



## **EFFECTIVE LEARNING WITH USAGE OF SIMULATORS – A CASE OF NCTUNS SIMULATOR IN COMPUTER NETWORKS**

**Subrahmanya Bhat\* & Dr. K. R. Kamath\*\***

\* Department of Computer Application, Srinivas Institute of Management Studies,  
Mangalore, Karnataka

\*\* Professor, Department of Computer Science, Srinivas Institute of Technology,  
Mangalore, Karnataka

### **Abstract:**

*The outcome of any good curriculum design should be better learning to student community. In this regard, some of the science subjects need to be taught with the help of simulator so that students can feel “hands on” learning instead of traditional way. In Computer Science which is one of the predominant branches under science stream, Computer Network is one such subject which makes the tutor job tough in understanding the concepts. In this regard we used a network simulator called NCTUns, as add on with our course. This network simulation is an important tool in developing, testing and evaluating network protocols. Simulation can be used without the target physical hardware, making it economical and practical for almost any scale of network topology and setup. It is possible to simulate a link of any bandwidth and delay, even if such a link is currently impossible in the real world. With simulation, it is possible to set each simulated node to use any desired software. Results are also easier to obtain and analyze, because extracting information from important points in the simulated network is also done by simply parsing the generated trace files.*

**Index Terms:** NCTUns & Simulator

### **Introduction:**

Curriculum design is one such process which needs a frequent change and also a backtracking at some point to know the learning outcome of student community. Always that Curriculum which results with better learning to student is always stands first. In making such Curriculum, one needs to incorporate new technologies like using Audio Visual tools, Projectors, Case study analysis, Industry visit so on. We have adopted usage of a Simulator Software – NCTUns 2.0; where in students can get some hands on learning exercise in the subject Computer Network in our MCA course. Network simulators implemented in software are valuable tools for students and also researchers to develop, test, and diagnose network protocols. Simulation is economical because it can carry out experiments without the actual hardware. It is flexible because it can, for example, simulate a link with any bandwidth and propagation delay. Simulation results are easier to analyze than experimental results because important information at critical points can be easily logged to help students diagnose network protocols.

NCTUns is open source, high quality, and supports many types of networks. The NCTUns is a high-fidelity and extensible network simulator and emulator capable of simulating various protocols used in both wired and wireless IP networks. Its core technology is based on the novel kernel re-entering methodology invented by Prof. S.Y. Wang from Harvard University. Due to this novel methodology, NCTUns provides many unique advantages that cannot be easily achieved by traditional network simulators such as ns-2. Useful features of NCTUns network simulator are listed as follows:

- ✓ It can be used as an emulator. An external host in the real world can exchange packets (e.g., set up a TCP connection) with nodes (e.g., host, router, or mobile

station) in a network simulated by NCTUns. Two external hosts in the real world can also exchange their packets via a network simulated by NCTUns. This feature is very useful as the function and performance of real-world devices can be tested under various simulated network conditions.

