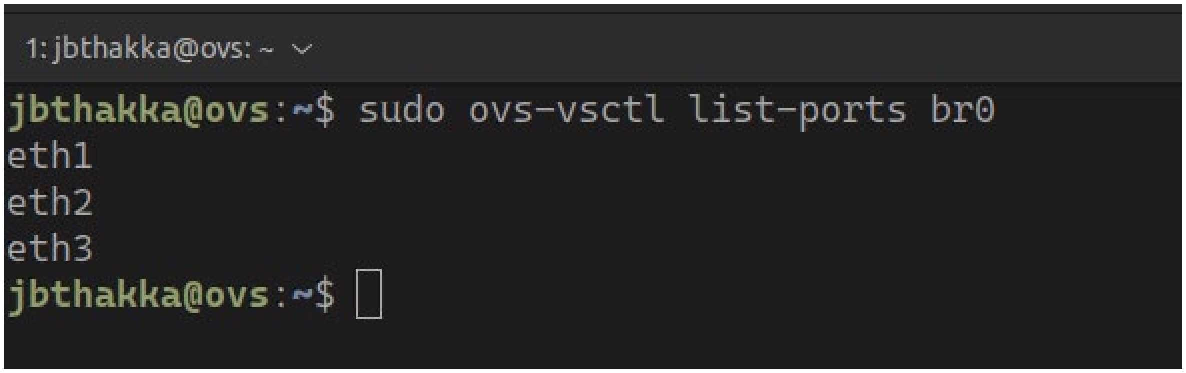
# CSE434 Lab 4 Report
# Group 82

## Group Members:
## Jeet Thakkar
## Wei Hng Yeo

**Exercise 1.1**



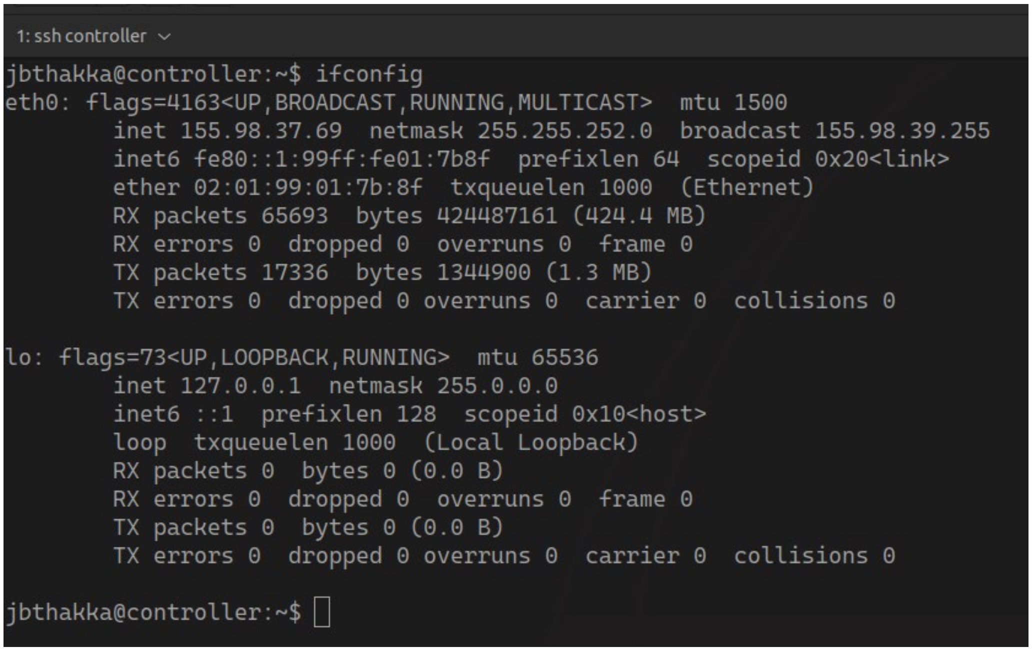
1: jbthakka@ovs: ~

```
jbthakka@ovs:~$ sudo ovs-vsctl list-ports br0
eth1
eth2
eth3
jbthakka@ovs:~$ 
```

Software Switch Configuration

From the above screenshot, we can see the data interfaces for the three hosts are are added to the bridge created on the software switch.

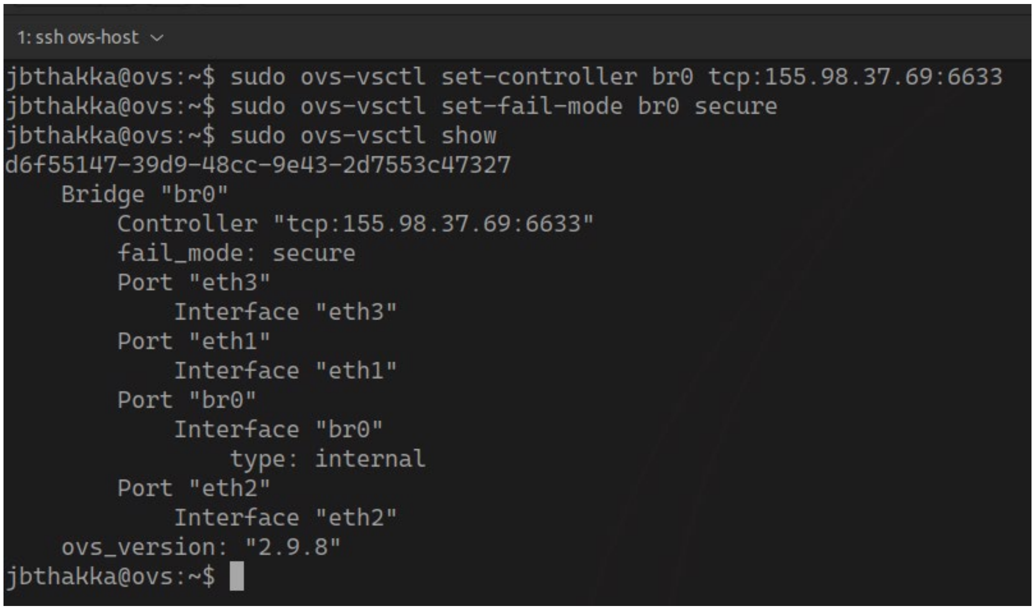
**Exercise 1.2**



```
jbthakka@controller:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 155.98.37.69  netmask 255.255.252.0  broadcast 155.98.39.255
        inet6 fe80::1:99ff:fe01:7b8f  prefixlen 64  scopeid 0x20<link>
        ether 02:01:99:01:7b:8f  txqueuelen 1000  (Ethernet)
        RX packets 65693  bytes 424487161 (424.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17336  bytes 1344900 (1.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

jbthakka@controller:~$ 
```

Ifconfig of Controller

The above screenshot shows the ifconfig of the controller.



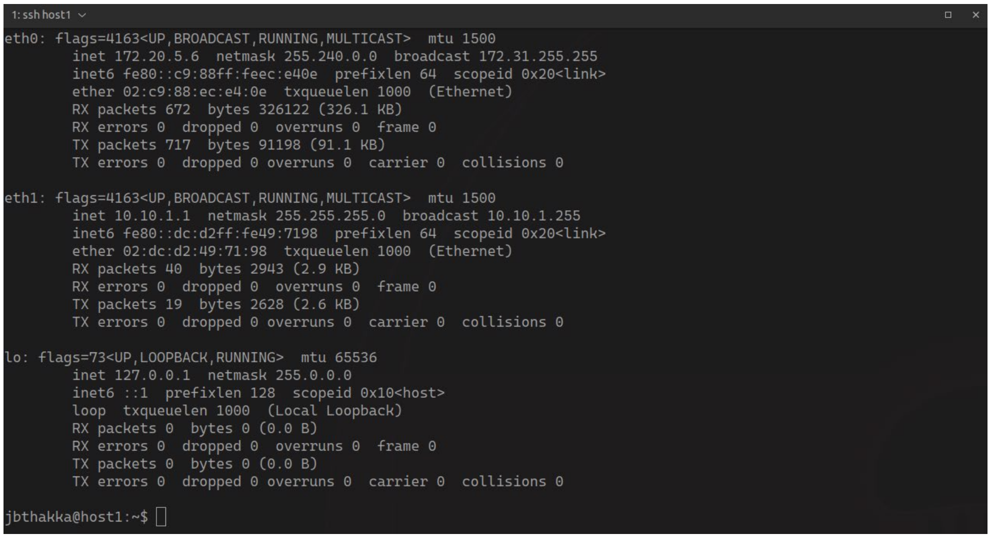
```
jbthakka@ovs:~$ sudo ovs-vsctl set-controller br0 tcp:155.98.37.69:6633
jbthakka@ovs:~$ sudo ovs-vsctl set-fail-mode br0 secure
jbthakka@ovs:~$ sudo ovs-vsctl show
d6f55147-39d9-48cc-9e43-2d7553c47327
    Bridge "br0"
        Controller "tcp:155.98.37.69:6633"
        fail_mode: secure
        Port "eth3"
            Interface "eth3"
        Port "eth1"
            Interface "eth1"
        Port "br0"
            Interface "br0"
                type: internal
        Port "eth2"
            Interface "eth2"
    ovs_version: "2.9.8"
jbthakka@ovs:~$ 
```

OVS settings shown

The above screenshot shows that the switch has been successfully pointed to the controller using the IP address, which corresponds to the IP address of the controller we’ve shown  in the previous ifconfig screenshot.

