International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 12

ISSN: 2454-4248
291 – 295

A Detailed Analogy of Network Simulators – NS1, NS2, NS3 and NS4

Ms. Avneet Kaur Saluja
Computer Science Department
ITM Universe Vadodara, India
aveefriend@gmail.com

Ms. Sweta A Dargad
Computer Science Department
ITM Universe Vadodara, India
swetamaheshwari02@gmail.com

Ms. Krupali Mistry
Computer Science Department
ITM Universe Vadodara, India
krupalimistry1993@gmail.com

***Abstract*—** Networking is a field of Computer Science where the researchers are dependent on simulators and simulation as the devices used in networking are very costly and complex. It is not easily possible to establish a computer network in real world easily, also direct installation of network devices and cables is not feasible. A simulator is a low cost mechanism which can be used to deploy a network and implement protocols and test the feasibility of the network. NS aka Network Simulator is one such low cost tool, which is available as open source software to network designers. With time simulators have evolved and now Network simulators can simulate wireless networks and advanced mobile networks. The Network Simulator evolution took place with the methodologies and coding technologies. Medium level language like C++ was used in NS1 and later in NS2 we started using easy modeling language like OTCL and C++. In NS3 we can now do coding with more powerful language Python, it also has support for OTCL and C++. High level and advanced language like P4 is the recent one to be used in NS4. The network simulators are now more powerful and fast, as compared to earlier generations of simulators. This paper talks about these advancements that have taken place in the history of Network Simulators and future scopes of NS.

***Keywords*-** *OTCL, NAM, NETAnim, TCL, NS2, NS3, NS4, P4,*

*****

I. INTRODUCTION

Simulation is a useful tool when one wants to implement new security solution. Increase the performance, estimate the performance. It is used when one considers the time and the resources. Simulation is a component that is used by the network researchers [3]. It is a component which is independent of the hardware and software. This component allows the proper understanding how the elements interact, how the individual elements affect the simulated environment [5]. The simulation model can be used with various variations.
Network simulator cannot be ignored in the research field. It is receiving its popularity due to the open source model. It has an advantage that due to open source every organization or every individual can find errors in it. It is flexible provides development in the new research technology can also be used for future improve it according to the bugs. Due to open source network simulator, it can be further used for future improvement. It provides a number of applications in different kind of protocols NMP, TL1, TFTP, FTP, Telnet and Cisco IOS device. It provides an easy to use GUI based network designer tool. It provides the package of tools that simulates analyze the various events to understand the network. Network simulator is a model by which we can model the behavior of the actual model [3]. Network simulator in research field is a methodology that can be implemented without real world implementation and can be used with various network protocols with different network topologies on this protocol. There are different types of network simulators available which can be classified based on simple to complex simulators. To decide the proper network simulator is a tough a complex task for the researchers. The different network simulator can be compared based on various parameters depending on the complexity, number of nodes, traffic on the nodes, can also be compared on CPU utilization, memory usage and the computation time it shows everything about the protocol that is to be used. There are a number of network simulators for instance NS1, NS2, NS3, OMNET+, SWAN, OPNET, JIST and GLOMOSIM, but we would be discussing about some of the simulators [5]. Thus simulators can evaluate the performance of the network protocols.
In this paper we have discussed about all the Network Simulators that are NS1, NS2, NS3 and NS4. NS4 is a project under design and development. The fascinating feature of NS4 which uses P4 driven network has lured our research towards itself. We have discussed some details, features and limitations of various network simulators. We have included comparison tables to easily identify the network simulator differences.

II. NS1

Network Simulator is the first version of Network simulators in the field of education and research. Network simulator-1 was researched by Lawrence Berkeley National Laboratory in 1995-97 by Steve McCanne, Sally Floyd, Kevin Fall, and other contributors. NS is an event-driven network simulator. By TCL program the network simulator is embedded. The simulator is brought into play by NS interpreter [9]. By new TCL procedure ‘ns’, the communication with the interpreter is done. In network topology, all the sources of traffic and the statistics required are defined in NS1 through ‘ns’ command. To invoke TCL procedures at arbitrary points at the time of simulations is done through NS1 commands.
Two classes of TCL Procedures are used for the simulation i.e. the core NS commands and object commands [11].