- ✓ It directly uses the real-life Linux's TCP/IP protocol stack to generate high-fidelity simulation results. By using a novel kernel re-entering simulation methodology, a real-life UNIX (e.g., Linux) kernel's protocol stack can be directly used to generate high-fidelity simulation results.
- ✓ It can use any real-life existing or to-be-developed UNIX application program as a traffic generator program without any modification. Any real-life program can be run on a simulated network to generate network traffic. This enables a researcher to test the functionality and performance of a distributed application or system under various network conditions. Another important advantage of this feature is that application programs developed during simulation studies can be directly moved to and used on real-world UNIX machines after simulation studies are finished. This eliminates the time and effort required to port a simulation prototype to a real-world implementation if traditional network simulators are used.
- ✓ It can use any real-life UNIX network configuration and monitoring tools. For example, the UNIX route, ifconfig, netstat, tcpdump, traceroute commands can be run on a simulated network to configure or monitor the simulated network.
- ✓ In NCTUns, the setup and usage of a simulated network and application programs are exactly the same as those used in real-world IP networks. For example, each layer-3 interface has an IP address assigned to it and application programs directly use these IP addresses to communicate with each other. For this reason, any person who is familiar with real-world IP networks can easily learn and operate NCTUns in a few minutes. For the same reason, NCTUns can be used as an educational tool to teach students how to configure and operate a real-world network.
- ✓ It can simulate fixed Internet, Wireless LANs, mobile ad hoc (sensor) networks, GPRS networks, and optical networks. A wired network is composed of fixed nodes and point-to-point links. Traditional circuit-switching optical networks and more advanced Optical Burst Switching (OBS) networks are also supported. A wireless networks is composed of IEEE 802.11 (b) mobile nodes and access points (both the ad-hoc mode and infra-structure mode are supported). GPRS cellular networks are also supported.
- ✓ It can simulate various networking devices. For example, Ethernet hubs, switches, routers, hosts, IEEE 802.11 (b) wireless stations and access points, WAN (for purposely delaying/dropping/reordering packets), Wall (wireless signal obstacle), GPRS base station, GPRS phone, GPRS GGSN, GPRS SGSN, optical circuit switch, optical burst switch, QoS DiffServ interior and boundary routers, etc.
- ✓ It can simulate various protocols. For example, IEEE 802.3 CSMA/CD MAC, IEEE 802.11 (b) CSMA/CA MAC, learning bridge protocol, spanning tree protocol, IP, Mobile IP, Diffserv (QoS), RIP, OSPF, UDP, TCP, RTP/RTCP/SDP, HTTP, FTP, Telnet, etc.
- ✓ It provides a highly integrated and professional GUI environment. This GUI can help a user (1) draw network topologies, (2) configure the protocol modules used inside a node, (3) specify the moving paths of mobile nodes, (4) plot

network performance graphs, (5) playing back the animation of a logged packet transfer trace, etc. All these operations can be easily and intuitively done with the GUI.

### **NCTUns 1.0 Simulator:**

The NCTUns 1.0 network simulator is a successor to the Harvard network simulator. The NCTUns 1.0 is a high fidelity and extensible discrete event network simulator capable of simulating various protocols used in both wired and wireless IP networks. Its core technology is based on a kernel reentering simulation methodology. In NCTUns 1.0, real-world TCP/IP protocol stacks are directly used to generate accurate simulation results, and all real-world application programs can directly run on any network simulated by this tool. The NCTUns 1.0. is equipped with a GUI environment to help a user to quickly (1) specify network topologies, (2) edit protocol parameters, (3) control the execution of simulations, (4) plot logged performance curves, and (5) play back logged packet transfer animations. It uses a distributed architecture to support remote and concurrent simulations on multiple machines. A user can just download the GUI program, use it to specify his (her) simulation job, and then submit the job to a remote simulation server for execution. When the job is finished, the results will be automatically transferred back to the GUI program for further analysis. NCTUns' simulation engine uses open system architecture to allow a user to easily add protocol modules. Adding a new protocol module into an existing protocol stack of a node or replacing an old one with a new one can be done via the GUI program's node editor.

Due to the kernel reentering simulation methodology, NCTUns 1.0 needs to modify the kernel of the underlying operating system. The NCTUns 1.0 is written in C++. It is open source, free for nonprofit use, and has an active user community. The Web site provides the package, documentation (papers, GUI user manual, and protocol module developer manual), demo videos, mailing lists, and a free simulation center service. NCTUns 1.0 is a simulator, which attempts to make use of a real world network stack for simulation. NCTUns uses a tunnel network interface to use the local machines network stack. Tunnel devices are available on most UNIX machines and allow packets to be written to and read from a special device file. To the kernel, it appears as though packets have arrived from the link layer when data is written to the device file. This means the packet will go through all the normal processing routines: the TCP/IP stack. When a packet is read from the tunnel device, the first packet in the tunnel interfaces output queue is copied to the reading application.

### **Setting up the Environment:**

A user using the NCTUns in single machine mode needs to do the following steps before he/she starts the GUI program:

- ✓ Set up environment variables: Before the user can run up the dispatcher, coordinator, or NCTUns GUI program he/she must set up the NCTUNSHOME environment variable.
- ✓ Start up the dispatcher on terminal 1.
- ✓ Start up the coordinator on terminal 2.
- ✓ Start up the nctunsclient on terminal 3.

After the above steps are followed, the starting screen of NCTUns disappears and the user is presented with the working window.

### **Drawing a Network Topology:**

To draw a new network topology, a user can perform the following steps: Choose Menu->File->Operating Mode-> and make sure that the "Draw Topology" mode is checked. This is the default mode of NCTUns when it is launched. It is only in this mode

that a user can draw a new network topology or change an existing simulation topology. When a user switches the mode to the next mode “Edit Property”, the simulation network topology can no longer be changed.

- ✓ Move the cursor to the toolbar.
- ✓ Left-Click the router icon on the toolbar.
- ✓ Left-Click anywhere in the blank working area to add a router to the current network topology. In the same way we can add switch, hub, WLAN access point, WLAN mobile node, wall (wireless signal obstacle) etc.
- ✓ Left-Click the host icon on the toolbar. Like in step 4, add the required number of hosts to the current topology.
- ✓ To add links between the hosts and the router, left-click the link icon on the toolbar to select it.
- ✓ Left-Click a host and hold the mouse button. Drag this link to the router and then release the mouse left button on top of the router. Now a link between the selected host and the router has been created.
- ✓ Add the other, required number of links in the same way. This completes the creation of a simple network topology.
- ✓ Save this network topology by choosing Menu->File->Save. It is saved with a .tpl extension.
- ✓ Take the snapshot of the above topology.

#### **Editing Node's Properties:**

A network node (device) may have many parameters to set. For example, we may have to set the maximum bandwidth, maximum queue size etc to be used in a network interface. For another example, we may want to specify that some application programs (traffic generators) should be run on some hosts or routers to generate network traffic. Before a user can start editing the properties of a node, he/she should switch the mode from the “Draw Topology” to “Edit Property” mode. In this mode, topology changes can no longer be made. That is, a user cannot add or delete nodes or links at this time. The GUI automatically finds subnets in a network and generates and assigns IP and MAC addresses to layer 3 network interfaces. A user should be aware that if he/she switches the mode back to the “Draw Topology” mode when he/she again switches the mode back to the “Edit Topology” mode, node's IP and MAC addresses will be regenerated and assigned to layer 3 interfaces. Therefore the application programs now may use wrong IP addresses to communicate with their partners.

#### **Running the Simulation:**

When a user finishes editing the properties of network nodes and specifying application programs to be executed during a simulation, he/she can start running the simulation. In order to do so, the user must switch mode explicitly from “Edit Property” to “Run Simulation”. Entering this mode indicates that no more changes can (should) be made to the simulation case, which is reasonable. This simulation is about to be started at this moment; of course, any of its settings should be fixed. Whenever the mode is switched to the “Run Simulation” mode, the many simulation files that collectively describe the simulation case will be exported. These simulation files will be transferred to the (either remote or local) simulation server for it to execute the simulation. These files are stored in the “main File Name.sim” directory, where main Filename is the name of the simulation case chosen in the “Draw Topology” mode.

