INTEGRATION OF R2LAB WITH NS-3

Practical Training Report

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

INDUKALA NALADALA

(15CO230)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE - 575025

August, 2018

DECLARATION

I hereby declare that the Practical Training report entitled **INTEGRATION OF R2LAB WITH NS-3** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **BACHELOR OF TECHNOLOGY** in **Computer Science and Engineering** is a *bonafide report of the work carried out by me*. The material contained in this report has not been submitted to any University or Institution for the award of any degree.

(15CO230, INDUKALA NALADALA)
Department of Computer Science and Engineering

Place: NITK, Surathkal.

Date: 19-07-2018

CERTIFICATE

This is to *certify* that the Practical Training report entitled **INTEGRATION OF R2LAB WITH NS-3** submitted by **INDUKALA NALADALA**, (Register Number: 15CO230) as the record of the work carried out by her, is *accepted as the Practical Training report submission* in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology**.

WALID DABBOUS
Supervisor

THIERRY TURLETTI
Supervisor

MOHIT TAHILIANI
Guide

Chairman - DUGC

Acknowledgment

I am highly indebted to my mentors, Thierry Turletti and Walid Dabbous for their guidance and support as well as for providing necessary information regarding the project.

My thanks and appreciations also go to my colleagues at INRIA in developing the project and willingly helping me out to the best of their abilities.

Place: Surathkal INDUKALA NALADALA

Date: 19-07-2018

Abstract

Researchers have long faced a fundamental tension between the experimental realism of wireless testbeds on one hand, and the control and repeatability of simulation on the other hand. To overcome the stark tradeoff of these traditional alternatives, we are developing a wireless emulator that enables both realistic and repeatable experimentation by leveraging physical layer emulation. Compared to simulation, our emulator-based approach provides us with a better understanding of real-world wireless network performance, and enables us to quickly deploy our research into an operational wireless network, while still allowing us to enjoy the benefits of a controlled experimental environment.

The primary objective of the proposed research is to help researchers in emulating any wired/wireless network topologies in R2lab. R2lab is an open wireless tested, located in an anechoic chamber for reproducible research in wireless Wi-Fi and 4G/5G networks. It hosts 37 wireless/wired nodes on the ceiling that are remotely controllable, as well as efficient software tools to allow reproducible research on scenarios of small size network topologies. The objective is to extend those software tools by interfacing R2lab nodes with the ns-3 network simulator, e.g., how to connect a real Wi-Fi network to a large-scale wired ns-3 simulated network (by using the ns-3 extensions to interact with the real world).

Keywords: Network emulation, R2lab testbed, ns-3.

Contents

Chapter 1

Introduction

The goal of the internship is to help researchers in emulating any wired/wireless network topologies in R2lab (ns 3 [b]). The project involves interfacing R2lab nodes with the ns-3 network simulator, e.g., how to connect a real Wi-Fi network to a large-scale wired ns-3 simulated network (by using the ns-3 extensions to interact with the real world (INRIA).

What is R2Lab ?

R2lab is an open tested located in an anechoic chamber for reproducible research in wireless WiFi and 4G/5G networks.

R2lab is part of the FIT federation, an open large-scale testing infrastructure for systems and applications on wireless and sensor communications.

Located at INRIA Sophia-Antipolis, R2lab proposes thirty seven customisable commercial off-the-shelf wireless devices, together with USRP nodes and commercial LTE phones, fit to create rich experimental setups. The testbed also features advanced software like leverage GnuRadio and OpenAirterface, as well as efficient software tools, to support easy experimentation.

These tools allow to book the whole testbed, to remotely control the wireless devices, to easily deploy various scenarios and to collect results.

What is ns-3 ?

ns-3 is a discrete-event network simulator, targeted primarily for research and

educational use. ns-3 is a free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

The goal of the ns-3 project is to develop a preferred, open simulation environment for networking research: it should be aligned with the simulation needs of modern networking research and should encourage community contribution, peer review, and validation of the software.

Integration of ns-3 with R2lab

ns-3 is a simulator. Its primary function is to simulate networks of communicating nodes and the traffic between them. To do this, ns-3 provides its primary abstractions of computing nodes with applications to generate traffic; and net devices and channels to move the traffic.