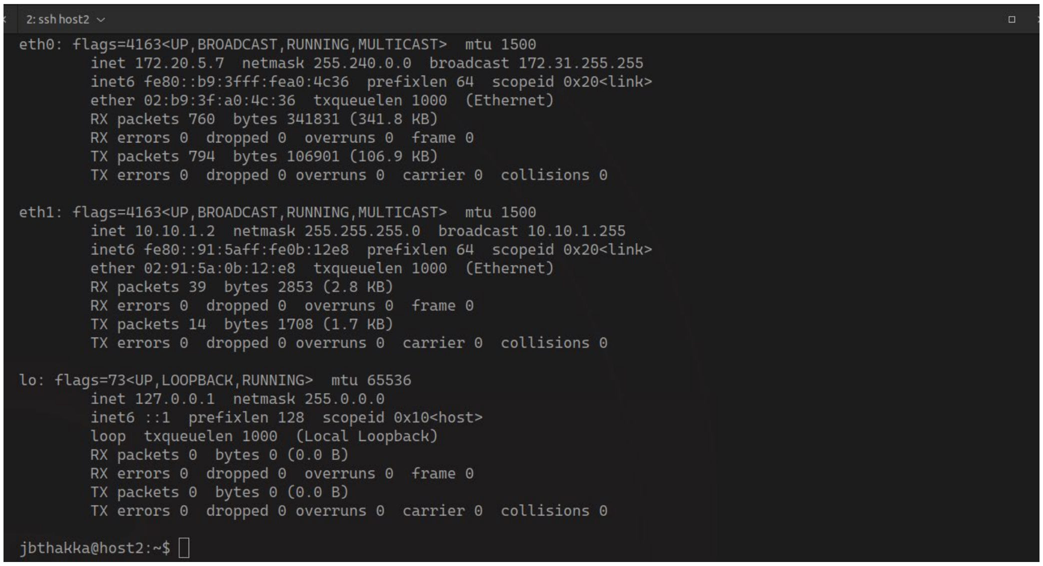
**Exercise 1.3**

Host 1: IP and MAC address



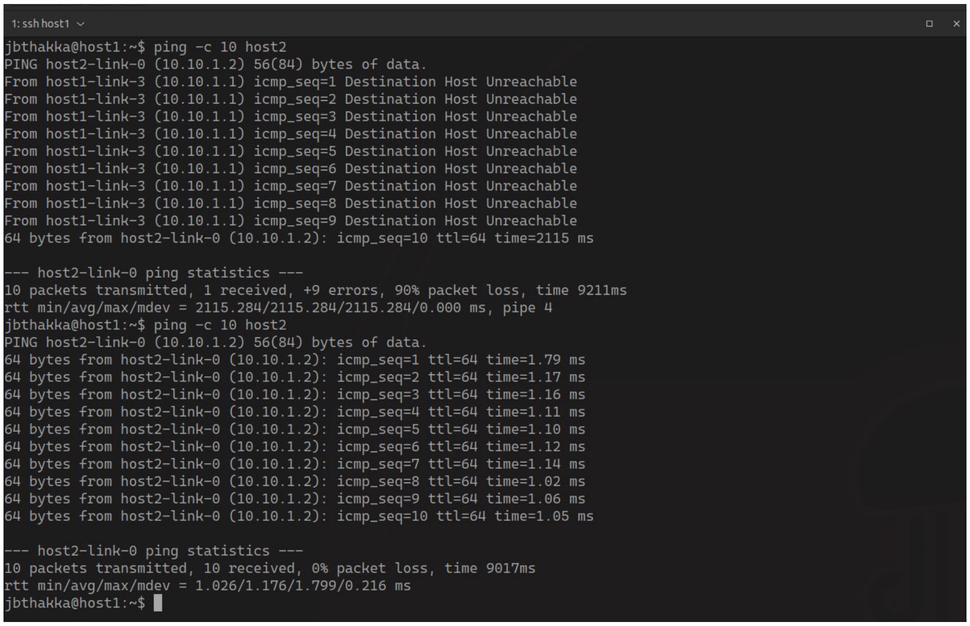
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.20.5.6  netmask 255.240.0.0  broadcast 172.31.255.255
        inet6 fe80::c9:88ff:feec:e40e  prefixlen 64  scopeid 0x20<link>
        ether 02:c9:88:ec:e4:0e  txqueuelen 1000  (Ethernet)
        RX packets 672  bytes 326122 (326.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 717  bytes 91198 (91.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.1.1  netmask 255.255.255.0  broadcast 10.10.1.255
        inet6 fe80::dc:d2ff:fe49:7198  prefixlen 64  scopeid 0x20<link>
        ether 02:dc:d2:49:71:98  txqueuelen 1000  (Ethernet)
        RX packets 40  bytes 2943 (2.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 19  bytes 2628 (2.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

jbthakka@host1:~$ 
```

Host 2: IP and MAC address



```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.20.5.7  netmask 255.240.0.0  broadcast 172.31.255.255
        inet6 fe80::b9:3fff:fea0:4c36  prefixlen 64  scopeid 0x20<link>
        ether 02:b9:3f:a0:4c:36  txqueuelen 1000  (Ethernet)
        RX packets 760  bytes 341831 (341.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 794  bytes 106901 (106.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.1.2  netmask 255.255.255.0  broadcast 10.10.1.255
        inet6 fe80::91:5aff:fe0b:12e8  prefixlen 64  scopeid 0x20<link>
        ether 02:91:5a:0b:12:e8  txqueuelen 1000  (Ethernet)
        RX packets 39  bytes 2853 (2.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1708 (1.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

jbthakka@host2:~$ 
```

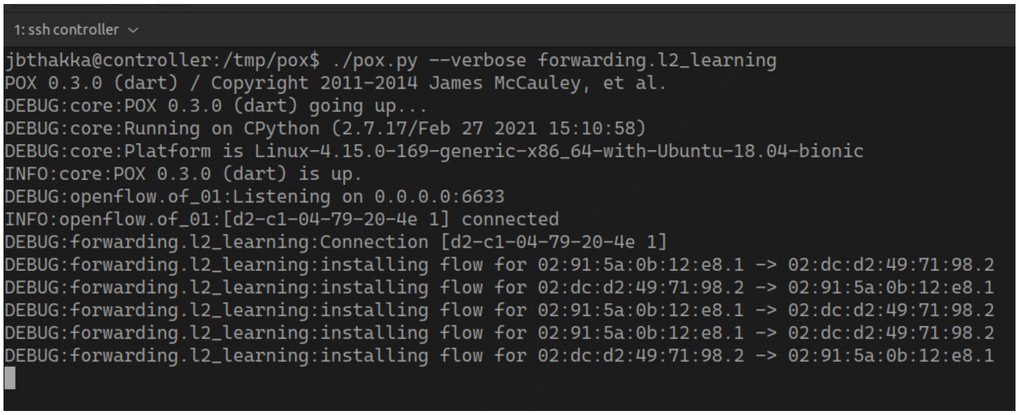
The ping from host 1 to host 2 shown below fails initially since the controller hasn’t started running yet. Once the controller starts running, the ping becomes successful because pox is initialized, and it sees that the software switch has received an ARP request from hostA for hostB - then it install the correct flows which make it possible for hostA to communicate with hostB.



```
jbthakka@host1:~$ ping -c 10 host2
PING host2-link-0 (10.10.1.2) 56(84) bytes of data.
From host1-link-3 (10.10.1.1) icmp_seq=1 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=2 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=3 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=4 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=5 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=6 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=7 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=8 Destination Host Unreachable
From host1-link-3 (10.10.1.1) icmp_seq=9 Destination Host Unreachable
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=10 ttl=64 time=2115 ms

--- host2-link-0 ping statistics ---
10 packets transmitted, 1 received, +9 errors, 90% packet loss, time 9211ms
rtt min/avg/max/mdev = 2115.284/2115.284/2115.284/0.000 ms, pipe 4
jbthakka@host1:~$ ping -c 10 host2
PING host2-link-0 (10.10.1.2) 56(84) bytes of data.
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=1 ttl=64 time=1.79 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=2 ttl=64 time=1.17 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=3 ttl=64 time=1.16 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=4 ttl=64 time=1.11 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=5 ttl=64 time=1.10 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=6 ttl=64 time=1.12 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=7 ttl=64 time=1.14 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=8 ttl=64 time=1.02 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=9 ttl=64 time=1.06 ms
64 bytes from host2-link-0 (10.10.1.2): icmp_seq=10 ttl=64 time=1.05 ms

--- host2-link-0 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9017ms
rtt min/avg/max/mdev = 1.026/1.176/1.799/0.216 ms
jbthakka@host1:~$ 
```

Ping output between host1 and host2

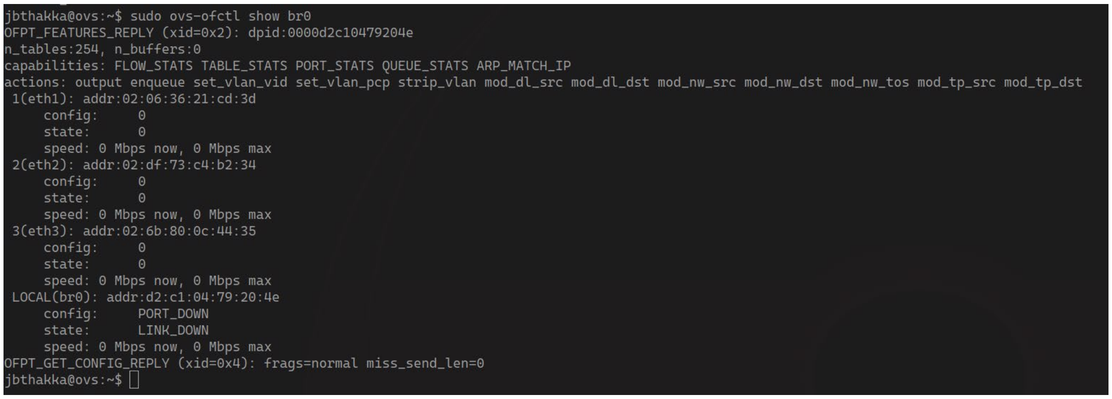
Controller pox initialization and output is shown below. The controller is installing flows on the software switch that allow traffic from from host 1 to host 2 and vice versa.



```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose forwarding.l2_learning
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:forwarding.l2_learning:Connection [d2-c1-04-79-20-4e 1]
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
```

## Exercise 1.4

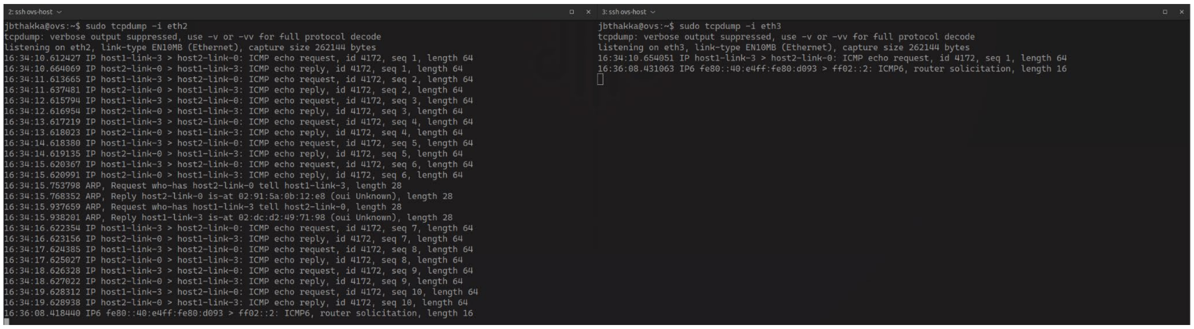
Manifest from the openvswitch



```
jbthakka@ovs:~$ sudo ovs-ofctl show br0
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000d2c10479204e
n_tables:254, n_buffers:0
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: output enqueue set_vlan_vid set_vlan_pcp strip_vlan mod_dl_src mod_dl_dst mod_nw_src mod_nw_dst mod_nw_tos mod_tp_src mod_tp_dst
 1(eth1): addr:02:06:36:21:cd:3d
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
 2(eth2): addr:02:df:73:c4:b2:34
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
 3(eth3): addr:02:6b:80:0c:44:35
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
 LOCAL(br0): addr:d2:c1:04:79:20:4e
     config:     PORT_DOWN
     state:      LINK_DOWN
     speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
jbthakka@ovs:~$
```

Manifest

MAC addresses:
eth1@OVS - 02:06:36:21:cd:3d - 1 (eth1)
eth2@OVS - 02:df:73:c4:b2:34 - 2 (eth2)
eth3@OVS - 02:6b:80:0c:44:35 - 3 (eth3)
eth1@host1 - 02:dc:d2:49:71:98
eth1@host2 - 02:91:5a:0b:12:e8
eth3@host3 - 02:b0:83:6c:35:fb

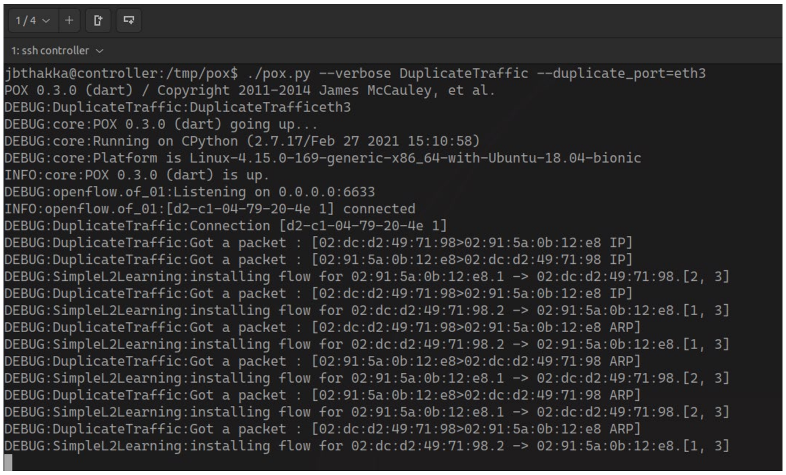


```
jbthakka@ovs:~$ sudo tcpdump -i eth2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
16:34:10.612427 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 1, length 64
16:34:10.664069 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 1, length 64
16:34:11.613665 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 2, length 64
16:34:11.637481 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 2, length 64
16:34:12.615794 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 3, length 64
16:34:12.616954 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 3, length 64
16:34:13.617219 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 4, length 64
16:34:13.618023 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 4, length 64
16:34:14.618380 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 5, length 64
16:34:14.619135 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 5, length 64
16:34:15.620367 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 6, length 64
16:34:15.620991 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 6, length 64
16:34:15.753798 ARP, Request who-has host2-link-0 tell host1-link-3, length 28
16:34:15.768352 ARP, Reply host2-link-0 is-at 02:91:5a:0b:12:e8 (oui Unknown), length 28
16:34:15.937659 ARP, Request who-has host1-link-3 tell host2-link-0, length 28
16:34:15.938201 ARP, Reply host1-link-3 is-at 02:dc:d2:49:71:98 (oui Unknown), length 28
16:34:16.622354 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 7, length 64
16:34:16.623156 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 7, length 64
16:34:17.624385 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 8, length 64
16:34:17.625027 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 8, length 64
16:34:18.626328 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 9, length 64
16:34:18.627022 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 9, length 64
16:34:19.628312 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 10, length 64
16:34:19.628938 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4172, seq 10, length 64
16:36:08.418440 IP6 fe80::40:e4ff:fe80:d093 > ff02::2: ICMP6, router solicitation, length 16
```

```
jbthakka@ovs:~$ sudo tcpdump -i eth3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth3, link-type EN10MB (Ethernet), capture size 262144 bytes
16:34:10.654051 IP host1-link-3 > host2-link-0: ICMP echo request, id 4172, seq 1, length 64
16:36:08.431063 IP6 fe80::40:e4ff:fe80:d093 > ff02::2: ICMP6, router solicitation, length 16
```

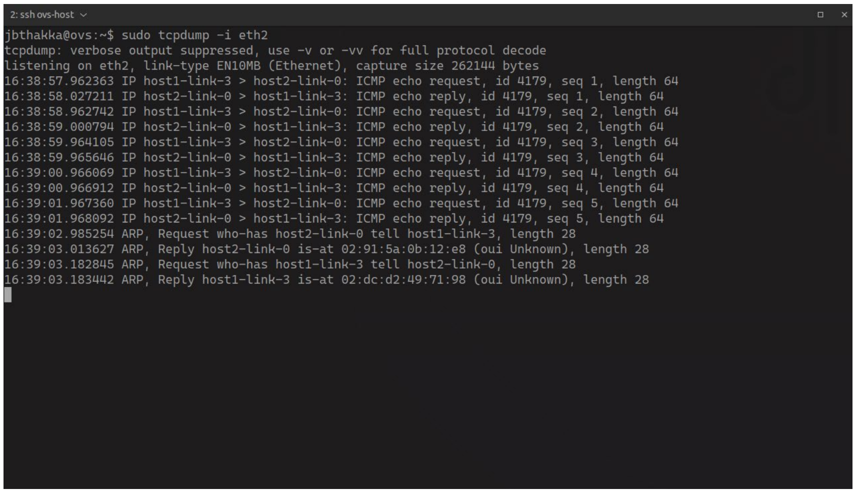
There is no duplication of traffic as shown on the above diagram when we simply run the learning switch program.

However when we run the DuplicateTraffic program with the eth3 specified as the duplicate interface we see the following output on the controller as shown below:



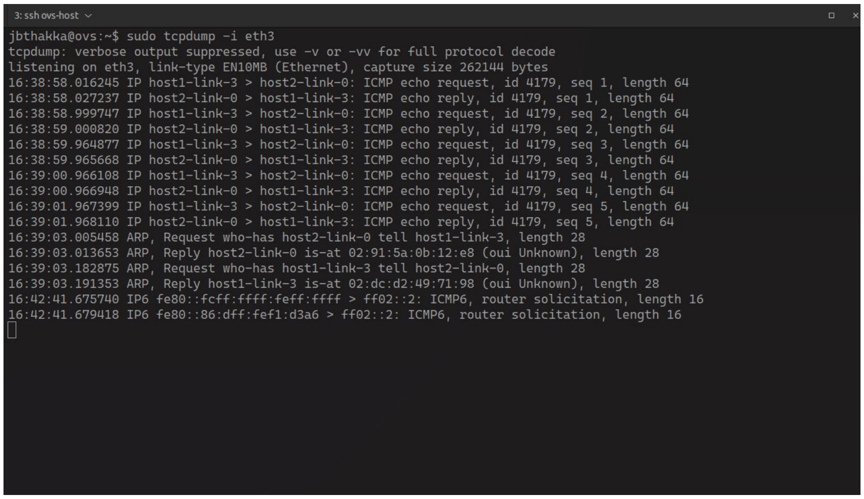
```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose DuplicateTraffic --duplicate_port=eth3
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:DuplicateTraffic:DuplicateTrafficeth3
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:DuplicateTraffic:Connection [d2-c1-04-79-20-4e 1]
DEBUG:DuplicateTraffic:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 IP]
DEBUG:DuplicateTraffic:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 IP]
DEBUG:SimpleL2Learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.[2, 3]
DEBUG:DuplicateTraffic:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 IP]
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.[1, 3]
DEBUG:DuplicateTraffic:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.[1, 3]
DEBUG:DuplicateTraffic:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.[2, 3]
DEBUG:DuplicateTraffic:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.[2, 3]
DEBUG:DuplicateTraffic:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.[1, 3]
```

And the following traffic on host2:



```
jbthakka@ovs:~$ sudo tcpdump -i eth2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
16:38:57.962363 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 1, length 64
16:38:58.027211 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 1, length 64
16:38:58.962742 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 2, length 64
16:38:59.000794 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 2, length 64
16:38:59.964105 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 3, length 64
16:38:59.965646 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 3, length 64
16:39:00.966069 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 4, length 64
16:39:00.966912 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 4, length 64
16:39:01.967360 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 5, length 64
16:39:01.968092 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 5, length 64
16:39:02.985254 ARP, Request who-has host2-link-0 tell host1-link-3, length 28
16:39:03.013627 ARP, Reply host2-link-0 is-at 02:91:5a:0b:12:e8 (oui Unknown), length 28
16:39:03.182845 ARP, Request who-has host1-link-3 tell host2-link-0, length 28
16:39:03.183442 ARP, Reply host1-link-3 is-at 02:dc:d2:49:71:98 (oui Unknown), length 28
```

And the duplicated traffic on host3:

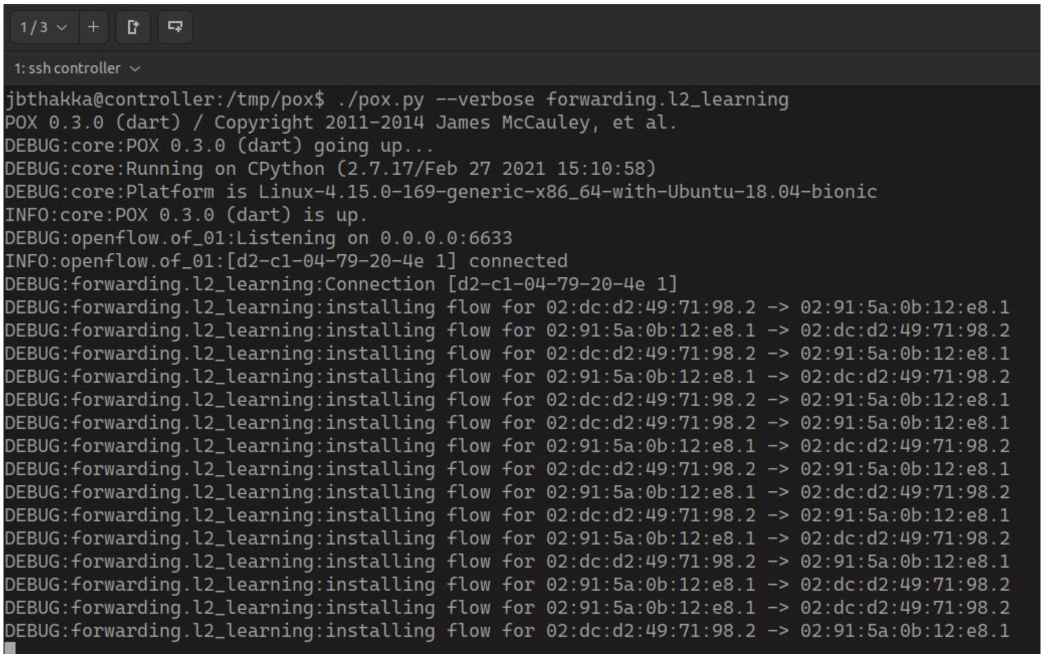


```
jbthakka@ovs:~$ sudo tcpdump -i eth3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth3, link-type EN10MB (Ethernet), capture size 262144 bytes
16:38:58.016245 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 1, length 64
16:38:58.027237 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 1, length 64
16:38:58.999747 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 2, length 64
16:38:59.000820 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 2, length 64
16:38:59.964877 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 3, length 64
16:38:59.965668 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 3, length 64
16:39:00.966108 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 4, length 64
16:39:00.966948 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 4, length 64
16:39:01.967399 IP host1-link-3 > host2-link-0: ICMP echo request, id 4179, seq 5, length 64
16:39:01.968110 IP host2-link-0 > host1-link-3: ICMP echo reply, id 4179, seq 5, length 64
16:39:03.005458 ARP, Request who-has host2-link-0 tell host1-link-3, length 28
16:39:03.013653 ARP, Reply host2-link-0 is-at 02:91:5a:0b:12:e8 (oui Unknown), length 28
16:39:03.182875 ARP, Request who-has host1-link-3 tell host2-link-0, length 28
16:39:03.191353 ARP, Reply host1-link-3 is-at 02:dc:d2:49:71:98 (oui Unknown), length 28
16:42:41.675740 IP6 fe80::fcff:ffff:feff:ffff > ff02::2: ICMP6, router solicitation, length 16
16:42:41.679418 IP6 fe80::86:dff:fef1:d3a6 > ff02::2: ICMP6, router solicitation, length 16
```

The controller receives a packet for host2, installs the correct flow to allow the packet to be delivered to host 2 and then runs a flow to duplicate the traffic to host 3.

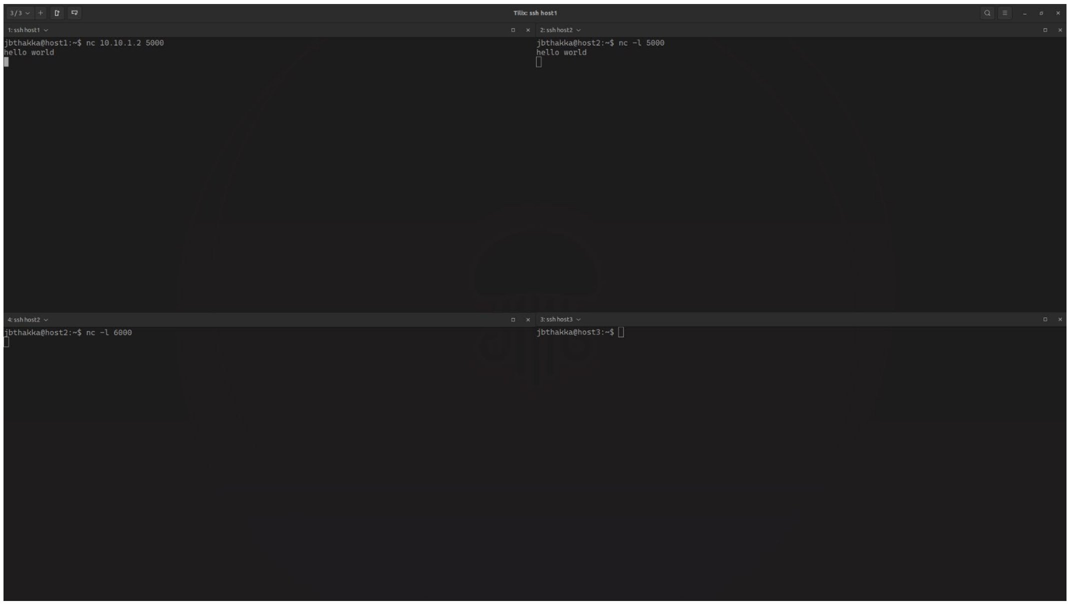
## Exercise 1.5

Before the rule for the port forwarding is inserted, when we send a message using netcat from host 1 to host 2, host 2 will receive the message on port 5000 - as it should go.



```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose forwarding.l2_learning
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:forwarding.l2_learning:Connection [d2-c1-04-79-20-4e 1]
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
```

Using l2_learning



```
jbthakka@host1:~$ nc 10.10.1.2 5000
hello world
```

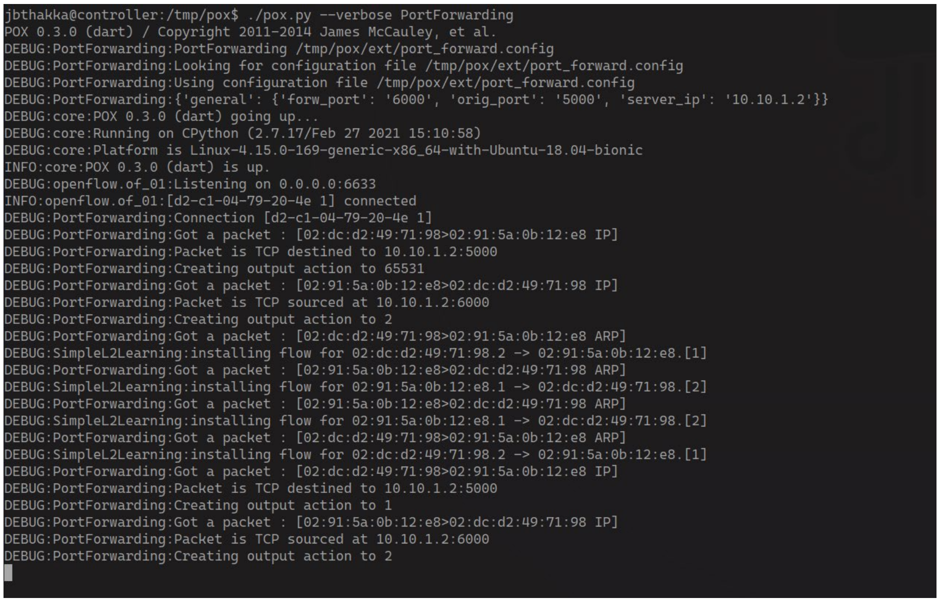
```
jbthakka@host2:~$ nc -l 5000
hello world
```

```
jbthakka@host2:~$ nc -l 6000
```

```
jbthakka@host3:~$
```

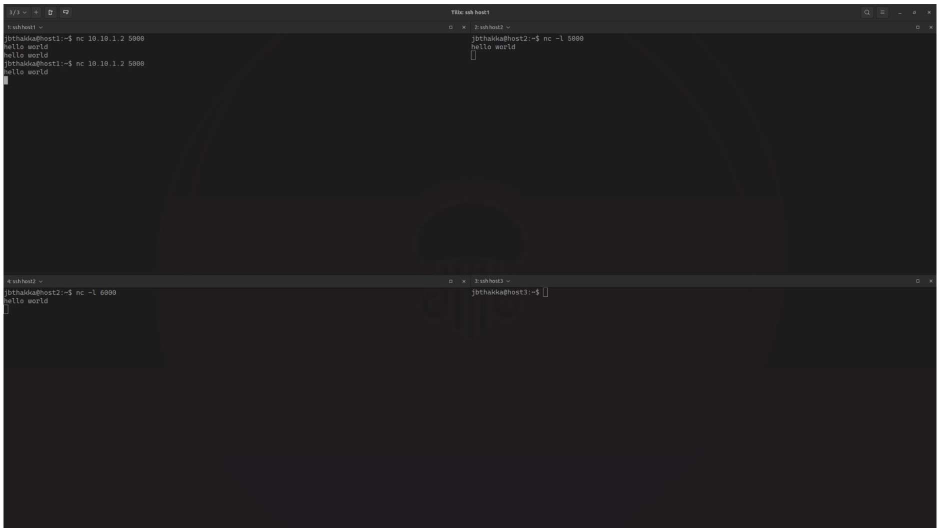
Netcat test

However, with port forwarding, when host 1 sends a message to host 2 on port 5000, the controller will forward the message to port 6000 on host 2, as shown below on the screenshot. The SDN rewrites the destination port on the packet and it gets delivered to the different port.



```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose PortForwarding
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:PortForwarding:PortForwarding /tmp/pox/ext/port_forward.config
DEBUG:PortForwarding:Looking for configuration file /tmp/pox/ext/port_forward.config
DEBUG:PortForwarding:Using configuration file /tmp/pox/ext/port_forward.config
DEBUG:PortForwarding:{'general': {'forw_port': '6000', 'orig_port': '5000', 'server_ip': '10.10.1.2'}}
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:PortForwarding:Connection [d2-c1-04-79-20-4e 1]
DEBUG:PortForwarding:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 IP]
DEBUG:PortForwarding:Packet is TCP destined to 10.10.1.2:5000
DEBUG:PortForwarding:Creating output action to 65531
DEBUG:PortForwarding:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 IP]
DEBUG:PortForwarding:Packet is TCP sourced at 10.10.1.2:6000
DEBUG:PortForwarding:Creating output action to 2
DEBUG:PortForwarding:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.[1]
DEBUG:PortForwarding:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.[2]
DEBUG:PortForwarding:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.[2]
DEBUG:PortForwarding:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 ARP]
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.[1]
DEBUG:PortForwarding:Got a packet : [02:dc:d2:49:71:98>02:91:5a:0b:12:e8 IP]
DEBUG:PortForwarding:Packet is TCP destined to 10.10.1.2:5000
DEBUG:PortForwarding:Creating output action to 1
DEBUG:PortForwarding:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 IP]
DEBUG:PortForwarding:Packet is TCP sourced at 10.10.1.2:6000
DEBUG:PortForwarding:Creating output action to 2
```

Using PortForwarding



```
jbthakka@host1:~$ nc 10.10.1.2 5000
hello world
hello world
jbthakka@host1:~$ nc 10.10.1.2 5000
hello world
```

```
jbthakka@host2:~$ nc -l 5000
hello world
```

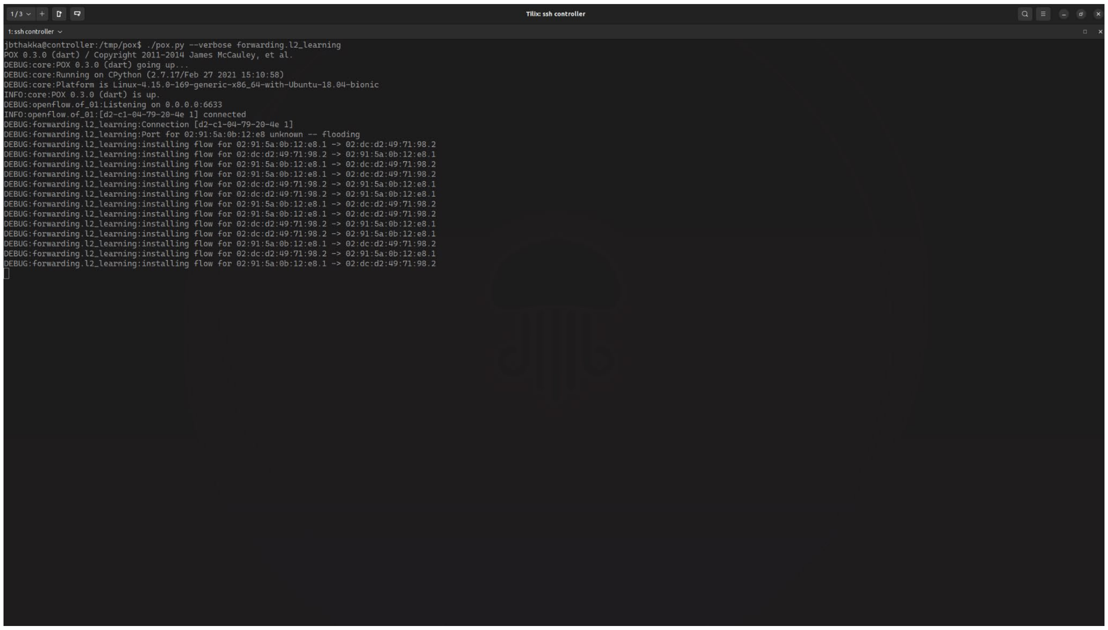
```
jbthakka@host2:~$ nc -l 6000
hello world
```

```
jbthakka@host3:~$
```

Netcat Test

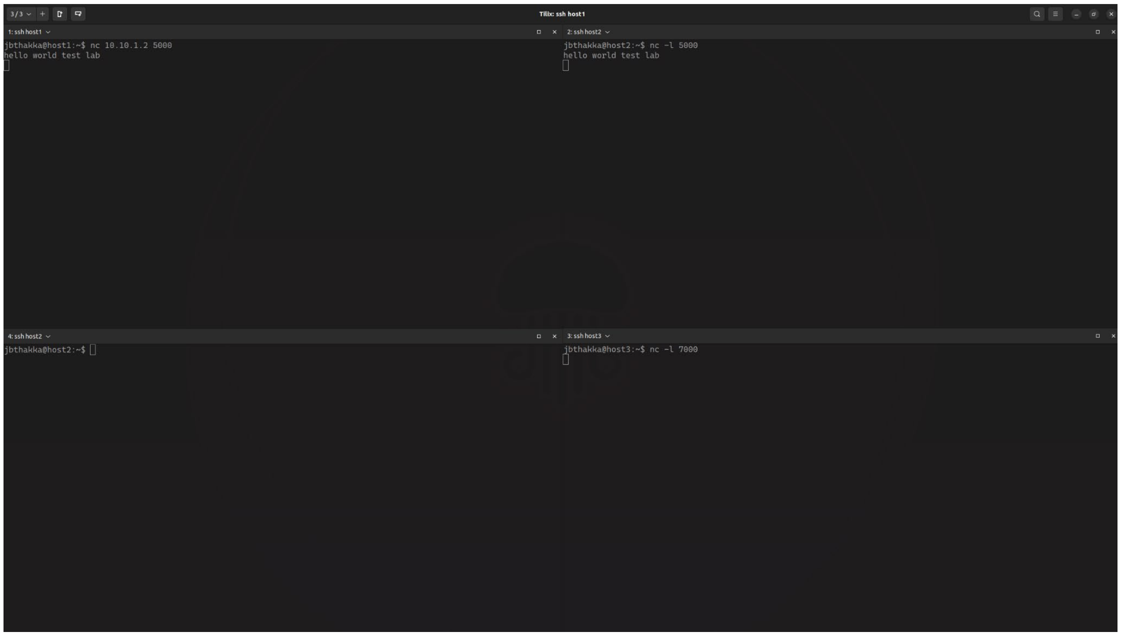
## Exercise 1.6

If we do not use the proxy, when host 1 messages host 2, the message will just appear on host 2 as shown below in the screenshot for the hosts. .



```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose forwarding.l2_learning
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:forwarding.l2_learning:Connection [d2-c1-04-79-20-4e 1]
DEBUG:forwarding.l2_learning:Port for 02:91:5a:0b:12:e8 unknown -- flooding
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
DEBUG:forwarding.l2_learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:91:5a:0b:12:e8.1
DEBUG:forwarding.l2_learning:installing flow for 02:91:5a:0b:12:e8.1 -> 02:dc:d2:49:71:98.2
```

Using l2_forwarding



```
jbthakka@host1:~$ nc 10.10.1.2 5000
hello world test lab
```

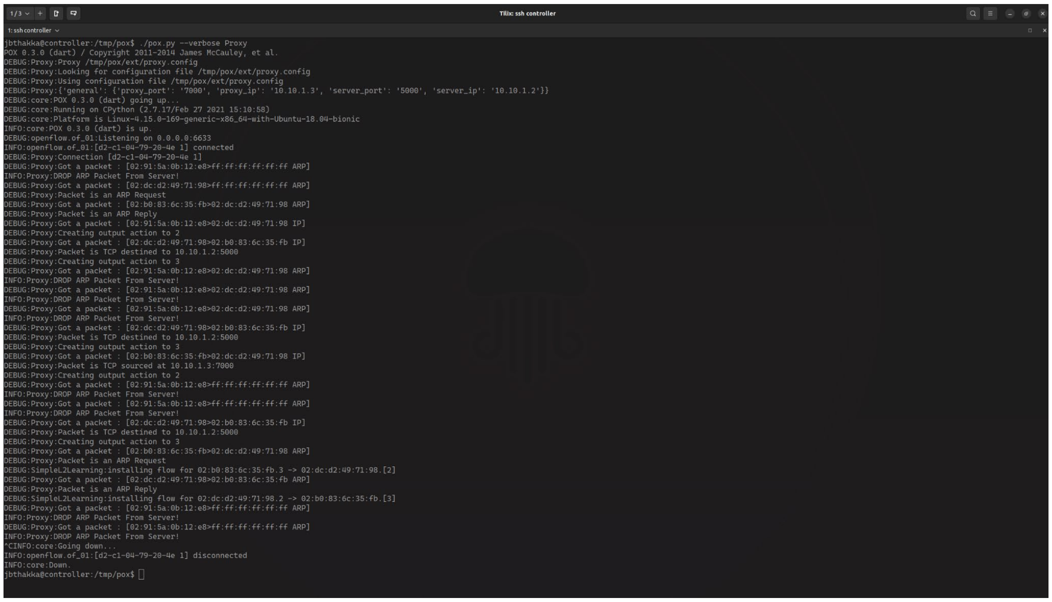
```
jbthakka@host2:~$ nc -l 5000
hello world test lab
```

```
jbthakka@host2:~$
```

```
jbthakka@host3:~$ nc -l 7000
```

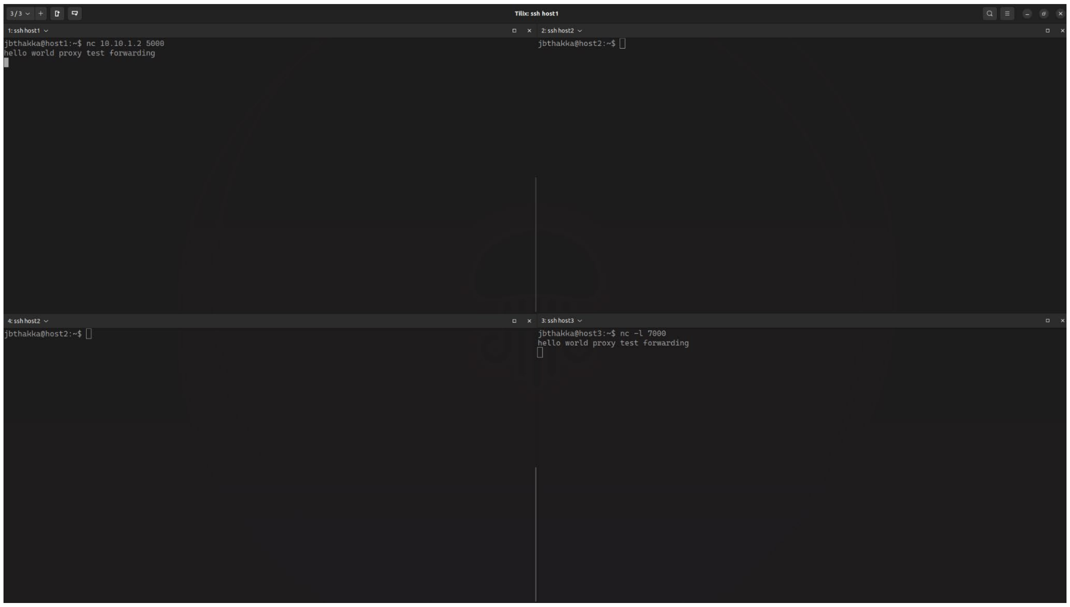
Netcat test

If we use proxy, when host 1 messages host 2, the message will appear on host 3 instead - because the pox controller is rewriting the packet destination in flight and routing the message to host3 as demonstrated by the screenshot.



```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose Proxy
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:Proxy:Proxy /tmp/pox/ext/proxy.config
DEBUG:Proxy:Looking for configuration file /tmp/pox/ext/proxy.config
DEBUG:Proxy:Using configuration file /tmp/pox/ext/proxy.config
DEBUG:Proxy:{'general': {'proxy_port': '7000', 'proxy_ip': '10.10.1.3', 'server_port': '5000', 'server_ip': '10.10.1.2'}}
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:Proxy:Connection [d2-c1-04-79-20-4e 1]
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>ff:ff:ff:ff:ff:ff ARP]
INFO:Proxy:DROP ARP Packet From Server!
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 IP]
DEBUG:Proxy:Creating output action to 2
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Packet is TCP destined to 10.10.1.2:5000
DEBUG:Proxy:Creating output action to 3
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 ARP]
INFO:Proxy:DROP ARP Packet From Server!
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 ARP]
INFO:Proxy:DROP ARP Packet From Server!
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>02:dc:d2:49:71:98 ARP]
INFO:Proxy:DROP ARP Packet From Server!
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Packet is TCP destined to 10.10.1.2:5000
DEBUG:Proxy:Creating output action to 3
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 IP]
DEBUG:Proxy:Packet is TCP sourced at 10.10.1.3:7000
DEBUG:Proxy:Creating output action to 2
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>ff:ff:ff:ff:ff:ff ARP]
INFO:Proxy:DROP ARP Packet From Server!
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>ff:ff:ff:ff:ff:ff ARP]
INFO:Proxy:DROP ARP Packet From Server!
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Packet is TCP destined to 10.10.1.2:5000
DEBUG:Proxy:Creating output action to 3
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>ff:ff:ff:ff:ff:ff ARP]
INFO:Proxy:DROP ARP Packet From Server!
DEBUG:Proxy:Got a packet : [02:91:5a:0b:12:e8>ff:ff:ff:ff:ff:ff ARP]
INFO:Proxy:DROP ARP Packet From Server!
^CINFO:core:Going down...
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] disconnected
INFO:core:Down.
jbthakka@controller:/tmp/pox$
```

Using Proxy



```
jbthakka@host1:~$ nc 10.10.1.2 5000
hello world proxy test forwarding
```

```
jbthakka@host2:~$
```

```
jbthakka@host2:~$
```

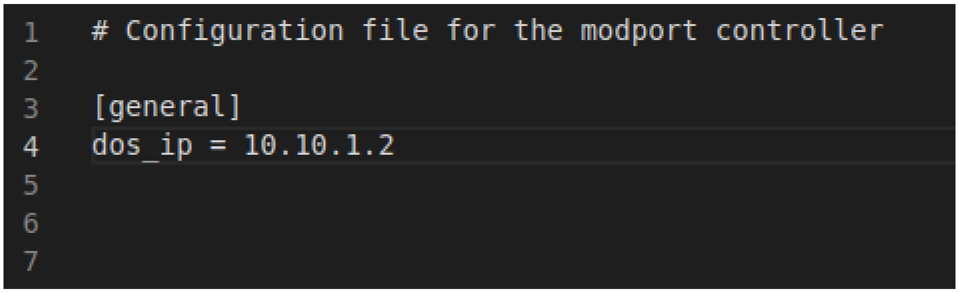
```
jbthakka@host3:~$ nc -l 7000
hello world proxy test forwarding
```

Netcat Test

**Exercise 1.7**

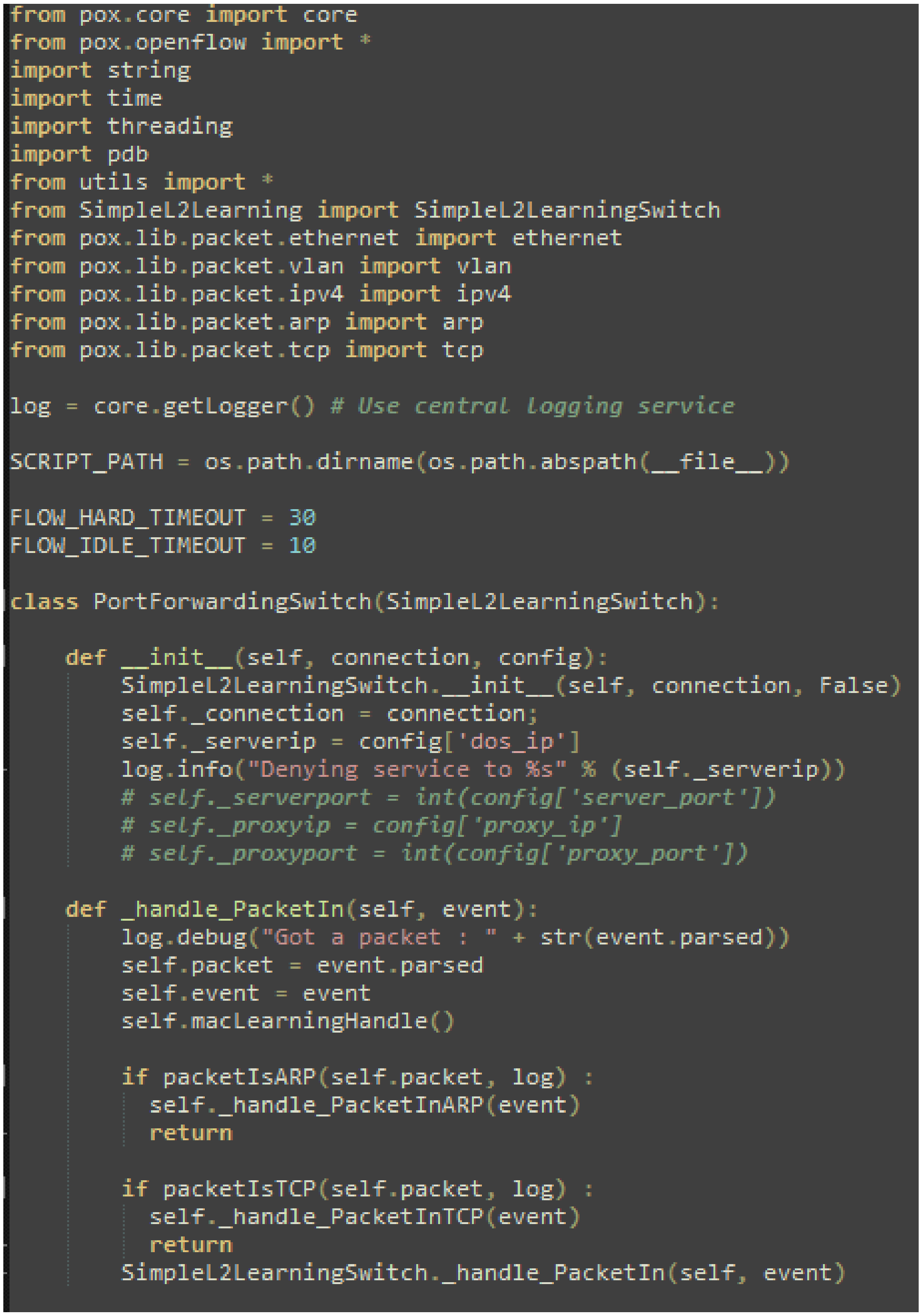
We choose to block traffic for host 2.

What we have done is modify the Proxy.py controller program to check if there is a ARP packet or TCP packet that is received from or intended for host 2 using packetDstIp and packetSrcIP functions inside the controller's function for the TCP and the ARP requests - basically we do not install any flows when encountering the packets for host 2, simply drop them, and log the message on the controller screen - this effectively prevents any host from connecting to/from host 2. You can DoS any other IP by providing a config file named "dos_config" with the following format:



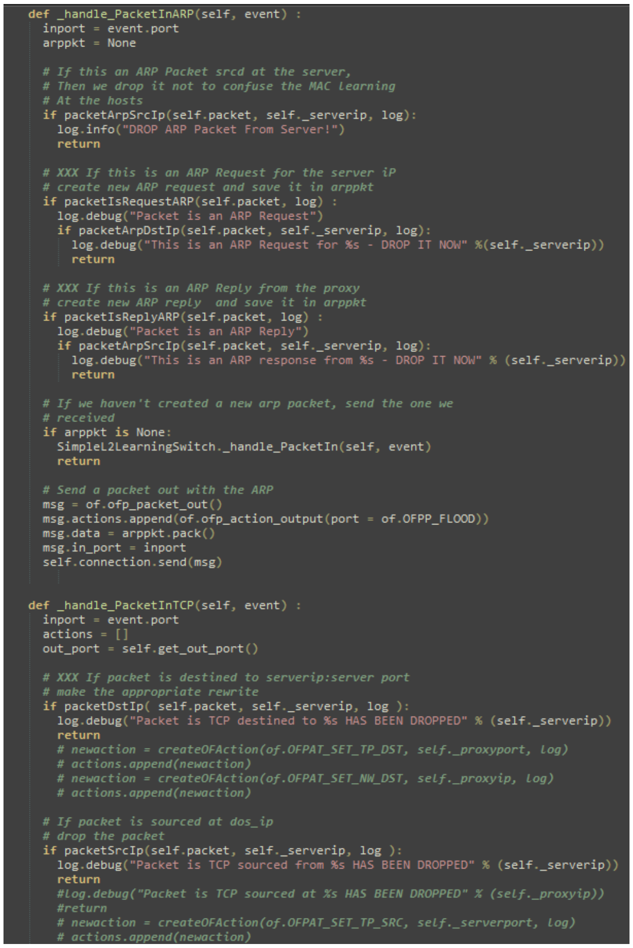
```
# Configuration file for the modport controller

[general]
dos_ip = 10.10.1.2
```

The source code is provided below:



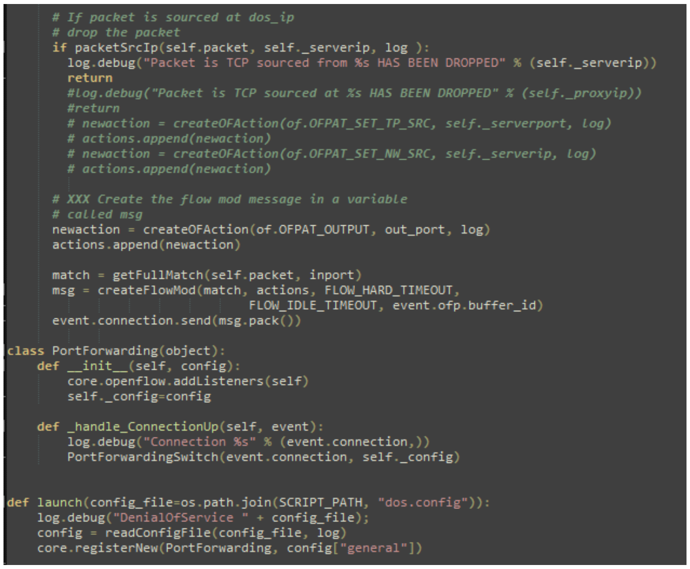
```
from pox.core import core
from pox.openflow import *
import string
import time
import threading
import pdb
from utils import *
from SimpleL2Learning import SimpleL2LearningSwitch
from pox.lib.packet.ethernet import ethernet
from pox.lib.packet.vlan import vlan
from pox.lib.packet.ipv4 import ipv4
from pox.lib.packet.arp import arp
from pox.lib.packet.tcp import tcp

log = core.getLogger() # Use central logging service

SCRIPT_PATH = os.path.dirname(os.path.abspath(__file__))

FLOW_HARD_TIMEOUT = 30
FLOW_IDLE_TIMEOUT = 10

class PortForwardingSwitch(SimpleL2LearningSwitch):

    def __init__(self, connection, config):
        SimpleL2LearningSwitch.__init__(self, connection, False)
        self._connection = connection;
        self._serverip = config['dos_ip']
        log.info("Denying service to %s" % (self._serverip))
        # self._serverport = int(config['server_port'])
        # self._proxyip = config['proxy_ip']
        # self._proxyport = int(config['proxy_port'])

    def _handle_PacketIn(self, event):
        log.debug("Got a packet : " + str(event.parsed))
        self.packet = event.parsed
        self.event = event
        self.macLearningHandle()

        if packetIsARP(self.packet, log) :
          self._handle_PacketInARP(event)
          return

        if packetIsTCP(self.packet, log) :
          self._handle_PacketInTCP(event)
          return
        SimpleL2LearningSwitch._handle_PacketIn(self, event)
```

Code Part 1



```
  def _handle_PacketInARP(self, event) :
    inport = event.port
    arppkt = None

    # If this an ARP Packet srcd at the server,
    # Then we drop it not to confuse the MAC learning
    # At the hosts
    if packetArpSrcIp(self.packet, self._serverip, log):
      log.info("DROP ARP Packet From Server!")
      return

    # XXX If this is an ARP Request for the server iP
    # create new ARP request and save it in arppkt
    if packetIsRequestARP(self.packet, log) :
      log.debug("Packet is an ARP Request")
      if packetArpDstIp(self.packet, self._serverip, log):
        log.debug("This is an ARP Request for %s - DROP IT NOW" %(self._serverip))
        return

    # XXX If this is an ARP Reply from the proxy
    # create new ARP reply  and save it in arppkt
    if packetIsReplyARP(self.packet, log) :
      log.debug("Packet is an ARP Reply")
      if packetArpSrcIp(self.packet, self._serverip, log):
        log.debug("This is an ARP response from %s - DROP IT NOW" % (self._serverip))
        return

    # If we haven't created a new arp packet, send the one we
    # received
    if arppkt is None:
      SimpleL2LearningSwitch._handle_PacketIn(self, event)
      return

    # Send a packet out with the ARP
    msg = of.ofp_packet_out()
    msg.actions.append(of.ofp_action_output(port = of.OFPP_FLOOD))
    msg.data = arppkt.pack()
    msg.in_port = inport
    self.connection.send(msg)


  def _handle_PacketInTCP(self, event) :
    inport = event.port
    actions = []
    out_port = self.get_out_port()

    # XXX If packet is destined to serverip:server port
    # make the appropriate rewrite
    if packetDstIp( self.packet, self._serverip, log ):
      log.debug("Packet is TCP destined to %s HAS BEEN DROPPED" % (self._serverip))
      return
      # newaction = createOFAction(of.OFPAT_SET_TP_DST, self._proxyport, log)
      # actions.append(newaction)
      # newaction = createOFAction(of.OFPAT_SET_NW_DST, self._proxyip, log)
      # actions.append(newaction)

    # If packet is sourced at dos_ip
    # drop the packet
    if packetSrcIp(self.packet, self._serverip, log ):
      log.debug("Packet is TCP sourced from %s HAS BEEN DROPPED" % (self._serverip))
      return
      #log.debug("Packet is TCP sourced at %s HAS BEEN DROPPED" % (self._proxyip))
      #return
      # newaction = createOFAction(of.OFPAT_SET_TP_SRC, self._serverport, log)
      # actions.append(newaction)
```

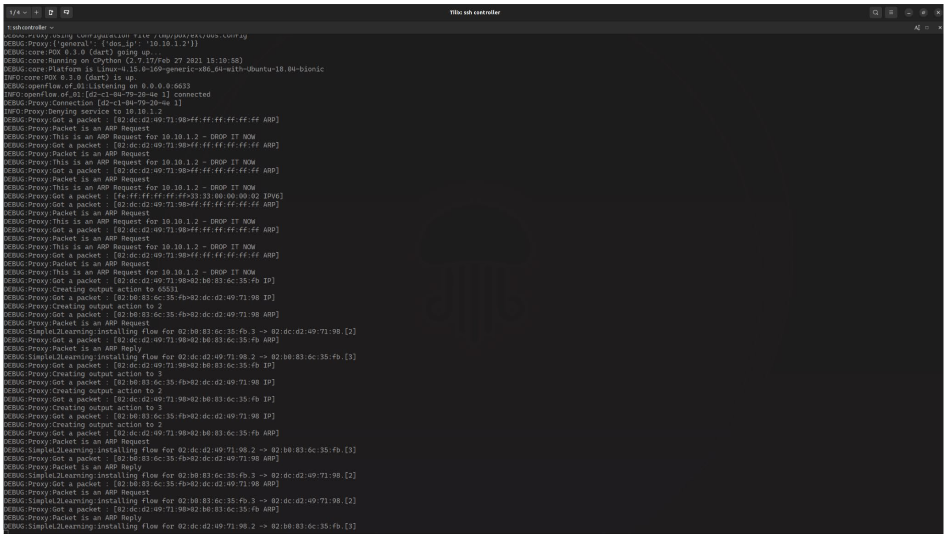
Code Part 2



```
      # If packet is sourced at dos_ip
      # drop the packet
      if packetSrcIp(self.packet, self._serverip, log ):
        log.debug("Packet is TCP sourced from %s HAS BEEN DROPPED" % (self._serverip))
        return
        #log.debug("Packet is TCP sourced at %s HAS BEEN DROPPED" % (self._proxyip))
        #return
        # newaction = createOFAction(of.OFPAT_SET_TP_SRC, self._serverport, log)
        # actions.append(newaction)
        # newaction = createOFAction(of.OFPAT_SET_NW_SRC, self._serverip, log)
        # actions.append(newaction)

      # XXX Create the flow mod message in a variable
      # called msg
      newaction = createOFAction(of.OFPAT_OUTPUT, out_port, log)
      actions.append(newaction)

      match = getFullMatch(self.packet, inport)
      msg = createFlowMod(match, actions, FLOW_HARD_TIMEOUT,
                                 FLOW_IDLE_TIMEOUT, event.ofp.buffer_id)
      event.connection.send(msg.pack())

class PortForwarding(object):
    def __init__(self, config):
        core.openflow.addListeners(self)
        self._config=config

    def _handle_ConnectionUp(self, event):
        log.debug("Connection %s" % (event.connection,))
        PortForwardingSwitch(event.connection, self._config)


def launch(config_file=os.path.join(SCRIPT_PATH, "dos.config")):
    log.debug("DenialOfService " + config_file);
    config = readConfigFile(config_file, log)
    core.registerNew(PortForwarding, config["general"])
```

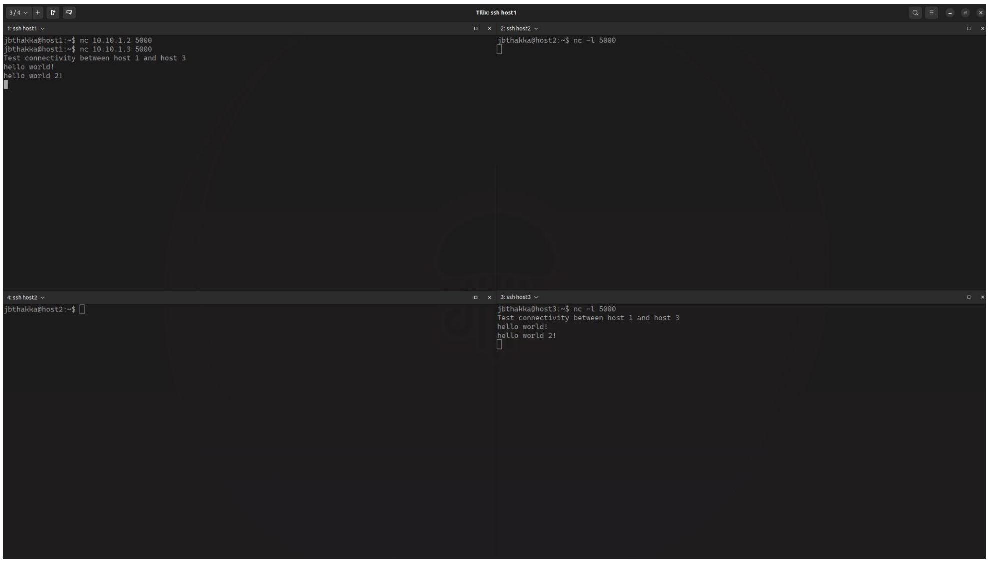
Code Part 3

Demo - When host 1 tries to message host 3, the message goes through successfully.



```
DEBUG:Proxy:Using configuration file /tmp/pox/ext/dos.config
DEBUG:Proxy:{'general': {'dos_ip': '10.10.1.2'}}
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:Proxy:Connection [d2-c1-04-79-20-4e 1]
INFO:Proxy:Denying service to 10.10.1.2
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [fe:ff:ff:ff:ff:ff>33:33:00:00:00:02 IPV6]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Creating output action to 65531
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 IP]
DEBUG:Proxy:Creating output action to 2
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Creating output action to 3
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 IP]
DEBUG:Proxy:Creating output action to 2
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Creating output action to 3
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 IP]
DEBUG:Proxy:Creating output action to 2
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
```

Host 1 message Host 3 (Controller View)



```
jbthakka@host1:~$ nc 10.10.1.2 5000
jbthakka@host1:~$ nc 10.10.1.3 5000
Test connectivity between host 1 and host 3
hello world!
hello world 2!
```

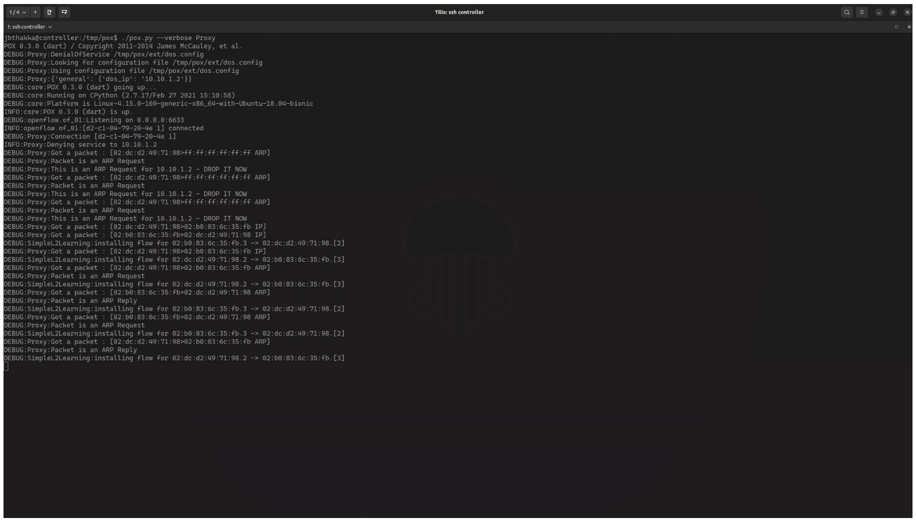
```
jbthakka@host2:~$ nc -l 5000
```

```
jbthakka@host2:~$
```

```
jbthakka@host3:~$ nc -l 5000
Test connectivity between host 1 and host 3
hello world!
hello world 2!
```

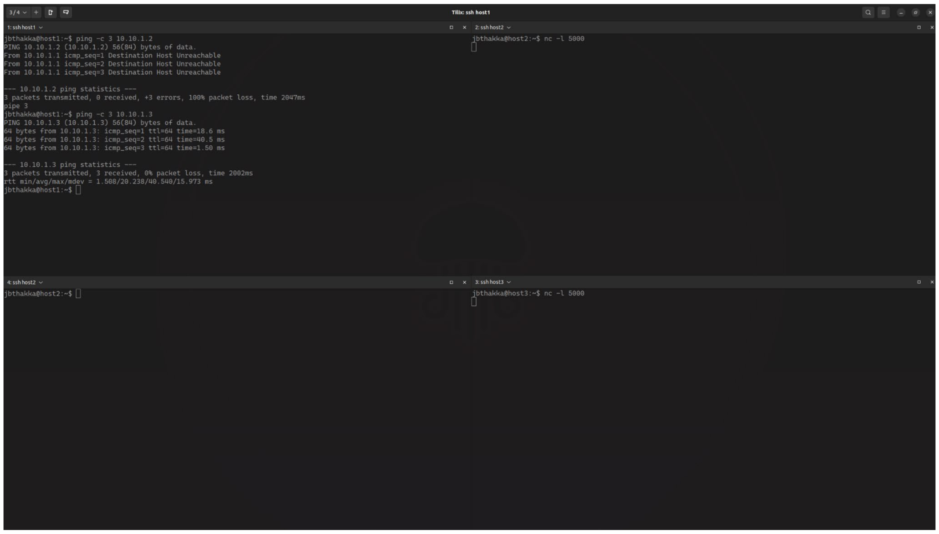
Host 1 message Host 3

When Host 1 tries to ping Host 2, all the packets will be dropped and hence Host 2 is unreachable from Host 1 as shown below.



```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose Proxy
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:Proxy:DenialOfService /tmp/pox/ext/dos.config
DEBUG:Proxy:Looking for configuration file /tmp/pox/ext/dos.config
DEBUG:Proxy:Using configuration file /tmp/pox/ext/dos.config
DEBUG:Proxy:{'general': {'dos_ip': '10.10.1.2'}}
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:Proxy:Connection [d2-c1-04-79-20-4e 1]
INFO:Proxy:Denying service to 10.10.1.2
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 IP]
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
```

Host 1 tries to ping Host 2 (Fails) (Controller View)



```
jbthakka@host1:~$ ping -c 3 10.10.1.2
PING 10.10.1.2 (10.10.1.2) 56(84) bytes of data.
From 10.10.1.1 icmp_seq=1 Destination Host Unreachable
From 10.10.1.1 icmp_seq=2 Destination Host Unreachable
From 10.10.1.1 icmp_seq=3 Destination Host Unreachable

--- 10.10.1.2 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2047ms
pipe 3
jbthakka@host1:~$ ping -c 3 10.10.1.3
PING 10.10.1.3 (10.10.1.3) 56(84) bytes of data.
64 bytes from 10.10.1.3: icmp_seq=1 ttl=64 time=18.6 ms
64 bytes from 10.10.1.3: icmp_seq=2 ttl=64 time=40.5 ms
64 bytes from 10.10.1.3: icmp_seq=3 ttl=64 time=1.50 ms

--- 10.10.1.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.508/20.238/40.540/15.973 ms
jbthakka@host1:~$
```

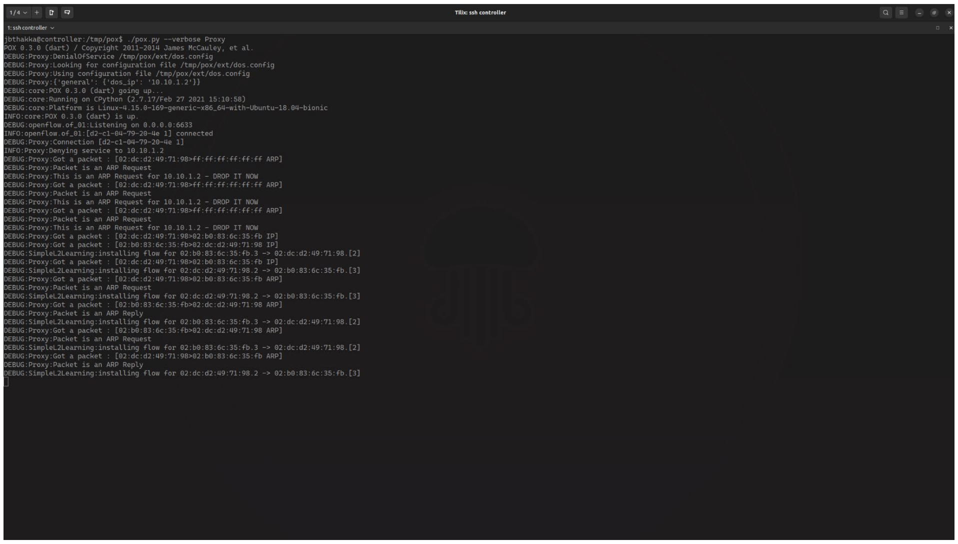
```
jbthakka@host2:~$ nc -l 5000
```

```
jbthakka@host2:~$
```

```
jbthakka@host3:~$ nc -l 5000
```

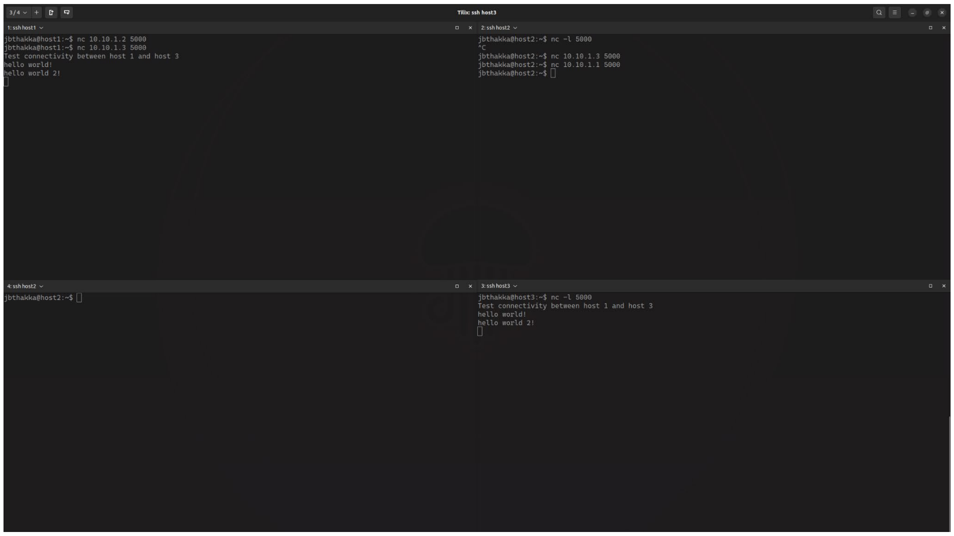
Host 1 tries to ping Host 2 (Fails)

When Host 2 tries to connect to Host 1 or 3, it wouldn't be able to as its packet is all dropped.



```
jbthakka@controller:/tmp/pox$ ./pox.py --verbose Proxy
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:Proxy:DenialOfService /tmp/pox/ext/dos.config
DEBUG:Proxy:Looking for configuration file /tmp/pox/ext/dos.config
DEBUG:Proxy:Using configuration file /tmp/pox/ext/dos.config
DEBUG:Proxy:{'general': {'dos_ip': '10.10.1.2'}}
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.17/Feb 27 2021 15:10:58)
DEBUG:core:Platform is Linux-4.15.0-169-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[d2-c1-04-79-20-4e 1] connected
DEBUG:Proxy:Connection [d2-c1-04-79-20-4e 1]
INFO:Proxy:Denying service to 10.10.1.2
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>ff:ff:ff:ff:ff:ff ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:Proxy:This is an ARP Request for 10.10.1.2 - DROP IT NOW
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 IP]
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb IP]
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:b0:83:6c:35:fb>02:dc:d2:49:71:98 ARP]
DEBUG:Proxy:Packet is an ARP Request
DEBUG:SimpleL2Learning:installing flow for 02:b0:83:6c:35:fb.3 -> 02:dc:d2:49:71:98.[2]
DEBUG:Proxy:Got a packet : [02:dc:d2:49:71:98>02:b0:83:6c:35:fb ARP]
DEBUG:Proxy:Packet is an ARP Reply
DEBUG:SimpleL2Learning:installing flow for 02:dc:d2:49:71:98.2 -> 02:b0:83:6c:35:fb.[3]
```

Host 2 tries to connect to host 1 and 3 via Netcat using TCP (fails) (Controller View)



```
jbthakka@host1:~$ nc 10.10.1.2 5000
jbthakka@host1:~$ nc 10.10.1.3 5000
Test connectivity between host 1 and host 3
hello world!
hello world 2!
```

```
jbthakka@host2:~$ nc -l 5000
^C
jbthakka@host2:~$ nc 10.10.1.3 5000
jbthakka@host2:~$ nc 10.10.1.1 5000
jbthakka@host2:~$
```

```
jbthakka@host2:~$
```

```
jbthakka@host3:~$ nc -l 5000
Test connectivity between host 1 and host 3
hello world!
hello world 2!
```

Host 2 tries to connect to host 1 and 3 via Netcat using TCP (fails)