



INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK



SPECIAL ISSUE FOR INTERNATIONAL CONFERENCE ON "INNOVATIONS IN SCIENCE & TECHNOLOGY: OPPORTUNITIES & CHALLENGES"

STUDY AND COMPARISON OF DIFFERENT RECENT NETWORK SIMULATORS AND THEIR INTERFACES IN MANET

PROF. SWAPNIL A. BOBADE, PROF. MANGESHKUMAR S. SHEGOKAR

Department of CSE, College of Engineering & Technology, Akola

Accepted Date: 07/09/2016; Published Date: 24/09/2016

Abstract: In the network research area, establishing of network in a real time scenario is very difficult. A single test bed takes a large amount of time and cost. So implementation of a whole network in real world is not easily possible and very costly too. A network simulator is a software program that imitates the working of computer network. In simulators, the computer network is typically modeled with devices; traffic etc. and the performance are analyzed. Typically, users can then customize the simulator to fulfill their specific analysis needs. Network simulators are used worldwide for education, commercial and Industrial purposes to simulate any part or entire network. There are numerous network simulators used day by day. Network simulators can be used to produce an approximation result of the network, which lays foundation for real time application or implementation. There are a number of network simulators, for instance, ns-2, ns-3, OMNET++, OPNET, JIST, J-SIM, and GLOMOSIM/QUALNET etc.

Keywords: Network, Manet



PAPER-QR CODE

Corresponding Author: PROF. SWAPNIL A. BOBADE

Co Author: PROF. MANGESHKUMAR S. SHEGOKAR

Access Online On:

www.ijpret.com

How to Cite This Article:

Swapnil A. Bobade. IJPRET. 2016; Volume 5 (2): 488-498

INTRODUCTION

In communication and computer network research, network simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts/packets, etc.) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab; various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions. When a simulation program is used in conjunction with live applications and services in order to observe end-to-end performance to the user desktop, this technique is also referred to as network emulation. In this paper, we have analyzed both open source and commercial simulators. Network simulation is without a doubt one of the most predominant evaluation methodologies in the area of computer networks. It is widely used for the development of new communication architectures and network protocols. So-called network simulators allow one to model an arbitrary computer network by specifying both the behavior of the network nodes and the communication channels. For example, in order to investigate the characteristics of a new routing protocol, it is usually implemented in a network simulator. Afterwards, the routing behavior can be easily studied in different topologies, given the fact that the network topology is merely a set of simulation parameters. Computer networks are being widely used in all fields of life, Net banking, Military operations, disaster relief and social networking. As the usage of various computer networks grows so does the need to make these networks faster, efficient and secure greatly increase in the early era, research work on communication networks involved both experimentation and mathematical modeling to prove feasibility and to establish bounds on expected performance of computer networks. However, in the last decade, computer networks have gone through a rapid revolution and have become too complicated for mathematical analysis. Computer-based simulation plays an important role in the research work to help the researchers and network designers to understand the behavior and performance of the networks and its protocols. Computer simulation is often used to test the planned capacity of networks and to meet customer requirements. In addition, simulation is also used to explore a wide range of potential protocol designs through rapid evaluation and iteration. However, different simulators require variable time, memory and computation power for evaluating proposed protocols/techniques. This paper presents performance comparison of following network simulators, i.e. ns-2, ns3, OMNET++, JiST and GloMoSiM, etc. because these are open source and well known network simulators in the research community. The main focus of this

paper is to assist researchers in choosing the most suitable simulators for their work in terms of memory usage, computation time and CPU utilization.

2. BACKGROUND OF PROPOSED WORK

There are a wide variety of network simulators, used in different situation as per researcher's requirements. Simulators may be complex, may be simple. As the matter of fact all the simulators must enable a user to represent a network topology, specifying the nodes on the network, the links between those nodes and the traffic between the nodes. More complex simulators may allow the user to specify everything about the protocols used to handle traffic in a network. Simulators can have graphical applications which allow users to easily visualize the workings of their simulated environment.

