Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN2150 - Communication Network Engineering

SDN Lab Report

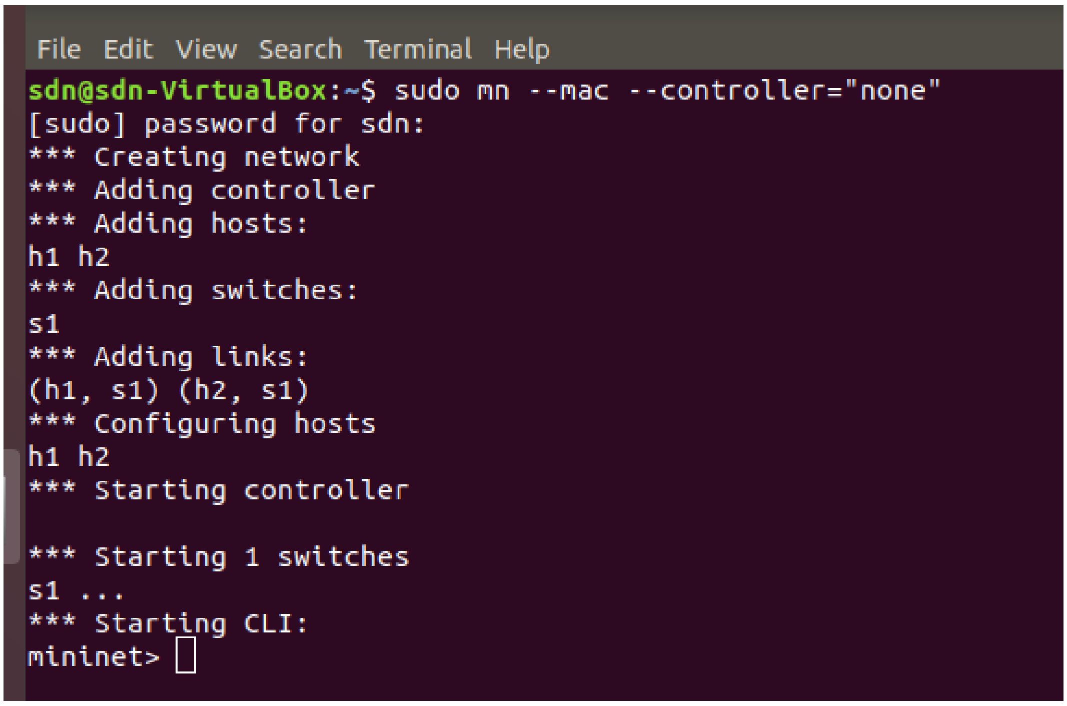
28 August 2024

210387D Mihiranga N.G.D.

Table of Contents

Part 1 : Build a standalone Mininet network.

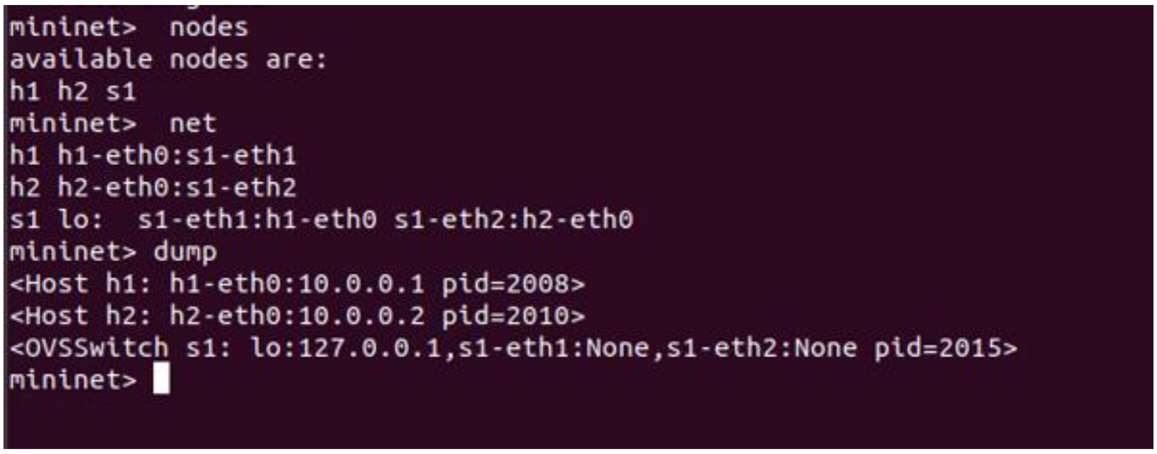
1. Create a simple network on the Mininet



```
sdn@sdn-VirtualBox:~$ sudo mn --mac --controller="none"
[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:
mininet> 
```

Observation: Creating the network

2.Network Connections



```
mininet>  nodes
available nodes are:
h1 h2 s1
mininet>  net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2008>
<Host h2: h2-eth0:10.0.0.2 pid=2010>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=2015>
mininet> 
```

Observation:

nodes : Lists available nodes
net : Shows links between nodes
dump: Provide detailed information on each node.