A. Features of NS1

- Routes are recomputed immediately if the network topology modifies. Here the routing protocol is not simulated.

 ISSN: 2454-4248
291 – 295

- It is less complicated because all the interactions are done only with the single TcL procedure
- TcL(Tool command Language) is used, so it is flexible for the simulation procedure.
- Once an object is created it can be manipulated many times.

B. Limitations of NS1

- Network simulator is also sometimes called as nightmare software due to its complex nature.
- It is also very time consuming method for the simulation.
- To work on NS1 user needs to learn TCL scripting language to estimate results through trace files.
- No debugging environment is there as C++ is used
- Network simulator is no longer developed or sustained.

III. NS2

NS2 is common and most widely used for research work also it is an open source software that was built in C++ that defines the internal mechanism that is the backend of the simulation objects and that run on Linux platform that uses TCL as a Scripting language [1]. It provides a simulation interface through OTCL that sets up simulation by accumulating and constructing the objects as well as scheduling different events. By Tcl c++ and Otcl are linked together [4]. NS2 is a discrete event simulator got support from DARPA,the VINT (Virtual Inter NetworkTested) project at LBL, Xerox PARC, UCB, and USC/ISI, targeted at networking research .NS2 provides substantial support for simulation of TCP, routing and multicast protocols over wireless as well as wired networks [10] . NS2 in research is also used heavily in the field of Ad-Hoc networking.The following figure displays the architecture of NS2 simulator where it has used even scheduler and OTCL scripts. [7]

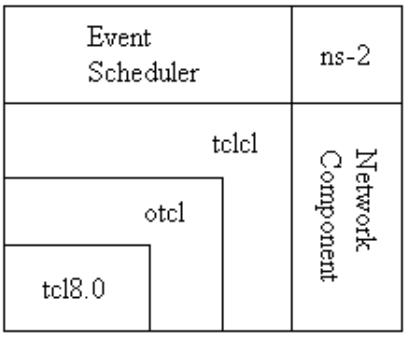

Figure 1. NS2 Architecture

A. Features of NS2

- The cost of NS2 software is Nothing as it is an open source software so anyone can download it for the research purpose, the preferably platform to run NS2 is LINUX and UNIX, but it can also be installed on windows system.
- NS2 is a valid simulator for many networks as it supports multiple protocols such as TCP, UDP, CBR, FTP etc. which is very much in demand which makes this simulator popular in research
- NS2 is user friendly because it is GUI.
- Two graphical representations are used in NS2 one is Nam (Network Animator) and other is Xgraph. Nam displays how the different nodes are communicating and how the exchange of packets has been done. It shows the topological design of network that is simulated; Xgraph is another method that represents the results in form of graph. It saved the result in trace file [16].
- It supports real-time simulation and also it can be used in education purpose.
- More Flexibility
- Availability of External Support

The following figure displays Nam Simulator where 12 wireless nodes are displayed connected via network protocol [17]. We can implement all he wireless and ad-hoc protocols using NS2 and Nam like AODV, DSDV, OLSR using Otcl scripts.