Often, however, it is desirable to mix and match real and simulated entities. For example, in order to validate a network model, one might create a simulation scenario, run the simulated model and gather statistics. Then one might move to real hardware and run the same scenario. One would then compare the results of the two run to verify that the simulation is indeed a faithful representation of the hardware to some specified degree under specified circumstances. One might also envision a scenario in which a network configuration is prohibitively expensive to create using real hardware. In this case, one might want to simulate a large network, but use real applications to drive the simulation through some virtualization scheme.

If we want to use a simulated network and real computing nodes, we use something called "ns-3 TAP." If you want to use simulated nodes and real networks, we use a technology called "ns-3 EMU."

1.1 Issues and Challenges

1. Less research has been done in this area - ORBIT testbed ns 3 [a] has been previously integrated with ns-3. But the examples of the mentioned integration are very basic.

2. Had to become familiar with R2lab and nepi-ng scripts.

3. Had to become familiar with the concept of EMU and TAP in ns-3.

1.2 Outline of the report

The report is divided into sections as follows:

1. Section 2 consists of Literature Review - the research done in this area until now.

2. Section 3 consists of the actual work done.

3. Section 4 consists of the results obtained.

4. Section 5 consists of the conclusion of the report.

Chapter 2

Literature Survey

ns-3 provides a realtime emulation package that allows us to connect ns-3 to real networks on real machines. Typically the real network will be a testbed of some kind. ORBIT is a two-tier laboratory emulator/field trial network project of WINLAB (Wireless Information Network Laboratory), at Rutgers. This wireless network emulator provides a large two-dimensional grid of 400 802.11 radio nodes as well as a number of smaller "sandbox" testbeds to allow one to test without reserving the main grid. ns-3 scripts available in the latest version of ns-3 show how basic client-server prorgams can be run on the ORBIT testbed (integrated with ns-3).

CMU Wireless Emulator uses real network cards along with a wireless signal propagation emulator. This allows for even more reproducible results than on a testbed with the advantage of not requiring a space the size of a hockey rink. This system uses a wireless network emulator that accurately emulates wireless signal propagation. The emulator takes in the signals generated by wireless network cards and subjects them to the same effects that occur in the real world (e.g. attenuation, multi-path fading, etc.). It then feeds the combined signals back into the wireless cards.

R2lab tutorials give an idea about how to use R2lab to perform network experiments - load images, simple ping, file transfers, etc.

ns-3 documentation gives details regarding the implementation of fd-net-device and tap-bridge in ns-3, which can be used in the integration of R2lab with ns-3.

2.1 Problem Statement

The objective of the internship is to help researchers in emulating any wired/wireless network topologies in R2lab. We have to interface R2lab nodes with the ns-3 network simulator, e.g., connect a real Wi-Fi network to a large-scale wired ns-3 simulated network (by using the ns-3 extensions to interact with the real world).

2.2 Objectives

1. Ping ns-3 simulated node running on a real node from another real node
2. Send multicast packets from a real node to ns-3 simulated network on another real node
3. Extend the integration of R2lab and ns-3 to support various topologies of ns-3 network - CSMA channel and OLSR protocol in WiFi.
4. Use VLC and multicast packets sending tool, mcsender to test the effectiveness of the integration of R2lab and ns-3

Chapter 3

Details of the work carried out in Industry

The following is a high level view of the work carried out as a part of this project:

The script prepared is used to integrate the network simulator, ns-3 with R2lab. Consider two nodes, fit03 and fit04. A network simulation is running on fit04. We try to ping a simulated node in fit04 from the real node fit03. Here, fit03 is the server node and fit04 is the client node. The following options are available:

Point to point scenario:

1. Ping

   We ping an ns-3 simulated node, running on fit04 from fit03. We can use the –target option to specify the ns-3 simulated node. By default, the ns-3 node with IP address 10.1.1.2 is pinged from fit03. This is the default scenario in the script.

2. VLC

   We send unicast packets from fit03 to the ns-3 simulated node with IP address 10.1.1.2 using VLC. This option can be enabled through –vlc.

Multicast scenario:

1. mcsender

We send multicast packets from fit03 to the ns-3 simulated nodes using mcsender, a multicast test tool to send multicast test packets. This option can be enabled using –multicast.

