Mininet:

```
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
C0
*** Starting 1 switches
s1 ..
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=264787>
<Host h2: h2-eth0:10.0.0.2 pid=264792>
<Host h3: h3-eth0:10.0.0.3 pid=264794>
<Host h4: h4-eth0:10.0.0.4 pid=264796>
<0VSSwitch{'protocols': 'OpenFlow13'} s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:Non</pre>
e,s1-eth3:None,s1-eth4:None pid=264801>
<RemoteController{'ip': '127.0.0.1'} c0: 127.0.0.1:6653 pid=264781>
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s1-eth3 (OK OK)
h4-eth0<->s1-eth4 (OK OK)
mininet>
```

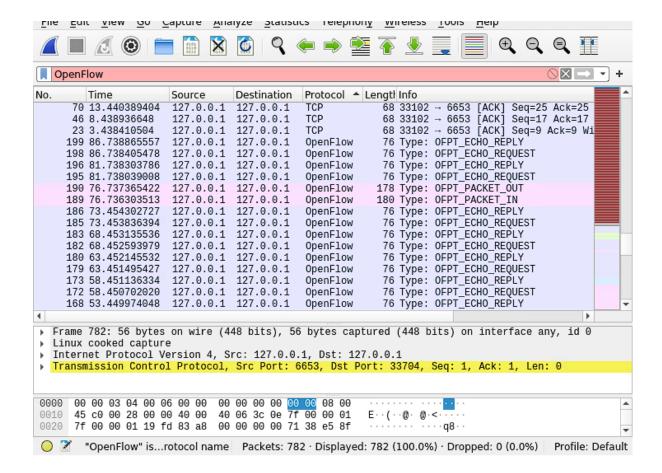
Controller and Openflow capture:

```
student@VM2:~$ sudo ovs-vsctl show
[sudo] password for student:
91003b0f-45e5-4b62-b5a6-287898f1e785
    Bridge s1
        Controller "ptcp:6654"
        Controller "tcp:127.0.0.1:6653"
            is connected: true
        fail mode: secure
        Port s1-eth3
            Interface s1-eth3
        Port s1-eth2
            Interface s1-eth2
 Ubuntu Software - eth4
            Interface s1-eth4
        Port s1-eth1
            Interface s1-eth1
        Port s1
            Interface s1
                type: internal
    ovs_version: "2.13.3"
```

```
cookie=0x0, duration=404.304s, table=0, n_packets=44, n_bytes=3464, priority=0
 actions=CONTROLLER:65535
 actions=controller:05535
student@VM2:~$ sudo ovs-ofctl -O OpenFlow13 dump-flows s1
cookie=0x0, duration=730.245s, table=0, n_packets=3, n_bytes=238, priority=1,i
n_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=outp
ut:"s1-eth1"
cookie=0x0, duration=730.241s, table=0, n_packets=2, n_bytes=140, priority=1,i
n port="s1-eth1",dl src=00:00:00:00:00:01,dl dst=00:00:00:00:00:02 actions=outp
ut:"s1-eth2"
cookie=0x0, duration=730.234s, table=0, n_packets=3, n_bytes=238, priority=1,i
n port="s1-eth3",dl src=00:00:00:00:00:03,dl dst=00:00:00:00:00:01 actions=outp
ut:"s1-eth1"
cookie=0x0, duration=730.233s, table=0, n_packets=2, n_bytes=140, priority=1,i
n port="s1-eth1",dl src=00:00:00:00:00:01,dl dst=00:00:00:00:00:03 actions=outp
ut:"s1-eth3"
cookie=0x0, duration=730.229s, table=0, n_packets=3, n_bytes=238, priority=1,i
n port="s1-eth4",dl src=00:00:00:00:00:04,dl dst=00:00:00:00:00:01 actions=outp
ut:"s1-eth1"
cookie=0x0, duration=730.227s, table=0, n packets=2, n bytes=140, priority=1,i
n port="s1-eth1",dl src=00:00:00:00:00:01,dl dst=00:00:00:00:00:04 actions=outp
ut:"s1-eth4"
cookie=0x0, duration=730.219s, table=0, n_packets=3, n_bytes=238, priority=1,i
n_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02 actions=outp
ut:"s1-eth2'
cookie=0x0, duration=730.218s, table=0, n_packets=2, n_bytes=140, priority=1,i
n_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03 actions=outp
ut:"s1-eth3"
cookie=0x0, duration=730.213s, table=0, n_packets=3, n_bytes=238, priority=1,i
n_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=outp
ut:"s1-eth2"
 cookie=0x0, duration=730.212s, table=0, n_packets=2, n_bytes=140, priority=1,i
n port="s1-eth2".dl src=00:00:00:00:00:02.dl dst=00:00:00:00:00:04 actions=outp
ut:"s1-eth4"
cookie=0x0, duration=730.202s, table=0, n packets=3, n bytes=238, priority=1,i
n_port="s1-eth4",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:00 actions=outp
ut:"s1-eth3"
cookie=0x0, duration=730.200s, table=0, n_packets=2, n_bytes=140, priority=1,i
n_port="s1-eth3",dl_src=00:00:00:00:00:03,dl dst=00:00:00:00:00:04 actions=outp
ut:"s1-eth4"
cookie=0x0, duration=1224.527s, table=0, n packets=70, n bytes=5116, priority=
0 actions=CONTROLLER:65535
student@VM2:~S
```

student@VM2:~\$ sudo ovs-ofctl -O OpenFlow13 dump-flows s1

CONTINUE TO NEXT PAGE PLEASE



Quick check on subnet knowledge: you have 2 VMs in vCloud as you created from template library. How can you identify from the ip that they are located in 2 different subnets?

A: We can use the IP address of the VMs to identify which subnet they are located in by looking at the subnet mask. The subnet mask is used to divide the IP address into two parts: the network address and the host address. By comparing the subnet mask of each VM's IP address, you can determine which subnet they belong to. For example, if the subnet mask for one VM is 255.255.255.0 and the subnet mask for the other VM is 255.255.0.0, we can say that the first VM is located in a subnet with a smaller network and the second VM is located in a subnet with a larger network.