

A COMPARISON OF NETWORK SIMULATION TOOLS

¹Kunal Thakur, ²Dr. Aman Kumar Sharma

¹M.Tech. Student, ²Professor
Department of Computer Science,
Himachal Pradesh University, Shimla, India.

Abstract: With the advance in network technology it is very costly and time-consuming to analyze big size networks. Network simulator software tools provide an environment in which the behavior of a real-world network can be intimated. Network simulators can be used to create simulation models for existing protocols, topologies, etc. New network protocols, topologies can also be developed and tested under predefined or specific parameters using network simulators. This paper focuses on the study and comparative analysis of three open source network simulators, namely NS-2, NS-3, OMNET++ by running simulation for Ad -Hoc On-Demand Vector (AODV) routing protocol. Results are obtained for CPU percentage use and memory usage while running the simulation.

Keywords: Network Simulator, Simulation, Protocol, Topology, NS-2, NS-3, OMNET++, AODV, MANET, Routing.

I. INTRODUCTION

Simulation is a technology with the help of which the unreal and real-life entities can be modeled on to the computer and run under certain predefined conditions [1]. The simulation models generated can be studied to access knowledge, measure behavior and functionality. Similarly, network simulators are used to model network traffic between nodes (network devices). The behavior of network entities in a simulation model is obtained by performing mathematical calculations that depend upon the parameters of the model [9]. Most of the network simulators support wired and wireless topologies. The simulation result may be presented by graphs and trace files. The trace files capture all events or activities occurred in simulation [2]. A large number of protocol models is provided by network simulators some of them are WLAN, WAN, UDP, TCP, IPv4, IPv6, MANET, VISNET, WSN. One of the important features of the network simulators is that many different simulation modules can be used together to observe the performance of a new and complex network.

II. NETWORK SIMULATORS

There are a variety of network simulators with different features both open source and commercially available. In this paper, three open source and discrete event simulators are compared so that the best among them in the context of performance (memory usage and CPU utilization) and ease of use could be found.

2.1 Ns-2

Network Simulator-2 (NS-2) is an open source discrete event simulator. It was first released in 1996, its architectural design is based on NS-1 and REAL network simulators [3]. For designing purpose two languages C++ and Object-oriented Tool command language (OoTcl), C++ is used to design internal working or backend whereas OoTcl is a scripting language used to design interface for simulations [15]

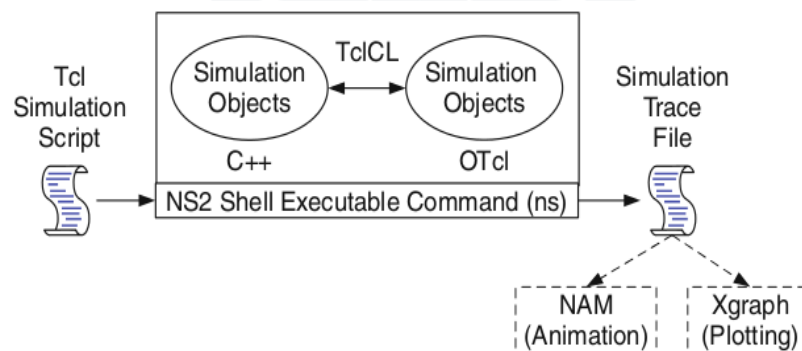


Fig. 2.1: Basic Architecture for NS-2 [5]

Fig 2.1 the basic architecture of NS-2 is depicted wherein TclCL stands for Tool Command Language with classes in NS-2, it acts as a linkage between C++ and OTcl simulation objects. Network Animator (NAM) is a tool which is used to visualize the simulation. Xgraph is the software that can plot simulation results graphically [4].

2.2 Ns-3

Network Simulator-3 (NS-3) is the advanced version of NS-2 simulator. It was released in 2006. NS-3 is available free of cost discreet event simulator, available for research and education purpose. Its basic architecture is based on NS-2, GTNETS, YANS network simulators [3]. Just like NS-2, NS-3 also supports discreet event simulation.