Network Topology:

- h1 is connected to s1 through h1-eth0 and s1-eth1.
- h2 is connected to s1 through h2-eth0 and s1-eth2.

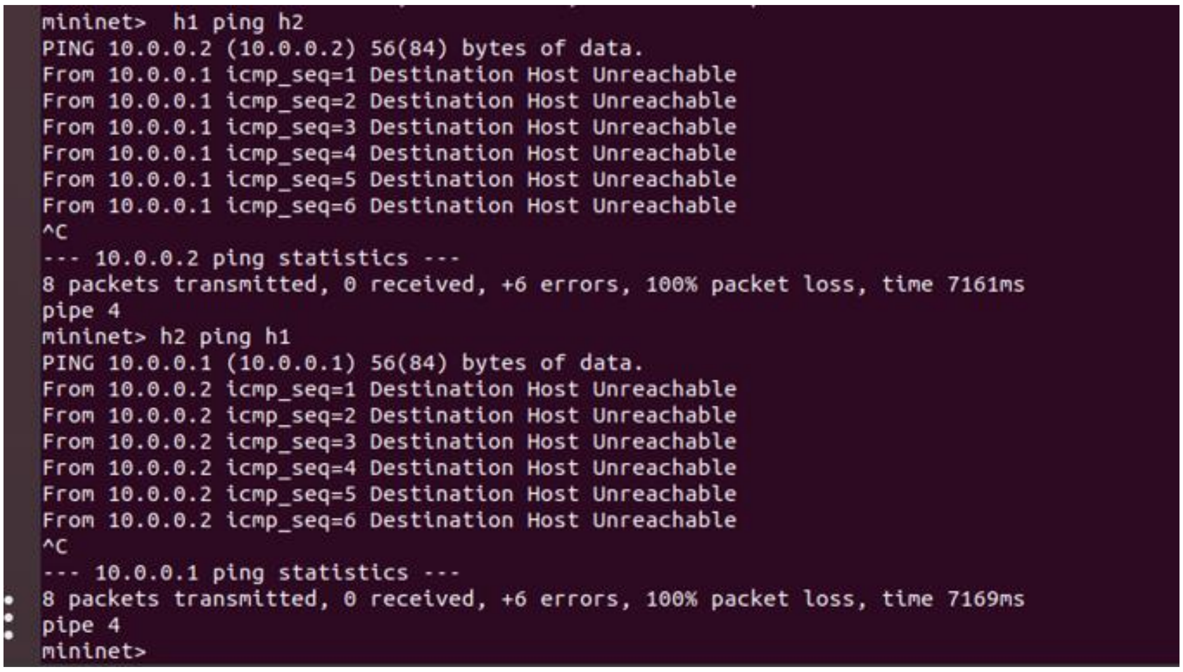
IP Addresses:

- h1 has the IP address 10.0.0.1.
- h2 has the IP address 10.0.0.2.

Switch (s1) Interface States:

- s1-eth1 and s1-eth2 interfaces currently have no peer connected (None).

3.Pinging between hosts



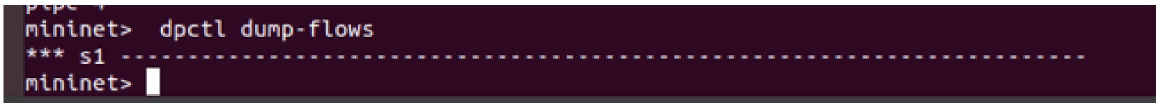
```
mininet>  h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable
From 10.0.0.1 icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7161ms
pipe 4
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
^C
--- 10.0.0.1 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7169ms
pipe 4
mininet>
```

Observations:

Ping did not succeed.

The switch's interfaces (s1-eth1 and s1-eth2) were reported as None, indicating that the switch was not recognizing any active connections on those interfaces.
Hence swich do not have rules to forward packets.