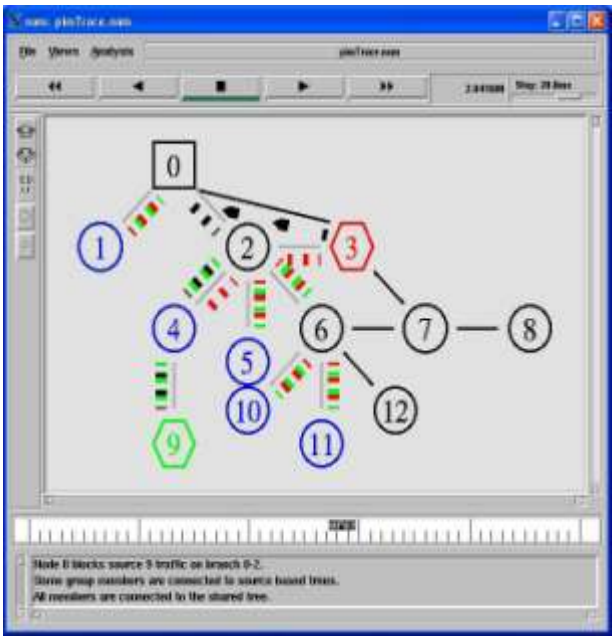

Figure 2. NAM Simulator for NS2

B. Limitations of NS2

- NS2 is widely used in research and much more enhancements in NS2 are to be done and still it is in development stage due to its popularity also it has some limitations.
- It has a complicated structure which is difficult to be reused in the real-life implementation.
- The bugs found in NS2 are sometimes unreliable
- Classless inter-domain routing and sub netting are not provided by the NS2 emulation.
- When the network is too large that is when the number of nodes increases then there is infinite wait and NS2 can't give satisfactory results of such large simulations for NS2 to work properly the number of nodes must be less .So the simulation process is slow when number of nodes increases
- It has Split object model (Otcl and C++) and use of Tcl. There is a large amount of abstraction at the

 ISSN: 2454-4248
291 – 295

network layer and below leads to big discontinuities when transitioning from simulation to experiments.

- Documentation of NS2 is also outdated as many more successor versions has been developed
- One of the main problems in NS2 is that to extract the results we need to parse the trace files and the tracing system in NS2 is not easy to use.

IV. NS3

NS3 simulator is also an open source project that was started in 2006 [4]. It is licensed under GNU GPLv2 licensing reduces the need to rewrite models for simulation. NS3 is also a discrete event network simulator that is also written in C++.It also provides python scripting API but it is optional (instead of Otcl script) .NS3 is a simulator that is written from scratch it is not an extension of NS2 [8].

NS3 is built as a library which may be statically or dynamically linked to a C++ main program does not provide any GUI i.e. graphical user interface but then also easy to handle. NS3 is compatible with Linux, but it can also run on windows by using MinGW. Using sockets NS3 supports emulation as well as simulation. To trace the network traffic standard tools like Wireshark can be used to read the trace files [5].

NS3 uses NetAnim as a GUI for users, where users can create and view the network graphically. NS3 has a support for all the networking technologies like Wired, Wireless, Ad-Hoc, cellular Networks. All these networks can be easily simulated in NS3 using C++ or python scripts. The input can be fed easily and results can be retrieved in GUI as well. Protocols like GPS, GPRS and LTE used in cellular networks can be simulated via NS3 and viewed in NetAnim. The following figure shows eleven nodes of network devices connected by some hybrid topology, where NetAnim provides utility of interface to graphically represent these nodes at different x and y dimensions [11].

