Survey on Simulation Tools for Mobile Ad-Hoc Networks

Sujata V. Mallapur
Department of Information Science and Engineering
Appa Institute of Engineering and Technology
Gulbarga, India

Siddarama . R. Patil
Department of Electronics and Communication Engineering
P. D. A College of Engineering
Gulbarga, India

***Abstract* - Simulation is economical because it can carry out experiments without the actual hardware and provides a good compromise between complexity and accuracy. Simulation can be done by using the network simulators implemented in software which are valuable tools for researchers to develop, test and diagnose network protocols. Simulation is used to verify and evaluate the performance of networks. Research on Network depends on the following techniques- Analytical model, Emulation, Testbed and Simulation to measure the behavior and performance of protocols for wired/wireless networks. There are many different network simulators available; it is extremely difficult to choose an appropriate tool for performance testing without the complete analysis of existing tools. This paper presents the short survey about network simulators for Mobile Ad-Hoc Networks (MANETs). It helps the researchers to find a simulation model that is reliable and efficient. We have described four different network simulators (NS-2, OMNET++, NCTUns, GlomoSim) features, advantages, disadvantages of and the comparisons of them.**

Keywords- Discrete-Event Simulation, Emulation, Open Source, Testbed.

I. INTRODUCTION

During the last ten years, Mobile Ad-hoc NETworks [1] (MANET's) have become more and more popular. This still growing interest requires adapting solutions from the traditional wired networks to the wireless environment. Performance evaluation of algorithms and protocols becomes challenging at different stages of design and deployment and implementation. Performance analysis of newly designed algorithms and protocols are extremely desirable for efficient MANET's deployment. For the performance analysis of MANETs the follow techniques that can be used are Analytical Modelling, Testbeds and Simulation/Emulation. Analytical methods require certain simplification to model and predict the performance. Therefore, they are inappropriate due to inherent complexity and diverse nature of MANETs. Oversimplified models may lead to inaccurate results that are not desirable.

Testbed- It can be implemented and deployed on actual hardware. While testbeds might yield the most accurate results, there are several drawbacks, such as: the need to obtain hardware and the severely limited monitoring and debugging possibilities, as well as, high effort needed to create an artificial environment resembling the real application scenario. Hence, testbed implementations will generally be an option only for smaller numbers of nodes and during the later stages of the implementation phase.

Emulation is hybrid approach that combines hardware and software where some components are implemented on real hardware and some are simulated.

Simulation is economical because it can carry out experiments without the actual hardware. It is flexible because it can, for example, simulate a link with any bandwidth and propagation delay or a router with any queue size and queue management policy. Simulation results are easier to analyse than experimental results because important information at critical points can be easily logged to help researchers diagnose network protocols. There are three types of simulation: Monte Carlo Simulation, Trace-Driven Simulation and Discrete-Event Simulations. The last two simulations are used commonly in MANETs. Discrete-event simulation is widely used for MANETs, because it can easily simulate lots of jobs running on different sensor nodes. This simulation can list pending events, which can be simulated by routines. Trace-Driven Simulation provides different services. This kind of simulation is commonly used in real system. The simulation results have more credibility. It provides more accurate workload; these detail information allow users to deeply study the simulation model.

Simulation can be done by using the Network Simulators implemented in software and are valuable tools for researchers to develop, test and diagnose network protocols. Different types of network simulators can be categorized and explained based on some criteria such as commercial or free. Some of the network simulators are commercial which means that they would not provide the source code of its software or the affiliated packages to the general users for free. All the users have to pay to get the license to use their software or pay to

order specific packages for their own specific usage requirements. Some of the typical examples are the OPNET [2] and QualNet [3]. Commercial simulator has its advantage and disadvantage. The advantage is that it generally has complete and up-to-date documentations and they can be consistently maintained by some specialized staff in that company. However, the open source network simulator is disadvantageous in this aspect, and generally there are not enough specialized people working on the documentation. This problem can be serious when the different versions come with many new things and it will become difficult to trace or understand the previous codes without appropriate documentations. On the contrary, the open source network simulator has the advantage that everything is very open and Every individuals or organization can contribute to it and find bugs in it. The interface is also open for future improvement. It can also be very flexible and reflect the most new recent developments of new technologies in a faster way than commercial network simulators. We can see that some advantages of commercial network simulators, however, are the disadvantage for the open source network simulators. Lack of enough systematic and complete documentations and lack of version control supports can lead to some serious problems and can limit the applicability and life-time of the open source network simulators. Some of the open source network simulators include NS-2 [4], NS-3 [5], OMNeT++ [6], SSFNet [7], NCTUns [8], and J-Sim [9].