4.Flow Table

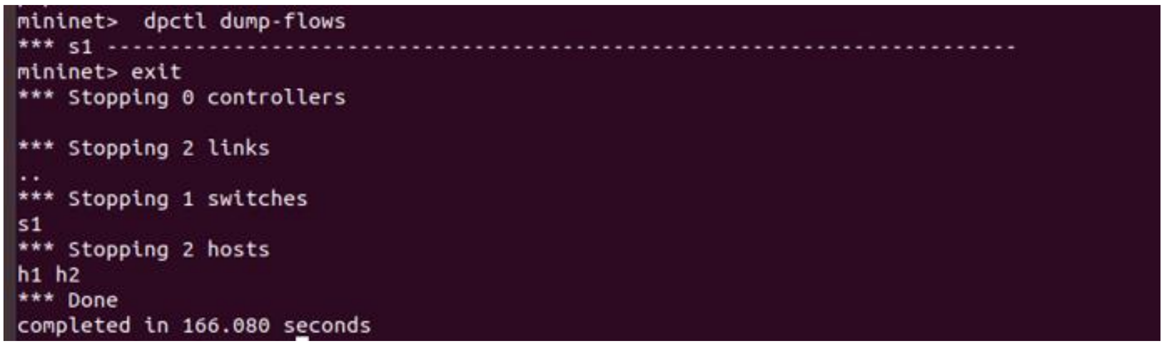


```
mininet>  dpctl dump-flows
*** s1 -----------------------------------------------------------------
mininet> █
```

Observations:

The absence of flow entries means that the switch s1 does not have any flow rules installed in its flow table due to there is no controller configured.

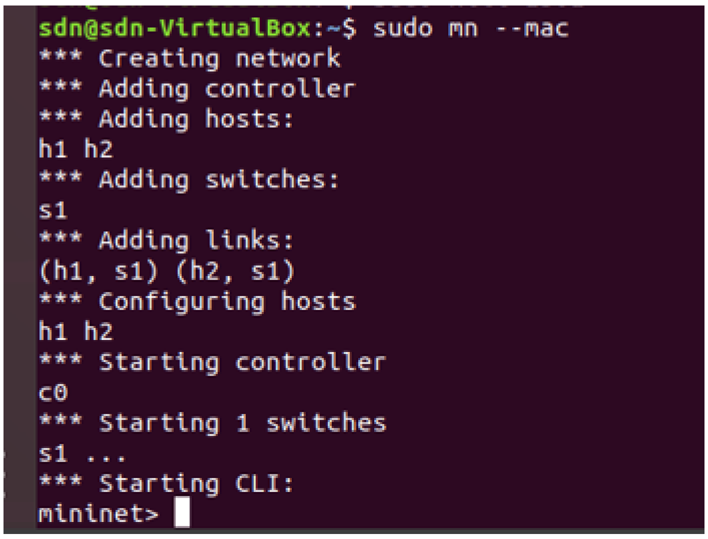
5.Exiting form mininet



```
mininet>  dpctl dump-flows
*** s1 ------------------------------------------------------------------------
mininet> exit
*** Stopping 0 controllers

*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 166.080 seconds
```

Observations: Existing from the Network

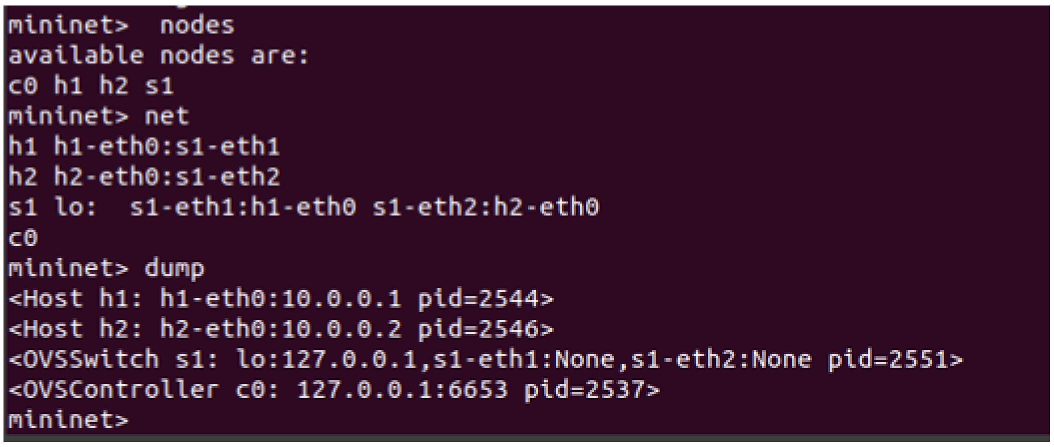
6. Create a simple network on the Mininet



```
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>
```

Observation: Creating the network with a controller.