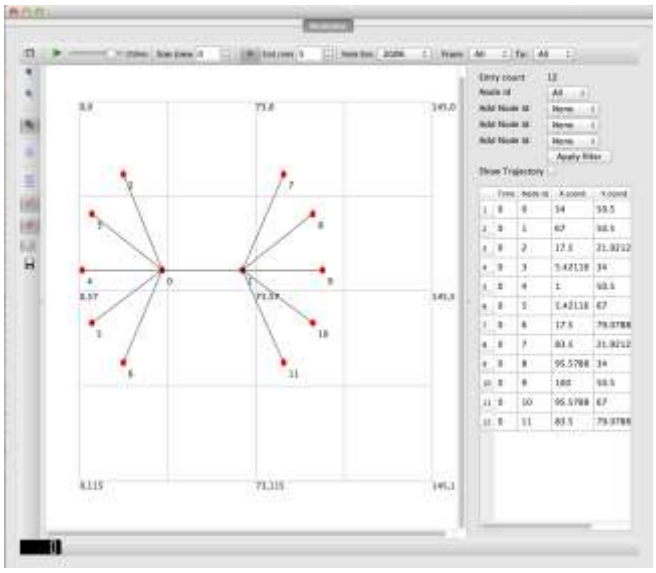

Figure 3. NetAnim for NS3

A. *NS3 Features*

- NS3 was mainly designed to improve the documentation that was weak in NS2, style of coding, improve the modularity and scalability also to enable easier model and packet extension object aggregation capabilities are used. It is mainly written in C++ and instead of Otcl optional python scripting interface is used [5].
- NS3 provides more attention to realism. NS3 simulation system is close to real time environment
- Open-source networking software such as kernel protocol stacks, routing daemons, and packet trace-analyzers, reduces the need to rewrite models and tools for simulation.
- It supports virtualization therefore it is a current topic of research that runs on a wireless simulation network.
- It provides the facility to incorporate more networking software that is open source so that less.
- NS-3 have more new features it can handle multiple nodes and their communication, it better uses the internet protocols with more detailed 802.11modelds
- It is highly reliable even if the complexity of the network is too high.
- It can carry large network simulations effectively.
- It increases the memory performances, also reduces the computational time as it is the most efficient tool for simulation.
- In NS2, two languages are used C++/TCL, so the debugging process is tough and complex in NS2 but in NS3 only one language C++ is required therefore robustness is more.

B. *Limitation of NS3*

- There is limited scope for visualization due to use of python in Network simulator 3.
- NS3 requires powerful community contribution in order to improve it and so that many users can use it.
- For wireless Systems real time animators are required
- It is complex to use.
- NS3 is in development but it is not compatible for work done on NS2.

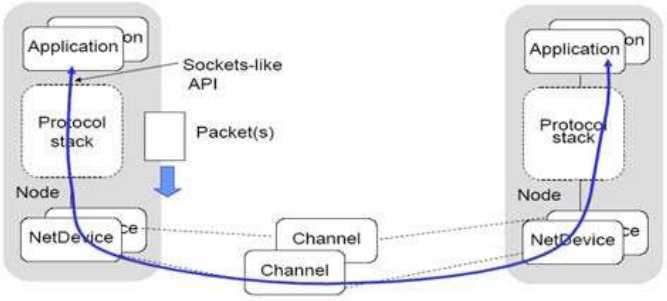

Figure 4. NS3 Basic Architecture [5]

V. NS4

NS4 is a P4 driven network simulator.NS4 has implied P4 into network simulations. NS4 has used P4 which is a target independent language.NS4 is designed to reduce the laborious work which was because of the redundancy of codes in NS-3. NS4 is seamlessly compatible with NS-3 [6].It works as a complete tool-set which provides network simulation. It provides better scalability.

 ISSN: 2454-4248
291 – 295

NS4 design consists of control plane and data plane. The control plane consists of real world, controlling application which is the GUI for user. This application gives input to Network OS. This controller application gives input to the agent model. NS4 allows network simulation for Bluetooth, 802.11a, CSMA, LTE, etc. protocols. The input program is fed into Channel Manager and a packet decapsulator and encapsulator. The channel manager expands the scope of simulate table P4 devices. It converts the input programs into P4 binary later, which creates P4 program. The following figure explains the design overview of NS4 [6][12].

