



A Comprehensive Overview of Simulation Tools for Virtual Network Implementation

Er. Kailash Aseri

Research Scholar, Faculty of Engineering & Technology
Jodhpur National University, Jodhpur,
Rajasthan, India

Abstract— *In the network research area, creating 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. The simulator helps the network developer to check whether the network is able to work in the real time. Thus both the time and cost of testing the functionality of network have been reduced and implementations are made easy.*

Network simulators are used worldwide for education, commercial and Industrial purposes to simulate any part or entire network. There are various network simulators used day by day. Network simulators can be used to produce an estimate result of the network, which sets basis for real time application or execution.

In this paper, we introduce the main features of different simulator and consider their advantages and disadvantages. We hope this survey prove to be a good reference source for those people who feel difficult to select the appropriate network simulators for their research.

Keywords- NS2, OPNET, NETSIM, JSIM

I. INTRODUCTION

We know that TCP/IP is the basic communication protocol of the Internet. It has higher layer which manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message and lower layer which handles the address part of each packet so that it gets to the right destination. Simulation is a very important modern technology. It can be applied to different science, engineering, or other application fields for different purposes. Computer assisted simulation can model proposed and real-life objects or activities on a computer so that it can be studied to see how the system function. Different variables can be used to forecast the behaviour of the system. Computer simulation can be used to assist the display and study in many normal systems. Computer networks are being widely used in all fields of life, Net banking, Military operations, disaster relief and social networking. As the usage of various computer networks grows so does the need to make these networks faster, efficient and secure seriously increase [1][2]. Various research and studies leads to development of various new protocols claiming to be more efficient than the other.

A network is more than just end user devices; it settlement of routers, gateways, servers as hardware components, several routing protocols, security mechanisms, Error correction and detection techniques. When we consider all these components, setting up a network is more a costly deal. Besides cost; the factors that need to be taken into considerations are, efficiency, security, speed etc. therefore when introducing a new techniques into a network is quite a risk. The size of the computer network, makes to almost impossible to conduct experiments directly on it, this could lead to serious damage of the network and loss of data and would require a lot of effort and cost for the damage recovery. Therefore there is a need to have test beds which can accurately imitate the working and interactions of various network components.

To simulate and get nearly accurate results we require powerful simulation tools, these tools are known as network simulators. A network simulator is a piece of software or hardware that forecasts the behaviour of a network, without an actual network being present [3]. Since a network simulator imitates the working of a computer network, it is a well suited tool to test all the newly proposed mechanisms. One can define the various types of devices used in the network and also model the network links, capacity of the links, the traffic load capacity, the type of traffic, behaviour of each device and also various attacks can be simulated to observe the behaviour in the presence of an attacker.

Today there exist various simulators; some are NS2, OPNET and Net sim. These network simulators are mostly GUI driven and can be easily installed on a desktop pc or a laptop. Some network simulators require input scripts, some other require command and still other are fully GUI based where the user can click and drag network components into a simulated topology example of one such Simulator is OPNET.

II. NETWORK SIMULATORS

A. OPNET (Optimized Network Engineering Tools)

OPNET is another network simulator, which again a best User Interface. It can be downloaded in IT Guru website. Test technology design in realistic conditions and evaluate enhancement to standard based protocols, develop new protocols and technologies [15]. OPNET supports four simulation technologies as

1. Discreet Event Simulator
2. Flow Analysis
3. Ace Quick Predict
4. Hybrid Simulation

Discreet event simulators provide models that clearly simulate protocols and simulate messages. [15].It executes in same way as production environment. Flow analysis provides analytical techniques and algorithms to model Stead state network behaviour. [15]ACE Quick Predict uses an analytical technique for studying the impact on application response time of changing network parameters (e.g., bandwidth, latency, utilization packet loss) this technique is supported within the OPNET Application Characterization Environment (ACE). OPNET is a high level event based network level simulation tool. Simulation operates at “packet-level” initially built for the simulation of fixed networks. OPNET contains a huge library of accurate models of commercially available fixed network hardware and protocols. Nowadays the possibilities for wireless network simulations are also very wide .Accurate radio transmission pipeline stage for the modelling of the physical layer (radio interface) .The simulator has a lot of potentiality but here exists typically a lack of the one self. OPNET can be used as a research tool or as a network design/analysis tool (end user).The beginning for the usage is high for the developer but low for the end user.

