Mininet/Faucet

{{ :2015:nsrc-tein-ait-noc:reannz-logo-grey-scale-v2-300x110.jpg |Thanks REANNZ}}

Objectives

In this session, we will use another python SDN application. This time the application is known as Faucet.

Faucet is an Openflow controller designed by the New Zealand research and education network REANNZ, to be a layer 2 switch based on OpenvApour's Valve. It handles MAC learning and supports VLANs.

It supports:

- OpenFlow v1.3
- Multiple datapaths
- Mixed tagged/untagged ports
- Port statistics
- Coexisting with other OpenFlow controllers

In this lab we will: * Run the Ryu controller with the Faucet application * Use various commands to gain experience with the increased feature set of the Faucet software. * Run the Gauge software to see how a datapath can exist with multiple controller applications.

Stop ryu-manager

Stop the ryu-manager with Ctrl-C if it is running.

Stop the Mininet environment

```
mininet> quit
*** Stopping 1 switches
s1 ..........
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Stopping 1 controllers
c0
*** Done
completed in 740.498 seconds
root@mininet-vm:~#
```

Look at the first Faucet config file

```
# cat /etc/opt/faucet/faucet.yaml
```

It should look something like this:

```
dp_id: 0x0000000000000001   # The id of the datapath to be controlled
name: "test-faucet-1" # The name of the datapath for use with logging
description: "Initial Test Faucet"    # Purely informational
hardware: "Open_vSwitch"  # used to determine which valve implementation to use
monitor_ports: True # whether gauge should monitor stats for ports
monitor_ports_file: "/var/log/faucet/ports.out" # The file to record ports
statistics
monitor_ports_interval: 10  # the polling interval for port stats in seconds
monitor_flow_table: True    # whether gauge should take periodic flow table dumps
monitor_flow_table_file: "/var/log/faucet/ft.out"   # the file to record flow table
dumps
monitor_flow_table_interval: 10 # the polling interval for flow table monitoring
interfaces:
    1:
        native_vlan: 100
        name: "port1"   # name for this port, used for logging/monitoring
        description: "Port 1"    # informational
    2:
        native_vlan: 100
        name: "port2"
        description: "Port 2"
    3:
        native_vlan: 100
        name: "port3"
        description: "Port 3"
    4:
        native_vlan: 100
    5:
        native_vlan: 100
    6:
        native_vlan: 100
    7:
        native_vlan: 100
    8:
        native_vlan: 100
    9:
        native_vlan: 100
    10:
        native_vlan: 100
vlans:
    100:
        description: "Test vlan" # informational
        name: "test_vlan"  # used for logging/monitoring
```

As you can see this places 10 ports on the device with Data Path ID 1 into vlan 100 (untagged).

Start Ryu and the Faucet controller

You should still have 2 ssh sessions open. Start Ryu and the Faucet controller in the controller window.

```
# ryu-manager --verbose ./faucet.py
loading app ./faucet.py
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
instantiating app ./faucet.py of Faucet
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK dpset
  PROVIDES EventDP TO {'Faucet': set(['dpset'])}
  CONSUMES EventOFPStateChange
  CONSUMES EventOFPPortStatus
  CONSUMES EventOFPSwitchFeatures
BRICK Faucet
  CONSUMES EventOFPPortStatus
  CONSUMES EventOFPPacketIn
  CONSUMES EventDP
BRICK ofp_event
  PROVIDES EventOFPStateChange TO {'dpset': set(['main', 'dead'])}
  PROVIDES EventOFPPortStatus TO {'dpset': set(['main']), 'Faucet': set(['main'])}
  PROVIDES EventOFPSwitchFeatures TO {'dpset': set(['config'])}
  PROVIDES EventOFPPacketIn TO {'Faucet': set(['main'])}
  CONSUMES EventOFPEchoRequest
  CONSUMES EventOFPSwitchFeatures
  CONSUMES EventOFPHello
  CONSUMES EventOFPErrorMsg
  CONSUMES EventOFPPortDescStatsReply
```

Start Mininet

We'll now start the Mininet environment with 10 hosts once more in the mininet window.

```
# mn --topo=tree,1,10 --mac --controller=remote --switch ovsk,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1) (h7, s1) (h8, s1) (h9, s1)
(h10, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
*** Starting 1 switches
s1
```

```
*** Starting CLI:
mininet>
```

Ensure that the bridge is using OpenFlow13

```
mininet> sh ovs-vsctl set bridge s1 protocols=OpenFlow13
```

Dump flows again to view differences.

