CS 470 Programming Assignment 2

A Simple SDN Controller Design using Trema

Due Time: 06/03/2016 @ 23:59:59

1. Objective

Getting Started: Familiarize yourself with Trema and Ruby.

Implement: Use Trema to develop a simple SDN controller that performs the basic functionality of layer 2 switching.

2. Getting Started

2.1 Install Trema

Option 1 (Recommended): Install from scratch

Install Ruby

Step 1: Install dependencies for Ruby

```
1. sudo apt-get update
2. sudo apt-get install git-core curl zlib1g-dev build-essential
   libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3
   libxml2-dev libxslt1-dev libcurl4-openssl-dev
   python-software-properties libffi-dev libpcap-dev
```

Step 2: Download Ruby 2.2.2 and install

```
1. cd
2. wget http://ftp.ruby-lang.org/pub/ruby/2.2/ruby-2.2.2.tar.gz
3. tar -xzvf ruby-2.2.2.tar.gz
4. cd ruby-2.2.2/
5. ./configure
6. make
7. sudo make install
8. ruby -v (ruby 2.2.2 should be shown)
```

Reference: https://gorails.com/setup/ubuntu/14.04

Install Openvswitch

```
1. sudo apt-get install openvswitch-switch
```

Install Trema

Step 1: Install bundler

```
1. (sudo) gem install rubygems-update
```

2. (sudo) update_rubygems
3. (sudo) gem install bundler

Step 2: Install Trema

1. git clone https://github.com/trema/trema-edge
2. cd trema-edge
3. bundle install
4. rake

Note: If you encounter any errors, please read the output message carefully and google the solution online, and you can learn a lot through these processes. Some representative errors and fix are linked below:

1. missing pcap.h or pcap-bpf.h
   sudo apt-get install libpcap-dev
2. missing unf_exf
   sudo apt-get install build-essential
3. missing ffi
   sudo apt-get install ruby-dev

Option 2 (simple): Use customized virtual machine

Step 1: Install an Oracle Virtual Box (free) https://www.virtualbox.org/wiki/Downloads

Step 2: Download the customized virtual machine (VM) from

Step 3: Install the VM, start Virtual Box, select File->Import Appliance-> and select the downloaded .ova file. The VM requires 1G RAM and 8G disk.

Step 4: **username: cs470, password: trema**

2.2 Test Packet In Application

2.2.1 Hello Trema

```
cs470@cs470-VirtualBox:~$ cd trema-edge/
cs470@cs470-VirtualBox:~/trema-edge$ ./trema run
src/examples/hello_trema/hello_trema.rb -c
src/examples/hello_trema/hello_trema.conf
Hello 0xabc from src/examples/hello_trema/hello_trema.rb!
```

2.2.2 Packet In

```
cs470@cs470-VirtualBox:~/trema-edge$ ./trema run
src/examples/packet_in/packet-in.rb -c
src/examples/packet_in/packet_in.conf
received a packet_in!
datapath_id: 0xabc
transaction_id: 0
buffer_id: 0
total_len: 110
reason: 0
```

```
table_id: 0
cookie: 0x1
in_port: 2
……..
```

2.2.3 Familiar with Trema APIs

1. Familiarize yourself with Trema APIs http://www.rubydoc.info/github/trema/trema/master/frames
2. Familiarize with Trema commands https://www.relishapp.com/trema/trema/docs/trema-commands
3. Learn and try to execute other examples under src/example/, e.g., the learning_switch application.
4. Other information about Trema: http://trema.github.io/trema/

3. Implementation

3.1 Packet Filter

The existing packet in application shows all the messages that come in the SDN controller. Since many of them are not very useful in this assignment (e.g., a lot of broadcasting messages from 0.0.0.0 to 255.255.255.255), we need to modify the packet in application to filter out the messages that we focus on. The functionality we need to achieve is: when we type in ./trema run src/examples/packet_in/packet-in.rb -c src/examples/packet_in/packet_in.conf, NO packet in messages will be shown. After we open a new terminal and type in ./trema send_packets --source host1 --dest host2 --n_pkts 10, only messages from 192.168.0.1 to 192.168.0.2 will be shown.

3.2 L2 Switch

After we finish the packet filter and test the send_packet command, we can open another new terminal and type in ./trema show_stats host1 or ./trema show_stats host2. We can find out that no packet is successfully transferred from host1 to host2. As shown in the result below, host 1 has sent 10 packets, but host2 do not receive any packets.

```
cs470@cs470-VirtualBox:~/trema-edge$ ./trema show_stats host1
Sent packets:
ip_dst,tp_dst,ip_src,tp_src,n_pkts,n_octets
192.168.0.2,1,192.168.0.1,1,10,500
Received packets:
cs470@cs470-VirtualBox:~/trema-edge$ ./trema show_stats host2
Sent packets:
Received packets:
```

The reason is that no routing entry has been inserted in the open vswitch yet. In this implementation, we will modify the packet filter application to develop a Layer 2 switching controller. The L2 switch can use Trema APIs (e.g., send_flow_mod_add, send_flow_mod_modify and send_packet_out and etc.) to insert a routing entry (from host1 to host2) into the routing table of the open vswitch. After L2 switch is run, if we run the send_packets command and show_stats host2, we should see that host2 has received 10 packets.

4. Submission and Grading

4.1 What to Submit

Your submission should contain a tar file – Name it as < your CIN number_your name>.tar:

- All source files (.rb, .conf and .c or .cc files), including Makefile and a README if you use C/C++.
- A report that summarize your observation and analysis in Section 3.3. (**No more than 5 pages**).

4.2 How to submit

Submit your .tar or .zip file on CSNS.

Each team only needs to make one submission.

4.3 Grading Criteria

- Correctness of output
- Your analysis
- Organization and documentation of your code

4.4 Important Key Points:

- DON’T ASSUME. If you have any doubts in project description, please come to my office hour.
- NO multiple submissions please. If you have more than 1 submission will not consider any of your submissions.
- Submission deadline is hard. No extension.
- Please do not submit any binaries or object files or any test files.