2.1 GLOMOSIM/QUALNET

Glomosim known as Global Mobile Information System simulator is a public domain simulator developed by UCLA. It has scalable environment for large wireless and the communication for wire line networks. GlomoSim is used for parallel discrete-event simulation. There are also various functions which can be used to send message through different layers. GloMoSim runs only on Parsec compiler. New protocols should be added in Parsec and C language.

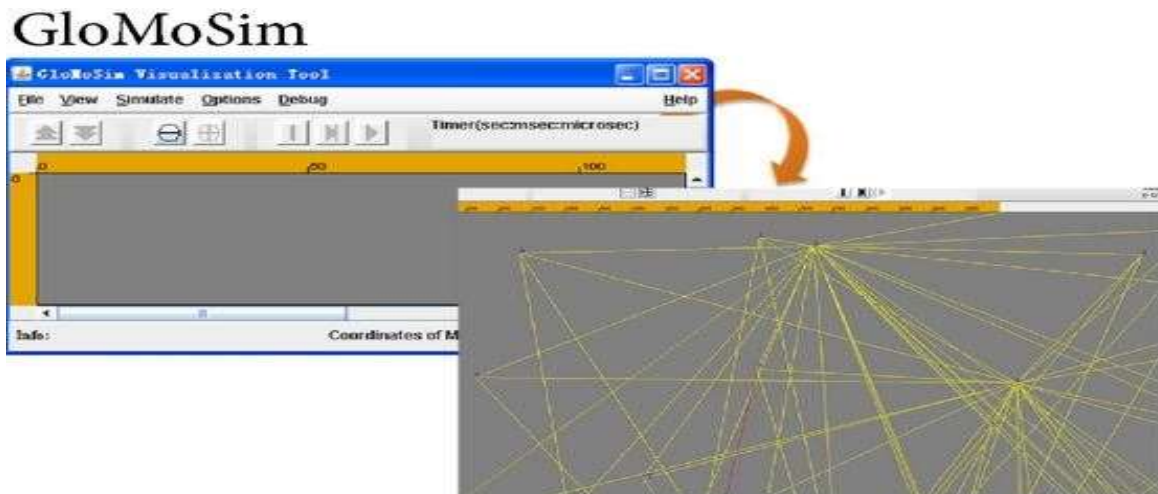


Figure 1: GloMoSim simulation Interface

Parsec code is used extensively in the GloMoSim kernel. GloMoSim is a network simulator which tries to mimic the various layers of the OSI layers. In GloMoSim, each entity represents a geographical area of the simulation. Hence the network nodes which a particular entity

represents are determined by the physical position of the nodes. Since GloMoSim implements different levels new protocols at various levels can be easily integrated.

2.2 OMNet++

OMNeT++ is an open source, component-based, modular and open architecture environment for discrete event simulation. It is free of charge for academic and non-profit use. OMNeT++ is not a network simulator; however, it is currently gaining popularity as a network simulation platform in the scientific community as well as in industrial ones, and building up a large user community.