7. Network Connections



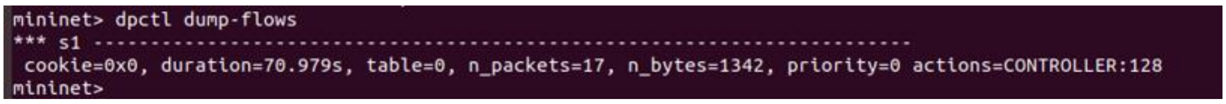
```
mininet>  nodes
available nodes are:
c0 h1 h2 s1
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2544>
<Host h2: h2-eth0:10.0.0.2 pid=2546>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=2551>
<OVSController c0: 127.0.0.1:6653 pid=2537>
mininet>
```

Observation:

nodes : Lists available nodes
net : Shows links between nodes
dump: Provide detailed information on each node.

In here when compared to previous network he controller c0 is listed as being at 127.0.0.1:6653.

8.Flow Table



```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=70.979s, table=0, n_packets=17, n_bytes=1342, priority=0 actions=CONTROLLER:128
mininet>
```

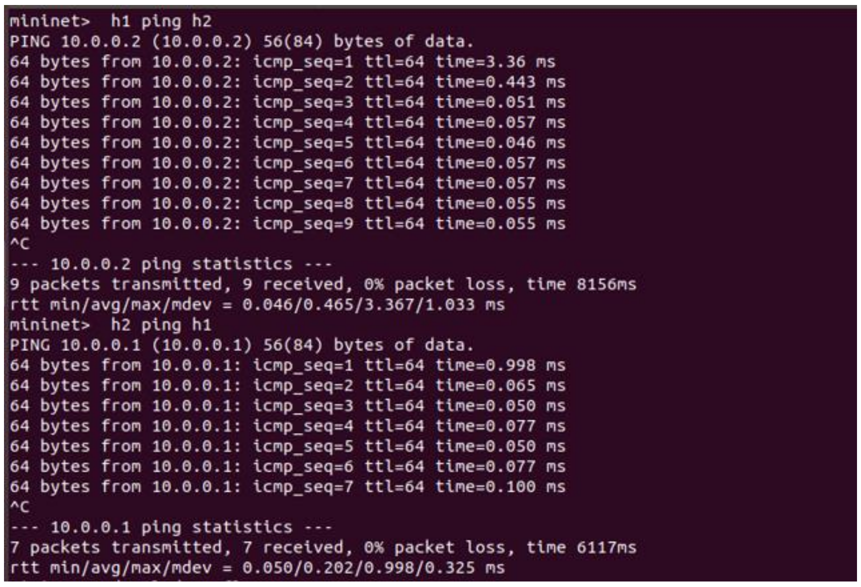
Reason :

The flow table includes a default entry that directs unmatched packets to the controller. This approach is standard in SDN environments, enabling the controller to make decisions about new or unknown traffic types. Upon detecting such traffic, the controller can install more specific flow rules tailored to the observed patterns. The default rule suggests that the switch is in a passive mode, awaiting instructions from the controller when it encounters new traffic. Since there are no predefined flow rules for host communication, any initial packets sent from h1 to h2 (or vice versa) will be forwarded to the controller by default.

Observation:

The flow entry in switch `s1`'s flow table is a default rule with a cookie value of `0x0`, indicating no special identification. It has been active for approximately 410 seconds and has matched 22 packets, processing a total of 1732 bytes. The entry is in the first flow table (`table=0`) and has the lowest priority (`priority=0`), acting as a catch-all rule. The action associated with this flow entry is to send packets to the controller, forwarding up to 128 bytes of each matching packet.

