ANALYSIS STUDY AND COMPARISON OF RECENT TRENDS IN NETWORK SIMULATOR

[1]Prof. M. S. Shegokar, [2]Prof. V. R. Tripathi,

[1]Assistant Professor, [2] Assistant Professor
[1]Computer Science and Engineering,
[1]College of Engineering and Technology, Akola, India

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.

***IndexTerms* – NS-2, NS-3, OMNET++, OPNET.**

INTRODUCTION

Network Simulation is a basic research tool in networking for the program frames up and controls the performance of a network. It is done in two different ways. It is a method to analyze the diverse activities of a network under various presumed network conditions. It has multifaceted benefits; one such is its ability to consume lower cost when weighed against real time implementation. It is first method to calculate the interaction among the different network entities such as hosts or packets. The other method is done by using mathematical formulas, or by captivating and then playing back these captured observations in order to form a production network. The performance of the network and the multiple applications associated with it and its services and technical support can be analyzed through test lab.

II. TYPES OF NETWORK SIMULATORS

The performance of certain computer network can be envisaged by certain network simulators that results in viewing the software that are used in networking areas. In case of conventional methodical techniques of communication networks have turned out to be way too complex to present an absolutely precise understanding of system behavior of network simulator that are made use of. The network of computer is characteristically modeled on the basis of devices, links, applications and many more in simulators. Fundamentally the network simulators are categorized into two broad types they are :(Egea-Lopez & Vales-Alonso, 2005)

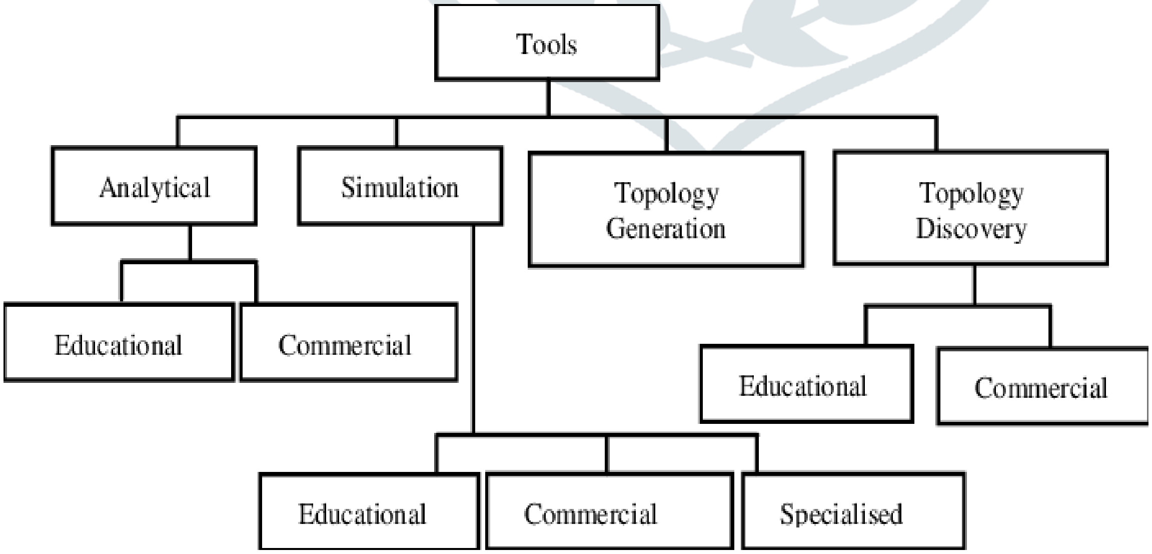

Fig. 1.1: Classification of network design and simulation tools

● GUI driven
● CLI driven

In today's scenario the user makes use of simulators normally so that it is supported by means of the most popular technologies. In the field of communication among the computer network research area, network simulation is a prominent modus operandi that are programmed models of the performance of a network each by calculating the interaction among the unusual network entities such as hosts or packets, and so forth. This process involves making use of mathematical formulas, or essentially incarcerating and staging back surveillance regarding a production network. The performance of the network and all the mixture of applications and services it chains can later on be observed by means of a test lab. Also multiple features of the environment can also be personalized in a prescribed approach in order to evaluate the working of network and how will it perform underneath diverse conditions. A network simulator can also be considered as software that forecasts the manners of a computer network's working. Every single network simulators possess multiple features; it is only these features that determine its implementation on various applications. For instance OMNET++ provides GUI support and NS3 offers attribute system support and OPNET offers grid computing support. The hardware of network simulator can never be modified or its usage cannot be minimized even with the help of any external hardware. It is also impossible to reduce the difficulties that come with it.