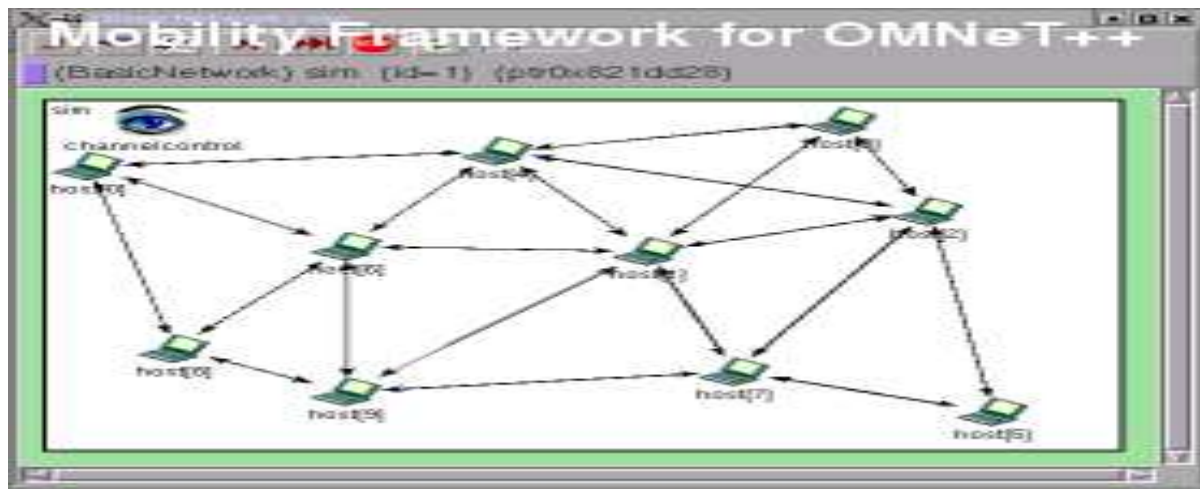


Figure 2: OMNet++ Simulation framework

In contrast to ns-2 and ns-3, OMNeT++ is not a network simulator by definition, but a general purpose discrete eventbased simulation framework. OMNeT++ simulations consist of so-called simple modules which realize the atomic behavior of a model, e.g. a particular protocol. OMNeT++ is a discrete event simulation environment. Its primary application area is the simulation of Communication networks, but because of its generic and flexible architecture, is successfully used in other areas like the simulation of complex IT systems, queuing networks or hardware architectures as well.

2.3 NS-2

It is a public domain event-driven network simulator developed at UC Berkeley. It is available on UNIX, Free BSD and Windows OS platforms. NS-2 is currently a part of the VINT (Virtual Inter-Network Test bed) project [VINT]. NS-2 is designed to simulate small-scale networks. NS-2 is based on three languages: TCL writes simulation script, OTCL defines simulation parameter, C++

implements the schedulers. The outputs produced by NS-2 can be: General format trace files, NAM format trace files, personalized trace files.

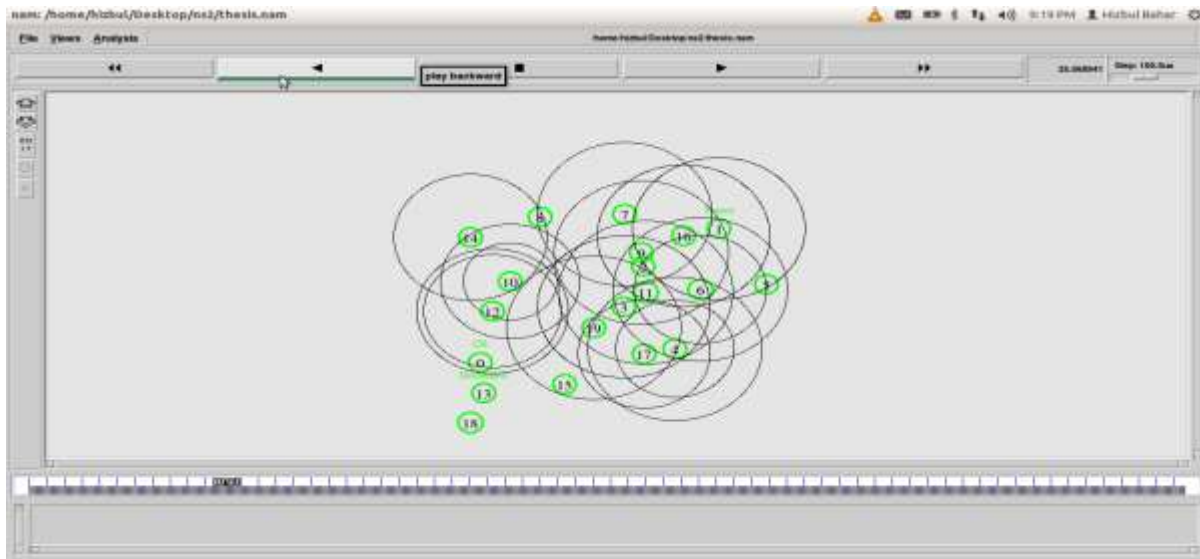


Figure 3: NS-2 Simulation Interface

Network simulations for ns-2 are composed of C++ code, which is used to model the behavior of the simulation nodes, and oTcl scripts that control the simulation and specify further aspects, for instance the network topology. Statistics says about 70% education purposes use ns2. It is open source, discrete event simulators for computer networks. NS2 code comprises of OTCL and C++. OTCL is an interpreter used to execute the commands. NS2 follows two levels of hierarchy namely C++ Hierarchy and the interpreted OTcl, which is one to one correspondence. Two languages are linked because to achieve efficiency. In ns-2, network animator (NAM) is used for the graphical view of the network. ns-2 is the most common and widely used network simulator for research work. NAM interface contains control features that allow users to forward, pause, stop and play the simulation.

2.4 OPNET

It provides a global environment to model, simulate and evaluate performances of all kinds of wired and wireless communication networks and distributed systems. It is available on Windows 2000, XP, Linux and Solaris platforms. The OPNET environment includes graphical tools for scenarios and models conception, scenarios simulation, data collection and data analysis. A simulation within OPNET is represented by a project including a set of scenarios. This project is created through the project editor also known as the OPNET central interface.

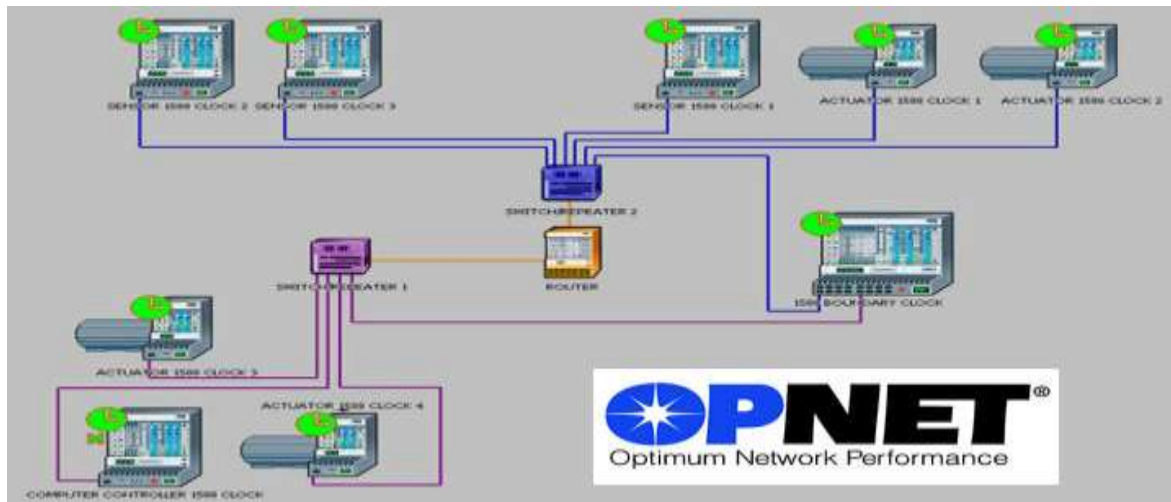


Figure 4: OPNET simulation Interface

OPNET (Optimized Network Engineering Tools) it is extensive and powerful simulation software with wide variety of possibilities to simulate entire heterogeneous networks with various protocols. OPNET is another network simulator, which again a best User Interface. It can be downloaded in IT Guru website. Test technology design in realistic conditions and evaluate enhancement to standard based protocols, develop new protocols and technologies. OPNET modeling modularity is an asset to model critical infrastructures. It offers the possibility to define and add new models specific to the infrastructure to simulate. New parameters can also be specified to model actions that might occur within the infrastructure (management, processing, etc.).

2.5 J-SIM

J-Sim is a real-time process-driven simulator, in other words, a simulation runs in the same manner as a real system does, in the sense that event executions are carried out in real time as opposed to at fixed time points in discrete event simulation. As in NS-2, two languages are used in J-Sim Java to describe and implement models and a script language to construct, configure and/or control the simulation at run-time.

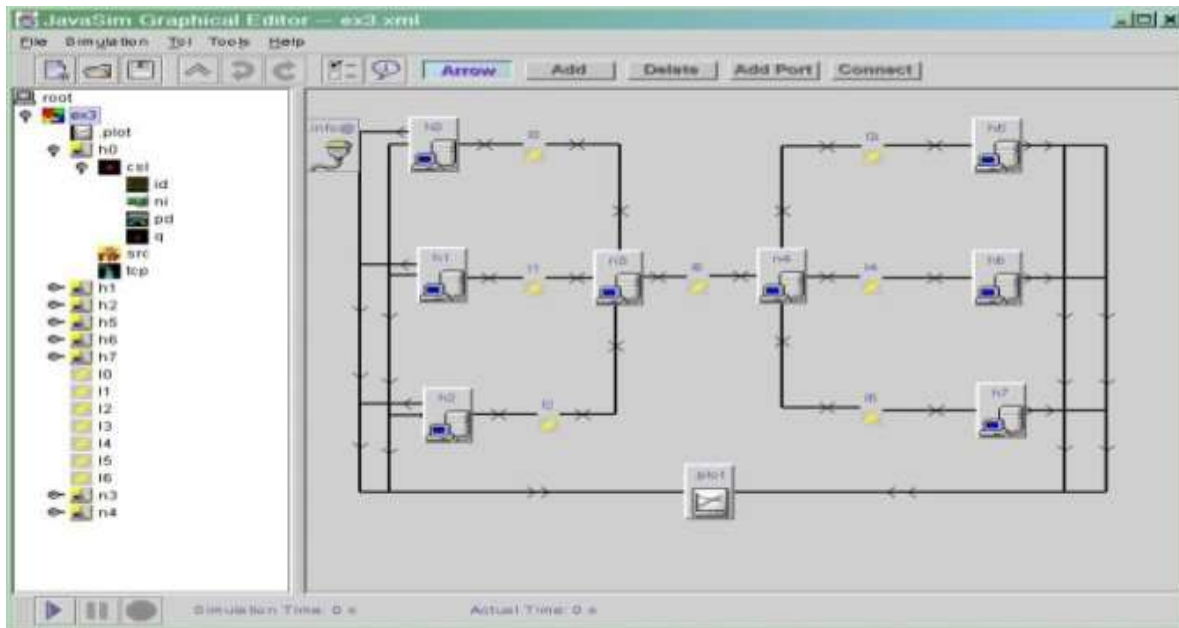


Figure 5:J-SIM simulation Interface

J-Sim has been designed to support Tcl, Perl or Python script languages, however, the available implementation is based on Tcl. J-Sim provides specific Tcl commands, the Runtime Virtual (RUV) commands, to simplify the manipulation and the configuration of the network components during simulation runtime. JSIM (Java-based simulation) it is a Java-based simulation system for building quantitative numeric models and analyzing them with respect to experimental reference data. J-Sim is an application development environment based on the component-based software architecture.

2.6 JiST

A fresh approach to network simulation is JiST ("Java in Simulation Time"), which in compliance with its name allows the implementation of network simulations in standard Java. Network simulations in JiST are made up of entities which represent the network elements, for example nodes, with simulation events being formed by method invocations among those entities.

2.7 NS-3

NS3 (Network Simulator version3) is also an open sourced discrete-event network simulator which targets primarily for research and educational use. NS3 is licensed under the GNU GPLv2 license, and is available for research and development. Like its predecessor, ns-3 relies on C++ for the implementation of the simulation models. However, ns-3 no longer uses oTcl scripts to

control the simulation, thus abandoning the problems which were introduced by the combination of C++ and oTcl in ns-2. Instead, network simulations in ns-3 can be implemented in pure C++, while parts of the simulation optionally can be realized using Python as well. Ns-3 is a discreet-event simulators primarily targeted for research and educational purposes. It was started in 2006 and NS-3 is not an extension of ns2. NS3 is a new simulator. The similarity between ns2 and ns3 are both written in C++ Codes but ns3 does not support ns2 API. In ns3 simulators is written in C++ and python. The new modules ns3 has when compared with ns2 are as follows:

1. NS 3 can handle multiple interfaces or nodes correctly
2. Use of IP Addressing and more alignment with Internet protocols and more detailed 802.11 models
3. Ns2 can be ported to ns3.

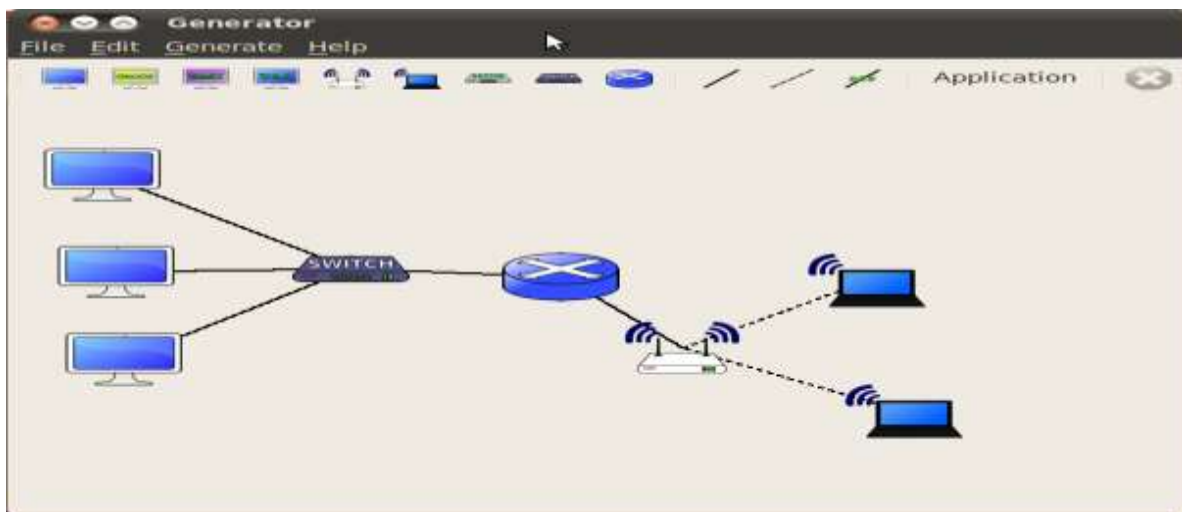


Figure 6: NS-3 simulation Interface

Like ns-2, it does not have an OTcl API. It is written in C++ language and python. The latest version of ns-3 is ns-3.10 that supports parallel simulation and has an enhanced feature set. In addition, ns-3 network simulations can be implemented in pure C++, while some parts of the simulation can also be written using Python.

