Mininet/Openflow

Objectives

In this lab, you will start by learning the basics of running Mininet in a virtual machine. Mininet facilitates creating and manipulating Software Defined Networking components. Through mininet you will explore OpenFlow, which is an open interface for controlling the network elements through their forwarding tables. A network element may be converted into a switch, router or even an access points via low-level primitives defined in OpenFlow. This lab is your opportunity to gain hands-on experience with the platforms and debugging tools most useful for developing network control applications on OpenFlow.

- Access Mininet on the hosted virtual machine
- Run the Ryu controller with a sample application
- Use various commands to gain experience with OpenFlow control of OpenvSwitch

Network Topology

The topology we are using is similar to the one in the Openflow Tutorial (https://github.com/osrg/ryu/wiki/OpenFlow_Tutorial). It has three hosts named h1, h2 and h3 respectively. Each host has an Ethernet interface called h1-eth0, h2-eth0 and h3-eth0 respectively. The three hosts are connected through a switch names s1. The switch s1 has three ports named s1-eth1, s1-eth2 and s1-eth3. The controller is connected on the loopback interface (in real life this may or may not be the case, it means the switch and controller are built in a single box). The controller is identified as c0 and connected through port 6633.

```
     +---------------------------+
     |                           |
     |      C0 - Controller      |
     |                           |
     +-------------+-------------+
                   |
     +-------------+-------------+
     |                           |
     |      S1 - OpenFlow        |
     |          Switch           |
     |                           |
     +-+----------+----------+---+
     s1-eth0    s1-eth1    s1-eth2
      +            +            +
      |            |            |
      |            |            |
      v            v            v
h1-eth0         h2-eth0         h3-eth0
+-+--+          +-+--+          +-+--+
| H1 |          | H2 |          | H3 |
+----+          +----+          +----+
```

You will need a Number

The instructor should have given everyone a different number. This will dictate which virtual machine you will be using. If the instructor happened to forget, then this is the time to remind them.

Write this number down on a piece of paper.

Anytime this lab mentions substitute it with the number you have written down.

Connect to your SDN VM with SSH

Open a terminal window on your machine. If you don't know how to do this ask an instructor for help.

At the prompt type:

```
ssh mininet@10.10.0.<NUMBER>
```

When you see the warning that looks something like this:

```
The authenticity of host '10.10.0.XXX (10.10.0.XXX)' can't be established.
RSA key fingerprint is e8:05:43:d5:9a:4b:72:ad:c9:92:97:ca:df:32:86:ab.
Are you sure you want to continue connecting (yes/no)? yes

Type "yes" and press ENTER.
```

When prompted with "mininet@10.10.0.XXX's password:" enter 'mininet'.

That should be it. You are now connected to a terminal session on your machine.

```
$ ssh mininet@10.10.0.XXX
mininet@192.168.56.101's password:
Welcome to Ubuntu 13.04 (GNU/Linux 3.8.0-19-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '13.10' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Sep  3 08:22:23 2014
mininet@mininet-vm:~$
```

To log out you can type:

```
exit
```

Open Two SSH windows

You will need two windows open to your VM for this lab. Go ahead and open the second one now.

Become root

All of the actions in this exercise are done as "root", so if you are not root already type the following in **both** windows:

```
mininet@mininet-vm:~$ sudo bash
root@mininet-vm:~#
```

Starting the RYU Openflow controller

Ensure that no other controller is present

```
root@mininet-vm:~# killall controller
controller: no process found
root@mininet-vm:~#
```

Note that 'controller' is a simple OpenFlow reference controller implementation in linux. We want to ensure that this is not running before we start our own controller.

Clear all mininet components

```
root@mininet-vm:~# mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-
openflowd ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
***  Removing OVS datapathsovs-vsctl --timeout=1 list-br
ovs-vsctl del-br s1
ovs-vsctl del-br s2
ovs-vsctl del-br s3
ovs-vsctl del-br s4
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '(\w+-eth\w+)'
*** Cleanup complete.
root@mininet-vm:~#
```