III SIMULATION TOOLS

3.1 Network Simulator 3 (NS3)

A type of open source discrete-event network simulator used for targeted primarily for research and educational purposes and Internet systems are named as Network Simulator version3. NS3 is a free open source network simulator licensed under the GNU GPLv2 license that is not an extended version of NS2. NS3 is available for all development and research work in now a day; it can rely on to design a new models, debug and maintain an existing one and also share the resultant output. NS3 is built as a library which is linked to a main C++ program dynamically or statically that also defines the type of simulation topology and then it starts the simulators. OMNET++ provides GUI support and NS3 offers attribute system support and OPNET offers grid computing support. NS3 is still under various developments. It has certain chief challenges. One of the most prominent and prevalent challenges is that NS3 requires contribution from the research community. Apart from this its primary concern is attaining the reliability of simulation by means of necessitating the changes that are to be enhanced. The predominating big issue in NS3 is its comparison to NS2. It is created with the intention of reproducing the successful mode of NS2. It is due to this various lot of organizations of elite classes contributed to this model and its components based on the framework of NS2. They both have unlike software core; NS3 is comparatively more attentive to realism. It has Integrated software and is more supportive of virtualization. They both have dissimilar software core. NS3 is comparatively more attentive to realism. It has integrated software and is more supportive of virtualization. The NS3 simulator is the third and improved version of network simulator. Unlike the second version of network simulator known as the NS2 this NS3 is a discrete-event network simulator aimed to benefit the research and educational purposes. It is accredited with the license called GNU GPLv2, and as mentioned before it is mainly accessible for research and development-3 purposes. It also serves as a definition of the representation of working course of action of packet data networks, and it also endows with an engine for simulation. A number of users are making use of NS3 to frame non-Internet-based systems. NS3 makes use of two key languages similar to the previous version of network simulator called NS2. This NS3 simulator is exclusively written in C++ along with some discretionary python bindings. Thus how simulation scripts can consequently be written in C++ or in Python. In order to display the results visually animators are made use. The above mentioned both languages work nicely on it. NS3 also supplies a strong library which is very helpful for the user to do their work by editing NS3 itself. For the sake of wired technology, the NS3 presents a device model of an uncomplicated network of Ethernet which makes use of CSMA/CD as its protocol scheme with exponentially increasing back-off to contend for the shared transmission medium. It also provides a set of 802.11 models that attempt to provide an accurate MAC-level implementation of the 802.11 specification and a PHY-level model of the 802.11a specification.

Type: Open source.
Languages: C++ and Optional Python Bindings.
Supported Operating System: GNU/Linux, FreeBSD, and Mac OS X, Windows XP, Win-Vista and Windows7.
Hardware Requirement: Free disk space of 5GB required, minimum 256MB RAM suggested
Features

- It condenses the memory footprint of the simulation and assign no memory for virtual zero bytes. Mobility model is not needed, as the wired devices do not need to know its node position.
- The core of NS3 is written in C++ and with Python scripting interface (compared with OTCL in NS2). Several advanced C++ design patterns are also used.
- It generates PCAP packet trace files, other utilities can be used to analyze traces as well. Users can write program for designing a simulation either in C++ or in Python Programming Language.
- The protocol entities are designed to be closer to real computers.
- It supports the incorporation of more open-source networking software and reduce the need to rewrite models for simulation.
- Alight weight virtual machines are used.

Merits

- It posses high modularity compared to its ancestor NS2.
- Support simulation for TCP, UDP, ICMP, IPv4, multicast routing, P2P and CSMA protocols.
- The supplied support for ported code should make model validation easier and more credible.
- It is incredibly flexible than any other simulators.
- Its extensive range of use in both optimization and expansion of the existing networks.

Demerits

- NS3 has to put up with from lack of credibility.
- NS3 is intended to replicate the successful mode of NS2 in which a lot of different organizations contributed to the models and components based on the framework of NS2.
- NS3 needs a lot of specialized maintainers in order to avail the merits of NS3 as the commercial OPNET network simulators.
- Active maintainers are required to respond to the user questions and bug reports, and to help test and validate the system.

