



---

# LAB4

---

ECE444/544



# Introduction

In this emulation lab, we are going to use the OpenFlow protocol analysis in Mininet.

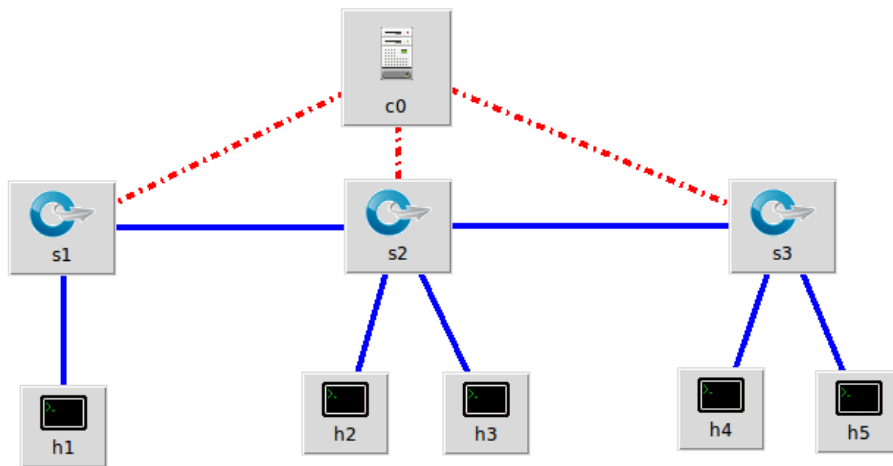
Generally, if you create a chart, all the flows are defaulted as all the ports are connected, no matter in or out. As long as the controller, switches and hosts are connected together. The switches will give the flow to the hosts, then the flows are given to the controller, again the controller gives the flows back to switches if the hosts from different switches want to be connected.

## Procedure

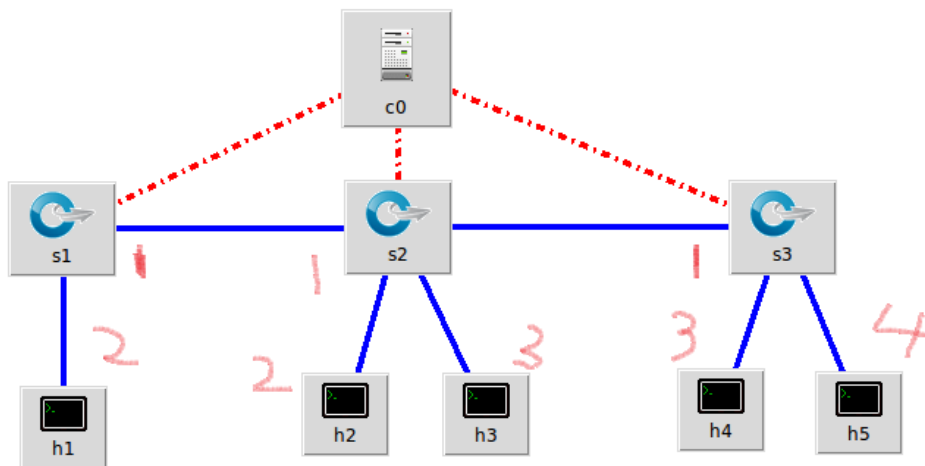
Before we formally start, here are the commands of the OpenFlow protocol analysis.

- Check the flow: `dpctl dump-flows`
- Add the flow: `dpctl add-flow in_port=2,actions=output:1`
- Delete all the flows in all switches: `dpctl del-flows`
- Delete the specific flow in all switches: `dpctl del-flows in_port=2`
- Delete the specific flow in the specific switches: `sh ovs-ofctl del-flows s1 in_port=2`

Then we draw a new schematic and set up the data.



Here are the ports:



And we analyze it.

```

root@ubuntu: /home/ubuntu/mininet
ting will prevent Mininet from quitting and will prevent you from starting the
network again during this session.

*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h5 -> h4 h2 h3 h1
h4 -> h5 h2 h3 h1
h2 -> h5 h4 h3 h1
h3 -> h5 h4 h2 h1
h1 -> h5 h4 h2 h3
*** Results: 0% dropped (20/20 received)
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 s1 s2 s3
mininet> net
h5 h5-eth0:s3-eth2
h4 h4-eth0:s3-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s2-eth2
h1 h1-eth0:s1-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth3
s2 lo: s2-eth1:h2-eth0 s2-eth2:h3-eth0 s2-eth3:s1-eth2 s2-eth4:s3-eth3
s3 lo: s3-eth1:h4-eth0 s3-eth2:h5-eth0 s3-eth3:s2-eth4

packet in 3 ee:5c:5d:da:e3:b8 33:33:00:00:00:02 3
packet in 1 02:53:f1:16:b0:14 33:33:00:00:00:02 2
packet in 3 b2:f6:66:7e:05:bd 33:33:00:00:00:fb 3
packet in 2 ee:5c:5d:da:e3:b8 33:33:00:00:00:fb 3
packet in 3 ee:5c:5d:da:e3:b8 33:33:00:00:00:fb 3
packet in 1 02:53:f1:16:b0:14 33:33:00:00:00:fb 2
packet in 2 de:77:d5:96:5b:eb ff:ff:ff:ff:ff:ff 4
packet in 1 de:77:d5:96:5b:eb ff:ff:ff:ff:ff:ff 2
packet in 2 ee:5c:5d:da:e3:b8 ff:ff:ff:ff:ff:ff 3
packet in 3 ee:5c:5d:da:e3:b8 ff:ff:ff:ff:ff:ff 3
packet in 2 de:77:d5:96:5b:eb 33:33:00:00:00:fb 4
packet in 1 de:77:d5:96:5b:eb 33:33:00:00:00:fb 2
packet in 3 b2:f6:66:7e:05:bd 33:33:00:00:00:fb 3
packet in 2 ee:5c:5d:da:e3:b8 33:33:00:00:00:fb 3
packet in 3 ee:5c:5d:da:e3:b8 33:33:00:00:00:fb 3
packet in 3 b2:f6:66:7e:05:bd ff:ff:ff:ff:ff:ff 3
packet in 1 02:53:f1:16:b0:14 33:33:00:00:00:fb 2
packet in 2 de:77:d5:96:5b:eb 33:33:00:00:00:02 4
packet in 1 de:77:d5:96:5b:eb 33:33:00:00:00:02 2
packet in 3 b2:f6:66:7e:05:bd 33:33:00:00:00:02 3
packet in 2 ee:5c:5d:da:e3:b8 33:33:00:00:00:02 3
packet in 3 ee:5c:5d:da:e3:b8 33:33:00:00:00:02 3
packet in 1 02:53:f1:16:b0:14 33:33:00:00:00:02 2

```

Now all the pings are connected (0% dropped).

Let's check the flow by typing in 'dpctl dump-flows'

```

mininet> pingall
*** Ping: testing ping reachability
h5 -> h3 h1 h4 h2
h3 -> h5 h1 h4 h2
h1 -> h5 h3 h4 h2
h4 -> h5 h3 h1 h2
h2 -> h5 h3 h1 h4
*** Results: 0% dropped (20/20 received)
mininet> dpctl dump-flows
*** s2 -----
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=31.17s, table=0, n_packets=3, n_bytes=238, idle_age=26, in
_port=1,dl_src=9a:d5:43:72:b0:b0,dl_dst=a2:06:db:58:6b:a9 actions=output:4
cookie=0x0, duration=31.037s, table=0, n_packets=2, n_bytes=140, idle_age=26, i
n_port=3,dl_src=76:15:fe:a3:8f:fe,dl_dst=9a:d5:43:72:b0:b0 actions=output:1
cookie=0x0, duration=31.011s, table=0, n_packets=3, n_bytes=238, idle_age=25, i
n_port=1,dl_src=9a:d5:43:72:b0:b0,dl_dst=e2:fc:9e:92:c2:95 actions=output:4
cookie=0x0, duration=31.098s, table=0, n_packets=3, n_bytes=238, idle_age=26, i
n_port=1,dl_src=9a:d5:43:72:b0:b0,dl_dst=c6:f0:4d:8c:44:e3 actions=output:2
cookie=0x0, duration=31.136s, table=0, n_packets=2, n_bytes=140, idle_age=26, i
n_port=2,dl_src=c6:f0:4d:8c:44:e3,dl_dst=76:15:fe:a3:8f:fe actions=output:3
cookie=0x0, duration=31.163s, table=0, n_packets=2, n_bytes=140, idle_age=26, i
n_port=4,dl_src=a2:06:db:58:6b:a9,dl_dst=9a:d5:43:72:b0:b0 actions=output:1
cookie=0x0, duration=31.043s, table=0, n_packets=3, n_bytes=238, idle_age=26, i

```

All of them are connected. Let's delete all of them. And check again, you'll find there is nothing is S2, S1 and S3. Type in 'dpctl del-flows' and 'dpctl dump-flows'

```

mininet> dpctl del-flows
*** s2 -----
*** s1 -----
*** s3 -----
mininet> dpctl dump-flows
*** s2 -----
NXST_FLOW reply (xid=0x4):
*** s1 -----
NXST_FLOW reply (xid=0x4):
*** s3 -----
NXST_FLOW reply (xid=0x4):
mininet>

```

Now you can edit the flows you want, here I add the input port 1, 2 and output port 1,2.

