Transferring a Video via ns-3 using the ORBIT (Open-Access Research Testbed for Next-Generation Wireless Networks) Testbed Portal

Submitted by:  
Ankit Deepak 13CO206  
Shravya KS 13CO243  
Chirag Jamadagni 13CO117  
Meenakshy B 13CO231  
Amita Kamath 13CO203

Submitted to:  
Dr. Mohit P Tahiliani  
Assistant Professor, CSE NITK

December 2015
Introduction

An emulator is a hardware/software that enables one computer system to behave like another. The computer system attempting to emulate is called the host and the system being imitated is called the guest. ns-3 is an open source simulator for research and development in the field of computer networks. In order to verify ns-3 simulations in a near real life scenario, we need to integrate ns-3 with testbeds. Currently, Emulab, ORBIT, and PlanetLab are the testbeds predominantly used by researchers. ORBIT is the abbreviation for Open Access Research Testbed for next generation wireless networks. The goal of orbit is to support the evaluation of different networking protocols in the real world.

ORBIT or Open-Access Research Testbed for Next-Generation Wireless Networks was developed and is maintained by Rutgers University. ORBIT is a two-tier wireless network emulator/field trial designed to achieve reproducible experimentation, while also supporting realistic evaluation of protocols and applications. The RADIO GRID TESTBED uses a novel approach based on a 20x20 two-dimensional grid of programmable radio nodes which can be interconnected into specified topologies with reproducible wireless channel models. Once the protocol or application have been validated on the radio grid emulator, users can migrate their experiments to the OUTDOOR ORBIT network which provides a configurable mix of both high-speed cellular (WiMAX, LTE) and 802.11 wireless access in a real-world setting. The ORBIT testbed also includes a number of SANDBOX networks used for debugging and controlled experimentation on specific aspects.

The first aim of this experiment is to understand the working and successfully integrate the ns-3 simulator with the ORBIT testbed, enabling users to emulate instead of just simulate. The second aim is to transfer a video from client to server on two separate nodes using ns-3.
**Session 1**  
12 October 2015

We began by generating an RSA key pair using the appropriate Linux command. The public key was uploaded onto the Orbit website (this was done for each laptop from which we ran experiments: including multiple at once). We logged onto the Orbit-Labs with the given credentials, after having booked sandbox-1 (sb1) consisting of two nodes (node1-1 and node1-2). These nodes are both Ubuntu 14.04.3 systems and hence have the same instruction set.

We initially encountered an error, "Permission denied (public key)" when trying to generate the ssh key. This is because we had moved the private key from its default location, i.e. in a file named id_rsa in a hidden folder ssh in the home directory. Once we moved the file back to its default location, the error was resolved.

The account profile maintains a list of the public keys registered. The first, ending with @internal, is used for internal communication between systems and should not be tampered with.

Initially, both nodes are switched off. We switched them on.

We began by accessing the systems following instructions given in the paper XXX. However, we noted some differences, including that the omf command does not require the version number to follow it. Additionally, instead of transferring ns-3 from the host machine to the sandbox nodes using the scp command, we used mercurial to install ns-3 on each node.

**Session 2**  
14 October 2015

We installed ns-3 on both nodes of the sandbox s1 and stored the system images with the ns-3 installation. The sandboxes continued to function smoothly at the end of the first 2-hour slot. We tried to execute the fd-emu example in ns-3, but faced a "broken pipe" error.
**Session 3**  
**December 1 2015**

In this session, we executed the ping application of `ns-3`.

We began by trying to ping one node from another using `ns-3`. Initially, we faced the same problem we had last time: a broken pipe between the server and client. However, ping was working correctly without `ns-3`. Additionally, if we simultaneously and continuously pinged one node from another without `ns-3`, we were able to ping using `ns-3` partially (2 of 8 packets returned) before the pipe broke.

We realized that the problem was one of IP conflict between the nodes and the `ns-3` client and server on the nodes. We changed the IP addresses of the `ns-3` entities to belong to the same network as the nodes, but with different addresses. The ping application then worked smoothly.

After completing ping, we attempted to transfer a video from one node to another using `ns-3`. We used an existing code to transfer a video between two `ns-3` entities, and achieved the transfer keeping in mind the previous issues with IP conflict.

We then tried to achieve the transfer via a wireless connection between the two nodes. To do so, we installed ath9k using the `modprobe` command, then brought the wireless card up using `sudo ifconfig wlan0 up`.

We then searched for WiFi networks using `sudo iwlist wlan0 scan`.

This command listed several networks. However, we were unable to connect to them as we did not have the appropriate WiFi keys.

We then tried to turn one node into an access point for the other node to connect to wirelessly, but the hardware did not support the same:
List of Commands Used

To enter the first node:
ssh root@node1-1

To show the statistics of both nodes:
omf stat -t all

To turn on both nodes:
omf tell -a on -t all

Output of the above command:
Talking to the CMC service, please wait

 Node: node1-1.sb1.orbit-lab.org State: POWERON
 Node: node1-2.sb1.orbit-lab.org State: POWERON

Pinging one node from another (without using ns-3):
ping root@node1-1
or
ping -c 1 node1-1

Installing mercurial:
apt-get install mercurial

Installing ns-3-dev using mercurial:
hg clone http://code.nsnam.org/ns-3-dev

Building ns-3 with examples and running the first example:
cd ns-3-dev
./waf configure --enable-examples
./waf build
./waf --run examples/tutorial/first

Saving the image of a node:
omf save -n node1-1.sb2.orbit-lab.org

Output of the above command:
INFO exp: - Saving disk image of 'node1-2.sb1.orbit-lab.org' finished with success.

Loading the saved image:
omf load -t node1-2.sb1.orbit-lab.org -i baseline.ndz
Logging out of a node:
logout

Obtaining details of the node:
lscpu

Output of the above command:
Architecture: x86_64
CPU opmode(s): 32bit, 64bit
Byte Order: Little Endian
CPU(s): 8
Online CPU(s) list: 07
Thread(s) per core: 2
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 60
Stepping: 3
CPU MHz: 800.000
BogoMIPS: 6784.44
Virtualization: VTx
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 8192K
NUMA node0 CPU(s): 07

Installing the pre-requisites for ns-3:
apt-get install gcc g++ python python-dev qt4-devtools libgtk3-dev python pygocanvas python-graphviz wireshark

Checking the interface configuration:
ifconfig

Copying the fd-emu UDP echo example into the scratch folder:
cp src/fdnetdevice/examples/fdemuudpecho.cc scratch/echotest.cc

Running the example in server mode:
./waf run "scratch/echotest server=true"

Running the example in client mode:
./waf run "scratch/echotest client=true"