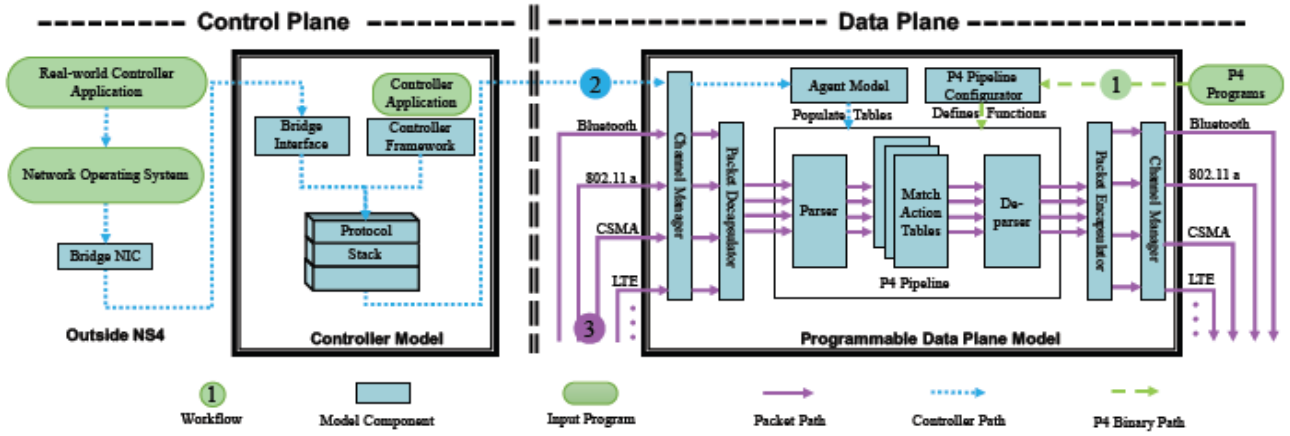

Figure 5. NS4 Design Overview

TABLE I. COMPARISON OF NS1, NS2, NS3 AND NS4

Features	Network Simulators			
	NS1	*NS2*	*NS3*	*NS4*
First Release	1995	1996	2008	Underdevelopment
Based On	----	NS-1 & Real-world Simulator	NS-2, GTNets, YANS	Integrating a P4 behavioral Model
License Type	Free, Open Source	Free, Open Source	Open source, GNU General Public License	----
Language	TCL (Tool Command Language)	C++ and OTCL(Object oriented Tool Command Language)	C++ and Optional Python Bindings	C++
Platform	Unix Systems, Linux, Free BSD SunOS /Solaris Windows 95/98/NT/ 2000/XP	GNU/Linux, FreeBSD, Mac OS X, Windows XP, Windows Vista, Cygwin and Win. 7.	GNU/Linux , FreeBSD, Mac OS X, Windows XP, Windows Vista and Windows 7.	GNU/Linux, FreeBSD, Mac OS X, Windows XP, Windows Vista and Windows 7
GUI Support	No	Limited	Yes	Yes
Document Available	Moderate	Excellent	Excellent	Moderate
Ease of Use	Hard	Hard	Hard	Hard

Features	Network Simulators			
	NS1	*NS2*	*NS3*	*NS4*
Simulation Event Type	Event-driven	Discrete-event	Discrete-event	P4 driven Discrete-event
Available Module	Wired and Wireless networks	Wired/Wireless Network, Ad-Hoc mode, Cannot simulate problems of the bandwidth or the power consumption in Wireless Sensor Network	Wired, Wireless, Ad-Hoc and Wireless Sensor Networks	Wired, Wireless, Ad-Hoc and Wireless Sensor Networks
Scalability	No	Sequential Simulation	Distributed Simulation	Better Scalability
Number of Node support	Upto 400	Upto 3000	----	----
Parallelism	No	No	-----	Yes
Design and Implementation protocols	Wired and wireless	Supports both wired and wireless Simulation of protocols	Supports both wired and wireless Simulation of protocols	Supports both wired and wireless Simulation of protocols
Fast Simulation capabilities	Low	Moderate	Moderate	Ultrahigh
Visualization	----	NAM	NS-3-viz, pyviz, nam, inspect	Under development
Modeling Environment	Command Editor	Command Editor	Command Editor	Command Editor
Scope of Application	Network Protocol	Network Protocol	Network Protocol	Network Programming Protocol

