Problems faced in Communicate set up of Coordinator with GUI and Dispatcher in NCTUns network simulator

D.Satish Kumar
Department of Computer Science and Engineering
Anna university of Technology Coimbatore
Tel: +91-9843723396 E-mail: satishcoimbatore@yahoo.co.in

Dr.N.Nagarajan
Department of Computer Science and Engineering
Coimbatore Institute of engineering and Technology
E-mail: swekalnag@gmail.com

Abstract
Distributed Emulation can be carried out between real world applications with a simulator. NCTUns-6.0 is a simulator based on kernel re-entering and distributed Emulation simulation methodology, with this methodology a more realistic network model can be simulated for real time applications. Coordinator is a Program that sets up communication between the GUI and Dispatcher in NCTUns-6.0 I, a network simulator running on Fedora-13. We have given in this paper how problems faced in communication set up of Coordinator with GUI and Dispatcher are overcome. This paper focuses on how to overcome the above problems and execute simulations in real time. This paper also deals with KDE, SELINUX, iptables, environment variables and X-server for NCTUns-6.0 simulation engine

Keywords: Coordinator, Dispatcher, nctunsc client, NCTUns-6.0, Fedora-12, SELinux, IPtables

1. Introduction
Network Emulation can be conducted by connection real world devices to a single machine. Distributed emulation is a novel methodology that allows multiple machines to cooperatively conduct an emulation case. By using this methodology, the System resource required to run an emulation case can come from multiple machines (e.g., multiple servers).

NCTUns is a kernel based network simulator that integrates user-level processes, operating system kernel, and the user-level simulation engine into a cooperative network simulation system. To meet the paper limitation limit NCTUns is briefly explained in

The purpose of our research is to find effective and efficient means of communication setup between the the Coordinator and Dispatcher and GUI program. In this paper, we use some simple basic solutions to solve the problems. We further claim that it will improve the quality and performance of Network Emulations. This paper gives solution for all users of NCTUns, but still there are problems which we are focusing in future.

The paper is organized as follows. In section 2 we describe the brief concept of Distribution Emulation Architecture. In section 3 we portray the Coordinator and Dispatcher and GUI concepts. Section 4 gives the details about the Problems faced and solutions, conclusion and future works are discussed in the last section
2. Distribution Emulation Architecture

The Distributed Emulated architecture for NCTUns is shown above, it has a main controller called NCTUns GUI connected to dispatcher. A Dispatcher is a program which connects several emulation machines (coordinator). The main controller manages all the emulated machines.

3. Related Work.

We have referred several papers and web sites to solve all problems occurred between. To meet the paper length limitation, we listed below.

3.1 Definition and Examples

Definition

We first give simple definitions for key terms as they will be used in the rest of this paper:

- **Coordinator**: Coordinator is a program to communicate with GUI and dispatcher, the coordinator is responsible for reporting the status of simulation study.

- **Dispatcher**: Dispatcher program can simultaneously manage and use multiple simulation servers [i.e., emulation machines] to increase the aggregate simulation throughput. It can be run on a separate machine or on a simulation server.

- **GUI**: GUI is a graphics user interface, helps the user to edit a network topology and execute it in an easy way. It uses TCP/IP sockets to communicate with other components.

- **NCTUnsclient**: It is a program in NCTUns that performs the following six main functions: 
  - *Topology Editor*: An environment to specify a network topology.
  - *Edit Property*: An environment to configure which modules to be used inside a node.
  - *Simulation Operation Front-End Interface*: An interface to control the execution of simulations.
  - *Packet Animation Player*: A player to play back logged packet transfer trace.
  - *Performance Monitor*: It is for monitoring and used to plot and display logged performance metrics over time.
  - *Retrieve Results*: An interface to view and control submitted background jobs.
SELinux: SELinux (Security-Enhanced Linux) in Fedora is an implementation of mandatory access control in the Linux kernel using the Linux Security Modules (LSM) framework. It is set to permissive or enforcing mode.