2. VLC

   We send multicast packets from fit03 to the ns-3 simulated nodes using VLC. This option can be enabled using –multicast –vlc.

The following two topology options are available for all the scenarios listed above:

1. CSMA

   The ns-3 simulated network in fit04 is a CSMA channel consisting of 4 simulated nodes, with IP addresses ranging from 10.1.1.1 to 10.1.1.4. CSMA channel is the default topology for all scenarios.

2. OLSR

   The ns-3 simulated network in fit04 uses the Optimized Link State Routing (OLSR) protocol, which is a dynamic mobile ad hoc unicast routing protocol. This topology can be enabled by using –olsr.

Example:

To run the scenario where the network simulated in fit04 uses OLSR and receives multicast packets sent from fit03 using VLC, the command is:

python3 ns3-scenario.py –vlc –multicast –olsr

Additional details:

1. The server node and client node for ns-3 scenario can be mentioned using the arguments, –server and –client.

2. The target ns-3 simulated node, where the tap device is to be created can be mentioned using the argument –target.

3. The duration for which ns-3 simulation has to run on the client node can be mentioned using the argument –duration.

4. If images have to be loaded on the server and client nodes, -l has to be used.

For example, to have fit01 as the server node and fit02 as the client node, with ns-3 simulated node with IP address 10.1.1.3 as the target, we use the command: (images are loaded on fit01 and fit02) python3 ns3-scenario.py –server=1 –client=2 –target=3 -l

Here, fit01 will ping the ns-3 simulated node with IP address 10.1.1.3, which is running on fit02. The ns-3 network topology would be CSMA channel.

Output:

1. The script retrieves pcap file generated for the target ns-3 simulated node to the local machine.

2. tcpdump output at tap0 on the client node is also retrieved at the local machine. (currently gives output only for CSMA channel)

3.1 Existing Method/System

Currently, R2lab has not been integrated with any simulator. However, there are a few tesbeds, like ORBIT which have been partially integrated with ns-3.