The aim of this survey is to find a MANET simulator that provides a good balance between features, efficiency, extendibility, accuracy, and easiness of use. Such a simulator will allow researchers to take the steps described above without much pester, and allow them to concentrate on their research rather than the simulator. Therefore, this survey contains information about features, strengths and weakness of different network simulators and comparison based on a collection of papers and surveys, some of which compare different Network Simulators according to defined criteria and some of which analyze simulators performance to a specific project or area.

The reminder of this paper is organized as follows. In Section II, we introduce previous work related to our study. In Section III is the main section of our paper where six Network Simulators are described in detail. In Section IV, Comparison between the Simulators is discussed. The paper is concluded in Section V.

II. Related Work

There are several surveys, comparisons, and also some case studies about wireless network simulators. They all differ with respect to the selection of evaluated simulators, the intention of the work, the focus of the potential comparison and the level of detail.

In [10] the author described a Case Study in which four popular Network Simulators, J-Sim, OMNeT++, NS-2 and ShoX, were used to evaluate a topology control protocol. The author described the outstanding features and also compared the effect needed for the installation familiarization, implementation visualization from features and useability point of view.

In [11] the author gives map characteristicss that MANETs Simulation tools should feature and current support of these gives the description of a DIANEmu, GlomoSim, GTNets, J-Sim, Jane, NAB, NS-2, OMNet++, OPNET Modeler, QualNet, and SWANS Simulators. And provides the estimation of their popularity, and gives some hints on which simulator to use for what needs.

In [12] author simply presents a comparative study of two network simulators, OPNET Modeler and NS-2, and provides a guide to researchers undertaking packet-level network simulations. The simulator outputs were compared to the output from a live network testbed.

In [13] is a deliverable of a project called IRRIIS, Integrated Risk Reduction of Information-based Infrastructure Systems, the purpose of which is to identify, list and compare tools and components suitable for simulation of critical infrastructures. The simulators used are OPNET Modeler, NS-2, QualNet, OMNeT++, J-Sim, SSF and Backplane.

In [14] author gives an overview about different issues in wireless sensor networks on a general basis. Only at the end a table is presented comparing the considered simulators according to their language, the available modules, and whether they have GUI support or not.

In [15] the author described a comparative study by considering the installation, familiarization, implementation and visualization of popular Network Simulators NS-2, J-sim, and OMNET++ and Shox.

In [16] author described a case study of Simulation tools for wireless Networks, compared strengths and weakness of NS-2, GloMoSim, J-Sim, OMNeT++, OPNET, QualNet.

III. Network Simulations Tools

A. NS-2

NS-2 [4] is a discrete event simulator which provides support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. It is the most popular network simulator used by researchers. The Network Simulator began as a variant of the REAL [17] network simulator in 1989 and has evolved over the past years.

NS-2 is written in C++ and is based on two languages: C++ which is used to extend the simulator (i.e. to define protocol behaviors), and OTcl [18], an object-oriented extension of Tcl developed by David Wetherall as part of the VUsystem project at Massachusetts Institute of Technology,

which is used for scenario configuration, manipulation of existing C++ objects, periodic or triggered actions, etc.