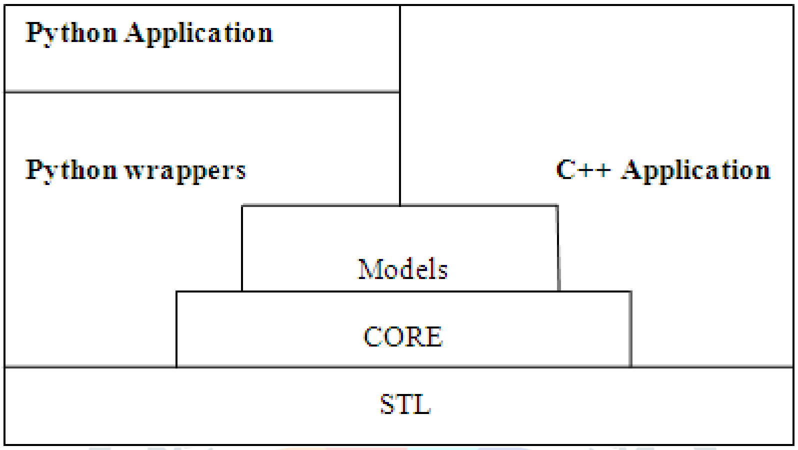

Fig. 1.2: Architecture of NS3

The above fig. 1.2 presents a pictorial representation of the architecture of NS3. Fig. 1.2 shows that the Python programs are predominantly used for bringing in the NS3 modules. Apart from this the python binding have also been modularized in the past few years. When it comes to using sockets the NS3 maintains both simulation and emulation equally. In the case of NS3 the protocol entities that are already planned to be closer to real world computers originates more consideration to the practicality. Multiple open source networking software's are supported by NS3 as result of this there is a software integration specialty to it (Rajankumar, Nimisha, & Kamboj, 2014). Nowadays in NS3 there are multiple types of virtual machines that are made use. The virtual machine that lends its supporting the features to NS3 is as follows:

- Testbed integration
- Attribute system
- Tracing Architecture
- Topology

Features of NS3

The Major and notable features of NS3 are as follows:

- Modular, documented core
- C++ programs and Python scripting
- Alignment with real systems
- Software integration
- Virtualization and test-bed integration
- Attribute system
- Updated models

Apart from the above mentioned features NS3 simulator is also used for research and education purposes for internet systems. Some other additional and equally significant features of NS3 are listed under statements:

Attention to realism

Each protocol entities are designed to be closer to the real time computer which includes the support for key interfaces such as sockets, multiple interfaces per nodes, use of IP address, network devices and other similarities.

Software Integration

NS3 architecture supports more open sourced networking software's which are routing daemons, kernel protocol stacks, packet trace analyze which are reducing the need to port or rewrite the tools and models for simulation.

Software core

The core is mainly written in C++ language and it has an optional python scripting interface. The main aim of designing such a core is to design to improve coding style, modularity, scalability, and documentation. It also supports several C++ design patterns such as templates, callbacks; smart pointers and copy-on-write are most important advantages of NS3. It also supports an object aggregation capability which enables easier model and packet extensions.

3.2 Objective Modular Network Tested in C++ Simulator(OMNET++)

OMNET++ can be described as a component based simulation package. This network simulator is constructed on the basis of C+++ language. This OMNET++ is corresponded by the means of small, reusable modules. All of these modules are well written in C++. OMNET++ which is expanded as Objective Modular Network Tested in C++ (Varga & Hornig, 2008) is an open source, extensible, modular and component based discrete event simulator tool which is similar to both NS-2 and NS-3 in its act of simulating both wired and wireless networks. This network simulator is entirely written in C++. Just like NS3 it is also mainly used for the sake of research and educational purposes that is beneficial for the entire global scientific community. It also presents an Eclipse-based IDE, a graphical runtime environment and a host of some other tools. This OMNET++ is a general-purpose simulator which has the capability of simulating any system unruffled of devices that interact with each other. The support offered by this network simulator for wire and wireless simulation is reasonably incomplete. OMNET++ also endows with a component-based, hierarchical, modular and extensible architecture. Components and modules are programmed in C++, and then assembled into larger components and models using a high-level language (NED*)*. It also presents reusability of model for free. OMNET++ has vast GUI support and because of this modular architecture, the simulation kernel and other models can be embedded easily into applications. Beside 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.

Type: Free for academic and non-profit use. Commercial user must obtain its license.

Language: Full system is developed with C++. But it also supports C# and JAVA in its IDE environment.