9.Pinging between hosts



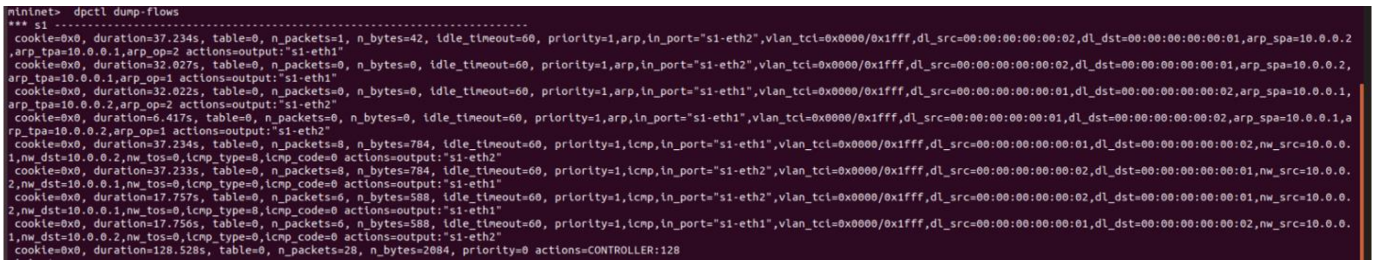
```
mininet>  h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.36 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.443 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.055 ms
^C
--- 10.0.0.2 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8156ms
rtt min/avg/max/mdev = 0.046/0.465/3.367/1.033 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=0.998 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.065 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.077 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.077 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.100 ms
^C
--- 10.0.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6117ms
rtt min/avg/max/mdev = 0.050/0.202/0.998/0.325 ms
```

Observations:

Ping succeeds.

The ping between h1 and h2 succeeds because the default flow entry in the switch s1's flow table is set to forward packets to the controller

10. Flow table (After Pinging, Before timeout)



```
mininet>  dpctl dump-flows
*** s1 -----------------------------------------------------------------------
 cookie=0x0, duration=37.234s, 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=output:"s1-eth1"
 cookie=0x0, duration=32.027s, table=0, n_packets=0, n_bytes=0, 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=output:"s1-eth1"
 cookie=0x0, duration=32.022s, 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=2 actions=output:"s1-eth2"
 cookie=0x0, duration=6.417s, 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,a
rp_tpa=10.0.0.2,arp_op=1 actions=output:"s1-eth2"
 cookie=0x0, duration=37.234s, table=0, n_packets=8, n_bytes=784, idle_timeout=60, priority=1,icmp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=37.233s, table=0, n_packets=8, n_bytes=784, idle_timeout=60, priority=1,icmp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=17.757s, table=0, n_packets=6, n_bytes=588, idle_timeout=60, priority=1,icmp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=17.756s, table=0, n_packets=6, n_bytes=588, idle_timeout=60, priority=1,icmp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=128.528s, table=0, n_packets=28, n_bytes=2084, priority=0 actions=CONTROLLER:128
```

Observations:

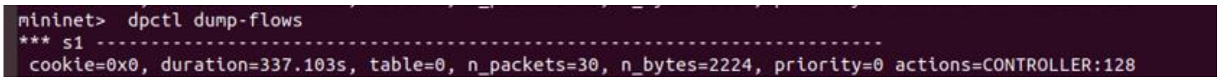
The flow table contained several flow entries related to ARP and ICMP. These entries had an idle_timeout=60, they would be removed from the flow table after 60 seconds of inactivity.

ARP Rules: The flow table includes entries for managing ARP requests and replies between h1 and h2. These rules enable the switch to process ARP packets, ensuring proper address resolution by directing them to the appropriate port.

ICMP Rules: The ICMP rules handle echo requests (ping) and their corresponding replies. These rules are configured with an idle_timeout=60, meaning they will be removed from the flow table if no matching packets are detected within 60 seconds.

Controller Rule: A default rule with priority 0 is also present, which forwards any unmatched packets to the controller (actions=CONTROLLER:128). This rule ensures that the controller handles any new or unanticipated traffic, allowing it to decide on the appropriate actions.

11. Flow table (After Pinging, After timeout)

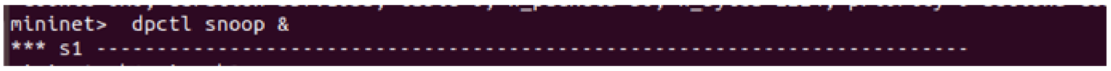


```
mininet>  dpctl dump-flows
*** s1 -----------------------------------------------------------------------
 cookie=0x0, duration=337.103s, table=0, n_packets=30, n_bytes=2224, priority=0 actions=CONTROLLER:128
```

Observations:

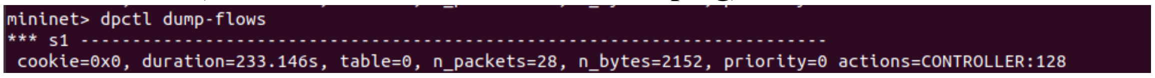
The flow entries were removed due to inactivity (idle_timeout=60). As a result, the flow table was left with only the default flow entry that forwards packets to the controller.