In NS-2, to create the topology of the network for simulation some of topology generators may use they are Inet Topology Generator, GT-ITM (Georgia Tech Internetwork Topology Models) topology generator or Tiers Topology Generator and convert their outputs to NS-2 format. Generation of topologies by hand is another option. The simulation event scheduler of the simulator, contained in OTcl script interpreter, is either a non real-time scheduler or a real time scheduler which is mainly used for real-time synchronization of an emulated network. The user indicates in the event scheduler when network elements should start or stop transmitting packets. In order to visualize a network simulation in NS-2, traffic and movement patterns should be generated and references as inputs into the OTcl code configuring the simulation scenario. The simulation can then be visualized by Nam [19] which is shown in figure 1. Using the trace file generated by the simulator. However, states that Nam cannot be used for accurate simulation analysis. Node appearance can be changed according to node's inner state, but it is limited. Extended Nam Editor can be used to graphically create simple scenarios and save them as script files. For statistics plotting, external tools like gnuplot or Xgraph must be used. There is a function in OTcl configuration file which can periodically call itself to collect statistical data of simulation and write them into a file. The file generated by the function can be referenced as input into external tools for plotting of data like Xgraph [21] and Gunplot [22].

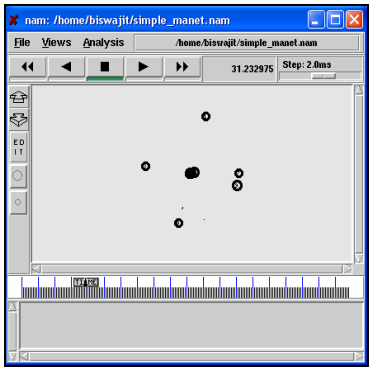

Figure1: NAM of NS-2

1. *Features*

- NS-2 supports deterministic or probabilistic packet loss in queues attached to network nodes as well as it supports deterministic and stochastic modeling of traffic distribution.
- NS-2 provides emulation facility; NS-2 can be connected to a real network and capture live packets just like a common node. It can also inject packets into the live network.
- The simulator can generate personalized trace files by allowing users to select parameters to be traced, therefore saves CPU resource.
- NS-2 offers a comprehensive documentation and a regularly updated manual as well as an API for C++ and OTcl classes.
- Other features of the simulator include models for different network architectures including Wireless LAN, MANET and satellite, built-in traffic models with support for development of new ones, plugging of new pseudorandom number generators, and net- work state estimation .

2. *Advantages and disadvantages*

Advantages:

1. NS-2 has advantages of large number of available models, realistic mobility models, powerful and flexible scripting and simulation setup, large user community, and ongoing development.
2. NS-2 includes an energy model and it allows user to easily generate traffic and movement patterns.
3. It provides a set of randomized mobility model and there are several projects to bring advanced mobility models to the simulators.

Disadvantages:

1. NS-2 needs to be recompilation every time if there is a change in the user code.
2. It is not so well structured software architecture and the mixture of compilation and interpretation made it difficult to analyze and understand the code.
3. Simulation running will be very slow especially when the network simulated contains many nodes.

B. OMNeT++

OMNeT++ [6] [28] (Objective Modular Network Testbed in C++) is a open source, discrete event simulator tool written in C++. OMNET++ is a general-purpose simulator capable of simulating any system composed of devices interacting with each other. OMNeT++ supports wireless and mobile simulations within OMNeT++. This support is said to be fairly incomplete. OMNet++ is for academic and educational use.

OMNeT++ [28] provides a component-based, hierarchical, modular and extensible architecture. Components, or modules, are programmed in C++ and new ones are developed using the C++ class library which consists of the simulation kernel and utility classes for random number generation, statistics collection, topology discovery etc. New modules may be derived from basic object classes like module, gate or connection. A high-level language called Network Description (NED) is used to assemble individual components into larger components and models. Alongside the simulation kernel library, the simulation environment contains a Graphical Network Editor (GNED), a NED compiler, graphical (Tkenv) and command line (Cmdenv) interfaces for simulation

execution, graphical tools for simulation result analysis (Plove, Scalars), a model documentation tool, utilities (random number seed generation tool, etc.), documentation and sample simulations. the simulator includes modules for Application Layer and Network Layer of OSI model as well as a Network Interface Card module which encapsulates MAC and PHY layers. OMNeT++ has extensive GUI support. This is shown in the figure 2. GNED allows building of network topologies graphically, therefore provides easy definition of network simulation scenarios. It can generate topologies as NED files. Tkenv supports interactive execution of the simulation, tracing and debugging. The user is able to start or stop the simulation execution in Tkenv and can change variables or objects inside the model at runtime. It provides the user with a detailed picture of the simulation state at any point of execution.

