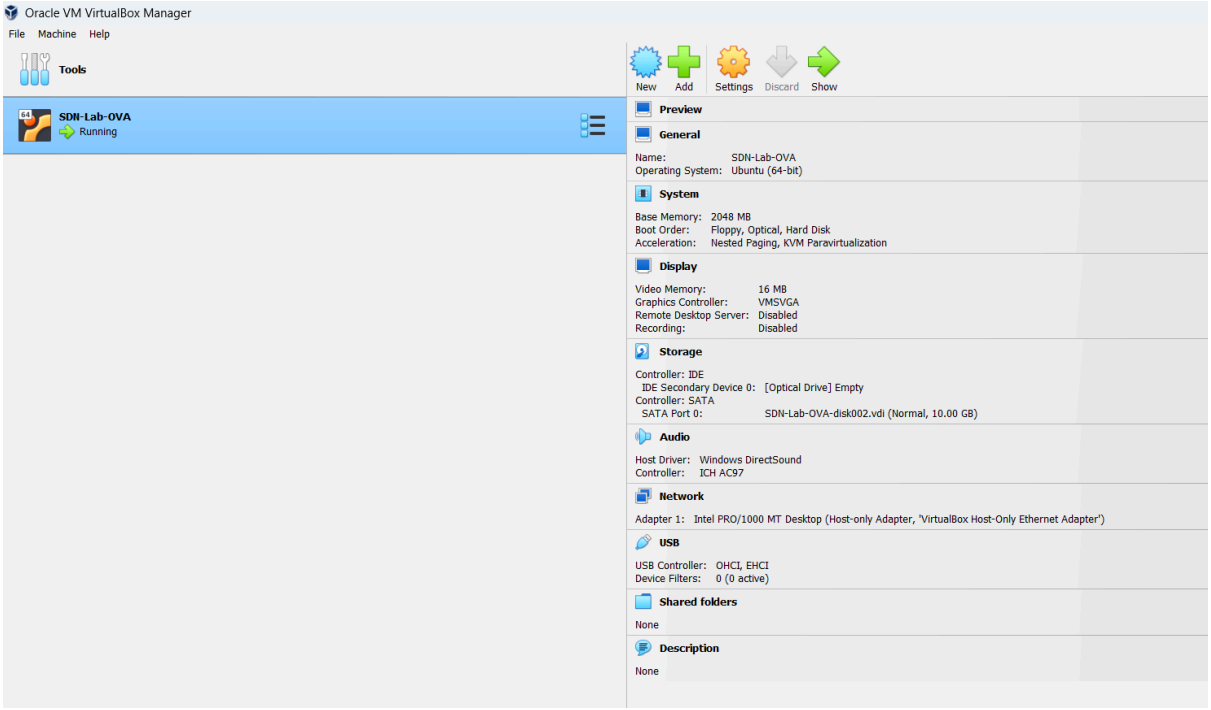
University of Moratuwa
Faculty of Engineering
Department of Electronic & Telecommunication Engineering

EN2150-Communication Network Engineering
OpenFlow Configuration Lab

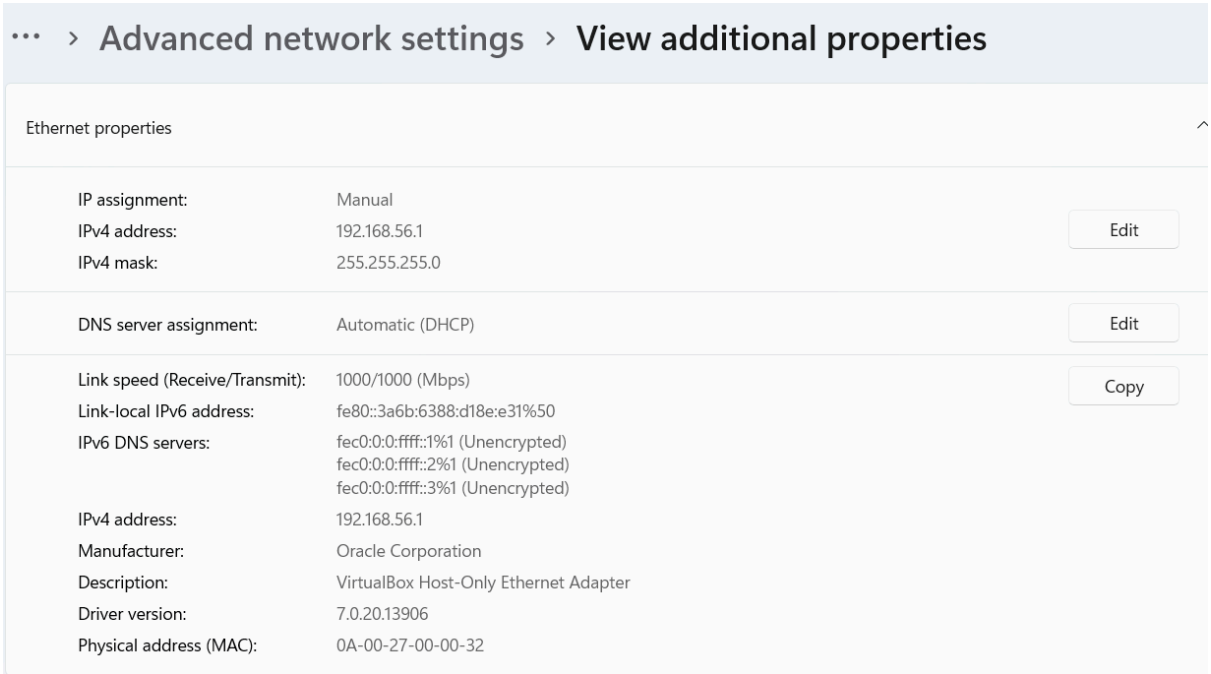
210031H A.A.W.L.R.Amarasinghe

Setup the Environment in Local PC

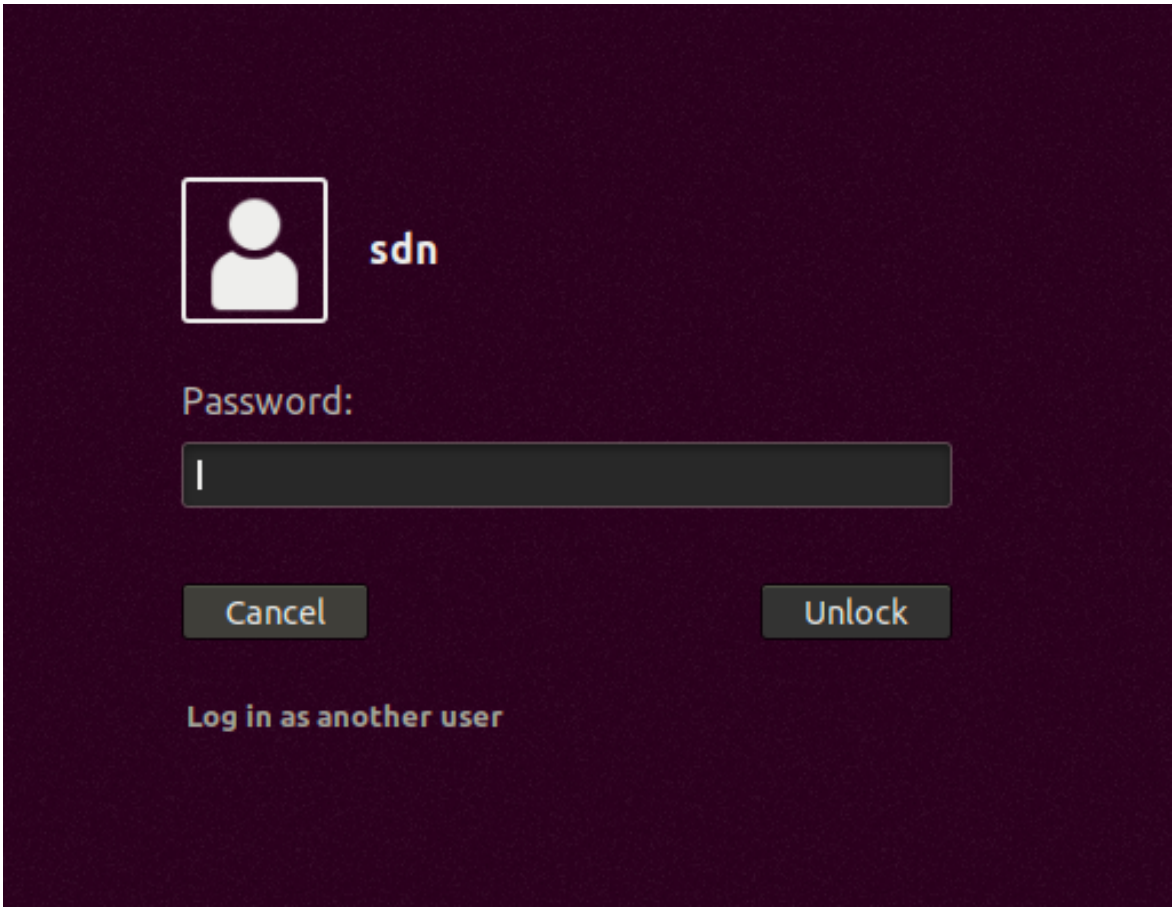
Virtual Box Software



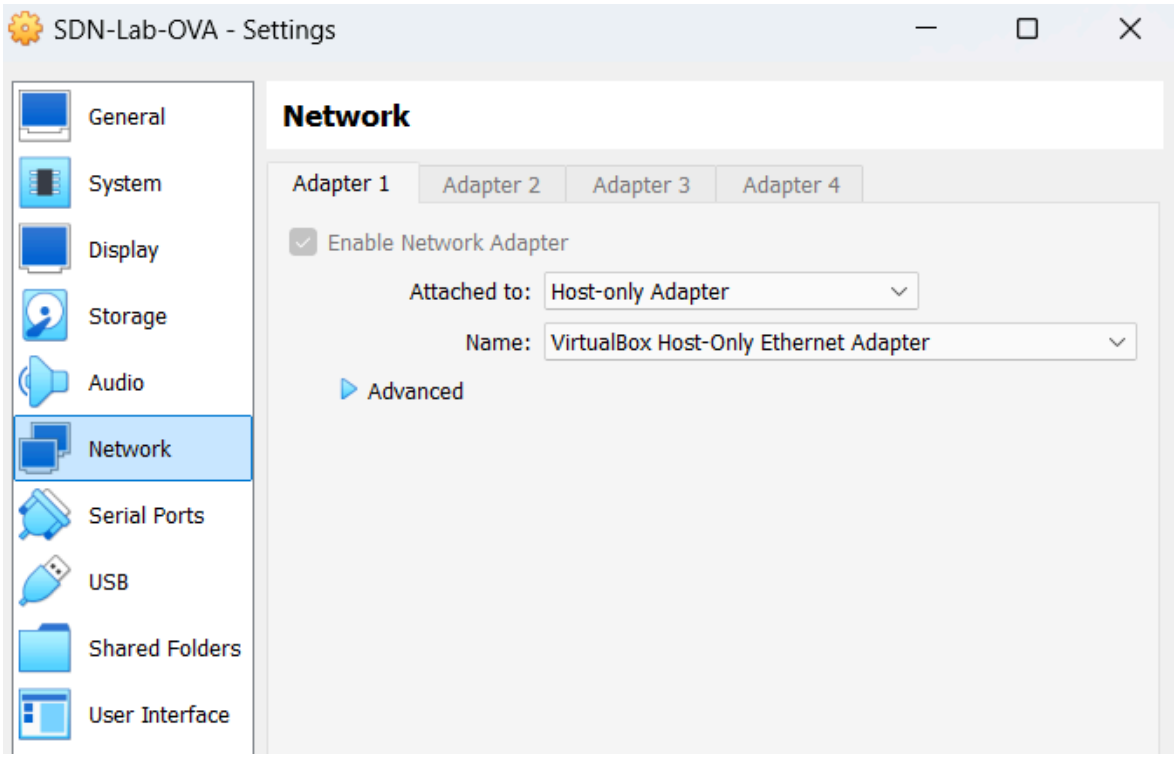
Network Adapter Settings in local PC



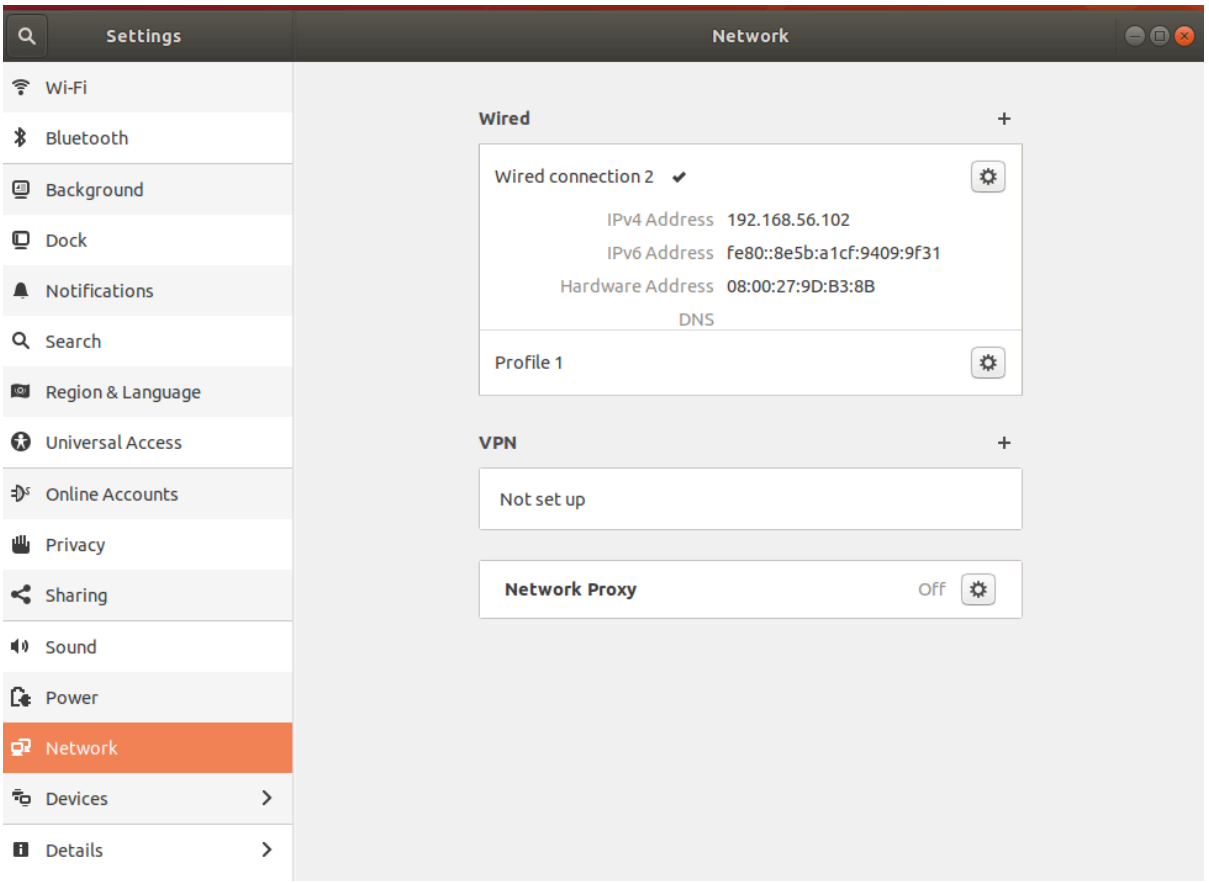
Login to the Virtual Machine
password: SDN@123



VirtualBox Network Settings



Virtual Machine IP address setup



Lab Exercise

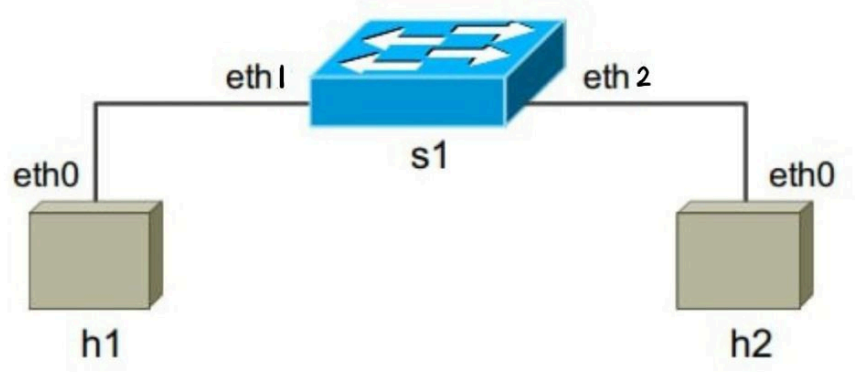
Part 1

1.1 Creating a Simple Network

Created the small Mininet network using the given command.

```
File  Edit  View  Search  Terminal  Help
sdn@sdn-VirtualBox:~$ sudo mn --mac --controller="none"
[sudo] password for sdn:
Sorry, try again.
[sudo] password for sdn:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
```



1.2 Mininet Commands

nodes

This command gives the list of the nodes in the network.
This network has three nodes. (Two hosts and a switch)

```
mininet> nodes
available nodes are:
h1 h2 s1
```

net

This command outputs the connections of all nodes.

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
```

dump

This command gives IP addresses of the hosts and loop back IP address of the switch.

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1902>
<Host h2: h2-eth0:10.0.0.2 pid=1904>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=1909>
```

1.3 Attempting Pinging

The ping attempts failed. In this network, there is no controller, and the switch is not configured, so packets can't be forwarded without either a controller or a properly configured switch. Without a controller, the switch cannot learn the MAC addresses of the hosts and therefore cannot forward packets between them. Therefore, the destination is unreachable.

```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable
From 10.0.0.1 icmp_seq=6 Destination Host Unreachable
From 10.0.0.1 icmp_seq=7 Destination Host Unreachable
From 10.0.0.1 icmp_seq=8 Destination Host Unreachable
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
From 10.0.0.1 icmp_seq=12 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
13 packets transmitted, 0 received, +12 errors, 100% packet loss, time 12285ms
pipe 4
mininet>
```

```
mininet> h2 ping h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
From 10.0.0.2 icmp_seq=4 Destination Host Unreachable
From 10.0.0.2 icmp_seq=5 Destination Host Unreachable
From 10.0.0.2 icmp_seq=6 Destination Host Unreachable
From 10.0.0.2 icmp_seq=7 Destination Host Unreachable
From 10.0.0.2 icmp_seq=8 Destination Host Unreachable
From 10.0.0.2 icmp_seq=9 Destination Host Unreachable
From 10.0.0.2 icmp_seq=10 Destination Host Unreachable
From 10.0.0.2 icmp_seq=11 Destination Host Unreachable
From 10.0.0.2 icmp_seq=12 Destination Host Unreachable
^C
--- 10.0.0.1 ping statistics ---
13 packets transmitted, 0 received, +12 errors, 100% packet loss, time 12279ms
pipe 4
mininet>
```

1.4 Dump-flows

Dump-flows command gives information about current identified flows in the network. As there is no OpenFlow controller and no flow is identified. The switch operates as an unmanaged device and does not perform any forwarding.

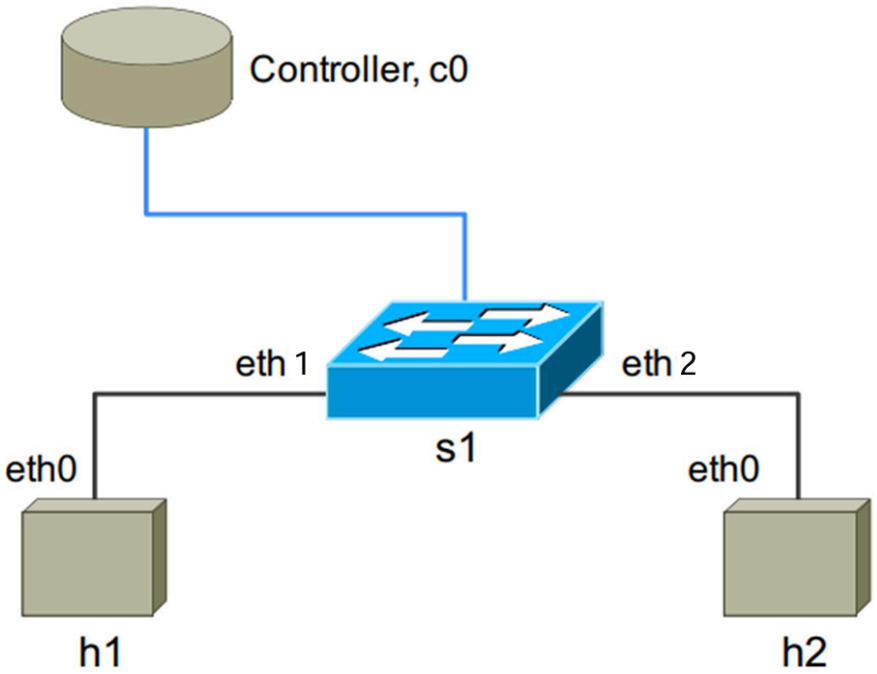
```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
mininet>
```

1.5 Exit from the Mininet

```
mininet> exit
*** Stopping 0 controllers

*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 164.715 seconds
```

1.6 Mininet network with default controller



This creates the previous network with the default controller

```
sdn@sdn-VirtualBox:~$ sudo mn --mac
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

1.7 Mininet Commands

These commands give the details about the network.

nodes

```
mininet> nodes
available nodes are:
c0 h1 h2 s1
mininet>
```

net

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
```

dump

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=6153>
<Host h2: h2-eth0:10.0.0.2 pid=6155>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=6160>
<OVSController c0: 127.0.0.1:6653 pid=6146>
mininet>
```

1.8 Flow Table Checking

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=248.712s, table=0, n_packets=20, n_bytes=1592, priority=0 actions=CONTROLLER:128
mininet>
```

A flow was identified even without performing a ping. This may be because the hosts exchanged packets, such as ARP packets. The flow has been active for 248.712s, is stored in table 0 and that 20 packets matched this flow rule. 1592 bytes were transferred and they had a priority of 0.

1.9 Pinging

The pings are successful. Previously there was no any controller but now there is the default OpenFlow controller and it does forward packets. The controller sets up flow rules and the packets are forwarded according to the rules.

```
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_seq=1 ttl=64 time=2.28 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.265 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.131 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.084 ms
^C
--- 10.0.0.2 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10206ms
rtt min/avg/max/mdev = 0.048/0.296/2.280/0.629 ms
mininet>
```

```
mininet> h2 ping h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=4.88 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.078 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.072 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.083 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.059 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.085 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.067 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.081 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=0.083 ms
64 bytes from 10.0.0.1: icmp_seq=11 ttl=64 time=0.044 ms
^C
--- 10.0.0.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10193ms
rtt min/avg/max/mdev = 0.044/0.508/4.887/1.384 ms
mininet>
```

1.10 Analysing dump-flows

The flow table now contains entries created by the controller. These entries match the source and destination MAC addresses of h1 and h2, allowing traffic to pass between them. The controller dynamically installs these flow rules based on traffic patterns to enable communication between the two hosts.

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=40.869s, table=0, n_packets=1, n_bytes=42, idle_timeout=60
, priority=1,arp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00:
02,dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=o
utput:"s1-eth1"
 cookie=0x0, duration=35.672s, table=0, n_packets=1, n_bytes=42, idle_timeout=60
, priority=1,arp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00:
02,dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=o
utput:"s1-eth1"
 cookie=0x0, duration=35.662s, table=0, n_packets=1, n_bytes=42, idle_timeout=60
, priority=1,arp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00:
01,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=o
utput:"s1-eth2"
 cookie=0x0, duration=6.488s, table=0, n_packets=0, n_bytes=0, idle_timeout=60,
priority=1,arp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00:01
,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=1 actions=out
put:"s1-eth2"
 cookie=0x0, duration=40.868s, table=0, n_packets=7, n_bytes=686, idle_timeout=6
0, priority=1,icmp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:0
0:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type
=8,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=40.867s, table=0, n_packets=7, n_bytes=686, idle_timeout=6
0, priority=1,icmp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:0
0:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type
=0,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=11.734s, table=0, n_packets=6, n_bytes=588, idle_timeout=6
0, priority=1,icmp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:0
0:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type
=8,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=11.726s, table=0, n_packets=6, n_bytes=588, idle_timeout=6
0, priority=1,icmp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:0
0:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type
=0,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=110.340s, table=0, n_packets=27, n_bytes=2054, priority=0
actions=CONTROLLER:128
mininet>
```

1.11 Analysing dump-flows after flows timeout

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=317.324s, table=0, n_packets=31, n_bytes=2334, priority=0
actions=CONTROLLER:128
mininet>
```

1.12 Enabling snooping

```
mininet> dpctl snoop &
*** s1 ------------------------------------------------------------------------
```

1.13 Analysing dump-flows after flows timeout

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
 cookie=0x0, duration=211.350s, table=0, n_packets=20, n_bytes=1592, priority=0 actions=CONTROLLER:128
```

1.14 Checking the flow table

After enabling snooping and rechecking the flow table, we can see that the table have some new entries or updated entries due to new packets being processed. Snooping allows us to see OpenFlow messages between the switch and the controller, such as PACKET_IN, PACKET_OUT, and FLOW_MOD.

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=70 in_port=2 (via no_match) data_len=70 (unbuffered)
icmp6,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=33:33:00:00:00:02,ipv6_src=fe80::200:ff:fe00:2,ipv6_dst=ff02::2,
cmp6_csum:7b2a
OFPT_PACKET_OUT (OF1.5) (xid=0x19): in_port=2 actions=FLOOD data_len=70
icmp6,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=33:33:00:00:00:02,ipv6_src=fe80::200:ff:fe00:2,ipv6_dst=ff02::2,
cmp6_csum:7b2a
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=70 in_port=1 (via no_match) data_len=70 (unbuffered)
icmp6,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=33:33:00:00:00:02,ipv6_src=fe80::200:ff:fe00:1,ipv6_dst=ff02::2,
cmp6_csum:7b2c
OFPT_PACKET_OUT (OF1.5) (xid=0x1a): in_port=1 actions=FLOOD data_len=70
icmp6,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=33:33:00:00:00:02,ipv6_src=fe80::200:ff:fe00:1,ipv6_dst=ff02::2,
cmp6_csum:7b2c
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
```

1.15 Pinging between the hosts

```
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_seq=1 ttl=64 time=2.85 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.283 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.054 ms
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7121ms
rtt min/avg/max/mdev = 0.043/0.442/2.857/0.915 ms
```

```
mininet> h2 ping h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=7.65 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.076 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.072 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.098 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.072 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.083 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.083 ms
^C
--- 10.0.0.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7136ms
rtt min/avg/max/mdev = 0.048/1.023/7.657/2.507 ms
```

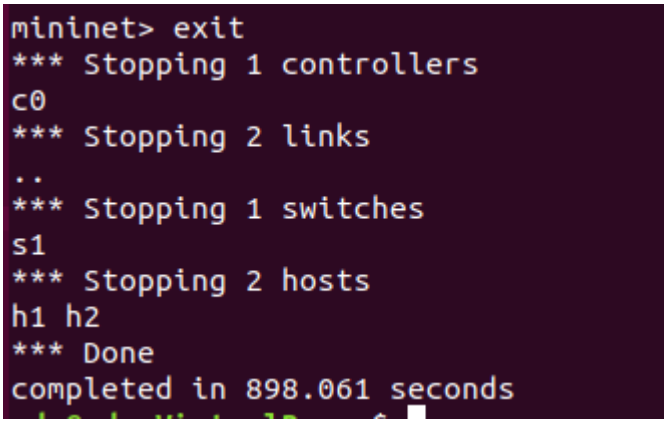
1.16 Analysing dump-flows

The flow table shows new flow entries corresponding to the ping traffic. These entries have been set up by the controller to match the MAC addresses of the hosts and ensure that packets are forwarded correctly between h1 and h2. This demonstrates the controller's role in dynamically managing the switch's flow table based on network traffic.

```
rtt min/avg/max/mdev = 0.084/0.561/2.331/0.885 ms
mininet>  dpctl dump-flows
*** s1 ------------------------------------------------------------
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
```

```
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=42 in_port=1 (via no_match) data_len=42 (unbuffered)
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_
OFPT_PACKET_OUT (OF1.5) (xid=0x1b): in_port=1 actions=FLOOD data_len=42
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=42 in_port=2 (via no_match) data_len=42 (unbuffered)
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_
OFPT_FLOW_MOD (OF1.5) (xid=0x1c): ADD priority=1,arp,in_port=2,vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00:
OFPT_PACKET_OUT (OF1.5) (xid=0x1d): in_port=2 actions=output:1 data_len=42
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=98 in_port=1 (via no_match) data_len=98 (unbuffered)
icmp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_to
OFPT_FLOW_MOD (OF1.5) (xid=0x1e): ADD priority=1,icmp,in_port=1,vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00
le:60 actions=output:2
OFPT_PACKET_OUT (OF1.5) (xid=0x1f): in_port=1 actions=output:2 data_len=98
icmp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_to
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=98 in_port=2 (via no_match) data_len=98 (unbuffered)
icmp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_to
OFPT_FLOW_MOD (OF1.5) (xid=0x20): ADD priority=1,icmp,in_port=2,vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00
le:60 actions=output:1
OFPT_PACKET_OUT (OF1.5) (xid=0x21): in_port=2 actions=output:1 data_len=98
icmp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_to
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=42 in_port=2 (via no_match) data_len=42 (unbuffered)
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_
OFPT_FLOW_MOD (OF1.5) (xid=0x22): ADD priority=1,arp,in_port=2,vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00:
OFPT_PACKET_OUT (OF1.5) (xid=0x23): in_port=2 actions=output:1 data_len=42
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=42 in_port=1 (via no_match) data_len=42 (unbuffered)
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_
OFPT_FLOW_MOD (OF1.5) (xid=0x24): ADD priority=1,arp,in_port=1,vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00:
OFPT_PACKET_OUT (OF1.5) (xid=0x25): in_port=1 actions=output:2 data_len=42
arp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (OF1.5) (xid=0x0): 0 bytes of payload
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=98 in_port=2 (via no_match) data_len=98 (unbuffered)
icmp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_to
OFPT_FLOW_MOD (OF1.5) (xid=0x26): ADD priority=1,icmp,in_port=2,vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00
le:60 actions=output:1
OFPT_PACKET_OUT (OF1.5) (xid=0x27): in_port=2 actions=output:1 data_len=98
icmp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_to
OFPT_PACKET_IN (OF1.5) (xid=0x0): cookie=0x0 total_len=98 in_port=1 (via no_match) data_len=98 (unbuffered)
icmp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_to
OFPT_FLOW_MOD (OF1.5) (xid=0x28): ADD priority=1,icmp,in_port=1,vlan_tci=0x0000/0x1fff,dl_src=00:00:00:00:00
le:60 actions=output:2
```

1.17 Exit from the Mininet



Part 2

Connecting Mininet network to Opendaylight controller

2.1 IP address configuration

The network interfaces on the system are as follows:

• lo (Loopback Interface): This is used for internal communication within the host. It has an MTU of 65536 and IP addresses:
– IPv4: 127.0.0.1/8
– IPv6: ::1/128

• enp0s3 (Ethernet Interface): This is used for network communication. It has an MTU of 1500 and IP addresses:
– IPv4: 192.168.56.102/24 (Broadcast: 192.168.56.255)
– IPv6: 46e2:d4b6:499d:8c6b/64

```
sdn@sdn-VirtualBox:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:9d:b3:8b brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute
 enp0s3
       valid_lft 450sec preferred_lft 450sec
    inet6 fe80::46e2:d4b6:499d:8c6b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

2.2 OpenDaylight

```
100% [========================================================================]
Karaf started in 20s. Bundle stats: 376 active, 377 total

    ________                       ________                .__  .__       .__     __
    \_____  \ ______   ____   ____ \______ \ _____  ___.__.|  | |__| ____ |  |___/  |_
     /   |   \\____ \_/ __ \ /    \ |    |  \\__  \<   |  ||  | |  |/ ___\|  |  \   __\
    /    |    \  |_> >  ___/|   |  \|    `   \/ __ \\___  ||  |_|  / /_/  >   Y  \  |
    \_______  /   __/ \___  >___|  /_______  (____  / ____||____/__\___  /|___|  /__|
            \/|__|        \/     \/        \/     \/\/            /_____/      \/


Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.
```

2.1 Installing OpenDaylight features

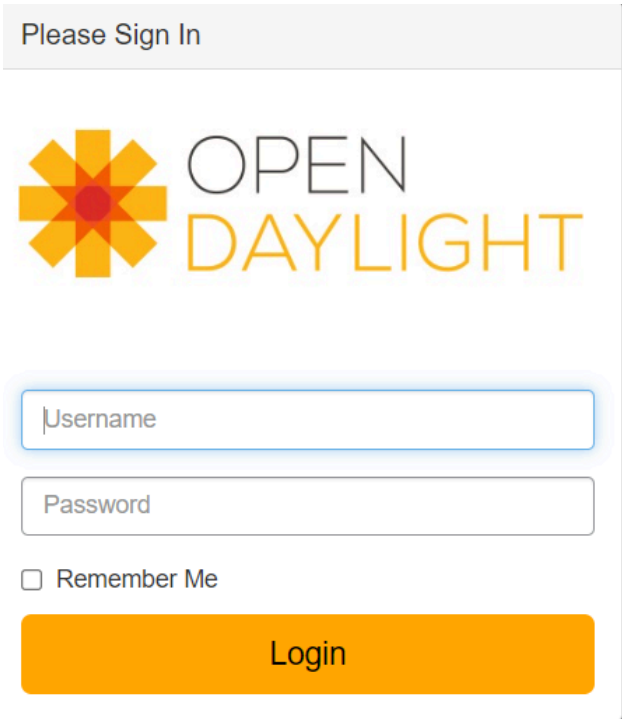
```
Name                                    | Version          | Required | State   | Repository
                                        | Description
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------
odl-openflowjava-protocol               | 0.5.2            |          | Started | odl-openflowja
va-0.5.2                                | ODL :: openflowjava :: odl-openflowjava-protocol
odl-mdsal-broker                        | 1.6.2            |          | Started | odl-mdsal-1.6.
2                                       | odl-mdsal-broker
features-dluxapps                       | 0.6.2            | x        | Started | features-dluxa
pps                                     | ODL :: dluxapps :: features-dluxapps
odl-mdsal-eos-binding                   | 2.3.2            |          | Started | odl-mdsal-eos-
binding                                 | OpenDaylight :: MD-SAL :: EOS :: Binding
jdbc                                    | 4.0.10           |          | Started | enterprise-4.0
.10                                     | JDBC service and commands
odl-l2switch-switch                     | 0.6.2            | x        | Started | odl-l2switch-s
witch                                   | OpenDaylight :: L2Switch :: Switch
pax-jdbc-spec                           | 1.0.1            |          | Started | org.ops4j.pax.
jdbc-1.0.1                              | Provides OSGi JDBC Service spec
pax-jdbc                                | 1.0.1            |          | Started | org.ops4j.pax.
jdbc-1.0.1                              | Provides JDBC Service support
pax-jdbc-config                         | 1.0.1            |          | Started | org.ops4j.pax.
jdbc-1.0.1                              | Provides JDBC Config support
odl-mdsal-common                        | 1.6.2            |          | Started | odl-mdsal-comm
on                                      | OpenDaylight :: Config :: All
pax-jetty                               | 9.2.21.v20170120 |          | Started | org.ops4j.pax.
web-4.3.4                               | Provide Jetty engine support
pax-http-jetty                          | 4.3.4            |          | Started | org.ops4j.pax.
web-4.3.4                               |
pax-http                                | 4.3.4            |          | Started | org.ops4j.pax.
web-4.3.4                               | Implementation of the OSGI HTTP Service
pax-http-whiteboard                     | 4.3.4            |          | Started | org.ops4j.pax.
web-4.3.4                               | Provide HTTP Whiteboard pattern support
pax-war                                 | 4.3.4            |          | Started | org.ops4j.pax.
web-4.3.4                               | Provide support of a full WebContainer
odl-yangtools-yang-data                 | 1.2.2            |          | Started | odl-yangtools-
yang-data                               | OpenDaylight :: Yangtools :: Data Binding
odl-mdsal-clustering-commons            | 1.6.2            |          | Started | odl-mdsal-clus
tering-commons                          | odl-mdsal-clustering-commons
odl-akka-system-2.4                     | 2.0.5            |          | Started | odl-akka-syste
m-2.4                                   | Akka Actor Framework System Bundles
odl-mdsal-models                        | 0.11.2           |          | Started | odl-mdsal-mode
ls                                      | OpenDaylight :: MD-SAL :: Models
odl-openflowplugin-app-topology         | 0.5.2            |          | Started | odl-openflowpl
ugin-app-topology                       | OpenDaylight :: Openflow Plugin :: Application -
odl-karaf-feat-jdbc                     | 2.0.5            |          | Started | odl-karaf-feat
-jdbc                                   | ODL :: odlparent :: odl-karaf-feat-jdbc
odl-lmax-3                              | 2.0.5            |          | Started | odl-lmax-3
                                        | OpenDaylight :: LMAX Disruptor
odl-guava-22                            | 2.0.5            |          | Started | odl-guava-22
                                        | OpenDaylight :: Guava 22 (for Karaf 4)
odl-mdsal-remoterpc-connector           | 1.6.2            |          | Started | odl-mdsal-remo
terpc-connector                         | odl-mdsal-remoterpc-connector
odl-openflowplugin-flow-services        | 0.5.2            |          | Started | odl-openflowpl
ugin-flow-services                      | OpenDaylight :: Openflow Plugin :: Flow Services
odl-akka-clustering-2.4                 | 2.0.5            |          | Started | odl-akka-clust
ering-2.4                               | Akka Clustering
odl-mdsal-binding-dom-adapter           | 2.3.2            |          | Started | odl-mdsal-bind
ing-dom-adapter                         | OpenDaylight :: MD-SAL :: DOM Adapter
odl-yangtools-common                    | 1.2.2            |          | Started | odl-yangtools-
common                                  | OpenDaylight :: Yangtools :: Common
odl-config-manager                      | 0.7.2            |          | Started | odl-config-man
ager                                    | odl-config-manager
odl-openflowplugin-app-config-pusher    | 0.5.2            |          | Started | odl-openflowpl
ugin-app-config-pusher                  | OpenDaylight :: Openflow Plugin :: Application -
odl-mdsal-binding-api                   | 2.3.2            |          | Started | odl-mdsal-bind
ing-api                                 | OpenDaylight :: MD-SAL :: Binding API
odl-aaa-shiro                           | 0.6.2            |          | Started | odl-aaa-0.6.2
                                        | ODL :: aaa :: odl-aaa-shiro
odl-mdsal-binding-base                  | 2.3.2            |          | Started | odl-mdsal-bind
ing-base                                | OpenDaylight :: MD-SAL :: Binding Base Concepts
odl-restconf                            | 1.6.2            | x        | Started | odl-restconf
                                        | OpenDaylight :: Restconf
odl-l2switch-arphandler                 | 0.6.2            |          | Started | odl-l2switch-a
rphandler                               | OpenDaylight :: L2Switch :: ArpHandler
odl-akka-leveldb-0.7                    | 2.0.5            |          | Started | odl-akka-level
db-0.7                                  | LevelDB
odl-dluxapps-yangman                    | 0.6.2            | x        | Started | odl-dluxapps-y
angman                                  | ODL :: dluxapps :: odl-dluxapps-yangman
odl-aaa-api                             | 0.6.2            |          | Started | odl-aaa-api
                                        | ODL :: aaa :: odl-aaa-api
odl-karaf-feat-jetty                    | 2.0.5            |          | Started | odl-karaf-feat
-jetty                                  | ODL :: odlparent :: odl-karaf-feat-jetty
odl-mdsal-singleton-dom                 | 2.3.2            |          | Started | odl-mdsal-sing
leton-dom                               | OpenDaylight :: MD-SAL :: Singleton :: DOM
odl-openflowplugin-app-forwardingrules-manager  | 0.5.2            |          | Started | odl-openflowpl
ugin-app-forwardingrules-manager        | OpenDaylight :: Openflow Plugin :: Application -
odl-config-startup                      | 0.7.2            |          | Started | odl-config-sta
```

```
odl-mdsal-distributed-datastore                | 1.6.2            |          | Started | odl-mdsal-dist
ributed-datastore                  | odl-mdsal-distributed-datastore
odl-mdsal-apidocs                              | 1.6.2            | x        | Started | odl-mdsal-apid
ocs                                | OpenDaylight :: MDSAL :: APIDOCS
odl-l2switch-loopremover                       | 0.6.2            |          | Started | odl-l2switch-l
oopremover                         | OpenDaylight :: L2Switch :: LoopRemover
odl-mdsal-dom                                  | 2.3.2            |          | Started | odl-mdsal-dom
                                   | OpenDaylight :: MD-SAL :: DOM
odl-mdsal-dom-broker                           | 2.3.2            |          | Started | odl-mdsal-dom-
broker                             | OpenDaylight :: MD-SAL :: DOM Broker
odl-mdsal-dom-api                              | 2.3.2            |          | Started | odl-mdsal-dom-
api                                | OpenDaylight :: MD-SAL :: DOM API and SPI
odl-mdsal-common                               | 2.3.2            |          | Started | odl-mdsal-comm
on                                 | OpenDaylight :: MD-SAL :: Common
odl-mdsal-binding-runtime                      | 2.3.2            |          | Started | odl-mdsal-bind
ing-runtime                        | OpenDaylight :: MD-SAL :: Binding Generator
aries-proxy                                    | 4.0.10           |          | Started | standard-4.0.1
0                                  | Aries Proxy
aries-blueprint                                | 4.0.10           |          | Started | standard-4.0.1
0                                  | Aries Blueprint
feature                                        | 4.0.10           |          | Started | standard-4.0.1
0                                  | Features Support
shell                                          | 4.0.10           |          | Started | standard-4.0.1
0                                  | Karaf Shell
shell-compat                                   | 4.0.10           |          | Started | standard-4.0.1
0                                  | Karaf Shell Compatibility
deployer                                       | 4.0.10           |          | Started | standard-4.0.1
0                                  | Karaf Deployer
bundle                                         | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide Bundle support
config                                         | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide OSGi ConfigAdmin support
diagnostic                                     | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide Diagnostic support
instance                                       | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide Instance support
jaas                                           | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide JAAS support
log                                            | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide Log support
package                                        | 4.0.10           |          | Started | standard-4.0.1
0                                  | Package commands and mbeans
service                                        | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide Service support
system                                         | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide System support
```

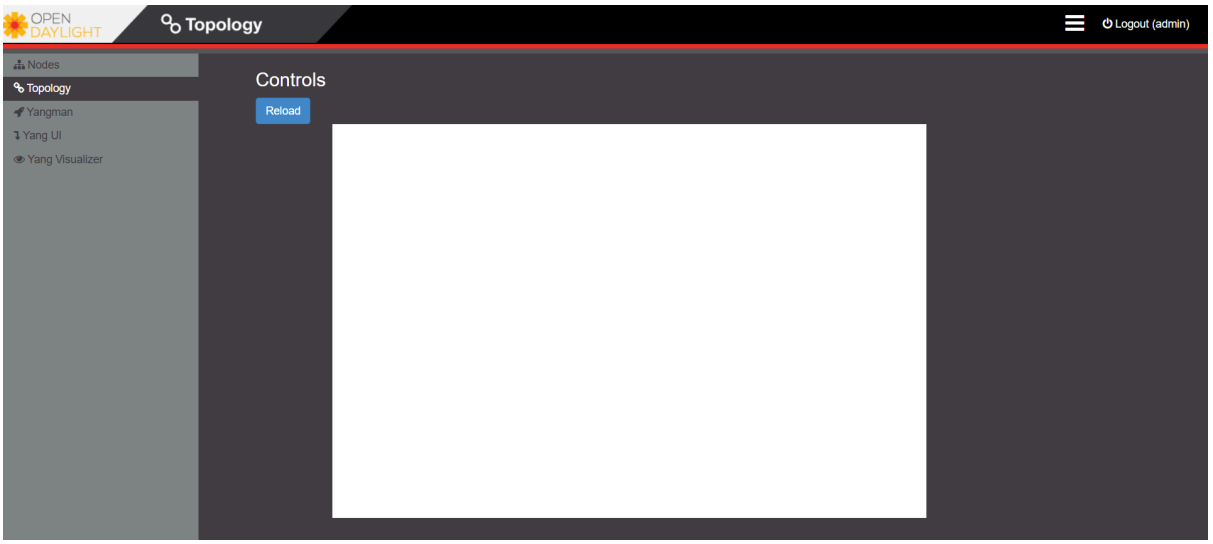
```
kar                                            | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide KAR (KARaf archive) support
ssh                                            | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide a SSHd server on Karaf
management                                     | 4.0.10           |          | Started | standard-4.0.1
0                                  | Provide a JMX MBeanServer and a set of MBeans in
wrap                                           | 0.0.0            | x        | Started | standard-4.0.1
0                                  | Wrap URL handler
standard                                       | 4.0.10           | x        | Started | standard-4.0.1
0                                  | Wrap feature describing all features part of a st
odl-config-netty-config-api                    | 0.7.2            |          | Started | odl-config-net
ty-config-api                      | odl-config-netty-config-api
odl-mdsal-singleton-common                     | 2.3.2            |          | Started | odl-mdsal-sing
leton-common                       | OpenDaylight :: MD-SAL :: Singleton :: Common
odl-config-api                                 | 0.7.2            |          | Started | odl-config-api
                                   | odl-config-api
odl-netty-4                                    | 2.0.5            |          | Started | odl-netty-4
                                   | OpenDaylight :: Netty
odl-l2switch-packethandler                     | 0.6.2            |          | Started | odl-l2switch-p
ackethandler                       | OpenDaylight :: L2Switch :: PacketHandler
odl-dluxapps-topology                          | 0.6.2            | x        | Started | odl-dluxapps-t
opology                            | ODL :: dluxapps :: odl-dluxapps-topology
odl-javassist-3                                | 2.0.5            |          | Started | odl-javassist-
3                                  | OpenDaylight :: Javassist
odl-aaa-encryption-service                     | 0.6.2            |          | Started | odl-aaa-0.6.2
                                   | ODL :: aaa :: odl-aaa-encryption-service
odl-akka-scala-2.11                            | 2.0.5            |          | Started | odl-akka-scala
-2.11                              | Scala Runtime for OpenDaylight
odl-mdsal-eos-dom                              | 2.3.2            |          | Started | odl-mdsal-eos-
dom                                | OpenDaylight :: MD-SAL :: EOS :: DOM
odl-openflowplugin-nsf-model                   | 0.5.2            |          | Started | odl-openflowpl
ugin-nsf-model                     | OpenDaylight :: OpenflowPlugin :: NSF :: Model
odl-dluxapps-nodes                             | 0.6.2            | x        | Started | odl-dluxapps-n
odes                               | ODL :: dluxapps :: odl-dluxapps-nodes
odl-dlux-core                                  | 0.6.2            | x        | Started | odl-dlux-0.6.2
                                   | Opendaylight dlux minimal feature
odl-l2switch-addresstracker                    | 0.6.2            |          | Started | odl-l2switch-a
ddresstracker                      | OpenDaylight :: L2Switch :: AddressTracker
odl-aaa-cert                                   | 0.6.2            |          | Started | odl-aaa-0.6.2
                                   | ODL :: aaa :: odl-aaa-cert
odl-yangtools-yang-parser                      | 1.2.2            |          | Started | odl-yangtools-
yang-parser                        | OpenDaylight :: Yangtools :: YANG Parser
```

2.2 OpenDaylight Log In

Username: admin
Password: admin

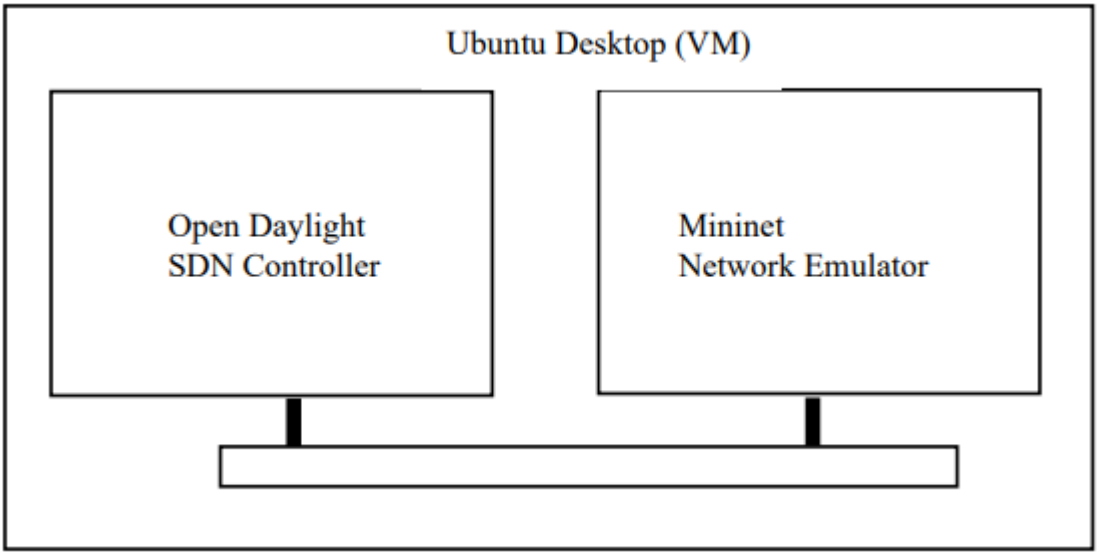


OpenDaylight GUI



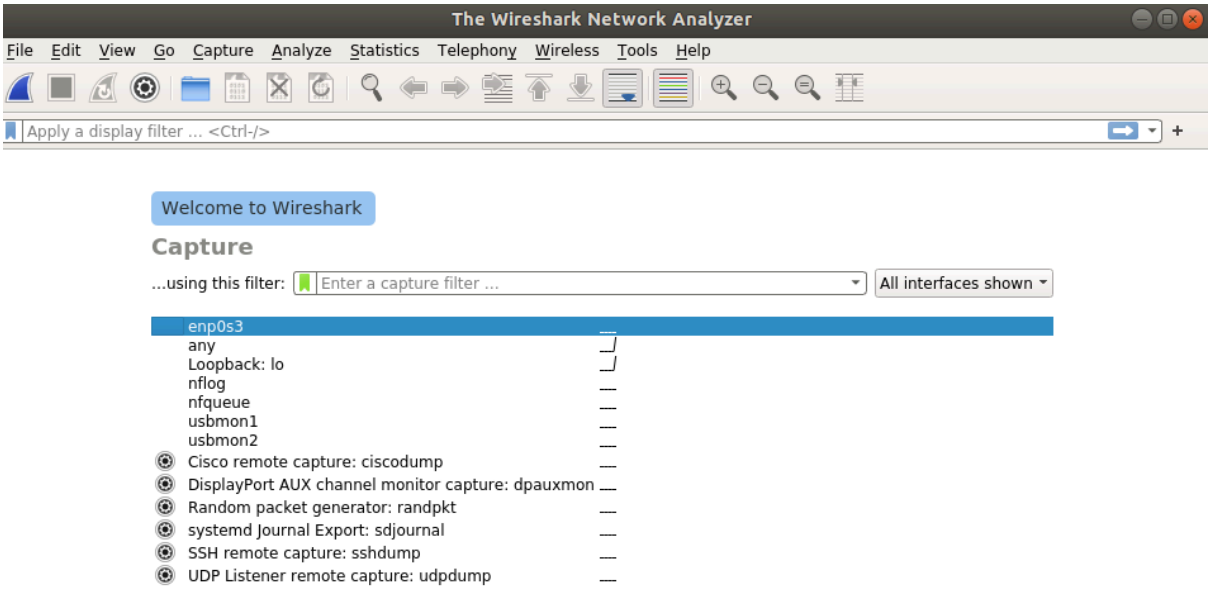
Part 3

Build a simple Mininet network using the OpenDaylight OpenFlow controller



3.1 Starting Wireshark in Virtual Machine

Select Loopback:lo as the interface



3.3 Creating a simple network on the Mininet VM

```
sdn@sdn-VirtualBox:~$ sudo mn --mac --controller=remote,ip=192.168.56.102,port=6
633 --switch ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
```

3.4 Checking the flow table

It contains the following:

cookie=0x2b00000000000000, duration=955.777s, table=0, n_packets=0, n_bytes=0, priority=100,dl_type=0x88cc actions=CONTROLLER:65535

This flow rule has been active for 955.777 seconds and it has a high priority of 100, ensuring it takes precedence over other rules. The entry specifically matches packets with an Ethernet type of 0x88cc, which corresponds to LLDP packets used for network discovery. No packets have matched this rule so far. The action associated with this entry is to send any matching packets directly to the controller with a buffer size of 65535 bytes.

cookie=0x2b00000000000000, duration=953.896s, table=0, n_packets=7, n_bytes=490, priority=2,in_port="s1-eth1" actions=output:"s1-eth2",CONTROLLER:65535

This flow rule has been active for about 953.896 seconds. It matches packets arriving on port s1-eth1 with a lower priority of 2. 7 packets, totaling 490 bytes, have matched this rule. The action defined in this entry is to forward the packets to port s1-eth2 and simultaneously send a copy of these packets to the controller with a maximum buffer size of 65535 bytes.

cookie=0x2b00000000000001, duration=953.889s, table=0, n_packets=7, n_bytes=490, priority=2,in_port="s1-eth2" actions=output:"s1-eth1",CONTROLLER:65535

This flow rule has been active for approximately 953.889 seconds. It is similar to the second entry but applies to packets arriving on port s1-eth2. Like the second entry, it has a priority of 2 and has matched 7 packets, totaling 490 bytes. The action for this rule is to forward the packets to port s1-eth1 and also send a copy to the controller with a buffer size of 65535B.

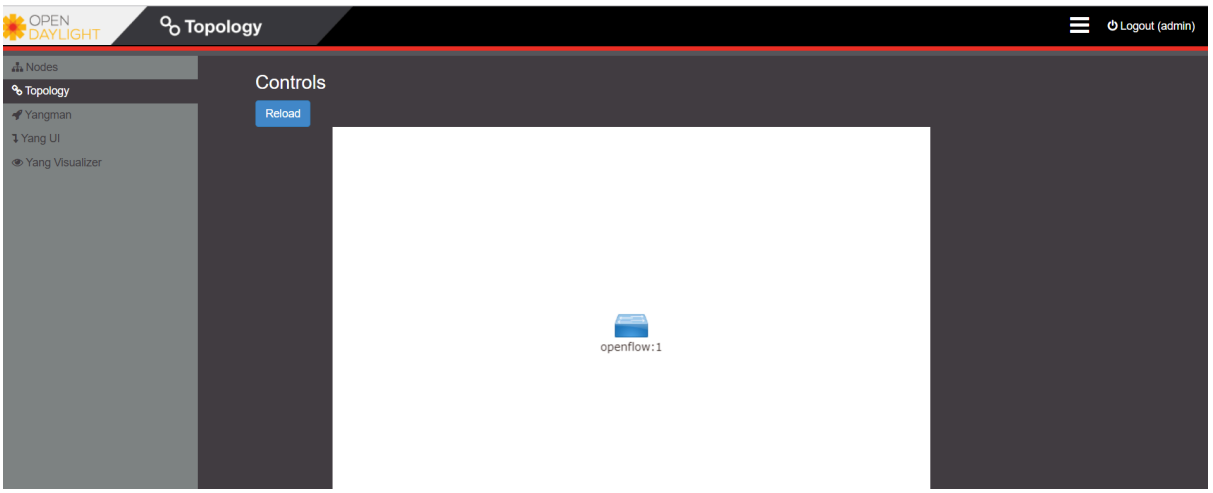
cookie=0x2b00000000000000, duration=955.777s, table=0, n_packets=2, n_bytes=180, priority=0 actions=drop

The final flow rule has been active for about 955.777 seconds. This rule, with the lowest priority of 0, acts as a catch-all for any packets that do not match higher-priority rules. So far, it has matched two packets, totaling 180 bytes. The action specified for this entry is to drop any matching packets, effectively discarding them from the network.

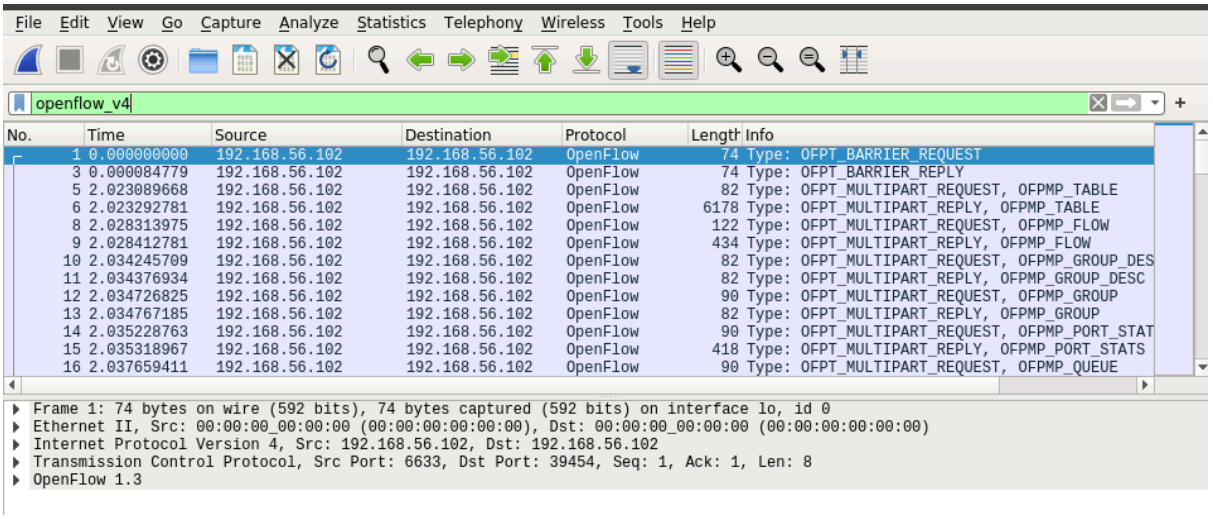
```
mininet> dpctl dump-flows --protocols=OpenFlow13
*** s1 ------------------------------------------------------------------------
 cookie=0x2b00000000000000, duration=955.777s, table=0, n_packets=0, n_bytes=0,
priority=100,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x2b00000000000000, duration=953.896s, table=0, n_packets=7, n_bytes=490
, priority=2,in_port="s1-eth1" actions=output:"s1-eth2",CONTROLLER:65535
 cookie=0x2b00000000000001, duration=953.889s, table=0, n_packets=7, n_bytes=490
, priority=2,in_port="s1-eth2" actions=output:"s1-eth1",CONTROLLER:65535
 cookie=0x2b00000000000000, duration=955.777s, table=0, n_packets=2, n_bytes=180
, priority=0 actions=drop
```

3.5 Initial OpenDaylight GUI

GUI shows only a switch. This is because the controller has not recognized any flows to be established between the devices connected to the switch. Therefore, the network activity is not reflected visually in the GUI.



3.6 Wireshark capture



3.8 Pinging

Yes, the pings succeed. This is because the OpenDaylight controller actively manages the switch. When a ping request is sent, the switch sends a PACKET_IN message to the controller, which then decides how to handle the packet and responds with a FLOW_MOD message to install the necessary flow rule. This process allows for successful communication between the hosts.

```
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_seq=1 ttl=64 time=0.221 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.066 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.060/0.105/0.221/0.067 ms
```

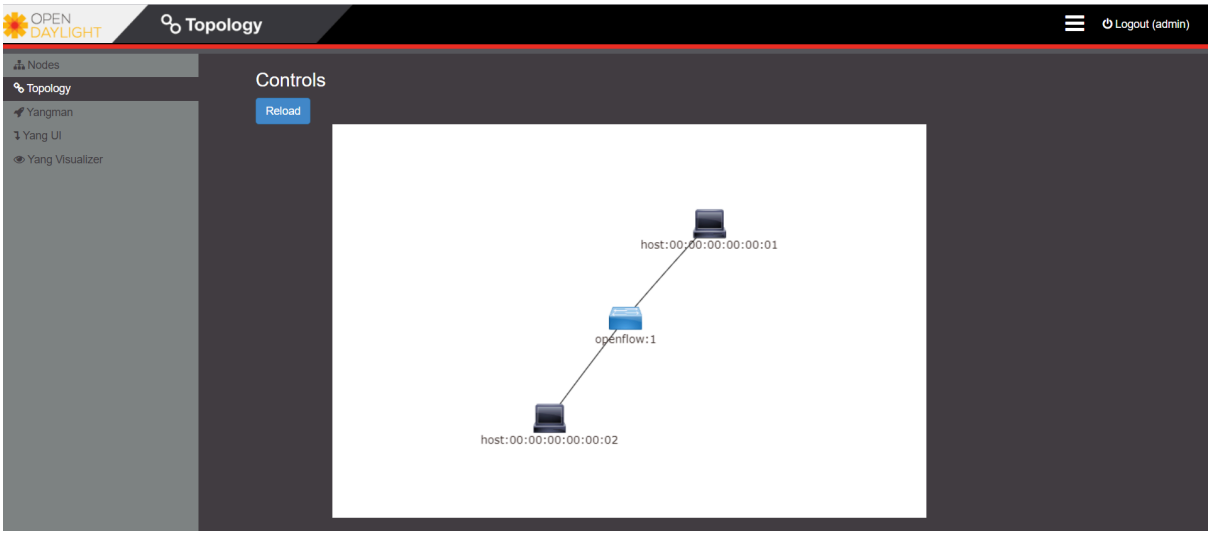
3.9 Checking the flow table

The flow table has several entries corresponding to the ping traffic. These entries were added by the OpenDaylight controller to manage the flow of packets between the hosts. The rules typically match on Ethernet headers and direct the packets to the appropriate ports, allowing bidirectional communication.

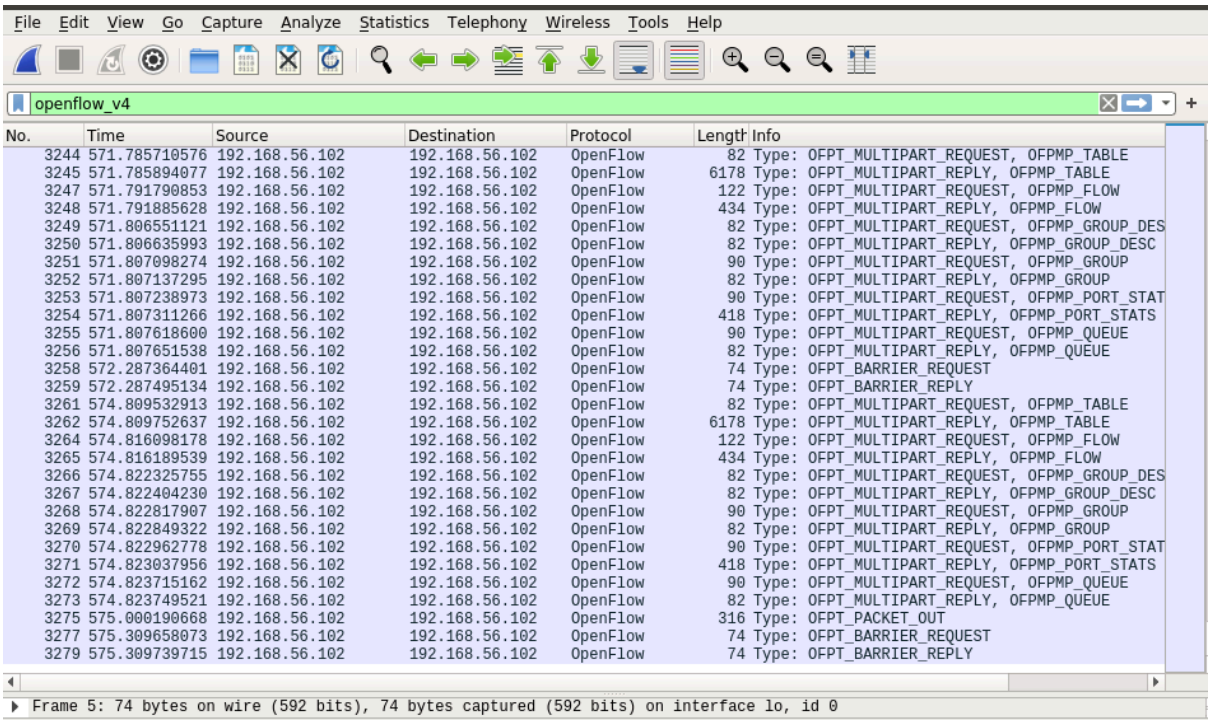
```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
2024-08-27T09:27:25Z|00001|vconn|WARN|unix:/var/run/openvswitch/s1.mgmt: version
 negotiation failed (we support version 0x01, peer supports version 0x04)
ovs-ofctl: s1: failed to connect to socket (Broken pipe)
```

3.10 OpenDaylight GUI after pinging

Yes, The topology has changed. Previously only the controller was shown. Now the controller and 2 hosts are also shown

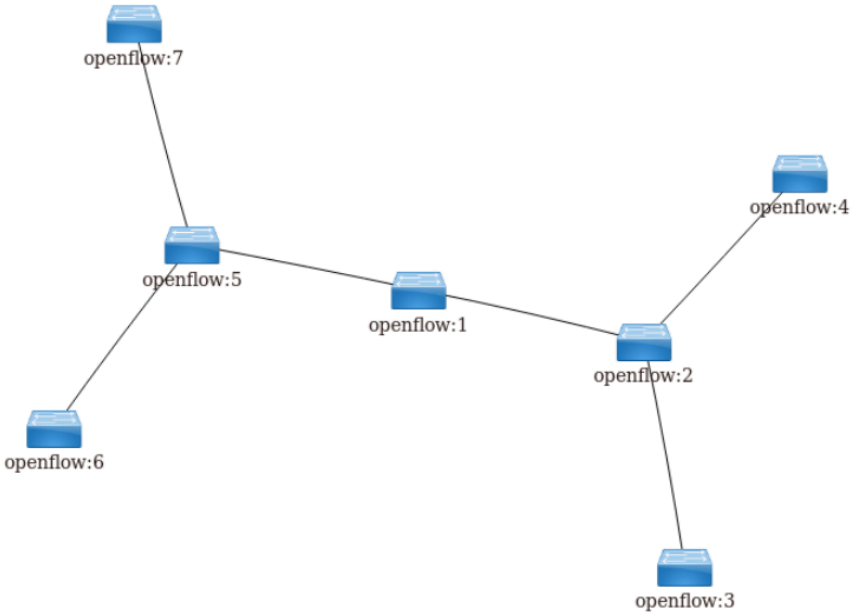


3.11 Wireshark capture

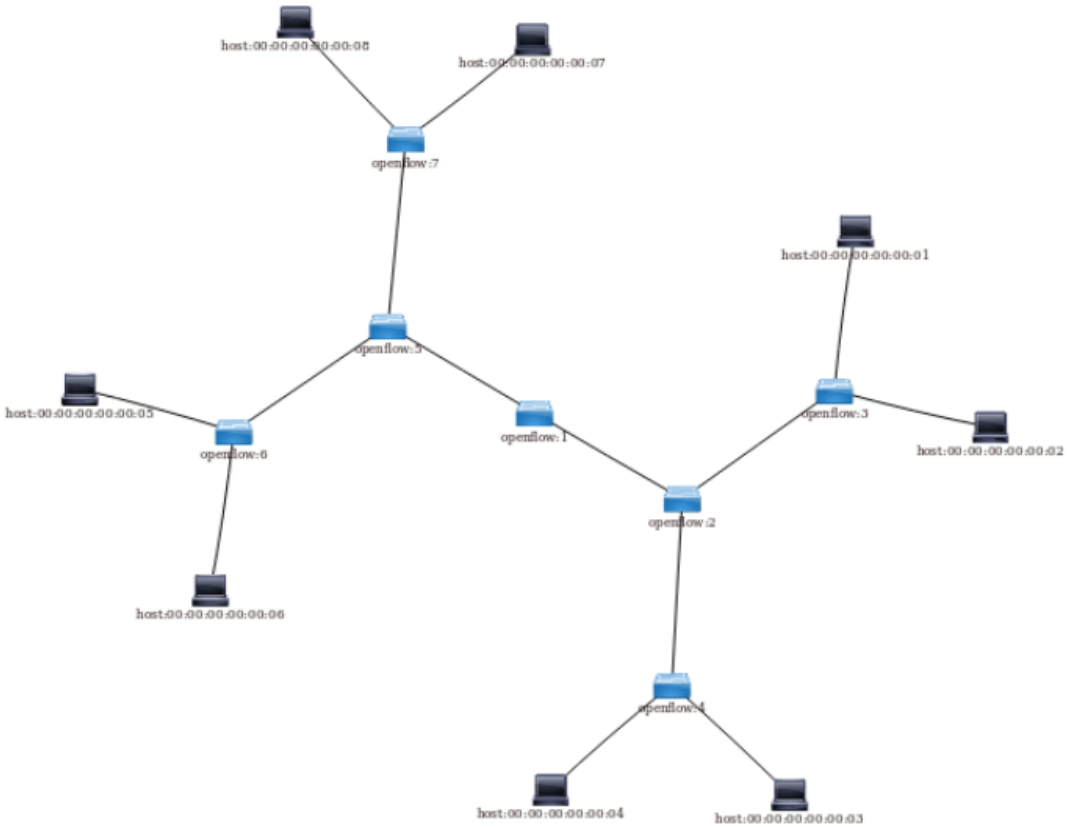


3.13 More network topologies

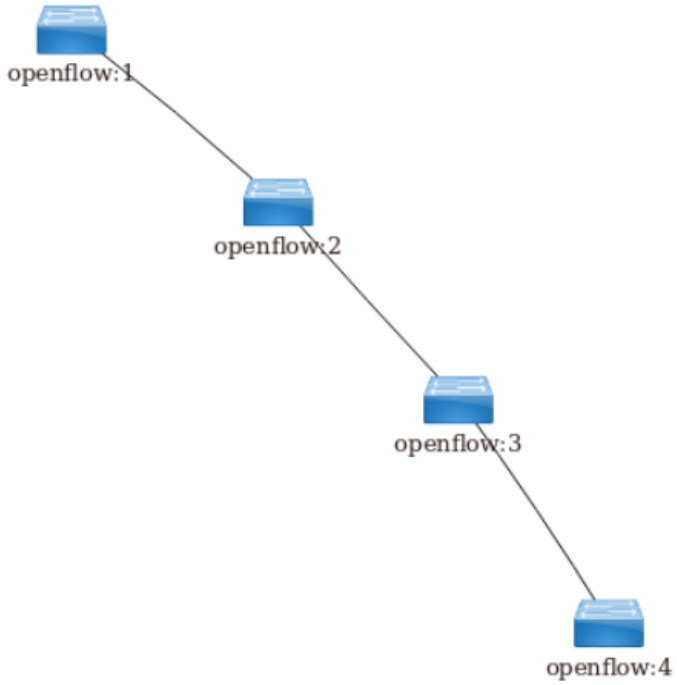
Case 1 - Before ping



Case 1 - After ping



Case 2 - Before ping



Case 2 - After ping