Start the Ryu controller

```
root@mininet-vm:~# ryu-manager --verbose ./simple_switch_13.py
loading app ./simple_switch_13.py
loading app ryu.controller.ofp_handler
instantiating app ./simple_switch_13.py of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK SimpleSwitch13
  CONSUMES EventOFPSwitchFeatures
  CONSUMES EventOFPPacketIn
BRICK ofp_event
  PROVIDES EventOFPSwitchFeatures TO {'SimpleSwitch13': set(['config'])}
  PROVIDES EventOFPPacketIn TO {'SimpleSwitch13': set(['main'])}
  CONSUMES EventOFPHello
  CONSUMES EventOFPErrorMsg
  CONSUMES EventOFPEchoRequest
  CONSUMES EventOFPPortDescStatsReply
  CONSUMES EventOFPSwitchFeatures
```

Understanding simple_switch.py

We have now started the RYU controller with the simple_switch application. The simple switch keeps track of where the host with each MAC address is located and accordingly sends packets towards the destination and not flood all ports.

Packet logic:

```
Create a table called mac_to_port ;
If {packet_in to switch}
{    Parse packet to reveal src and dst MAC addr;
     Store in the dictionary the mapping between src_mac and the in_port;
     Lookup dst_mac in mac_to_port dict of switch s1 to find next hop;
        If { next hop is found}
             { create flow_mod ;
             send;
            }
       else
             flood all ports ≠ in_port;
```

Starting the Mininet environment

Start mininet with 3 hosts connected to 1 switch

In the other window

```
root@mininet-vm:~# mn --topo=tree,1,3 --mac --controller=remote --switch
ovsk,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet>
```

Monitor controller to ensure that the switch connects

In the RYU controller window you should see a message similar to the following to show that the switch has connected to the controller and has exchanged information about its capabilities.

```
connected socket:<eventlet.greenio.GreenSocket object at 0xa986c0c> address:
('127.0.0.1', 42733)
connected socket:<eventlet.greenio.GreenSocket object at 0xa986cec> address:
('127.0.0.1', 42734)
hello ev <ryu.controller.ofp_event.EventOFPHello object at 0xa9897ac>
move onto config mode
EVENT ofp_event->SimpleSwitch13 EventOFPSwitchFeatures
switch features ev version: 0x4 msg_type 0x6 xid 0xb15cb575
OFPSwitchFeatures(auxiliary_id=0,capabilities=71,datapath_id=1,n_buffers=256,n_t
ables=254)
move onto main mode
```

Dump flows on switch s1

A flow is the finest work unit of a switch. In Mininet, dpctl is a command that allows visibility and control over a single switch's flow table. It is especially useful for debugging, by viewing flow state and flow counters.

```
mininet> dpctl dump-flows -O OpenFlow13
*** s1 ------------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=2.481s, table=0, n_packets=0, n_bytes=0, priority=0
actions=FLOOD,CONTROLLER:64
mininet>
```

Passing packets

Start a ping from host h1 to host h2

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=5.10 ms
64 bytes from 10.0.0.2: icmp_req=2 ttl=64 time=0.238 ms
64 bytes from 10.0.0.2: icmp_req=3 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_req=4 ttl=64 time=0.051 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.051/1.360/5.100/2.160 ms
mininet>
```

Monitor new messages in the controller window

In the RYU controller window we want to ensure that we see the EventOFPPacketIn messages along with the controller telling us that it is adding unicast flows.

```
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
```

```
packet in from 00:00:00:00:00:01 port 1 to 00:00:00:00:00:02 on dpid 1
associate 00:00:00:00:00:01 with port 1 on dpid 1
packet in from 00:00:00:00:00:02 port 2 to 00:00:00:00:00:01 on dpid 1
associate 00:00:00:00:00:02 with port 2 on dpid 1
add unicast flow from 00:00:00:00:00:02 port 2 to 00:00:00:00:00:01 port 1 on
dpid 1
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
packet in from 00:00:00:00:00:01 port 1 to 00:00:00:00:00:02 on dpid 1
add unicast flow from 00:00:00:00:00:01 port 1 to 00:00:00:00:00:02 port 2 on
dpid 1
```

Dump flows again to view differences.

We can confirm that the unicast flows have been added by dumping the flow table on the switch