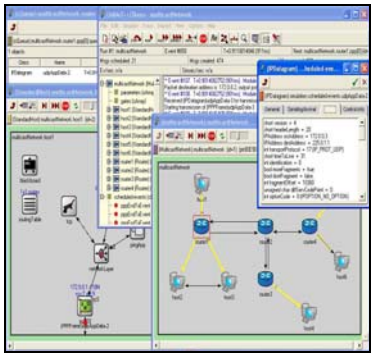

Figure 2: OMNeT++ GUI during simulation.

In OMNeT++, basic entity is a module. Modules can be atomic, which capture the actual behavior, or they can be composed of sub-modules. Modules communicate with each other by sending and receiving messages through their gates which are linked to each other via connections. Gates can be associated with propagation delay, error rate and data rate, and they support only 1-to-1 correspondence. Connected modules form compound modules. Every simulation model is an instance of a compound module type, i.e. hierarchically nested modules. This level of components and topology is dealt with in NED files. Each module is able to generate, read or react to messages (also known as events).

1. Features

- OMNeT++ is a feature-rich and powerful simulation tool.
- OMNeT++ has external extensions which enable it to provide support for simulation of wireless networks. Two most known and used extensions are INET Framework and Mobility Framework for mobile ad-hoc networks.
- Some features of the simulator are failure modeling at specified times at runtime for nodes and links, built-in traffic scenarios with support for custom development, creation of any form of hierarchical topology using NED language and multiple open source pseudorandom number generators some of which are quite feasible for large scale simulations
- A future development is that simulation executables created by the simulator are actually standalone programs that can be run on other machines without the simulators.

2. Advantages and Disadvantages

Advantages: 1. OMNeT++ provides a powerful GUI. This strong GUI makes the tracing and debugging much easier than using other simulator.
2. OMNeT++ accurately models most hardware and includes the modeling of physical phenomena.

Disadvantages:

1. It does not offer a great variety of protocols, and very few have been implemented, leaving users with significant background work if they want to test their own protocol in different environments.
2. Poor documentation and poor analysis of typical performance measures [25].
3. The mobility extension of the simulator fairly incomplete [26].

C. NCTUns

NCTUns [8] network simulator, which is a high-fidelity and extensible network simulator capable of simulating both wired and wireless IP networks. NCTUns stands for National Chiao Tung University network simulator. The predecessor of the NCTUns is the Harvard network simulator [22], which was authored by S.Y. Wang in 1999. The Harvard network simulator has several limitations and drawbacks that need to be overcome and solved. Some useful features and functions need to be implemented and added to it. For these reasons, S.Y. Wang developed the NCTUns.

The NCTUns uses a distributed architecture to support remote simulations and concurrent simulations. It also uses open-system architecture to enable protocol modules to be easily added to the simulator. Functionally, it is divided into eight separate components some of most important components are described: The first component is the fully-integrated GUI environment by which a user can edit a network topology, configure the protocol modules used inside a network node, specify mobile nodes' moving paths, plot performance curves, play back animations of logged packet transfers, etc. The component topology editor is used to generate topology [27], which is shown in Figure 3. The nodes are created using the node editor. A node in the NCTUns represents a network device such as a switch or an IEEE 802.11 (b) wireless LAN access point. This is shown in the figure 4. The node editor provides a convenient environment

to flexibly configure the protocol modules used inside a network node. By using this tool, a user can use the mouse to graphically add, delete, or replace a protocol module with his (her) own module. As such, the node editor enables a user to easily test the functionality and performance of a new designed protocol.

The second component is the simulation engine. A simulation engine is a user-level program. It functions like a small operating system. Through a defined API, it provides useful and basic simulation services to protocol modules (to be described soon).

The third component is various protocol modules. A protocol module is like a layer of a protocol stack. It performs a specific protocol or function.