Supported Operating System: Windows, Linux, Mac OS X, and other Unix-like system.

Hardware Requirement: 512 MB RAM and 400 MB space in hard disk.

The above mentioned modules of simulation and the relationships of communication links are stockpiled in the format of plaintext which goes by the name Network Description or NED files. This can be represented by a graphical format. OMNET++ provides a wide-ranging of GUI support. OMNET++ possesses the capability to handle any larger network size simulations in the most effective manner there is. As a result it can run in various operating systems namely LINUX, UNIX, WINDOWS link systems and MAC-OS. The simulation tool of OMNET++ contains a variety of file formats for the sake of writing C++ code. The type these file formats are as follows:

- Header files (.h)
- Initialization files (.ini)
- CC file

OMNET++ makes use of the INET framework. This specific type of framework is specially used for the sake of modelling spatial relation among mobile nodes. This INET framework is strong enough to be run on a regular basis. INET framework is an extensive branch as it supports all the types of protocol namely IP, IPv4, IPv6, TCP, UDP, SCTP and other protocol implementations. Various types compounds makes up the OMNET++ they are as follows,

- Compiler of network editor (NED)
- Simulation kernel library
- Topology description language
- Compound line interface
- Model documentation tool
- Graphical Output vector Plotting Tool
- Utilities
- Graphical User Interface (GUI)
- NED Files

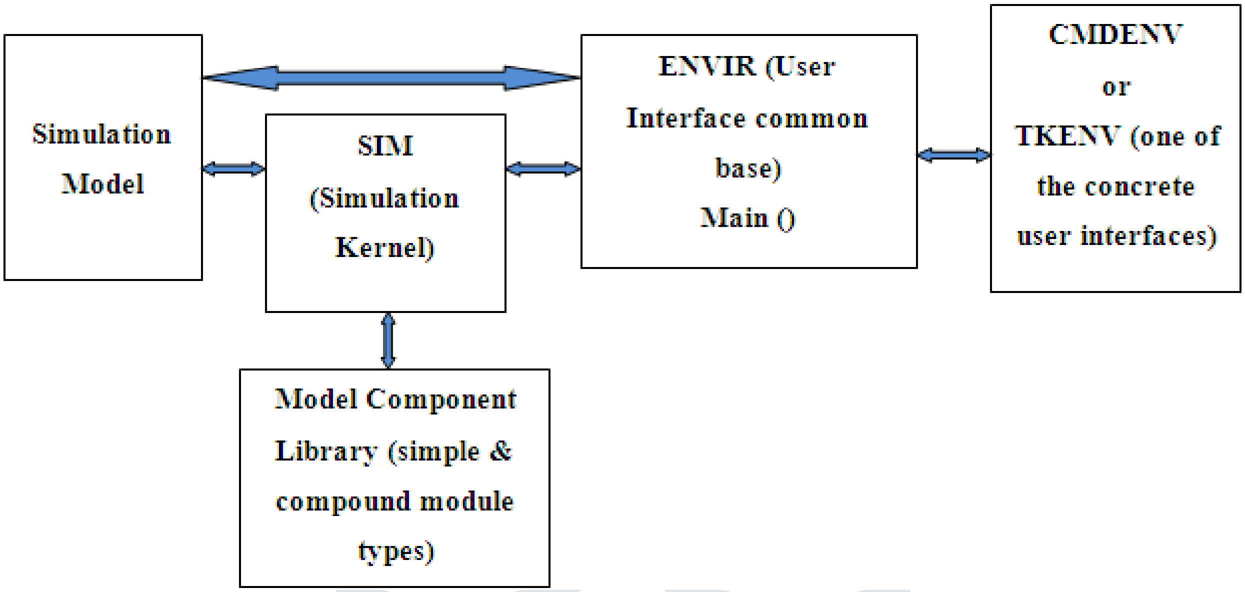
Simulation Model
SIM (Simulation Kernel)
ENVIR (User Interface common base) Main ()
CMDENV or TKENV (one of the concrete user interfaces)
Model Component Library (simple & compound module types)

Fig. 1.3: Simulation Architecture of OMNET++

OMNET++ is a type of network simulator that contains a multiple number of component modules as shown in Fig. 1.3.(Zhao & Braun, 2011) This network simulator is categorized into two entirely varied types of module which are as follows:

- Simple Module
- Compound Module

The first type of OMNET++ is the simple module which is mainly used for defining the algorithm. In this a specific type of algorithm occurs. This simple module contains an active component of the system which is actually the definition of simple module and it is only this module that defines the behavior which is related to the system. The collection of the simple module is put together to make up a compound module which possesses an atmosphere that allows it to interact with one another. This compound module is what represents the host node. The modules that communicate by means of message among one another consist of a great number of qualities such as timestamps. Simple module is can send messages all the way through gates. These gates are similar to the interfaces of modules such as input and output gates.

Messages that which are sent out by means of the output gate come up on the other side through the input gate. The input and output gates are linked by means of a network with a strong connection. In the case of OMNET++ the modules and interconnection connecting modules are illustrated through OMNET++ topology description language called NED. The most important narrative of NED is that it is a simple module declaration, compound module definition and network definitions. The characteristics that are related to the NED languages consist of: Inheritance, Interfaces, Packages, Inner types, Meta data annotations. When it comes to OMNET++ the NED language is beneficial similar to the benefits of XML language. This similarity is due to the fact that NED language is capable of converting XML representation

without any data loss. OMNET++ makes use of the object library as it helps by making available all very rich yet simple module implementation. Parallel simulation execution abbreviated as PDES support is offered by the OMNET++ network simulator. This specific and special parallel simulation feature is quiet helpful as it helps in the large scale simulations.

Features of OMNET++

- OMNET++ contains various modules which include all the basic individualities of OMNET++, the simulation tool.
- OMNET++ possesses multiple features which makes it quiet mainly a resourceful simulation tool nowadays.
- For the duration of running time these simulators are disappointment in modelling as at a particular time for nodes and links it becomes the most advantageous characteristic that is supported by OMNET++.
- OMNET++ has the capacity to produce any form of hierarchical topology for large scale simulation by making use of Network Description (NED) language.
- OMNET++ contains a numerous pseudorandom amount of generators that are a reasonable features and it is used for bulky size simulations.
- Each and every module of OMNET++ has the ability to produce, interpret and even act in response to the provided message. These messages are called as the events.
- OMNET++ can is supportive to both wireless and mobile based simulations that happen within an OMNET++ environment.
- OMNET++ is exceptionally appropriate for complex protocols and the implementations that come with it.
- It is highly rich-featured and also a very commanding simulation tool. It makes available infrastructure and tools for the sake of writing simulations. One of the essential components of this infrastructure is component architecture for simulation models.

- In order to form compound modules these modules can be connected with each other via gates and combined. The depth of module nesting is not inadequate. Modules communicate by means of message passing, where messages can carry arbitrary data structures.
- OMNET++ simulations can be run by means of multiple user interfaces. Graphical, animating user interfaces are highly useful for demonstration and debugging purposes. Command line user interfaces are the best for batch execution.
- Released with full source code. This is free to use.
- Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic net-works, etc. Are available in it.
- OMNET++ also aids in parallel distributed simulation. The parallel simulation algorithm can be effortlessly extended and/or new ones can be plugged in. Models do not necessitate any special instrumentation to run in parallel – all it matters is configuration.

INET framework

INET framework can be described as nothing but a set of simulation modules. These specific sets of simulation modules are unconstrained beneath the GPL. A GPL is makes up the definition that it can make available a range of modules and protocols that are suitable for OMNET++. Some of the dissimilar types of protocols that are supported by our OMNET++ simulation tools are TCP, UDP, IPv4, IPv6, ARP protocols. Apart from this the OMNET++ also supports MAC protocol suite of IEEE 802.11. The framework of INET has the ability to maintain a range of node groups such as wireless and mobile node (Varga & Hornig, 2008). The communication among these networks must be a derivative from the mobility of the INET framework. OMNET++ makes available an additional one framework for Mobile ad hoc networks which are called as the INET-MANET. The functionalities provided by INET framework are similar to that of INET-MANET. Other than this it also offers a number of other supplementary protocols and components that are extremely helpful when it comes to modeling the wireless communication networks.

Simulation Environment

The OMNET++ simulation tool that is proposed by us is aided by the systems which are self-possessed with multiple devices that are capable of interacting with each others in a simulated environment. OMNET++ is described as a general purpose simulation tool. OMNET++ is most accommodative for the environment of mobile ad hoc networks abbreviated
as MANET.