Look at the flow rules - Can you explain what they are all doing?

```
mininet> dpctl dump-flows -O OpenFlow13
*** s1 ------------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x5adc15c0, duration=46.669s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=3
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.668s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=7
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.668s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=8
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.669s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=1
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.668s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=9
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.667s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=10
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.669s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=5
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.668s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=6
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.669s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=2
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.669s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=4
actions=push_vlan:0x8100,set_field:100->vlan_vid,goto_table:1
 cookie=0x5adc15c0, duration=46.669s, table=0, n_packets=0, n_bytes=0,
priority=9099,dl_type=0x88cc actions=drop
 cookie=0x5adc15c0, duration=46.669s, table=0, n_packets=0, n_bytes=0, priority=0
actions=drop
 cookie=0x5adc15c0, duration=46.67s, table=0, n_packets=0, n_bytes=0,
priority=9099,dl_dst=01:80:c2:00:00:00 actions=drop
 cookie=0x5adc15c0, duration=46.67s, table=0, n_packets=0, n_bytes=0,
priority=9099,dl_dst=01:00:0c:cc:cc:cd actions=drop
```

```
 cookie=0x5adc15c0, duration=46.667s, table=1, n_packets=0, n_bytes=0, priority=0
actions=CONTROLLER:65509,goto_table:2
 cookie=0x5adc15c0, duration=46.669s, table=2, n_packets=0, n_bytes=0, priority=0
actions=goto_table:3
 cookie=0x5adc15c0, duration=46.667s, table=3, n_packets=0, n_bytes=0,
priority=0,dl_vlan=100
actions=strip_vlan,output:1,output:2,output:3,output:4,output:5,output:6,output:7,o
utput:8,output:9,output:10
```

Why might the rules no longer be using actions=ALL?

Ping between two hosts

In the Mininet window

```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=64 time=7.01 ms
64 bytes from 10.0.0.2: icmp_req=2 ttl=64 time=0.438 ms
64 bytes from 10.0.0.2: icmp_req=3 ttl=64 time=0.074 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.074/2.510/7.018/3.191 ms
mininet>
```

Dump flows again to view differences.

```
mininet> dpctl dump-flows -O OpenFlow13
```

Take note of the lines which look like the following: What do they do?

```
cookie=0x5adc15c0, duration=17.78s, table=1, n_packets=3, n_bytes=238,
hard_timeout=300, priority=9001,in_port=2,dl_vlan=100,dl_src=00:00:00:00:00:02
actions=goto_table:2
cookie=0x5adc15c0, duration=17.78s, table=1, n_packets=3, n_bytes=238,
hard_timeout=300, priority=9001,in_port=1,dl_vlan=100,dl_src=00:00:00:00:00:01
actions=goto_table:2
cookie=0x5adc15c0, duration=17.78s, table=2, n_packets=3, n_bytes=238,
idle_timeout=300, priority=9001,dl_vlan=100,dl_dst=00:00:00:00:00:01
actions=strip_vlan,output:1
cookie=0x5adc15c0, duration=17.78s, table=2, n_packets=3, n_bytes=238,
idle_timeout=300, priority=9001,dl_vlan=100,dl_dst=00:00:00:00:00:02
actions=strip_vlan,output:2
```

Ping between all the hosts

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Results: 0% dropped (90/90 received)
mininet>
```

Shutdown Mininet and Ryu

```
mininet> exit
*** Stopping 1 switches
s1 ..........
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Stopping 1 controllers
c0
*** Done
completed in 515.983 seconds
```

Use Ctrl-C in the Ryu window to shutdown the faucet application

Change the Faucet config file to create a new VLAN

We are going to edit the faucet config file and place half the ports into a new vlan. Edit the file with the following commands

```
$EDITOR /etc/opt/faucet/faucet.yaml
```

Change ports 6-10 to be in vlan 200

```
6:
        native_vlan: 200
7:
        native_vlan: 200
8:
        native_vlan: 200
9:
        native_vlan: 200
10:
        native_vlan: 200
```

Then create a new vlan in the `vlans:` section

```
vlans:
    100:
        description: "Test vlan" # informational
        name: "test_vlan"  # used for logging/monitoring
    200:
        description: "Test vlan 200" # informational
        name: "test_vlan_2"  # used for logging/monitoring
```

Restart Faucet

```
# ryu-manager --verbose ./faucet.py
```

Restart Mininet

```
# mn --topo=tree,1,10 --mac --controller=remote --switch ovsk,protocols=OpenFlow13
...
mininet> sh ovs-vsctl set bridge s1 protocols=OpenFlow13
```

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 X X X X X
h2 -> h1 h3 h4 h5 X X X X X
h3 -> h1 h2 h4 h5 X X X X X
h4 -> h1 h2 h3 h5 X X X X X
h5 -> h1 h2 h3 h4 X X X X X
h6 -> X X X X X h7 h8 h9 h10
h7 -> X X X X X h6 h8 h9 h10
h8 -> X X X X X h6 h7 h9 h10
h9 -> X X X X X h6 h7 h8 h10
h10 -> X X X X X h6 h7 h8 h9
*** Results: 55% dropped (40/90 received)
mininet>
```

Dump flows again to view differences.

```
mininet> dpctl dump-flows -O OpenFlow13
```

What do you notice about the table=3 rules this time?

```
cookie=0x5adc15c0, duration=175.413s, table=3, n_packets=89, n_bytes=3962,
priority=0,dl_vlan=200
actions=strip_vlan,output:6,output:7,output:8,output:9,output:10
cookie=0x5adc15c0, duration=175.411s, table=3, n_packets=93, n_bytes=4242,
priority=0,dl_vlan=100
actions=strip_vlan,output:1,output:2,output:3,output:4,output:5
```