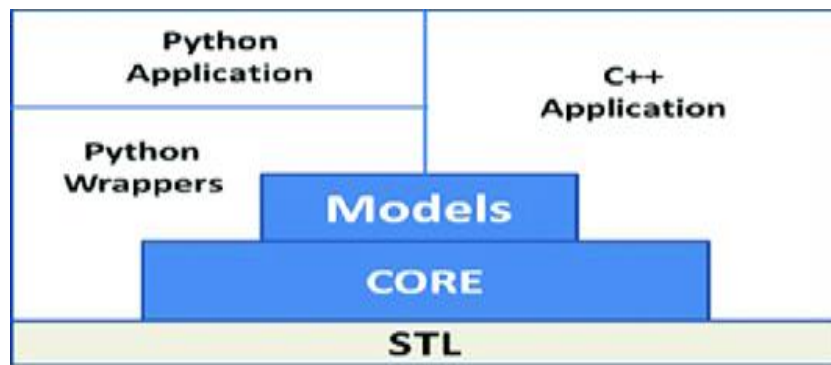


Fig. 2.2: Basic Architecture for NS-3 [7]

To develop simulation the programming language used in NS-3 is C++ or Python. Python is optional which can be used for scripting [6]. Network Animator (NetAnim) is the software package used by NS-3 for visualization purpose of simulations. NetAnim uses XML trace files to run network simulations. PyViz is a Python-based simulation visualizer that can run network simulations for NS-3.

2.3 Omnet++

OMNET++ stands for Optical Micro Network plus plus, it is a component-based, open source, discrete event simulator in which simulations are developed using C++ coding it also provides GUI support [8].

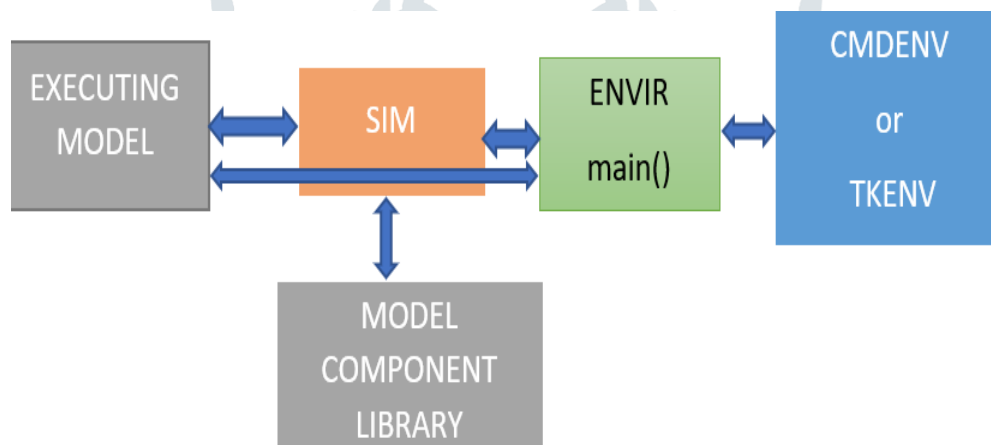


Fig. 2.2: Basic Architecture for OMNET++ [1].

Main components of OMNET++ are: - Simulation kernel library (C++), the NED (Network Definition) language used to create network topologies, Eclipse-based Integrated Development Environment (IDE), interactive simulation runtime GUI (Qtenv), Command-line interface to execute simulation (Cmdenv) [8].

III. GENERAL FEATURES OF NETWORK SIMULATORS

Basic features of the above three network simulators are represented with the help of table 3.1. The features included in the table are: object-oriented modeling which means support of object-oriented libraries, parallelism, and distributed simulation support which includes MPI (Message Passing Interface) libraries in NS3 and MPI/PVM (Parallel Virtual Machine) libraries in OMNET++, Integrated Development Environment (IDE) support means availability of source code editor, automation tool, debugger, etc. within a single framework. Other features mentioned in table 3.1 are GUI support, programming support for modeling simulations, ease of use, simulation visualization tools, operating system support, cost of availability.