Firewall or IPtables: IPtables is administration tool / command for IPv4 packet filtering and NAT. You need to use the following tools:

---

An real router is connected to two emulation machines. The IP address are shown in figure. The Dispatcher and main controller(GUI) are running on the emulated machine with IP address 192.168.1.1, and we are running the coordinator program in emulated machine with IP address 192.168.1.2.

Steps for communication set up:

Step 1: All the above machines are physically connected to form a network

Step 2: Dispatcher program is set to run on the emulated machine [192.168.1.1]

Step 3: Coordinator Program is set to run on the emulated machine [192.168.1.2], before it is set to run modify the Dispatcher IP parameter in configuration file of coordinator.

Step 4: Run the GUI on the emulated machine [192.168.1.1] and specify the IP address [192.168.1.1] to dispatcher.

Step 5: In GUI edit network topology, and run simulation.

The GUI will split in to two networks as shown I figure and give their configuration files to the corresponding emulated machine.

4 Problem Faced and solutions

4.1 Details about the Problem

In this section, we discuss the Problems that occurred in communication setup of NCTUns-6.0 Network simulator.
We have given Linux version development vs Year of release in the above graph, It is seen that the current version is Linux Kernel 2.6.33 and year of release is 2010
Initially all users will follow the rules, but still the problem persists and they don’t know the basic operation and how to overcome it. They have to avoid by cross checking whether the system accepts the rules or not and also the communication is carried out in specific IP address between the coordinator and Dispatcher.
4.1.1 Problem-1

Choose the correct version of NCTUns that matches the kernel version of Fedora Linux. We have given in table, the latest version and their kernel numbers that should be matched. There are several versions of it currently it is NCTUNs 6.0 [kernel numbers 2.6.28.9 and 2.6.31.6, for Fedora 11 and 12]

```
[root@Test ~]# cat /selinux/enforce
1
[root@Test ~]# echo 0 > /selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
[root@Test ~]# echo 0 >/selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
[root@Test ~]# echo 0 >/selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
[root@Test ~]# echo 0 >/selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
[root@Test ~]# echo 0 >/selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
[root@Test ~]# echo 0 >/selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
[root@Test ~]# echo 0 >/selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
[root@Test ~]# echo 0 >/selinux/enforce
[root@Test ~]# cat /selinux/enforce
0
```

4.1.2 Problem-2

Turning off SELinux temporarily

For the operating system as a whole, there is two kinds of disabling:

**Permissive** - switch the SELinux kernel into a mode where every operation is allowed. Operations that would be denied are allowed and a message is logged identifying that it would be denied. The mechanism that defines labels for files which are being created/changed is still active.

**Disabled** - SELinux is completely switched off in the kernel. This allows all operations to be permitted, and also disables the process which decides what to label files & processes with.

Disabling SELinux temporarily, if the problem you are experiencing is related to your SELinux settings. To turn it off, you will need to become the root users on your system and execute the following command:

```
# echo 0 > /selinux/enforce
```

This temporarily turns off SELinux until it is either re-enabled or the system is rebooted. To turn it back on you simply execute this command:

```
# echo 1 > /selinux/enforce
```
As you can see from these commands what you are doing is setting the file /selinux/enforce to either '1' or '0' to denote 'true' and 'false'.

Disabling SELinux could lead to problems if you want to re-enable it again later. When the system runs with file labelling disable it will create files with no label - which could cause problems if the system is booted into Enforcement mode.

4.1.3 Problem-3

How Do I Disable IPtables?

First login as the root user.

Next enter the following three commands to disable firewall.

```
# service iptables save
# service iptables stop
# chkconfig iptables off
```

If you are using IPv6 firewall, enter:

```
# service ip6tables save
```

Use the following procedure to disable the SELinux using the GUI.

