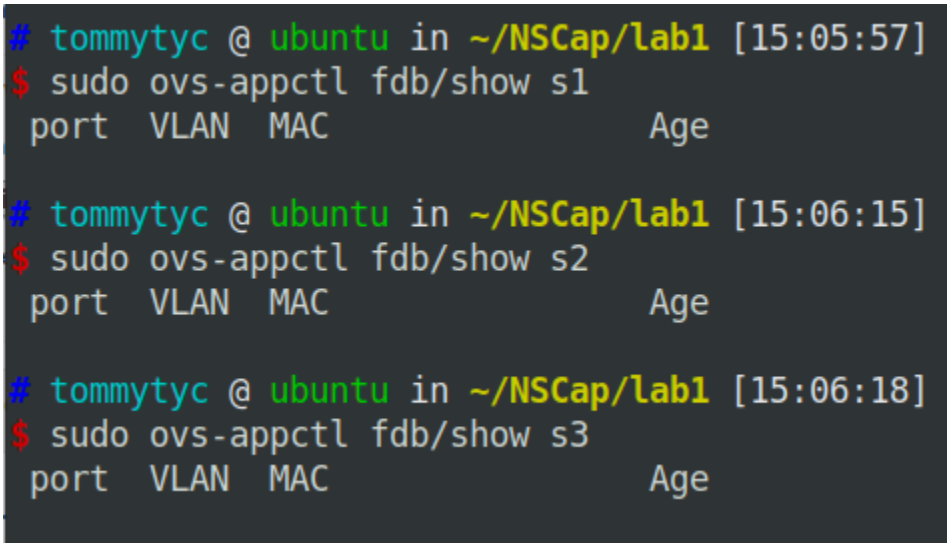
NSCap Lab Report

Lab 1

Part 1

1. Show the flushed MAC table of all switch
   - 

2. Show that h4 is trying to find out h1's MAC address
   - 

   - h4 sends ARP request broadcast to find out h1's MAC address.
3. Show that h1 is trying to find out h4's MAC address
   - 

   - h1 sends ARP request broadcast to find out h4's MAC address.
4. The first ping takes longer time than the others, because at first all of the switch have no idea the MAC address of each device. It costs a few time for each switch to "learn" the match of IP address and MAC address of each device.
5. Show the switch tables and identify the entries constitutethe path of Ping

```
# tommytyc @ ubuntu in ~/NSCap/lab1 [15:06:20]
$ sudo ovs-appctl fdb/show s1
 port  VLAN  MAC                Age
    2     0  5a:64:f7:a4:7c:0d   39
    1     0  9a:ed:a4:4b:c7:85   39

# tommytyc @ ubuntu in ~/NSCap/lab1 [15:17:21]
$ sudo ovs-appctl fdb/show s2
 port  VLAN  MAC                Age
    1     0  5a:64:f7:a4:7c:0d  125
    2     0  9a:ed:a4:4b:c7:85  125

# tommytyc @ ubuntu in ~/NSCap/lab1 [15:18:46]
$ sudo ovs-appctl fdb/show s3
 port  VLAN  MAC                Age
    3     0  9a:ed:a4:4b:c7:85  131
    1     0  5a:64:f7:a4:7c:0d  131
```

- h1 -> s1 port2 -> s1 port1 -> s2 port1 -> s2 port2 -> s3 port1 -> s3 port3 -> h4

Part 2

1. h1 can not successfully ping h4.

```
mininet> h1 ping h4 -c 5
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4100ms
```

2. h1 can succexxfully ping h4.

```
mininet> h1 ping h4 -c 5
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.908 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.081 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.353 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.092 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.293 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4101ms
rtt min/avg/max/mdev = 0.081/0.345/0.908/0.301 ms
```

3. s1 MAC table
   - before STP enable

```
# tommytyc @ ubuntu in ~/NSCap/lab1 [17:49:45]
$ sudo ovs-appctl fdb/show s1
 port  VLAN  MAC                Age
    2     0  16:51:2f:1d:94:2d    0
    2     0  ae:79:c8:57:18:9c    0
    2     0  be:93:ea:c5:01:0e    0
    1     0  16:28:c5:79:8d:40    0
    1     0  6e:63:74:bf:07:b4    0
    2     0  ae:28:2b:e9:a8:31    0
    1     0  46:ab:43:11:73:dc    0
    1     0  72:33:db:c2:f7:ec    0
    1     0  0e:cc:b1:84:2e:28    0
    1     0  3e:c8:0c:9a:48:48    0
    2     0  36:83:2a:1b:85:ef    0
    2     0  6a:d3:dd:2d:4b:97    0
```

- (so many switch info exchange despite no command being given…

- after STP enable

```
# tommytyc @ ubuntu in ~/NSCap/lab1 [17:54:24]
$ sudo ovs-appctl fdb/show s1
 port  VLAN  MAC                Age
    4     0  ee:00:a4:57:c7:f6   28
    2     0  86:c3:2e:f5:0c:c8   20
    2     0  5a:b0:a1:73:ad:97   18
    2     0  92:4e:61:42:78:bb   17
    2     0  de:7e:3d:30:fe:d3   17
    3     0  ee:68:d9:ea:a9:04   13
    2     0  8e:4b:cb:33:85:b3   13
    2     0  6a:b8:32:fa:a8:45    7
```

- At first the STP is not enabled, and the ARP request comes from h1 will be broadcasted in the switch loop from s1 to each of the other switches.
- After we enable the STP option, the switches will come up with a way to break down the loop and end up the broadcast storm.

4. I've learned about the MAC learning table and STP protocol.
   - MAC learning table can be observed in part 1. The first time h1 ping h4 will take a few time to send ARP request to find out the MAC address of h4, and the MAC address of each device will be recorded in the MAC learning table of switches.
   - STP protocol can break down the switch loop to avoid broadcast storm, which is caused by the ARP request sent in a switch topology with loop inside.