Table 3.1: General Features of Network Simulators

Simulators	NS-2	NS-3	OMNET++
Features			
The object-oriented modelling [9]	Present	Present	Present
Integrated development environment	Not Present	Not Present	Present
GUI Support [10]	Limited	Present	Present
Programming support for modelling [11]	C++/OTCL scripting to develop network topology	C++/Scripting optional with Python	C++/NED(Network Definition) language to develop network interface.
Simulation parallelism [10]	Not Present	MPI	MPI/PVM
Simulation visualization tool	NAM Animator	NetAnim Animator Or pyViz.	Runtime GUI (Qtenv)
Documentation support [10]	Good	Excellent	Excellent
Ease of use [10]	Hard	Moderate	Easy
Operating system support	GNU/Linux, FreeBSD, Mac OS X, Windows	GNU/Linux, FreeBSD, Mac OS X, Windows	Windows XP or Later, Linux, Mac OS X
Availability Cost [11]	Free of cost	Free of cost	Free of cost

IV. SIMULATION AND RESULTS

For evaluation of performance comparison, AODV routing protocol was simulated in NS-2, NS-3, OMNET++. The parameters taken for performance comparison are:

1. Memory usage by simulators while running simulation of AODV routing protocol. The metric for measurement is MB (Mega Byte).
2. CPU percentage used by the simulator while running simulation of AODV protocol. CPU percentage is the time of CPU allocated while running simulation.

4.1 Ad Hoc On-Demand Distance Vector (AODV) Routing

AODV is a reactive (on-demand connection-oriented) Mobile Ad-Hoc Network (MANET) protocol, each node contains the routing table and sequence numbers are used to update and maintain routes [13]. There are two basic steps in AODV, one is route discovery using R-REQ packets and other is route maintenance using HELLO packets [14]. The disadvantage of AODV is that it takes time to establish the initial route between nodes, also the intermediate nodes can lead to discrepant routes if the sequence number is not updated time to time [12].

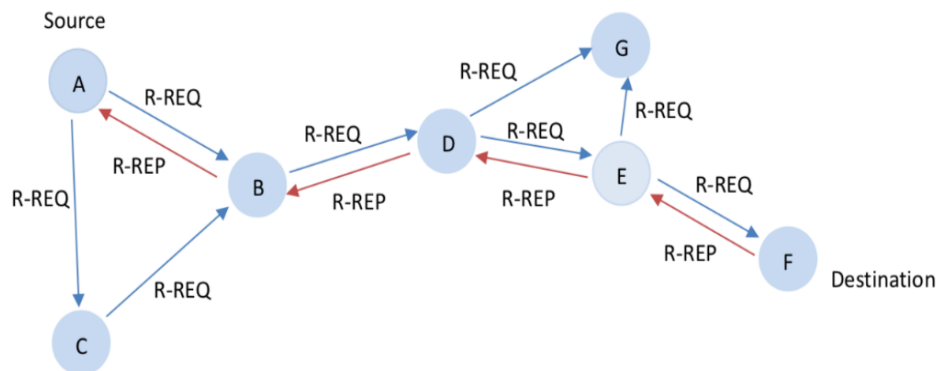


Fig 4.1 AODV Protocol Between Nodes.

In Fig 4.1 A is the source node wants to communicate with node F the destination node. R-REQ are route-request packets, R-REP is route-reply packets. Following steps show how R-REQ and R-REP packets are used to discover the destination node and establish a route with the shortest distance between source and destination.

4.1.1 Steps

1. Firstly source (A) node broadcasts R-REQ packet to its neighbor nodes (C, B) and it further gets transmitted to the next nodes until reaches destination node(F).
2. When destination node (F) receives R-REQ, it sends (unicasts) back the R-REP packet through intermediate nodes with the shortest distance to the source node and a temporary route gets established.
3. After establishing a route, HELLO message is sent by all node to its nearby node to keep the route maintained. If any moving node moves away or out of the established route the above steps are again implemented to set up a new route.

4.2 Simulation Environment

Simulation environment consists of basic parameters and their values for the development of simulation. As simulation starts, the source node starts sending data packets of size 1MB to the destination node. During the simulation, the number of nodes was varied from 150 to 900 and each simulation was run for 200 seconds.

Table 4.1 Simulation tool version

Simulator	Versions
NS-2	2.35
NS-3	3.28
OMNET++	5.4.1

The simulation tools were run on operating system Linux (Ubuntu18.8). Simulator versions are given in Table 4.1.