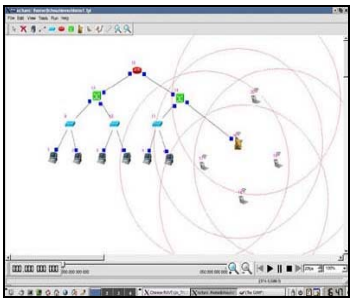

Figure 3 . Topology editor of the NCTUns simulator

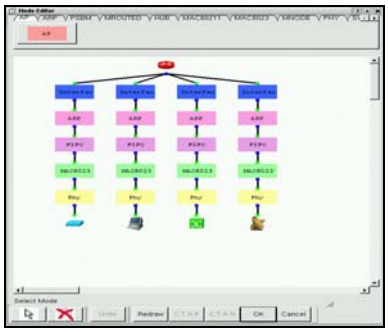

Figure 4 . Node editor of the NCTUns simulator

For example, the ARP protocol or a FIFO queue is implemented as a protocol module. A protocol module is composed of a set of functions. It needs to be compiled with the simulation engine to form a simulation server. Inside the simulation server, multiple protocol modules can be linked into a chain to form a protocol stack. The fourth component is the simulation job dispatcher, which is a user-level program. It should be executed and remain alive all the time to manage multiple simulation machines. The fifth component is the coordinator. This is a user level program. On every machine where a simulation server program resides, a coordinator program needs to be executed and remain alive. Its task is to let the job dispatcher know whether this machine is currently busy running a simulation or not. When executed, it immediately registers itself with the dispatcher to join the dispatcher's simulation machine farm. To plot the graph performance monitor component is used. This is shown in figure 4.

1. *Features*

- The NCTUns has a fully-integrated GUI environment by which a user can easily perform simulation studies.
- The topology editor provides a convenient and intuitive way to graphically construct a network topology, by specifying various parameters of network devices and protocols. A network can be either a fixed wired network or a mobile wireless network.
- The performance monitor, easily and graphically display the plots of some monitored performance metrics such as a link's utilization or a TCP connection's achieved throughput.

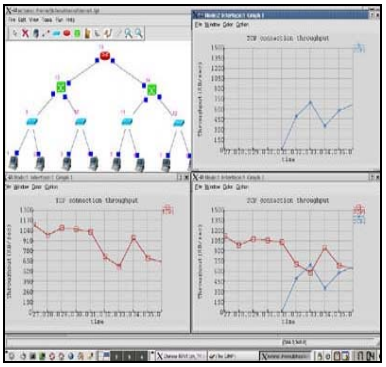

Figure 5 . The performance monitor of the NCTUns network simulator

- The packet animation player, using the packet animation player, a logged packet transfer trace can be graphically replayed at any speed. The packet animation player is a very useful tool because it can help a researcher to visually debug the behaviors of a protocol.
- The main features of NCTUns are Tunnel Network Interface, Simulating Single-hop Networks, and Simulating Multi-hop Networks.

2. *Advantages and Disadvantages*

Advantages:

1. It provides easy to use GUI Environment.
2. Its distributed and open-system architecture design supports remote simulations and concurrent simulations, and allows new protocol modules to be easily added to its simulation engine [27].
3. NCTUns [27] provides better functionality and performance.

Disadvantages:

1. The connection through dispatcher with the simulation server is not stable. Indeed it is frequent that the coordinator becomes busy. In this case, it notifies its state to the dispatcher, this will not be able to select the appropriate simulation machine. Therefore it is necessary to start again the coordinator, the dispatcher and the client.
2. The programming is not supported by the NCTUns. So simulation parameters set only by graphical user interface.
3. The manipulation at every node has to be done node by node, or all the nodes to the same time.

D. GloMoSim

GloMoSim [23] is a scalable simulation environment for wireless network systems. It is being designed using the parallel discrete-event simulation capability provided by PARSEC [24]. Most network systems are currently built using a layered approach that is similar to the OSI seven layer network architecture. The plan is to build GloMoSim using a similar layered approach. Standard APIs will be used between the different simulation layers. This will allow the rapid integration of models developed at different layers by different people. The goal is to build a library of parallelized models that can be used for the evaluation of a variety of wireless network protocols. The proposed protocol stack will include models for the channel, radio, MAC, network, transport, and higher layers. GloMoSim stopped releasing updates in 2000. Instead, it is now updated as a commercial product called QualNet [3].

