Mininet/Valve

Objectives

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

- Run the Ryu controller with the Valve application
- Use various commands to gain experience with the increased feature set of the Valve software.

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:~#
```

Change to the Valve directory

We have placed a copy of the Valve software into the VM. If you want to get this later then you can clone the repo from: https://github.com/openvapour/valve

Change to the directory containing the Valve software

```
root@mininet-vm:~# cd valve
root@mininet-vm:~/valve#
```

Look at the first Valve config file

```
root@mininet-vm:~/valve# cat valve.yaml-flat
---

0000000000000001:
default:
type: untagged
```

```
vlans: [10]
```

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

Copy the config file into place

```
root@mininet-vm:~/valve# cp valve.yaml-flat valve.yaml
```

Start Ryu and the Valve controller

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

Start Mininet

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

```
root@mininet-vm:~/valve# 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>
```

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=0x0, duration=3.399s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff
actions=output:1,output:2,output:3,output:4,output:5,output:6,output:7,output:8,
output:9,output:10
cookie=0x0, duration=2.559s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=8,dl_src=00:00:00:00:00:08,dl_dst=ff:ff:ff:ff:ff:ff
actions=output:1,output:2,output:3,output:4,output:5,output:6,output:7,output:8,
output:9,output:10
cookie=0x0, duration=1.215s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=5,dl_src=00:00:00:00:00:05,dl_dst=ff:ff:ff:ff:ff:ff
actions=output:1,output:2,output:3,output:4,output:5,output:6,output:7,output:8,
output:9,output:10
cookie=0x0, duration=2.455s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=6,dl_src=00:00:00:00:00:06,dl_dst=ff:ff:ff:ff:ff:ff
actions=output:1,output:2,output:3,output:4,output:5,output:6,output:7,output:8,
output:9,output:10
cookie=0x0, duration=2.143s, table=0, n_packets=0, n_bytes=0,
priority=9000,in_port=7,dl_src=00:00:00:00:00:07,dl_dst=ff:ff:ff:ff:ff:ff
actions=output:1,output:2,output:3,output:4,output:5,output:6,output:7,output: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>
```

We can see in the Ryu output, the Valve software has added flows

```
EVENT ofp_event->Valve EventOFPPacketIn
Jan 27 02:06:18 Valve  INFO     dpid:1                Packet_in
src:00:00:00:00:00:02 dst:00:00:00:00:00:01 in_port:2 vid:10
```

```
Jan 27 02:06:18 Valve  INFO     dpid:1                Adding unicast flow
dl_dst:00:00:00:00:00:01 vid:10
Jan 27 02:06:18 Valve  INFO     dpid:1                Packet_in
src:00:00:00:00:00:01 dst:00:00:00:00:00:02 in_port:1 vid:10
Jan 27 02:06:18 Valve  INFO     dpid:1                Adding unicast flow
dl_dst:00:00:00:00:00:02 vid:10
Jan 27 02:06:18 Valve  INFO     dpid:1                Packet_in
src:00:00:00:00:00:02 dst:00:00:00:00:00:01 in_port:2 vid:10
Jan 27 02:06:18 Valve  INFO     dpid:1                Adding unicast flow
dl_dst:00:00:00:00:00:01 vid:10
```

Dump flows again to view differences.

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

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 RYU and Mininet

Look at the next Valve config file

```
root@mininet-vm:~/valve# cat valve.yaml-vlan
---

0000000000000001:
default:
type: untagged
vlans: [10]

1:
type: untagged
vlans: [10]

2:
type: untagged
vlans: [10]
```

```
3:
type: untagged
vlans: [20]

4:
type: untagged
vlans: [20]
```

Copy the config file into place

```
root@mininet-vm:~/valve# cp valve.yaml-flat valve.yaml
```

Start Mininet and Ryu once again

```
root@mininet-vm:~/valve# mn --topo=tree,1,10 --mac --controller=remote --switch
ovsk,protocols=OpenFlow13


root@mininet-vm:~/valve# ryu-manager --verbose ./valve.py
```

Dump flows again to view differences.

Look at the flow rules - Can you explain what is different this time?

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

Ping all the hosts

As you can see by the config file - the hosts on ports 3 and 4 have been moved to a separate vlan.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X h5 h6 h7 h8 h9 h10
h2 -> h1 X X h5 h6 h7 h8 h9 h10
h3 -> X X h4 X X X X X X
h4 -> X X h3 X X X X X X
h5 -> h1 h2 X X h6 h7 h8 h9 h10
h6 -> h1 h2 X X h5 h7 h8 h9 h10
h7 -> h1 h2 X X h5 h6 h8 h9 h10
h8 -> h1 h2 X X h5 h6 h7 h9 h10
h9 -> h1 h2 X X h5 h6 h7 h8 h10
h10 -> h1 h2 X X h5 h6 h7 h8 h9
*** Results: 35% dropped (58/90 received)
mininet>


#Shutdown RYU and Mininet

# Look at the next Valve config file


root@mininet-vm:~/valve# cat valve.yaml-multiple
```

```
---
all:
type: untagged
vlans: [10]

0000000000000001:
default:
type: untagged
vlans: [10]

0000000000000002:
default:
type: untagged
vlans: [10]

0000000000000003:
default:
type: untagged
vlans: [10]

0000000000000004:
default:
type: untagged
vlans: [10]

0000000000000005:
default:
type: untagged
vlans: [10]

0000000000000006:
default:
type: untagged
vlans: [10]

0000000000000007:
default:
type: untagged
vlans: [10]

0000000000000008:
default:
type: untagged
vlans: [10]
```

Copy the config file into place

```
root@mininet-vm:~/valve# cp valve.yaml-multiple valve.yaml
```

Start Mininet and Ryu once again

NOTE THAT THE MININET TOPOLOGY HAS CHANGED

```
root@mininet-vm:~/valve# mn --topo=tree,3,2 --mac --controller=remote --switch
ovsk,protocols=OpenFlow13


root@mininet-vm:~/valve# ryu-manager --verbose ./valve.py
```

Dump flows rules

```
Look at the flow rules - You can now see the flow rules on each of the switches.

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

Ping all the hosts

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 0% dropped (56/56 received)
mininet>
```

Dump flows rules again

Look at the flow rules - You can now see the flow rules on each of the switches.

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