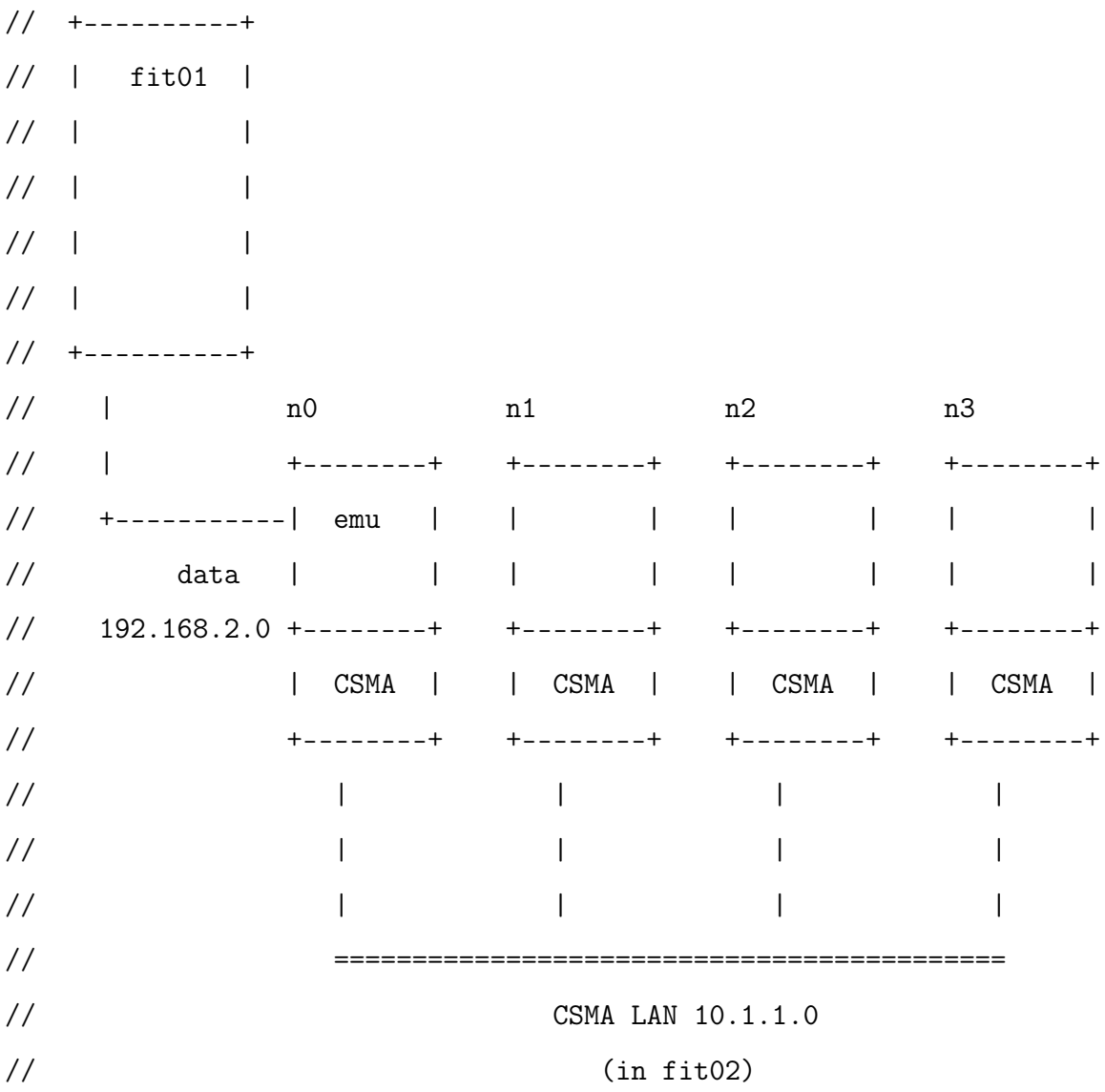
3.2 Proposed Method/System

Proposed method integrates R2lab with ns-3 allowing a mix of experimentation and simulation.

3.3 Implementation Details

The following is the explanation of **CSMA ns-3** script written:

```
// Network Topology:
//
```



Local Node (in R2lab), Remote Node (in R2lab), Destination Node (in ns-3) are passed as command-line arguments to the ns-3 script. Multicast option can also be turned on using command-line argument.

Calculation of checksums in protocols is enabled as we would be interacting with real-world machines via ns-3.

Emu netdevice is installed on the first ns-3 node, n0. Emu netdevice allows a simulation node to send and receive packets over a real network. The emulated net device relies on a specified interface - "data" being in promiscuous mode. It opens a raw socket and binds to that interface. We perform MAC spoofing to separate simulation

network traffic from other network traffic that may be flowing to and from the host.

CSMA channel (for nodes n0, n1, n2 and n3) is created.

In the ns-3 script, we use:

```
Ipv4Address gateway ("0.0.0.0");
Ipv4StaticRoutingHelper ipv4RoutingHelper;
Ptr<Ipv4StaticRouting> staticRouting = ipv4RoutingHelper.GetStaticRouting (ip

staticRouting->SetDefaultRoute (gateway, interface);
```

When the ping application sends its ICMP packet, it will send it down the ns-3 protocol stack. We set the IP address of the destination to the address corresponding to Remote Node (in R2lab). This address is off our local network so we have got to provide some kind of default route to ns-3 to be able to get that ICMP packet forwarded off of our network.

You have got to provide an IP address of a real host that you can send real packets to and have them forwarded off of your local network. We mention gateway as 0.0.0.0 .

If MULTICAST option is enabled using command-line argument, we set up multicast routing. The V4Ping application is used to send ICMP echo requests from the ns-3 simulated node, n2 to the Remote Node (in R2lab).

Additionally, in the **OLSR ns-3 script**:

We add the required routes into the Ipv4StaticRouting Protocol instance and have the node generate HNA messages for all these routes which are associated with non-OLSR interfaces specified in the code.

The OLSR code is a modification of the file: *examples/routing/manet-routing-compare.cc*

The **pcap files** generated by ns-3 can be viewed using tcpdump or wireshark.

Wireshark can be used to extract payload when we send a video from Local Node (in R2lab) to Destination Node (in ns-3).