```
mininet> dpctl dump-flows -O OpenFlow13
*** s1 ------------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=38.044s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=37.044s, table=0, n_packets=3, n_bytes=238,
priority=100,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02
actions=output:2
 cookie=0x0, duration=38.043s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=2,dl_src=00:00:00:00:00:02,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=38.043s, table=0, n_packets=4, n_bytes=336,
priority=100,in_port=2,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01
actions=output:1
 cookie=0x0, duration=38.043s, table=0, n_packets=0, n_bytes=0,
priority=5,in_port=2,dl_src=00:00:00:00:00:02,dl_type=0x88cc actions=drop
 cookie=0x0, duration=38.043s, table=0, n_packets=0, n_bytes=0,
priority=5,in_port=1,dl_src=00:00:00:00:00:01,dl_type=0x88cc actions=drop
 cookie=0x0, duration=38.043s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=2,dl_src=00:00:00:00:00:02,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=38.044s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=73.001s, table=0, n_packets=3, n_bytes=294, priority=0
actions=FLOOD,CONTROLLER:64
mininet>
```

Running with more hosts.

Stop the current Mininet simulation

```
mininet> exit
*** Stopping 1 switches
s1 ...
*** Stopping 3 hosts
h1 h2 h3
*** Stopping 1 controllers
c0
*** Done
```

```
completed in 3.678 seconds
root@mininet-vm:~#
```

Start a new simulation with a few more hosts (10 hosts, 1 switch)

```
root@mininet-vm:~# 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.

```
mininet> dpctl dump-flows -O OpenFlow13
*** s1 ------------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=49.513s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=8,dl_src=00:00:00:00:00:08,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=50.102s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=49.874s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=5,dl_src=00:00:00:00:00:05,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=49.77s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=6,dl_src=00:00:00:00:00:06,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=49.699s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=7,dl_src=00:00:00:00:00:07,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=49.73s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=3,dl_src=00:00:00:00:00:03,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
...
 cookie=0x0, duration=50.101s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=50.185s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=4,dl_src=00:00:00:00:00:04,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=49.77s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=6,dl_src=00:00:00:00:00:06,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=49.573s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=10,dl_src=00:00:00:00:00:0a,dl_dst=01:00:00:00:00:00/01:00:0
0:00:00:00 actions=ALL
 cookie=0x0, duration=49.797s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=2,dl_src=00:00:00:00:00:02,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=49.461s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=9,dl_src=00:00:00:00:00:09,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=49.873s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=5,dl_src=00:00:00:00:00:05,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=50.391s, table=0, n_packets=15, n_bytes=1158, priority=0
actions=FLOOD,CONTROLLER:64
mininet>
```

Ping from h1 to h2 once again

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=0.585 ms
64 bytes from 10.0.0.2: icmp_req=2 ttl=64 time=0.319 ms
64 bytes from 10.0.0.2: icmp_req=3 ttl=64 time=0.063 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.063/0.322/0.585/0.213 ms
mininet>
```

Dump flows again to view differences.

Rootshell Window

```
mininet> dpctl dump-flows -O OpenFlow13
*** s1 ------------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=147.663s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=8,dl_src=00:00:00:00:00:08,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=148.252s, table=0, n_packets=1, n_bytes=42,
priority=10,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=148.024s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=5,dl_src=00:00:00:00:00:05,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=147.92s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=6,dl_src=00:00:00:00:00:06,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=147.849s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=7,dl_src=00:00:00:00:00:07,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
 cookie=0x0, duration=147.88s, table=0, n_packets=0, n_bytes=0,
priority=10,in_port=3,dl_src=00:00:00:00:00:03,dl_dst=ff:ff:ff:ff:ff:ff
actions=ALL
...
 cookie=0x0, duration=147.879s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=3,dl_src=00:00:00:00:00:03,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
```

```
 cookie=0x0, duration=147.848s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=7,dl_src=00:00:00:00:00:07,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=148.251s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=148.335s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=4,dl_src=00:00:00:00:00:04,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=147.92s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=6,dl_src=00:00:00:00:00:06,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=147.723s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=10,dl_src=00:00:00:00:00:0a,dl_dst=01:00:00:00:00:00/01:00:0
0:00:00:00 actions=ALL
 cookie=0x0, duration=147.947s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=2,dl_src=00:00:00:00:00:02,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=147.611s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=9,dl_src=00:00:00:00:00:09,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=148.023s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=5,dl_src=00:00:00:00:00:05,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=148.541s, table=0, n_packets=18, n_bytes=1396, priority=0
actions=FLOOD,CONTROLLER:64
```

Ping all hosts

Mininet Window

```
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>
```

Monitor new messages in the controller window

RYU Window