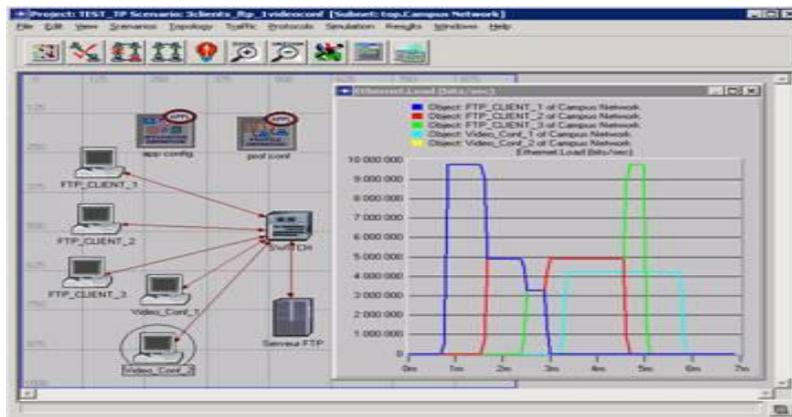


Fig.1

The main programming language in OPNET is C (recent releases support C++ development). The initial configuration (topology setup, parameter setting) is usually achieved using Graphical User Interface (GUI), a set of XML files or through C library calls. Simulation scenarios (e.g., parameter change after some time, topology update, etc.) usually require writing C or C++ code; although in simpler cases one can use special “scenario” parameters (e.g., link fail/restore time)[8]

B. NS-2 (Network Simulator version2)

The Statistics says about 70% education purposes use network simulator “ns2”. It is open source, discreet event simulators for computer networks [4]. Ns2 code comprises of OTCL and C++. OTCL is an interpreter used to execute the commands. NS2 follows two levels of hierarchy namely C++ *Hierarchy* and the *Interpreted OTcL*, which is one to one correspondence [5]. Two languages are linked because to achieve efficiency. C++ Hierarchy allows faster execution and to achieve efficiency. This gives detailed description, definition and operation of protocols, packets and processing time. On the other hand OTcL enables user to define network topology, protocols, applications that user tend to simulate. OTcL can make use compiled C++ Object through an *OTcL linkage* [7]. *OTcL Linkage* creates a matching between OTcL and C++ Objects. Whenever you run a *tcl* file it will produce two outputs or two files namely **trace file** and **namfile**. **Trace file** differs for both wired and wireless scenario. It defines the event discreet simulators. It records the data for each millisecond and gives an output regarding packets send, received, dropped, initial energy of nodes, consumption of energy for transmitting, receiving, idle power, sleep power. It denotes the traffic model, simulation packets, packet size, and mac address. **Nam file** is a visual graphical window which shows the node movements, radio range, and packet transfer including time. **Trace file** can be given input to a new scenario file called NS- VISUAL TRACE ANALYZER [6].

Apart from ns2 tcl code can be generated using two most important network tools like

1. nscript
2. NSG JAR File
3. Mannasim generate
4. TCL 830.exe

The above tools are user graphic tool to generate tcl node. The drag and drop components like Node, link, traffic, applications, parameters are predefined. **NSG Jar file** is a java applet file which has inbuilt java code for nodes, links, application, agent and parameters. **Fig 2** shows the user interface module of NSG Jar window to create tcl codes. As explained has different drop-down boxes with options to create nodes , agents , link and parameters like range , energy , Link layer , antenna can be modified. **nscrip** and **Mannasim** are similar like NSGJAR and drawback for those three tools is that we can't add any new protocols, C++ Code into the files. Ns2 can be executed only in Linux platform and windows needs **Cygwin** platform. To execute ns2 in windows environment we need a special tool called TCL 830.Exe, nam.exe and ns.exe. It enables to run the tcl command in windows command prompt. Again the main drawback of the tool, we can't include C++ Codes. The above tools are not simulators, but just an alternative and easy solution to generate tcl codes. For adding new routing protocols, modifying existing protocol we should stick to ns versions like ns2.29, 2.34 2.35. TCL codes were created by John Ousterhout.

