A Review on Network Simulator & its Installation

Himanshu Trivedi[1], Manoj Mali[2]

[1]*M.Tech Scholar, GITS, Udaipur*

[2]*Asst. Prof., Shri U.S.B. College of Engg & Mgmt. Aburoad*

Abstract: **NS2 is an open-source event-driven simulator designed specifically for research in computer communication networks.ns (network simulator) denotes series of discrete event network simulators, like ns-1, ns-2 and ns-3. Its first version is named as ns-1. developed at VJ,GEEKLIME, Madurai (LBNL) in the 1995-97 The code was written in c++ with tcl based scripting scenario.1996-97, ns 2 was introduced based on a refactoring by Steve McCanne.tcl is replaced by OTcl object-oriented dialect Tcl. C++ simulation objects are linked to shadow objects in OTclcan be used to simulate wireless sensor network. Here in this paper we will be taking into account an overview of ns2.**

Keywords: NS2, C++, Tcl, WSN.

I. INTRODUCTION

NS2 is widely used tool for simulate the network for wireless. It is a part of a software that predicts the performance of a network with its physically existence.
A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions such a temperature, vibration. Sensor mode can be considerd as small computers they usually consists of a processing units. Automation Energy is the scarcest resource of WSN nodes, and it determines the lifetime of WSNs. Algorithms and protocols are required to address issues such as lifetime maximization, robustness and fault tolerance and self configuration. Network simulation is a technique where a program models the behavior of a wired or wireless network by calculat ing the interaction between the different network entit ies (hosts/routers, data links, packets etc). The behavior of the network and the various applications and services it supports can then be observed in a test lab, various attributes can be modified in a controlled manner to assess how the network would behave under different conditions. Basically a network simulator is a piece of software or hardware that predicts the behavior of the network, without an actual network presence.

II. BRIEF OVERVIEW AND ARCHITECTURE OF NS2

NS, the free open-source network simulator, is the de-facto standard for research over a wide variety of networking areas. ns version 2 is widely used across both academia and industry as a way of designing, testing and evaluating new and existing protocols and architectures, and has also proven a very useful tool for teaching purposes.

Simulator targeted at networking research. It provides support for simulation of TCP, routing, and multicast protocols over all networks wireless.NS2 can be employed in most UNIX systems and windows (XP, VESTA and 7), and in this paper windows XP is used. Most procedure processes of the NS2 code are written in C++. It uses TCL as its scripting language, Otcl adds object orientation to TCL.NS (version 2) is an object oriented. Presently, ns-2 consists of over 300,000 lines of source code, and there is probably a comparable amount of contributed code that is not integrated directly into the main distribution.

A. Languages Used: languages which are used is as follows

1) Source Code C++: In ns2 Detailed protocol simulations is required so c++ is the first choice for Source Coce. More important it is a system programming language. It provides some features which are handy in simulation like byte manipulation, packet processing, algorithm implementation
2) Scripting language Tcl:Tool command language is used for Scripting of slightly varying parameters or configurations. It has a feature of quickly exploring a number of scenarios. iteration time (change the model and re -run) is more important in scripting.
3) Scalability:
Per-packet processing must be fast. Separating control and packet handling.

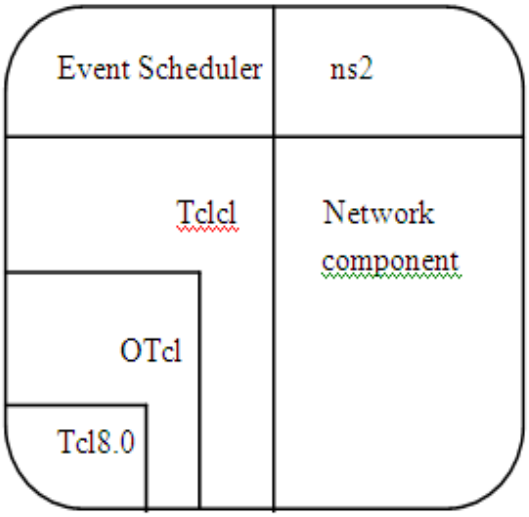

Fig: NS 2 architecture

Figure shows the general architecture of NS. designing and running simulation in the TCL using the simulator object in the OTcl library. The event schedulers and the

most of the network components are implemented in the C++ and available to OTcl through an OTcl linkage that is implemented using tclcl.