 ISSN: 2454-4248
291 – 295

Features	Network Simulators			
	NS1	***NS2***	***NS3***	***NS4***
Developed By	Lawrence Berkeley National Laboratory (LBNL)	DARPA. But currently development being done with SAMAN and through NSF with CONSER with other researchers including ACIRI	NS-3 Consortium	----
Official Website	https://ee.lbl.gov/NS/	http://www.isi.edu/NSnam/ NS/	http://www.NSnam.org/	https://NS-4.github.io/

VI. Conclusion

The comparative study of all the various NS versions reveals that selecting suitable network simulators, we can decrease the work of network administrator. The use of network simulator will not only ease the work, but also would be cost efficient. Using network simulators we would be able o simulate various network devices and applying various network protocols we can identify which is the best suitable network protocol with the given topology and user requirements has evolved with time and it has provided coherence with the current types of networks like MANET in Adhoc Networks, LTE in Cellular Network. It has also provided support of popular and highly efficient languages for implementation like Object Oriented, C++, OTCL, and Python.

References

[1] Rajankumar, Patel, Patel Nimisha, and Pariza Kamboj. "A comparative study and simulation of AODV MANET routing protocol in NS2 & NS3." Computing for Sustainable Global Development (INDIACom), 2014 International Conference on. IEEE, 2014.

[2] Kumar, AR Ashok, S. V. Rao, and Diganta Goswami. "NS3 simulator for a study of data center networks." Parallel and Distributed Computing (ISPDC), 2013 IEEE 12th International Symposium on. IEEE, 2013.

[3] Gupta, Suraj G., , Mangesh M. Ghonge, Parag D. Thakare, and P. M. Jawandhiya. "Open-source network simulation tools: An overview." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 2.4 (2013): pp-1629.

[4] Bilalb, Sardar M., and Mazliza Othmana. "A Performance Comparison of Network Simulators for Wireless Networks." arXiv preprint arXiv:1307.4129 (2013).

[5] Chaudhary, Rachna, Shweta Sethi, Rita Keshari, and Sakshi Goel. "A study of comparison of Network Simulator-3 and Network Simulator-2." IJCSIT) International Journal of Computer Science and Information Technologies3.1 (2012): 3085-3092.

[6] Fan, Chengze, Jun Bi, Yu Zhou, Cheng Zhang, and Haisu Yu. "NS4: A P4-driven Network Simulator." Proceedings of the SIGCOMM Posters and Demos. ACM, 2017.

[7] X. Zhou and H. Tian, "Comparison on Network Simulation Techniques," 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Guangzhou, 2016, pp. 313-316.

[8] Rampfl, Sebastian. "Network simulation and its limitations." Proceeding zum Seminar Future Internet (FI), Innovative Internet Technologien und Mobile communication (IITM) und Autonomous Communication Networks (ACN). Vol. 57. 2013.

[9] https://ee.lbl.gov/NS/

[10] http://www.isi.edu/NSnam/NS/

[11] http://www.NSnam.org/

[12] https://NS-4.github.io/

[13] Kabir, Mohammed Humayun, Syful Islam, Md Javed Hossain, and Sazzad Hossain. "Detail comparison of network simulators." International Journal of Scientific & Engineering Research 5 (2014): 203-218.

[14] Christhu, M. R., N. Marium, John Major, and D. Shibin. "A comprehensive overview on different network simulators." International journal of engineering and technology (IJET) 5.1 (2013): 325-332.

[15] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford,Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and DavidWalker. “P4: Programming protocol-independent packet processors.” SIGCOMM Comput. Commun. Rev., 44(3):87–95, July 2014.

[16] http://www.vidatek.co.uk/tutorials2/network-simulator-2-NS2/

[17] http://marco.uminho.pt/~joao/pim-NS2/