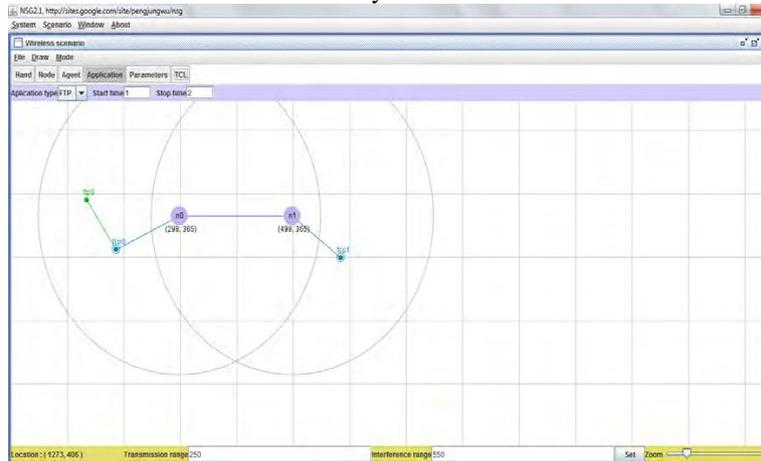


Fig 2. NSG Jar User interface to generate tcl code

C++:

C++ is fast to run but slower to change, making it suitable for detailed protocol implementation

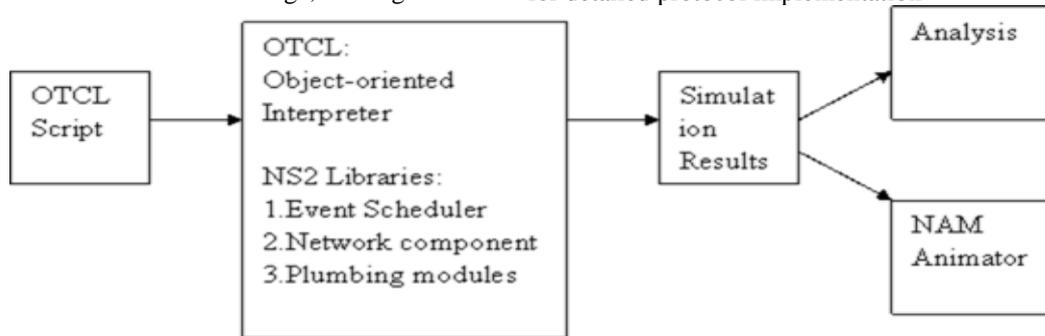


Fig 3. OTcl runs

OTcl runs much slower but can be changed very quickly (and interactively), making it ideal for simulation configuration. Ns provides glue to make objects and variables appear on both languages.

Example:

NS2: The code of ns2 should be written in tcl. And the output file is shown in nam window.

Here is an example:

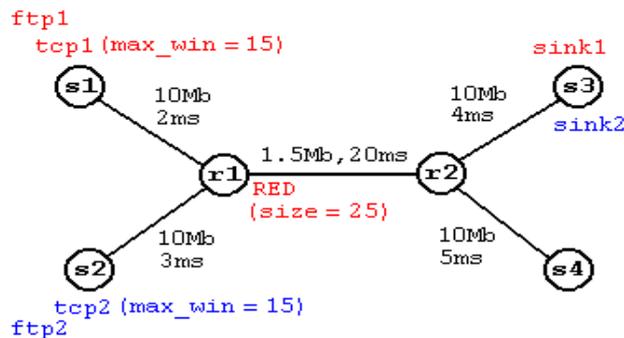
```

set node(s1) [$ns node]
set node(s2) [$ns node] set node(r1) [$ns node] set node(r2) [$ns node]
set node(s3) [$ns node] set node(s4) [$ns node]
$ns duplex-link $node(s1) $node(r1) 10Mb 2ms DropTail
$ns duplex-link $node(s2) $node(r1) 10Mb 3ms DropTail
$ns duplex-link $node(r1) $node(r2) 1.5Mb 20ms RED
$ns queue-limit $node(r1) $node(r2) 25
$ns queue-limit $node(r2) $node(r1) 25
$ns duplex-link $node(s3) $node(r2) 10Mb 4ms DropTail
$ns duplex-link $node(s4) $node(r2) 10Mb 5ms DropTail
$ns duplex-link-op $node(s1) $node(r1) orient right-down
$ns duplex-link-op $node(s2) $node(r1) orient right-up
$ns duplex-link-op $node(r1) $node(r2) orient right
$ns duplex-link-op $node(r1) $node(r2) queuePos 0
  
```

```

$ns duplex-link-op $node(r2) $node(r1) queuePos 0
$ns duplex-link-op $node(s3) $node(r2) orient left-down
$ns duplex-link-op $node(s4) $node(r2) orient left-up
set tcp1 [$ns create-connection TCP/Reno $node(s1) TCPSink
$node(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node(s2) TCPSink
$node(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
$ns at 0.0 "$ftp1 start"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"
$ns run
    
```

Output:



C. NETSIM (Network Based Environment for Modelling and Simulation):

NetSim is a discrete event simulator developed by Tetcos in 1997, in association with Indian Institute of Science. NetSim has also been featured with Computer Networks and Internets V edition by Dr. Douglas Comer, published by Prentice Hall. It has an object-oriented system modelling and simulation (M&S) environment to support simulation and analysis of voice and data communication scenarios for High Frequency Global Communication Systems (HFGCS).

D. JSIM (Java-based simulation):

JSim has been developed by a team at the Distributed Real-time Computing Laboratory (DRCL). The project has been sponsored by the National Science Foundation (NSF), DARPA’s Information Technology Office, Air Force Office of Scientific Research’s Multidisciplinary University Research Initiative, the Ohio State University and the University of Illinois at Urbana-Champaign. J-Sim is free and available with source code.

E. REAL

It is in Computer Science Department Technical Report 88/472, UC Berkeley, 1988. REAL is a simulator for studying the dynamic behaviour of flow and congestion control schemes in packet switch data networks. It provides users with a way of specifying such networks and to observe their behaviour.

III. LANGUAGE USED BY SIMULATORS

- NS2- Use C++, Otel
- OPNET- Use C (C++):
- NetSim- Use Java
- REAL- Use C
- J-Sim- Use Java, Tcl

VI. MERITS AND DEMERITS OF SIMULATORS

- NS2:** NS-2 provides emulation functionalities. NS-2 can be used for parallel and distributed simulation (PDNS).
- OPNET:** Founded in 1991, OPNET Technologies Co., Ltd. (OPNET) specializes in designing, manufacturing and marketing telecommunication transmission products for access networks and inter-office networks.
- NETSIM:** NetSim was used to create a successful national cyber exercise and considers. [11] NetSim has been used to create a fast, functional simulator. NetSim is intended for use within several different defence systems. It will support computer-based collaborative work, such as shared work areas and means of communication.
- JSIM:** JSIM, a Java-based simulation and animation environment supporting Web-Based Simulation, a rapidly emerging area of simulation research and development. [25]

REAL: *REAL* is a network simulator originally intended for studying the dynamic behavior of flow and congestion control schemes in packet-switched data network? *REAL* can be modified to analyze modifications to these protocols or alternate protocols.

V. CONCLUSION AND FUTURE SCOPE

The other network simulators we missing to explain are GosIP, NetViz, NIST, *REAL*, NEST, PTOLEMY with TCP Simulation. The various network simulators are explained. Various network simulators enable to simulate the network but only very few simulators are used by researchers. Our work stated many simulators apart from the traditional network which will explain network researchers. The future work may include explaining new simulators which we stated above.

REFERENCES

- [1] Kotilainen N., Vapa M., Weber M., Töyrylä J. and Vuori J., "P2PDisCo Java Distributed Computing for Workstations Using Cheddar Peer-to-Peer Middleware", *Proceedings of the 19th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2005)*, Denver, Colorado, USA, 2005.
- [2] Lv Q., Cao P., Cohen E., Li K. and Shenker S., Search and Replication in Unstructured Peer-to-Peer Networks, *Proceedings of the 16th International Conference on Supercomputing*, ACM Press, 2002, 84-95.
- [3] Ting N. and Deters R., "3LS - A Peer-to-Peer Network Simulator", *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P 2003)*, IEEE Press, 2003, 212-213.
- [4] Neal Charbonneau nealc.com Using Tcl and ns2 01/2010 Tcl and ns2: Tutorial
- [5] NS Simulation for Beginners, Lecture notes, 2003 -2004 by Eitan Altman ,Tania Jimenez
- [6] Fernando Rocha "Ns2 Visual Trace Analyzer Manual of ns2- Visual Trace Analyzer 0.272"
- [7] The *ns* Manual (formerly *ns* Notes and Documentation)1 The VINT Project "A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. Kevin
- [8] Marek Małowidzki, "Network Simulator: A Developer's perspective", Proc. of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), 2004, page no 73-83.
- [9] Roman Dunaytsev Department of *Communications Engineering Tampere University of Technology*: Network Simulators OPNET Overview and Examples
- [10] Sam Joseph Ph.D., NeuroGrid, <http://www.neurogrid.net/publications/publications.html>
- [11] en.wikipedia.org/wiki/NetSim
- [12] A Survey of Network Simulation Tools: Current Status and Future Development
- [13] Mrs. Saba Siraj & Other "Network Simulation Tools Survey" *IJARCCCE* Vol. 1, Issue 4, June 2012