Use this 'dpctl add-flow in\_port=2 ,actions=output:1'

The port number is random, you can choose whatever you want.

Don't forget to the check the conditions of the flows.

```

mininet> dpctl add-flow in_port=2,actions=output:2
*** s2 -----
*** s1 -----
*** s3 -----
mininet> dpctl dump-flows
*** s2 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=176.253s, table=0, n_packets=11, n_bytes=910, idle_age=168
, in_port=3,dl_src=76:15:fe:a3:8f:fe,dl_dst=9a:d5:43:72:b0:b0 actions=output:1
  cookie=0x0, duration=176.252s, table=0, n_packets=69, n_bytes=3346, idle_age=11
1, in_port=1,dl_src=9a:d5:43:72:b0:b0,dl_dst=76:15:fe:a3:8f:fe actions=output:3
  cookie=0x0, duration=6.432s, table=0, n_packets=0, n_bytes=0, idle_age=6, in_po
rt=2 actions=output:2
*** s1 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=176.257s, table=0, n_packets=11, n_bytes=910, idle_age=168

```

```

mininet> dpctl add-flow in_port=2,actions=output:1
*** s2 -----
*** s1 -----
*** s3 -----
mininet> dpctl dump-flows
*** s2 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=331.549s, table=0, n_packets=11, n_bytes=910, idle_age=323
, in_port=3,dl_src=76:15:fe:a3:8f:fe,dl_dst=9a:d5:43:72:b0:b0 actions=output:1
  cookie=0x0, duration=331.548s, table=0, n_packets=93, n_bytes=4354, idle_age=12
5, in_port=1,dl_src=9a:d5:43:72:b0:b0,dl_dst=76:15:fe:a3:8f:fe actions=output:3
  cookie=0x0, duration=2.508s, table=0, n_packets=0, n_bytes=0, idle_age=2, in_po
rt=2 actions=output:1
*** s1 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=331.553s, table=0, n_packets=11, n_bytes=910, idle_age=323
, in_port=1,dl_src=76:15:fe:a3:8f:fe,dl_dst=9a:d5:43:72:b0:b0 actions=output:2

```

Now we use 'h1 ping h2' to check the connection between h1 and h2.

```
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.  
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=5.28 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.122 ms  
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.057 ms  
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.084 ms  
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.037 ms  
^C  
--- 10.0.0.2 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4000ms  
rtt min/avg/max/mdev = 0.037/1.117/5.288/2.085 ms  
mininet>
```

H1 and H2 are connected because we've added this flows by hand.

```
mininet> h1 ping h3  
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.  
From 10.0.0.1 icmp_seq=9 Destination Host Unreachable  
From 10.0.0.1 icmp_seq=10 Destination Host Unreachable  
From 10.0.0.1 icmp_seq=11 Destination Host Unreachable  
^C  
--- 10.0.0.3 ping statistics ---  
13 packets transmitted, 0 received, +3 errors, 100% packet loss, time 12082ms  
pipe 3  
mininet>
```

However, we didn't add the other ports, so h1 failed to connect with h3.

'sh ovs-ofctl del-flows s2 in\_port=2'.

```
mininet> sh ovs-ofctl del-flows s2 in_port=2  
mininet> dpctl dump-flows  
*** s2 -----  
NXST_FLOW reply (xid=0x4):  
 cookie=0x0, duration=894.684s, table=0, n_packets=17, n_bytes=1442, idle_age=55  
5, in_port=3, dl_src=76:15:fe:a3:8f:fe, dl_dst=9a:d5:43:72:b0:b0 actions=output:1  
 cookie=0x0, duration=894.683s, table=0, n_packets=100, n_bytes=4928, idle_age=5  
55, in_port=1, dl_src=9a:d5:43:72:b0:b0, dl_dst=76:15:fe:a3:8f:fe actions=output:3  
 cookie=0x0, duration=442.229s, table=0, n_packets=10, n_bytes=812, idle_age=435  
, in_port=3, dl_src=76:15:fe:a3:8f:fe, dl_dst=c6:f0:4d:8c:44:e3 actions=output:2
```

All the input port of 2 have gone.

Finally, let's check the ping conditions in all the hosts. Type in 'pingall'

```
mininet> pingall  
*** Ping: testing ping reachability  
h5 -> X X h4 X  
h3 -> X h1 h4 h2  
h1 -> X h3 h4 h2  
h4 -> h5 h3 h1 h2  
h2 -> X h3 h1 h4  
*** Results: 30% dropped (14/20 received)  
mininet>
```

# Summary

- I hope these 5 labs can help you get familiar with the mininet and the ECE44/544 labs.