NEPI-NG SCRIPT:

NEPI is a tool that provides a uniform API to run experiments on many platforms. The entire experiment is automated using nepi-ng script. The nepi-ng script automates the following:

1. Load images on the R2lab nodes
2. Configure routes in the Remote node (in R2Lab)
3. Install mcsender, VLC on Remote and Local R2lab nodes (as required)
4. Run ns-3 scripts
5. Retrieve pcap traces and received video

The ns-3 and nepi-ng scripts are available in the github repository: **fit-r2lab/r2lab-demos**

Chapter 4

Results

I was successful in partially integrating R2lab with ns-3. The results of the scripts in the previous section are as follows:

1. **Result of Ping from fit03 to ns-3 simulated node (on fit04) with IP address 10.1.1.2 (when the ns-simulated network is CSMA channel)**:

```
fit03:PING 10.1.1.2 (10.1.1.2) 56(84) bytes of data.
fit03:64 bytes from 10.1.1.2: icmp_seq=7 ttl=63 time=21.3 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=8 ttl=63 time=4.69 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=9 ttl=63 time=5.00 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=10 ttl=63 time=4.97 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=11 ttl=63 time=6.29 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=12 ttl=63 time=4.98 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=13 ttl=63 time=5.01 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=14 ttl=63 time=4.94 ms
fit03:64 bytes from 10.1.1.2: icmp_seq=15 ttl=63 time=5.33 ms
```

The output of tcpdump at tap device running on the ns-3 simulated node with IP address, 10.1.1.2 is:

```
tcpdump -nn -tt -r tap0.pcap
reading from file tap0.pcap, link-type EN10MB (Ethernet)
```

```
1531814664.160478 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 1 gr
1531814664.436488 IP6 :: > ff02::1:ff00:1: ICMP6, neighbor solicitation, who has f
1531814664.608464 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 1 gr
1531814665.157968 ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.4,
1531814665.159826 ARP, Reply 10.1.1.1 is-at 00:00:00:00:00:03, length 46
1531814665.162286 IP 10.1.1.4 > 192.168.2.3: ICMP echo request, id 0, seq 0, lengt
1531814665.178078 ARP, Request who-has 10.1.1.4 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1,
1531814665.180346 ARP, Reply 10.1.1.4 is-at 00:00:00:00:00:06, length 46
1531814665.182304 IP 192.168.2.3 > 10.1.1.4: ICMP echo reply, id 0, seq 0, length
1531814665.331047 ARP, Request who-has 10.1.1.2 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1,
1531814665.333184 ARP, Reply 10.1.1.2 is-at 00:00:00:00:00:04, length 46
1531814665.335418 IP 192.168.2.3 > 10.1.1.2: ICMP echo request, id 12329, seq 7, l
1531814665.341379 ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.2,
1531814665.343586 ARP, Reply 10.1.1.1 is-at 00:00:00:00:00:03, length 46
1531814665.345724 IP 10.1.1.2 > 192.168.2.3: ICMP echo reply, id 12329, seq 7, leng
1531814666.154963 IP 10.1.1.4 > 192.168.2.3: ICMP echo request, id 0, seq 1, lengt
1531814666.157590 IP 192.168.2.3 > 10.1.1.4: ICMP echo reply, id 0, seq 1, length
```

2. **Result of sending multicast packets (using mcsender) from fit03 to ns-simulated network on fit04 (Topology: CSMA channel):**

```
tcpdump -nn -tt -r packets-2-0.pcap
reading from file packets-2-0.pcap, link-type EN10MB (Ethernet)
0.919687 IP 192.168.2.3.47052 > 225.1.2.4.1234: UDP, length 46
1.919705 IP 192.168.2.3.47052 > 225.1.2.4.1234: UDP, length 46
2.919878 IP 192.168.2.3.47052 > 225.1.2.4.1234: UDP, length 46
3.920000 IP 192.168.2.3.47052 > 225.1.2.4.1234: UDP, length 46
4.920128 IP 192.168.2.3.47052 > 225.1.2.4.1234: UDP, length 46
5.920215 IP 192.168.2.3.47052 > 225.1.2.4.1234: UDP, length 46
```

3. **Result of sending multicast packets (using VLC) from fit03 to ns-simulated network on fit04 (Topology: CSMA channel):**

```
tcpdump -nn -tt -r packets-2-0.pcap
reading from file packets-2-0.pcap, link-type EN10MB (Ethernet)
7.254588 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.258950 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.263889 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.268467 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.272977 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.277428 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.282213 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.287113 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
7.291654 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
```