- Open the System menu of Fedora. By using mouse point to Administration and then click Security Level and Firewall to display the Security Level Configuration dialog box.
- Then click the SELinux tab.
- In the SELinux Setting select either Disabled, Enforcing or Permissive, and then click OK.
- If you changed from Enabled to Disabled or vice versa, you need to restart the machine for the change to take effect.
Communication SetUp:

1. Dispatcher Program is set to run:

```bash
[tst@Test ~]$ ls
[tst@Test ~]$ cd /usr/local/nctuns/bin/
[tst@Test bin]$ ls
  coordinator  dispatcher  nctunsclient.bin  nctunsse
[tst@Test bin]$ ./dispatcher
Sorry!! dispatcher needs root's privilege to operate properly.
Please switch to root to run dispatcher
[tst@Test bin]$ su root
Password:
[root@Test bin]# ./dispatcher
Environment "NCTUNSHOME" not set
[root@Test bin]# export NCTUNSHOME=/usr/local/nctuns/
[root@Test bin]# export NCTUNS_TOOLS=/usr/local/nctuns/tools
[root@Test bin]# export NCTUNS_BIN=/usr/local/nctuns/bin
[root@Test bin]# ./dispatcher
ServerSocket listen to port:9810
ServerSocket listen to port:9800
(Active:0| fd:3)   (Active:1| fd:4)
---> New Server
Server accepted: 5   2
(Active:0| fd:3)   (Active:1| fd:4)   (Active:2| fd:5)
[From Server...] register[127.0.0.1]:9830|9840|IDLE
(Regiser Complete!)
<ServerList>:127.0.0.1:FREE
(Active:0| fd:3)   (Active:1| fd:4)   (Active:2|)
```

2. Coordinator Program is set to run
3. nctunsclient program is set to run

5. Conclusions and Future Works

Thus Distributed Emulation can be carried out between real world applications with a simulator. Thus NCTUns-6.0 is a simulator based on kernel re-entering and distributed Emulation simulation methodology, with this methodology a more realistic network model can be simulated for real time applications. Here Coordinator is a Program that sets up communication between the GUI and Dispatcher in NCTUns-6.0 I, a network simulator running on Fedora-13. This paper presents problems faced in communication set up of Coordinator with GUI and Dispatcher. This paper focuses on how to overcome the above problems and execute simulations in real time. This paper also deals with KDE, SELINUX, iptables, environment variables and X-server for NCTUns-6.0 simulation engine
References


Mirco Musolesi, and Cecilia Mascolo, “A Community Based Mobility Model for Ad Hoc Network Research”, REALMAN’06, May 26, 2006, Florence, Italy.


http://pcl.cs.ucla.edu/projects/glomosim

Authors

D.Satish Kumar received his B.E., degree in Electrical and Electronics Engineering from Bharathiar University and M.E. degree in Computer Science and Engineering from Manonmaniam Sundharnar University respectively. He is currently working towards the PhD degree in Computer Science and Engineering in Anna University of Technology Coimbatore, Tamil Nadu INDIA. His research interests accumulate in the area of IEEE 802.16j Multi hop Mobile networks, Routing, WIMAX networks, Network management in Wireless Networks.

Email: satishcoimbatore@yahoo.co.in

Dr.Nagarajan received his B.Tech and M.E. degrees in Electronics Engineering at M.I.T Chennai. He received his PhD in Faculty of Information and communication engineering from Anna University, Chennai. He is currently working as Principal, Coimbatore Institute of Engineering and Technology, Coimbatore. He is member of board of study of faculty of information Technology at Anna University of technology, Coimbatore. His specialization includes optical, wireless Adhoc and sensor networks. He is guiding assorted research scholars in optical networks and wireless networks.

Email: swekalnag@rediffmail.com
This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE’s homepage:
http://www.iiste.org

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. **Prospective authors of IISTE journals can find the submission instruction on the following page:**
http://www.iiste.org/Journals/

The IISTE editorial team promises to the review and publish all the qualified submissions in a fast manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

**IISTE Knowledge Sharing Partners**

EBSCO, Index Copernicus, Ulrich’s Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar