network intrusion detection software, and so on.
- GUI in Qualnet is upheld utilizing Qualnet Architect, Qualnet Analyzer, Qualnet Packet tracer, Qualnet Command Line interface, Qualnet File editorial manager, and so forth.
- For database storage purpose, it utilizes SQLite and MySQL 5.0 for the Statistics Database (Stats DB).

2.7.2 Advantages

- Reduces time: It has made possible to perform simulation in minutes as compared to other simulators which take months to perform the simulation due to their parallel execution capability.
- Provides high fidelity Simulation: It provides high fidelity for the simulation of network devices, antennas, human interaction etc.at a real-time speed.
- Cleanse the requirement for Abstraction: It kills the requirement for display reflection because of its loyalty and adaptability.
- Versatility through the help of parallel execution.
- GUI devices for framework/convention demonstrating [11].
- Rapid prototyping of protocols [11].

2.7.3 Disadvantages

- QualNet is not open source (commercial) simulator.
- It hides the source code of some files. So, we cannot do the modification for such files.
- QualNet is paid and much expensive as it is commercial.

2.7.4 Comparison with Ns-2

- QualNet is a commercial simulator whereas Ns-2 is open source (free).
- Parallel execution is supported in QualNet whereas Ns-2 runs on a single machine.
- QualNet is mainly developed for wireless simulations whereas Ns-2 is developed for wired simulations.
- QualNet uses GUI to create model specifications whereas in Ns-2 connections have to be established manually for model specifications.
- Ns-2 is not as fast and scalable as QualNet.
- Ns-2 is widely used in research as it is freely available whereas QualNet is not used for research purposes.
- Simulations of wireless sensor networks and GSM mobile networks supported in QualNet whereas Ns-2 does not support these.
- Qualnet incorporates an assortment of advanced libraries, for example, mesh networking, battery models, network security toolkit, and a substantial number of protocols at various layers.

3. CONCLUSION

The paper presents comparative study and the brief description of various network simulators such as Ns-2, Ns-3, OMNeT++, OPNET, NetSim, REAL, and QualNet. Each simulator has their domain of relative strength and weakness compared to other simulators. Amongst the available open source network simulators, OMNeT++ and Ns-2 are the most appropriate one. The main strength of OMNeT++ is its GUI and IDE support, while the strength for Ns-2 is that it provides a large number of models, also Ns-2 is most popular in academic research. OMNeT++ and Ns-3 can be used for large-scale network simulation. Also, Ns-3 provides better results. Ns-3 has ability like handling multiple interfaces on hubs accurately, utilization of IP addressing to and greater arrangement with Internet protocols and designs, more point by point 802.11 models and so forth etc. Ns-3 is quickly forming into an adaptable and simple to-utilize tool appropriate for simulation of wireless network. These

highlights makes Ns3; substitution of Ns2. Appropriate guidelines are provided about the network simulators which will be beneficial in selecting simulator to perform particular task or to build project with specified requirements.

4. REFERENCES

- [1] Dr. Vijay R. Ghorpade, Comparative Study of Network Simulator: NS2 and NS3, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 3, March 2016.
- [2] Mohammed Humayun Kabir, Syful Islam, Md. Javed Hossain and Sazzad Hossain, Detail Comparison of Network Simulators, International Journal of Scientific & Engineering Research, Volume 5, Issue 10, October-2014.
- [3] Nancy Garg, Network Simulators: A Case Study, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 1, January 2015.
- [4] Mrs Saba Siraj, A. Gupta, and Rinku Badgujar, Network simulation tools survey, International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, June 2012.
- [5] Atta ur Rehman Khana, Sardar M. Bilal and Mazliza Othmana, A Performance Comparison of Network Simulators for Wireless Networks, arXiv preprint arXiv:1307.4129. 2013 Jul 15.
- [6] Xiaodong Xian, Weiren Shi and He Huang, Comparison of OMNET++ and Other Simulator for WSN Simulation, 3rd IEEE Conference on Industrial Electronics and Applications 2008 on 3-5 june.
- [7] Andras Varga and Rudolf Hornig, Using the OMNeT++ Discrete Event Simulation System in Education, IEEE TRANSACTIONS ON EDUCATION, VOL. 42, NO. 4, NOVEMBER 1999.
- [8] Mrs. C. Gayathri and Dr.R. Vadivel, An Overview: Basic Concept of Network Simulation Tools, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Special Issue 1, January 2017.
- [9] Patel Rajan, and Pariza Kamboj, Investigation of Network Simulation Tools and Comparison Study: NS3 vs NS2, Journal of Network Communications and Emerging Technologies (JNCET), Volume 5, Special Issue 2, December (2015) transactions 14: 15.
- [10] Srinivasan Keshav, REAL: A Network Simulator, University of California at Berkeley, USA, December 1988.
- [11] Goyal, Anjali, Sandip Vijay, and Dharmendra Kumar Jhariya, Simulation and performance analysis of routing protocols in wireless sensor network using qualnet, International Journal of computer applications 52.2 (2012).