Table 4.2 Simulation Setup Parameters

Parameters	Values
Simulation time	200s
Mobility of nodes	Present
Packet size	1MB
Number of nodes	150-900
Routing protocol	AODV
Topology	Random wireless
Distance between nodes	250m

Simulation parameters mentioned in table 4.2 includes time for which simulation run, size of the packets transmitted among nodes, type of topology used which is random wireless. The number of nodes taken in the network is 150 to 900.

4.3 Results

The performance comparison graphs are drawn for memory usage (MB) and CPU percentage usage against the number of nodes.

4.3.1 Memory Utilization

AODV protocol was simulated with a simulation time of 200s and the number of nodes is varied from 150 to 900. The readings are taken in the System Monitor tool in Linux (Ubuntu). Fig 4.2 shows a graph for memory utilization by simulators NS-2, NS-3, and OMNET++.

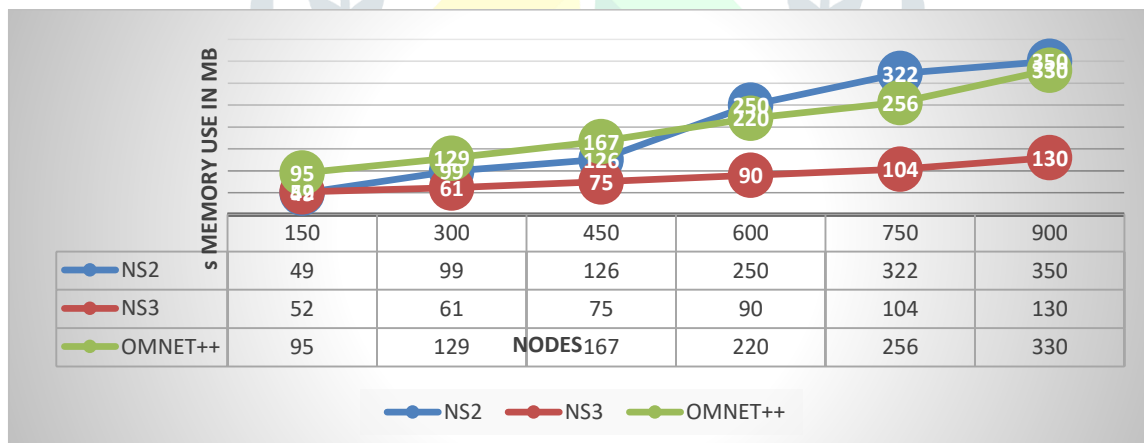


Fig 4.2 Memory Usage Graph

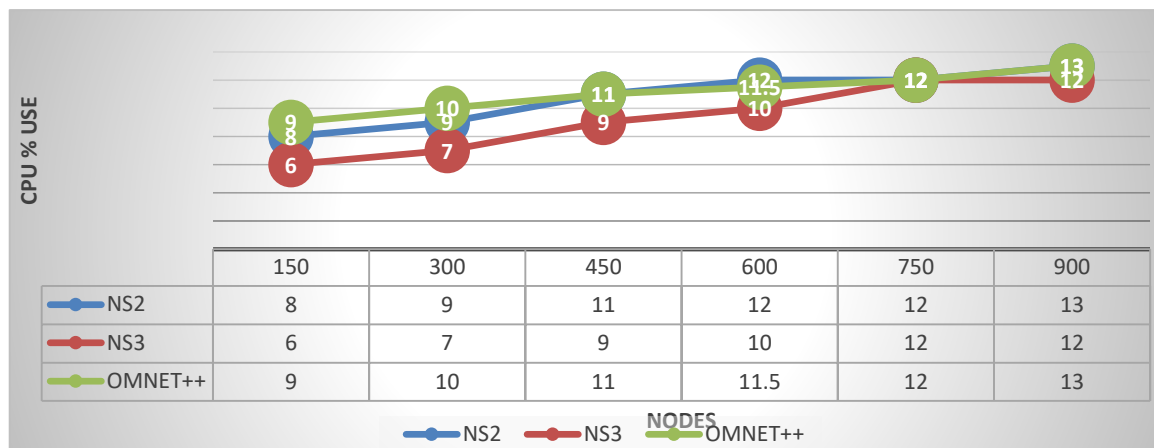
In Fig 4.2, the x-axis represents the number of nodes increasing from 150 to 900 and y-axis represents memory usage in MB (Mega Byte). The memory usage for NS-3 is lesser as compared to another two simulators (NS-2 and OMNET++).