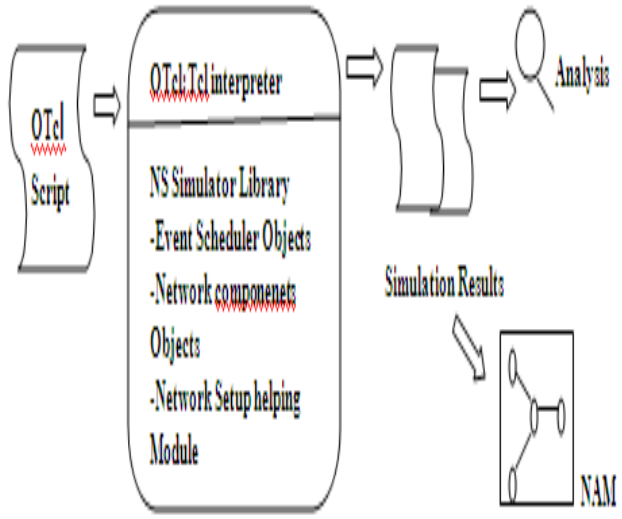

Figure 2 Simplified User"s View of NS-2

Above figure shows Simplified user"s view NS is object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup module libraries. To use NS we program in OTcl script language.

An OTcl script will do the following.

1) Initiates an event scheduler.
2) Sets up the network topology using the network objects.

III. INSTALLATION

In following steps we will discuss about the installation of NS2-2.35 open source operating system like ubuntu.

1) Download
2) Unzip or untar it to any folder that contains it using the command from GUI terminal
   $tar-xzvf ns-allinone-2.35.tar.gz
   After unrar it will create ns-allinone-2.35 folder go into the folder by using command
3) $ cd ns-allinone-2.35 Run this command
   $sudo apt-get install build-essential autoconf automake libxmu-dev
4) Run this command to start ns-2 installation wait until installation is over
   $./install
5) Once installed the PATH information will have to be provided. Copy the PATH and LD_LIBRARY__PATH variable to .bashrc
6) Input the path information in .bashrc file like this
   export PATH=$PATH:<Place your path here> export LD_LIBRARY_PATH=$LD_LIBRARY PATH:<place the LD_LIBRARY_PATHS> here.
7) Once done, save the file and close
   Execute the command and run. /validate
8) Restart the system open the terminal and type ns
9) If % sign appears it means ns-2 is installed in system.

IV. SIMULATON WORKFLOW

The general process of creating a simulation can be divided into several steps:

a) *Topology definition*: to ease the creation of basic facilities and define their interrelationships, ns-3 has a system of containers and helpers that facilitates this process.
b) *Model development*: models are added to simulation (for example, UDP, IPv4, point-to-point devices and links, applications); most of the time this is done using helpers.
c) *Node and link configuration*: models set their default values (for example, the size of packets sent by an application or MTU of a point-to-point link); most of the time this is done using the attribute system.
d) *Execution*: simulation facilities generate events, data requested by the user is logged.
e) *Performance analysis*: after the simulation is finished and data is available as a time-stamped event trace. This data can then be statistically analysed with tools like R to draw conclusions.
f) *Graphical Visualization*: raw or processed data collected in a simulation can be graphed using tools like Gnuplot, matplotlib or XGRAPH.

V. CONCLUSION

Some time NS 2 is criticized just because its modelling is a very complex and time-consuming task, it has no GUI and one needs to learn scripting language, queuing theory and modelling techniques. But moreover involves addressing a wide range of issues steaming from limited energy reserves, computation power, communication capabilities and self managing sensor nodes.It is a flexible tool for network engineers to vestigate how various protocols perform with different topologies and configurations.

NS provides support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

REFERENCES

[1] http://www.ijarcsse.com/docs/papers/May2012/Volum2_issue5/V2I500463.pdf.
[2] http://www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial1.pdf
[3] http://www.wns2.org/
[4] http://www.nsnam.org/overview/publications/
[5] Introduction to network Simulator NS2, Teerawat Issariyakul, Ekram Hossain, Springer, ISBN: 978-0-387-71759-3 e-ISBN: 978-0-387-71760-9