The result of tcpdump at tap device running on one of the ns-3 simulated nodes in the network is:

```
1531815709.748266 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.752462 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.756728 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.761520 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.765818 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.930782 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.935230 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.939488 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
1531815709.943780 IP 192.168.2.3.55387 > 225.1.2.4.1234: UDP, length 1316
```

4. **Result of sending multicast packets (using VLC) from fit03 to ns-simulated network on fit04 (Topology: WiFi - OLSR protocol):**

```
tcpdump -nn -tt -r packets-2-0.pcap
24.157520 IP 192.168.2.3.37130 > 225.1.2.4.1234: UDP, length 1316
24.159066 IP 192.168.2.3.37130 > 225.1.2.4.1234: UDP, length 1316
24.160672 IP 192.168.2.3.37130 > 225.1.2.4.1234: UDP, length 1316
24.162018 IP 192.168.2.3.37130 > 225.1.2.4.1234: UDP, length 1316
24.163444 IP 192.168.2.3.37130 > 225.1.2.4.1234: UDP, length 1316
24.172255 IP 10.1.1.3.698 > 10.1.1.255.698: OLSRv4, seq 0x000c, length 20
24.200851 IP 10.1.1.2.698 > 10.1.1.255.698: OLSRv4, seq 0x000c, length 44
24.221542 IP 10.1.1.1.698 > 10.1.1.255.698: OLSRv4, seq 0x000e, length 44
24.345536 IP 192.168.2.3.37130 > 225.1.2.4.1234: UDP, length 1316
24.346882 IP 192.168.2.3.37130 > 225.1.2.4.1234: UDP, length 1316
```

Chapter 5

Conclusions

Network emulation is a hybrid approach that combines real elements of a deployed network application, such as end hosts and protocol implementations with synthetic, simulated, or abstracted elements, such as the network links, intermediate nodes and background traffic. Which elements are real and which are partially or fully simulated will often differ, depending on the experimenter's needs and the available resources. A fundamental difference between simulation and emulation is that while the former runs in virtual simulated time, the latter must run in real time. Another important difference is that it is impossible to have an absolutely repeatable order of events in an emulation. That is due to its realtime nature and, often, a physically-distributed computation infrastructure.

Emulators are a middle ground between pure simulation and wireless testbeds that combine the repeatability, configurability, isolation and manageability of simulations and the realism of testbeds. They utilize a real MAC layer, provide a realistic physical layer and avoid adopting a uncontrollable or locale-specific architecture. They alos provide a high degree of control and fidelity.

The integration of R2lab testbed with ns-3 in the case of topologies like CSMA channel, WiFi (with OLSR protocol) was successful. Testing of code indicated successful transmission of packets from a real node, say fit03 to ns-3 simulated network running on another real node, say, fit04. This was checked using tools like ping (for unicast

packets), mcsender (for multicast packets) and VLC (for both unicast and multicast packets).

Future work would be to use the ns-3 - integrated R2lab for testing various protocols.

Bibliography

INRIA. R2lab: Reproducible Research Lab. *https://r2lab.inria.fr/*.

ns 3. How to use ns-3 in the ORBIT testbed. *www.nsnam.org*, a.

ns 3. Documentation : ns-3. *www.nsnam.org*, b.

Brief Bio-Data

Naladala Indukala
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
P.O. Srinivasnagar
Mangalore, 575025
Mobile: 9880692703
Email: indukala.nitk@gmail.com

Permanent Address

Naladala Indukala
8-E Malayagiri
Anushaktinagar
Mumbai, 400094

Education

10^{th} - Atomic Energy Central School-2, Anushaktinagar, Mumbai, Maharashtra, India, 2013. (10 CGPA)

12^{th} - SRP Junior College, Govandi, Mumbai, Maharashtra, India, 2015. (94.6%)
(B. Tech) -

- 1 Sem - 9.1 (CGPA)
- 2 Sem - 9.38 (CGPA)
- 3 Sem - 9.35 (CGPA)
- 4 Sem - 9.25 (CGPA)
- 5 Sem - 9.32 (CGPA)
- 6 Sem - 9.43 (CGPA)