4.3.2 CPU Utilization

CPU utilization by simulators NS-2, NS-3 and OMNET++ while running AODV protocol is measured in terms percentage. Fig 4.3 the x-axis shows the number of nodes increasing from 150 to 900, the y-axis shows the CPU % (percentage) use. The readings are taken in the System Monitor tool in Linux (Ubuntu).

The CPU utilization for three simulators does not have much difference. But for a large number of nodes, the CPU utilization may show a considerable difference. For 900 nodes, NS-3 uses less percentage of CPU as compared to the other two simulators

Fig 4.3 CPU Usage Graph.



V. CONCLUSION

Simulation of AODV routing protocol is performed on the simulators NS-2, NS-3, OMNET++. The comparison parameters taken are the percentage of CPU utilization and memory used in MB. From the simulation results, it was obtained that among all three simulators NS-3 was found utilizing a minimum amount of memory during simulation run and all three simulators with a maximum of 900 nodes for CPU percentage utilization is almost similar. But for a large number of nodes, the CPU utilization may show the difference.

REFERENCES

- [1] Siraj, Saba, A. Gupta, and Rinku Badgular. "Network simulation tools survey." *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 1, Issue 4, pp. 199-206, 2012.
- [2] Christhu raj M.R, Namrata Marium Chacko, John major and Shibin. D, "A Comprehensive Overview on Different Network Simulators" *International Journal of Engineering and Technology*, Vol 5 No 1, pp. 325-332, 2013.
- [3] Rachna Chaudhary, Shweta Sethi, Rita Keshari and Sakshi Goel, "A study of comparison of Network Simulator -3 and Network Simulator -2", *International Journal of Computer Science and Information Technologies*, Vol. 3, issue 1, 2012.
- [4] Suchitha. S Dr.B.G.Prasanthi, "A study of network Simulation tools", *Adarsh Journal of Management Research*, pp- 270-272, 2017.
- [5] https://www.researchgate.net/figure/Basic-Architecture-of-NS-2_fig3_259484464. [Accessed: 9-April-2019 at 14:00hrs].
- [6] <https://www.nsnam.org/docs/tutorial/html/introduction.html#about-ns-3>. [Accessed: 9-April- 2019 at 15:00 hrs].
- [7] https://www.researchgate.net/figure/Architecture-of-the-network-simulator-NS-3_fig4_320671648. [Accessed: 9-April-2019 at 15:02 hrs].
- [8] <https://omnetpp.org/intro/> [Accessed: 10-April- 2019 at 14:00 hrs].
- [9] Xiao Zhou, Hui Tian, "Comparison on network simulation techniques", 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE 2016.
- [10] Mohammed Humayun Kabir, Syful Islam, Md. Javed Hossain, and Sazzad Hossain, "Detail Comparison of Network Simulators", *International Journal of Scientific & Engineering Research*, Vol. 5, Issue 10, 2014.
- [11] Shikha Saxena, P Suraiya Husain, "Network Simulator, Technique of implementing the network on the computer", *International Journal of Scientific Engineering and Applied Science* Vol.1, Issue-9, 2015.
- [12] M.Khaleel Ullah Khan, Dr.K.S.Ramesh, " Performance of Wireless Sensor Network Simulators using Ad-hoc On-Demand Distance Vector (AODV) Protocol", *IOSR Journal of Electronics and Communication Engineering*, Vol. 12, Issue 1, pp. 13-19, 2017.
- [13] Khosrozadeh, Ali, Abolfazle Akbari, Maryam Bagheri, and Neda Beikmahdavi "A new algorithm AODV routing protocol in mobile ADHOC networks". *International Journal of Latest Trends in Computing*. Vol. 2, Issue3, pp. 457-464, 2011.
- [14] Neerja Khatri and Arvind Kumar. "Analysing Performance of AODV in MANET: A SURVEY." *International Journal of Scientific and Engineering Research*, Vol 3, Issue 6, 2012.
- [15] Neha Singh, Prof. Rajeshwar Lal Dua and Vinita Mathur, "Network Simulator NS2-2.35", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, Issue 5, pp. 224-228, 2012.