```
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
packet in from 00:00:00:00:00:09 port 9 to 00:00:00:00:00:08 on dpid 1
add unicast flow from 00:00:00:00:00:09 port 9 to 00:00:00:00:00:08 port 8 on
dpid 1
packet in from 00:00:00:00:00:08 port 8 to 00:00:00:00:00:09 on dpid 1
add unicast flow from 00:00:00:00:00:08 port 8 to 00:00:00:00:00:09 port 9 on
dpid 1
```

```
packet in from 00:00:00:00:00:09 port 9 to 00:00:00:00:00:08 on dpid 1
add unicast flow from 00:00:00:00:00:09 port 9 to 00:00:00:00:00:08 port 8 on
dpid 1
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
packet in from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:08 on dpid 1
add unicast flow from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:08 port 8 on
dpid 1
packet in from 00:00:00:00:00:08 port 8 to 00:00:00:00:00:0a on dpid 1
add unicast flow from 00:00:00:00:00:08 port 8 to 00:00:00:00:00:0a port 10 on
dpid 1
packet in from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:08 on dpid 1
add unicast flow from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:08 port 8 on
dpid 1
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
packet in from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:09 on dpid 1
add unicast flow from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:09 port 9 on
dpid 1
packet in from 00:00:00:00:00:09 port 9 to 00:00:00:00:00:0a on dpid 1
add unicast flow from 00:00:00:00:00:09 port 9 to 00:00:00:00:00:0a port 10 on
dpid 1
packet in from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:09 on dpid 1
add unicast flow from 00:00:00:00:00:0a port 10 to 00:00:00:00:00:09 port 9 on
dpid 1
```

Dump flows again to view differences.

```
mininet> dpctl dump-flows -O OpenFlow13
*** s1 ------------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=67.505s, table=0, n_packets=2, n_bytes=140,
priority=100,in_port=5,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:07
actions=output:7
 cookie=0x0, duration=67.614s, table=0, n_packets=2, n_bytes=140,
priority=100,in_port=2,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03
actions=output:3
 cookie=0x0, duration=67.619s, table=0, n_packets=2, n_bytes=140,
priority=100,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:0a
actions=output:10
 cookie=0x0, duration=67.644s, table=0, n_packets=2, n_bytes=140,
priority=100,in_port=4,dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01
actions=output:1
 ...
 cookie=0x0, duration=249.891s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=3,dl_src=00:00:00:00:00:03,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=249.86s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=7,dl_src=00:00:00:00:00:07,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=250.263s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=250.347s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=4,dl_src=00:00:00:00:00:04,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=249.932s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=6,dl_src=00:00:00:00:00:06,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=249.735s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=10,dl_src=00:00:00:00:00:0a,dl_dst=01:00:00:00:00:00/01:00:0
0:00:00:00 actions=ALL
 cookie=0x0, duration=249.959s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=2,dl_src=00:00:00:00:00:02,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=249.623s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=9,dl_src=00:00:00:00:00:09,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=250.035s, table=0, n_packets=4, n_bytes=300,
priority=10,in_port=5,dl_src=00:00:00:00:00:05,dl_dst=01:00:00:00:00:00/01:00:00
:00:00:00 actions=ALL
 cookie=0x0, duration=250.553s, table=0, n_packets=150, n_bytes=11868,
priority=0 actions=FLOOD,CONTROLLER:64
mininet>
```

Ping all hosts once again

Mininet Window

```
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>
```

Monitor new messages in the controller window

RYU Window

What happened that time?

Running a high bandwidth flow

Starting iperf between hosts

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h10
Waiting for iperf to start up...*** Results: ['5.52 Gbits/sec', '5.52
Gbits/sec']
mininet>
```

Dump flows to see the flows which match

```
mininet> dpctl dump-flows -O OpenFlow13
*** s1 ------------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
...
...
 cookie=0x0, duration=209.485s, table=0, n_packets=2384026, n_bytes=3609389036,
priority=100,in_port=1,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:0a
actions=output:10
...
...
 cookie=0x0, duration=209.485s, table=0, n_packets=27163, n_bytes=1792770,
priority=100,in_port=10,dl_src=00:00:00:00:00:0a,dl_dst=00:00:00:00:00:01
actions=output:1
...
...
 cookie=0x0, duration=392.419s, table=0, n_packets=150, n_bytes=11868,
priority=0 actions=FLOOD,CONTROLLER:64
```

Did any packets come to the controller? Where were most of the packets sent?