3 RESULTS AND CONCLUSION

Simulation of electricity and telecommunication can be done by Qualnet, Omnet++, Opnet, Ns-2 and J-Sim, jist. Fast simulation capabilities are only supported by Qualnet and Opnet. Qualnet

and Opnet are commercial whereas Ns-2, Omnet++ and J-Sim are open source. Our results show that three of them, ns-3, OMNeT++ and JiST are all capable of carrying out large-scale network simulations in an efficient way. JiST has proven to be the fastest simulator by far in our experiments; however the exhaustive memory consumption may limit its applicability in some simulation scenarios. The other network simulators we left to explain are GosIP, NetViz, NIST, REAL, NEST, PTOLEMY with TCP Simulation. The various network simulators are explained with some user interface figure. Various network simulators enable to simulate the network but only very few simulators are used by researchers. Our work stated many simulators apart from the traditional network which will explain network researchers. In addition, ns-2 and ns-2 fully utilize the CPU, but is able to reduce CPU utilization when other applications are executed in parallel. Despite being quite new and is still under development, ns-3 demonstrates the best performance among all.

Table: Comparison of different network simulators

features	Omnet++	OPnet	J-Sim	Qualnet	NS-2
Language supported	C++	C++/Java	Java	Parsec C++	C++/OTCL
License	Open source	Commercial	Open source	Commercial	Open source
GUI support	Good	Excellent	Good	Excellent	Poor
Time taken to learn	Moderate	Long	Moderate	Very easy	Long
Platform	Linux,mac-os,unix	C,C++,Opnet modeler software	Mat lab	Linux	Unix,mac-OS Microsoft window
Availability of analysis tool	✓	✓	✓	✓	✓
Network visualization	✓	✓	✓	✓	✓
Possibility to design and modify	✓	✓	✓	✓	✓
Create trace files	✓	✓	✓	✓	✓
Interaction with real world	✓	✓	✓	✓	✓
Communication with other		✓		✓	

REFERENCES

1. Kotilainen N., Vapa M., Weber M., Töyrylä J. and Vuori J., "P2PDisCo Java Distributed Computing for Workstations Using Chedar Peer-to-Peer Middleware", Proceedings of the 19th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2005), Denver, Colorado, USA, 2005.
2. Lv Q., Cao P., Cohen E., Li K. and Shenker S., Search and Replication in Unstructured Peer-to-Peer Networks, Proceedings of the 16th International Conference on Supercomputing, ACM Press, 2002, 84-95.
3. J. Heidemann, K. Mills, S. Kumar, "Expanding Confidence in Network Simulation," IEEE Network Magazine, Vol. 15, No. 5, Apr. 2000, pp. 58- 63.
4. Jianli Pan, Prof. Raj Jain, Project , "A Survey of Network Simulation Tools: Current Status and Future Developments", report.
5. G. S. Fishman. Principles of Discrete Event Simulation. John Wiley & Sons, Inc., New York, NY, USA, 1978.
6. Neal Charbonneau nealc.com Using Tcl and ns2 01/2010 Tcl and ns2: Tutorial
7. NS Simulation for Beginners, Lecture notes, 2003 -2004 by EitanAltman ,Tania Jimenez.
8. E. Weingarten, H. V. Lehn, K. Wehrle, "A performance comparison of recent network simulators," IEEE International Conference on Communications, 2009.
9. NS-3 Simulator ns-3 Tutorial Release ns-3.15 ns-3 project November 13, 2012.
10. The network simulator, <http://www.isi.edu/nsnam/ns>, accessed Sept 20th 2011.
11. Roman Dunaytsev Department of Communications Engineering Tampere University of Technology: Network Simulators OPNET Overview and Examples.
12. T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, "ns-3 project goals," workshop on ns-2: the IP network simulator, ACM, New York, USA, 2006.

13. G. Riley. Large scale network simulations with GTNetS. In Proceedings of the 2003 Winter Simulation Conference, 2003.
14. G. Carneiro, P. Fortuna, M. Ricardo, "FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3)," Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, 2009.
15. H. N. Pham, D. Pediaditakis, and A. Boulis. From simulation to real deployments in wsn and back. Proceedings of the 2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007)., pages 1–6, June 2007.