1. Features

- GlomoSim allows the simulation Scalability to simulate networks with a hundred and thousand of nodes.
- GloMosim Supports protocol for the wireless networks.
- GloMoSim provides the Random Waypoint mobility model, which may not be suitable for all types of simulations.
- The BonnMotion software provides a generator for other kinds of mobility models.
- GloMoSim is designed to be extensible, with all protocols implemented as modules in the GloMoSim library

2. Advantages and disadvantages

Advantage

1. Achievement of large scalability, good mobility models (“random drunken model” and “random distribution model”), specify for wireless simulation, many ad hoc networking protocols support and analysis and visualization tools are basic but sufficient for general studies.
2. The ability to use GloMoSim in a parallel environment distinguishes it from most other wireless network simulators.

Disadvantage:

1. Documentation is quite poor. No specific routing protocols for sensor network, No energy consumption models Still has transport layer and IP address support.
2. Hard for user is to simulate large sensor network since no hardware environment.
3. The main disadvantage of GloMosSim is update of this simulator is not regular.

3. COMPARISONS

In this section we summaries the main streams of MANET simulators in the table 1(see Appendix). Table 1 has six simulators considered in the previous section listed in the consecutive rows and special features in the consecutive columns.

Network simulators exhibit different features and models. Each has its own advantages and disadvantages and each is appropriate in different situations. All four simulators common feature is Discrete-Event Simulators and General Purpose Simulators.

4. CONCLUSION

We described the four network simulators that support the simulation of MANET’s (NS-2, OMNET++, NCTUns, GloMoSim) features, advantages and disadvantages. According to the survey, simulators have the many its features, but none of them offer the good support for all features for MANET simulation. Therefore we have searched a balanced simulator that would offer a good user experience for MANETs.

NS-2 and OMNET++ are the best choices for the MANETs. NS-2 profits from the large available models, Ns-2 supports wide range of protocols in all range of protocols in all layers for example, the Specific MANET routing protocols are provided by the NS-2. While the OMNET++ supports the powerful GUI, well defined simulation engine and supports hierarchical modeling, so it is better for development. GloMoSim is also a good and are strong scalability power and it is useful when the wireless network contains the large number of nodes.

REFERENCES

[1] Abolhasan, M., Wysocki, T., and Dutkiewicz, E., "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks 2*, pp. 1-22 (2004).

[2] OPNET Modeler, http://www.opnet.com/.

[3] The QualNet . http://www.qualnet.com.

[4] The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[5] NS3 official website, http://www.nsnam.org/documents.html.

[6] OMNeT++ Community Site. http://www.omnetpp.org/.

[7] Scalable Simulation Framework (SSF), SSFNet homepage, http://www.ssfnet.org /homePage.html.

[8] Shie-Yuan Wang*,y and Yi-Bing Lin, "NCTUns network simulation and emulationfor wireless resource management", *Wirel. Commun. Mob. Comput.* .Page No. 899–916, 2005.

[9] J-sim Official. http://sites.google.com/site/jsimofficial/.

[10] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. "Comparative Study of Wireless Network Simulator", IEEE *The Seventh International Conference on Networking, pages 517-523*, 2008.

[11] J. L. Hogie, P. Bouvry, and F. Guinand. "An Overview of MANETs Simulatio*", In Electronic Notes in Theoretical Computer Science, Proc. of 1st International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005), LNCS*, pages 81101, April 2005. Elsevier.

[12] G. F. Lucio, M. Paredes-Farrera, E. Jammeh, M. Fleury, and M. J. Reed. "OPNET Modeler and Ns-2 - Comparing the Accuracy of Network Simulators for Packet-level Analysis Using a Net-work Testbed*", WSEAS Transactions on Computers*, 2(3):700707, July 2003.