Merits of OMNET++

- The commanding GUI support of the OMNET+ makes available an effortlessness of tracing and debugging capability that is not available in any other network simulation tools.
- OMNET++ has the capacity to precisely model a variety of hardware.
- OMNET++ is accessible for any educational and research purpose that are done by a number of academic students and research scholars.
- Provides a powerful GUI environment.
- Tracing and debugging are much easier than other simulators.
- Accurately models most hardware and include the modeling of physical phenomena.

Demerits of OMNET++

- OMNET++ simulation tool lacks in offering any colossal diversity of the required network protocols for the sake of implementation.
- Poor analysis and management of typical performance.
- The mobility extension is relatively incomplete.
- The extension for mobility offered by the OMNET++ is quiet incomplete.
- Also the mobility extension is pretty deprived in OMNET++ when compared to others.
- OMNET++ has very poor performance analysis methods.

IV Summary of Conceptual Review of Literature

In past area we have explored distinctive Open Source Network Simulators devices. From the table 1.1, we have reviewed that NS3 and OMNET++ are fit for conveying vast scale reproduction in a productive way. OMNET++ bolsters a high GUI ability however it needs in execution concerning NS3. General execution among the 4 test systems NS3, OMNET++ and GloMoSim are the best decisions as per different authors[(Garg, 2015), (Borboruah & Nandi,
2014),(ur Rehman Khan, Bilal, & Othman, 2012)]. NS2 has rich accumulation of various system modules. All in all, the topic of which simulator to utilize is a troublesome one and the answer is to a great extent subject to the particular use case. In any case, if versatility is the principle concern NS3 and OMNET++ are brilliant decisions.

Table 1.1: Surveys of Simulators

Features	GloMoSim	NS-2	NS-3	OMNET++
Language support	Parsec C	C++/OTcl	C++	C++
GUI Support	Poor	Poor	Good	Good
Time taken to learn	Long	Long	Moderate	Moderate
Time spent in downloading and Installation	Long Time to download.	Moderate Time	Long Time to download & install all necessary patches & supporting software	Very Easy takes very less time to install and easily available
Platform	Linux, Windows	Linus, Unix, Windows	Linus, Unix, Windows	Linux, Unix, Windows, Mac OS
Network Visualization tool	Yes	Yes	Yes	Yes
Create Trace file	Yes	Yes	Yes	Yes
Availability of analysis Tool	Yes	Yes	Yes	Yes
Redesigning & Modification	Possible	Possible	Possible	Possible
Interaction with Real Time system	Possible	NO	Possible	Possible
Fast Simulation Capability	Poor	Moderate	Moderate	Moderate
Design &Implementation Protocols	Support Only wireless Protocols	Support both wired and wireless	Support both wired and wireless	Support both wired and wireless

V Conclusion

This paper focus on available open source network simulation tools. This paper shows network simulator features, languages, environment setup, modules, merits and demerits. This paper performs a theoretical survey of open source network simulators. This paper help in selecting the network simulators i.e. OMNET++ and NS3.It helps in understanding the network simulators in a better way, and this will helps in implementation of MANET. In future, the implementation of MANET routing protocols on Network simulator are studied.

References

[1] Shiva Prakash, J. P. Saini and S. C. Gupta “ Methodologies and Applications of Wireless Mobile Ad-hoc Networks Routing Protocols “IJAIS Volume 1– No.6, February 2012.

[2] Sunil Taneja and Ashwani Kush, “A Survey of Routing Protocols in Mobile Ad Hoc Networks”, International Journal of Innovation, Management and Technology, Vol. 1, No. 3, August 2010.

[3] Neha Singh, Prof. Rajeshwar Lal Dua, Vinita Mathur “Network Simulator NS2-2.35”IJARCSSE Volume 2, Issue 5, May 2012.

[4] Dr.S.S.Dhenakaran , A.Parvathavarthini “An Overview of Routing Protocols in Mobile Ad-Hoc Network” IJARCSSE Volume 3, Issue 2, February 2013.

[5] Dr. Aditya Goel, Ajaii Sharma “Performance Analysis of Mobile Ad-hoc Network Using AODV Protocol” IJCSS, Volume (3): Issue (5) 2009.

[6] Mrs. Saba Siraj, Mr. Ajay Kumar Gupta and Mrs Rinku-Badgujar “Network Simulation Tools Survey” IJARCCE Vol. 1, Issue 4, June 2012.

[7] Longowal, Sangrur “Performance Comparison of DSR and AODV Routing Protocols with Efficient Mobility Model in Mobile Ad-Hoc Network” IJCST Vol. 2, Iss ue 2, June 2011.