

Computer Comm & Networks - ITCS 6166

Assignment - 3

Due on : 04/03/2018 11:59pm

Goal: In this exercise, you will be learning how to build SDN application for implementing simple packet forwarding using RYU SDN controller. You'll also learn how to do performance testing of these custom topologies using ping and iperf.

Discussion Topic: RYU SDN controller – Post any questions and collaborate with your peers.

Note: Please go through the references first before going through this document.

Overview:

You should familiarize yourself with the types of openflow messages exchanged between the switch and controller. In particular, you should be able identify message type and the action that needs to be taken by the controller for the message.

Some key message types are OF-PacketIn, OF-PacketOut, OF-PacketMod,

- OF-PacketIn – Message sent by the switch to controller for requesting forwarding decision (flow rule)
- OF-PacketOut – Message sent by the controller to switch to forward the packet using the port on the switch
- OF-FlowMod – Message sent by the controller to switch with instruction to ADD/DELETE/MODIFY the existing forwarding decision (flow rule) on the flow table.

The switch will use OF-PacketIn messages to inform the controller whenever it receives a unknown destination MAC address.

Your RYU forwarding application should be able to install forwarding rule (flow rule) to the switches to enable communication between all the hosts. You should make use of OF-PacketIn, OF-PacketOut and OF-FlowMod to write this forwarding application.

Steps:

- Fire up your mininet python script which contains your network topology
- Start your RYU controller with `--verbose` option to enable logging

Use the network topology that you created in assignment-1 and modify the link parameters. Change the switch controller type to remote controller with IP address 127.0.0.1 and port number 6633.

ASSIGNMENT TOPOLOGY:

Background

Data center networks typically have a tree-like topology. End-hosts connect to top-of-rack switches, which form the leaves (edges) of the tree; one or more core switches form the root; and

one or more layers of aggregation switches form the middle of the tree. In a basic tree topology, each switch (except the core switch) has a single parent switch. Additional switches and links may be added to construct more complex tree topologies (e.g., fat tree) in an effort to improve fault tolerance or increase inter-rack bandwidth.

In this assignment, your task is to implement simple packet forwarding using RYU SDN controller. You will assume each level i.e., core, aggregation, edge and host to be composed of a single layer of switches/hosts with a configurable fanout value (k). For example, a simple tree network having a single layer per each level and a fanout of 2 looks like:

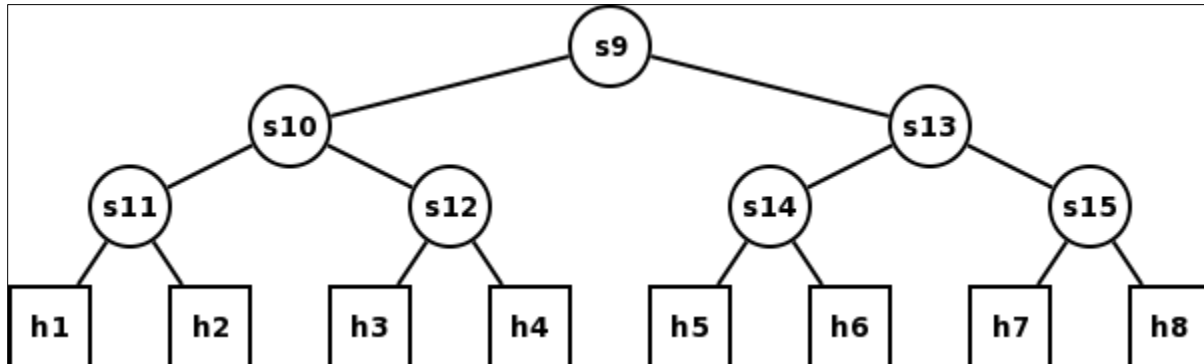


Figure 2: Simple Tree Topology

Verification:

1. Use pingall command in mininet terminal and ensure all hosts are reachable with 0% dropped.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> _
```

2. Use iperf command to ensure the link parameters are enforced properly as in the given network topology

Iperf is a traffic generation / network performance measurement tool that can operate in client / server mode. The server mode operation will open a port and accept the incoming connection requests. Client mode operation will generate requests towards the Iperf server. We will be using this tool extensively for measuring the end-to-end network bandwidth.

Example: To measure the end-to-end bandwidth between h1 and h2, do the following start iperf server on host h1 use the following command in terminal:

```
iperf -s
```

To start the iperf client on host h2 use the following command in terminal:

```
iperf -c 10.0.0.1
```

Now, you will should be able to see the end-to-end network bandwidth as shown in the below figure:

```
^Croot@mininet-vm:~# iperf -c 10.0.0.1
-----
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 13] local 10.0.0.2 port 41690 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 13] 0.0-10.0 sec  33.1 GBytes 28.5 Gbits/sec
root@mininet-vm:~#
```

If you see there is no packet drop then, you have successfully met the goal of this assignment.

Submitting your Code

You should turn in your python script on CANVAS course site assignment section with filename named in the following format:

Please include the below things in your assignment submission:

1. Python script: Save file as **ryu_yourfirstname.py**
2. Snapshot of pingall and iperf verification.

Upload the Compressed file as **Assignment3_yourfirstname.zip**

Note: Please follow the deadline strictly as we would not accept any submissions after the deadline.

Also in case of any question post your query in discussion portal.

References:

1. <http://mininet.org/api/hierarchy.html>
2. <https://github.com/mininet/mininet/wiki/Documentation>
3. <https://www.tutorialspoint.com/python/>
4. https://ryu.readthedocs.io/en/latest/writing_ryu_app.html