[13] S. Duflos, G. L. Grand, A. A. Diallo, C.Chaudet, A. Hecker, C. Balducelli, F. Flentge, C. Schwaegerl, and O. Seifert. Deliverable d "*List of Available and Suitable Simulation Component"*,Technical report, Ecole Nationale Supieure des Tommunications (ENST), September 2006.

[14] B. Schilling. "Qualitative comparison of network simulation tools*", Technical report, Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart*, January 2005.

[15] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. "Comparative Study of Wireless Network Simulator", *The Seventh International Conference on Networking*, pages 517-523, 2008.

[16] S. Mehta, Niamat Uallh, Md. Humaun Kabir , "A Case Study of Network Smulation Tools for Wireless Networks", 2009 *Third Asia international Conference on Modeling & Simulation IEEE*, Page No. 661-666, 2009.

[17] The REAL network simulator. http://www.cs.cornell.edu/skeshav/real/overview.html.

[18] OTcl. http://otcl-tclcl.sourceforge.net/otcl/.

[19] Nam: Network Animator. http://www.isi.edu/nsnam/nam/.

[20] Xgraph. http://www.isi.edu/nsnam/xgraph/index.html.

[21] Gnuplot Homepage. http://www.gnuplot.info/.

[22] Harvard TCP/IP network simulator 1.0, available at http:// www.eecs.harvard.edu/networking/simulator.html.

[23] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. "*GloMoSim: A Scalable Network Simulation Environment*", *UCLA Computer Science Department Technical Report 990027*, May 1999.

[24] R. Bagrodia, R. Meyer et al., "*PARSEC: A Parallel Simulation Environment for Complex Systems*", IEEE Computer, October 98.

[25] L. Begg,W. Liu, K. Pawlikowski, S. Perera, and H. Sirisena. "Survey of Simulators of Next Generation Networks for Studying Service Avail ability and Resilience. Technical Report TRCOSC 05/06", *Department of Computer Science & Software Engineering, University of Canterbury, Christchurch*, New Zealand, February 2006.

[26] J. L. Hogie, P. Bouvry, and F. Guinand. "An Overview of MANETs Simulation", *In Electronic Notes in Theoretical Computer Science, Proc. of 1st International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005*), LNCS, pages 81101, April 2005. Elsevier.

[27] S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, and C.C. Lin "The Design and Implementation of the NCTUns 1.0 Networ Simulator" .

[28] Murat Miran K¨oksa *"A Survey of Network Simulators Supporting Wireless Networks*"

[29] D. Johnson, D. A. Maltz, "Dynamic Source Routing Protocol in Ad Hoc wireless Networks," *Kluwer Academic Publishers*, 1996

[30] C. E. Perkins and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing (Internet-Draft)," *Proceedings of the Second IEEE Workshop on Mobile Computing System and Application*, New Orleans, LA, USA, pp. 90-100, February 1999.

[31] Zygmunt J. Haas, Cornell University Marc R. Pearlman, "Cornell University the Zone Routing Protocol (ZRP) for Ad Hoc Networks", *draft-ietf-manet-zone-zrp-02.txt>*, 2001.

APPENDIX

Sr. No	Main Streams	Network Simulators			
		NS-2	**OMNET++**	**NCTUns**	**GloMoSim**
1.	**Emulation Support**	Limited Support	Limited Support	Yes	No
2.	**License**	Open Source	Open Source	Open Source	Open Source
3.	**GUI**	No	Yes	Yes	Limited
4.	**Interface**	C++ , OTcl	C++ , NED	C	C
5.	**Available Modules**	Wired, Wireless, Ad-Hoc and Wireless Sensor Networks.	Wired, Wireless and Ad-Hoc Networks	Wired, Wireless and Ad-Hoc Networks	Wired, Wireless and Ad-Hoc Networks
6.	**Documentation and User Support**	Excellent	Medium	Good	Poor
7.	**Scalability**	Small	Large	Medium	Large
8.	**Extendable**	Yes	Yes	Yes	Yes
9.	**Simulation Technique**	Discrete Event Simulation	Discrete Event Simulation	Discrete Event Simulation	Discrete Event Simulation

Table 1: Comparison of Network Simulators