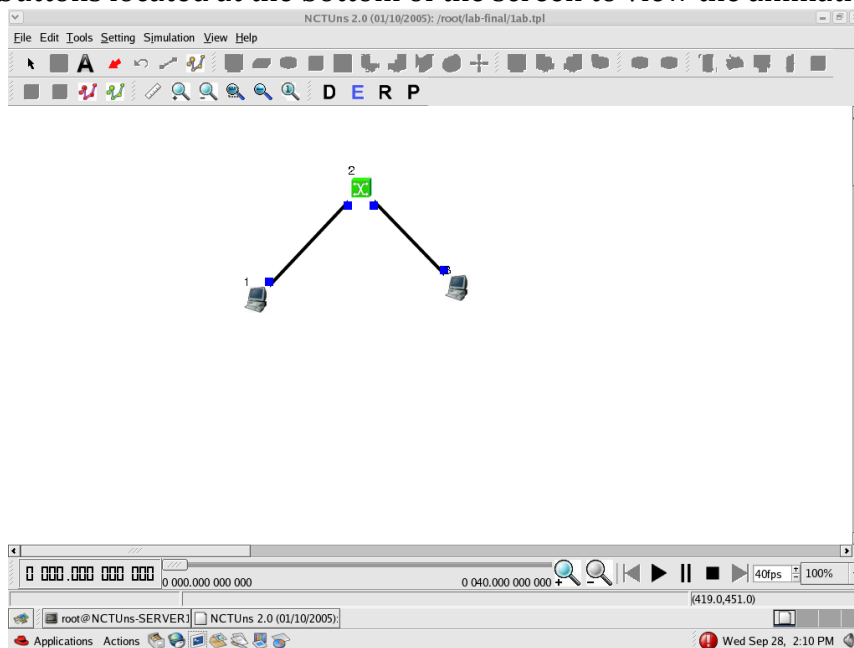
#### **Playing Back the Packet Animation Trace:**

After the simulation is finished, the simulation server will send back the simulation result files to the GUI program after receiving these files, the GUI program

will store these files in the “results directory”. It will then automatically switch to “play back mode”. These files include a packet animation trace file and all performance log files that the user specifies to generate. Outputting these performance log files can be specified by checking some output options in some protocol modules in the node editor. In addition to this, application programs can generate their own data files. The packet animation trace file can be replayed later by the packet animation player. The performance curve of these log files can be plotted by the performance monitor.

**Post Analysis:**

When the user wants to review the simulation results of a simulation case that has been finished before, he /she can run up the GUI program again and then open the case's topology file. The user can switch the mode directly to the “Play Back” mode. The GUI program will then automatically reload the results including the packet animation trace file and performance log file. After the loading process is finished, the user can use the control buttons located at the bottom of the screen to view the animation.



To simulate a three-node point-to-point network with a duplex link between them and Set the queue size and vary the bandwidth and to find the number of packets dropped one can follow these steps. Commands used: `stg -u 1024 100 1.0.1.2` and `rtg -u -w log1` By setting the bandwidth as 10 Mbps on both the links and queue size as 50 we obtain the following results: Out throughput n1-p1= 1177, in throughput n3-p1 = 1177, Collision and drop = 0

By changing bandwidth to 9Mbps in the destination link, we obtain the following results: Out throughput n1-p1 =1177, Input throughput n3-p1 = ~ 0, Collision and drop = 1100

**Conclusion:**

This network simulation is an important tool in developing, testing and evaluating network protocols. Simulation can be used without the target physical hardware, making it economical and practical for almost any scale of network topology and setup. It is possible to simulate a link of any bandwidth and delay, even if such a link is currently impossible in the real world. With simulation, it is possible to set each simulated node to use any desired software. Results are also easier to obtain and analyze, because extracting information from important points in the simulated

network is also done by simply parsing the generated trace files. Using such Simulator students can learn better by having hands on experiment in various topologies and connecting devices with different parameters and hence are in better stage to analyze the performance of any given network.

**References:**

1. [www.nctns.org](http://www.nctns.org)
2. [www.wikipedia.com](http://www.wikipedia.com)
3. [nsl.cs.nctu.edu.tw/NSL/nctuns.html](http://nsl.cs.nctu.edu.tw/NSL/nctuns.html)