12.Enable Snooping



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

14.Flow table (After timeout , After Enable Snooping)

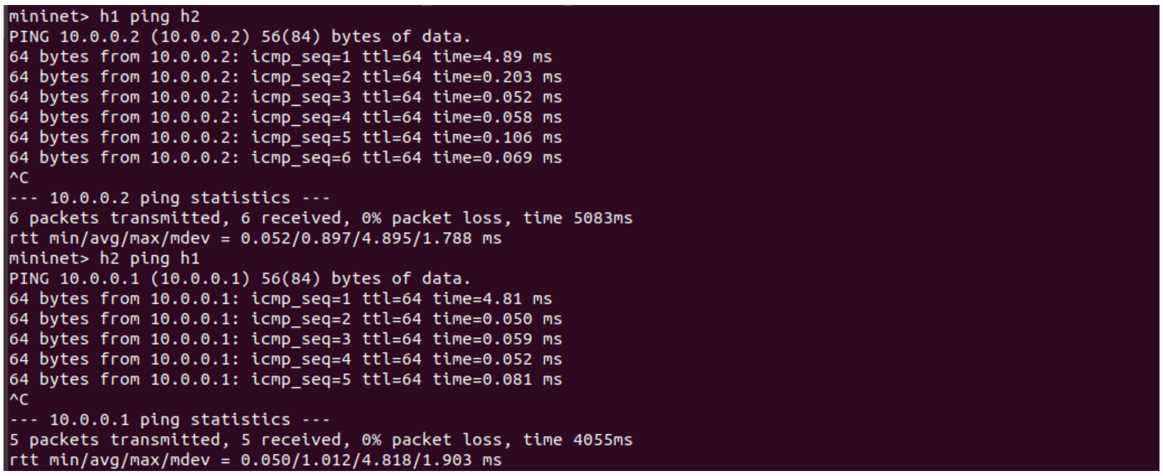


```
mininet> dpctl dump-flows
*** s1 -----------------------------------------------------------------------
 cookie=0x0, duration=233.146s, table=0, n_packets=28, n_bytes=2152, priority=0 actions=CONTROLLER:128
```

Observations:

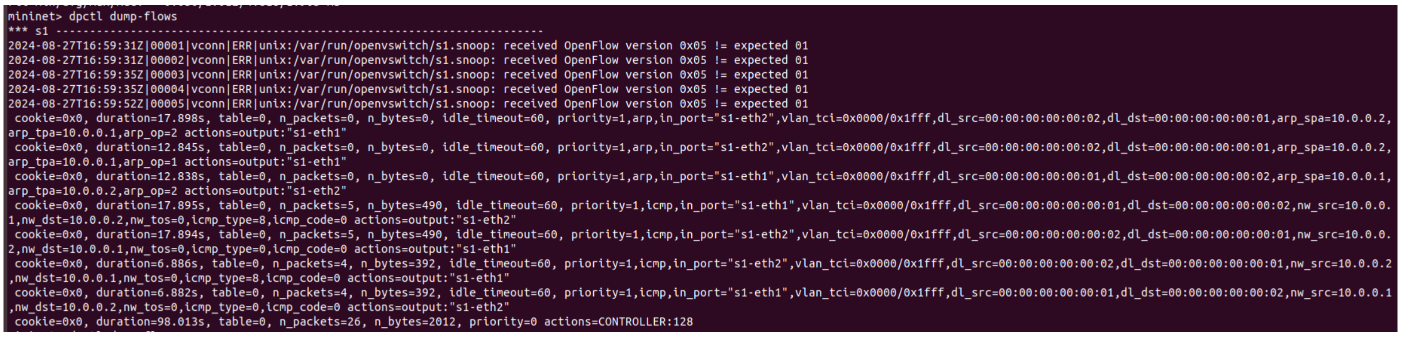
The flow entry on switch s1 has matched 28 packets and 2152 bytes over 233 seconds. With a priority of 0, it directs all matching packets to the controller, limiting the packet size to 128 bytes.

15.Pinging between 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=4.89 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.203 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.069 ms
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5083ms
rtt min/avg/max/mdev = 0.052/0.897/4.895/1.788 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=4.81 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.081 ms
^C
--- 10.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4055ms
rtt min/avg/max/mdev = 0.050/1.012/4.818/1.903 ms
```

16.Check Flow Table

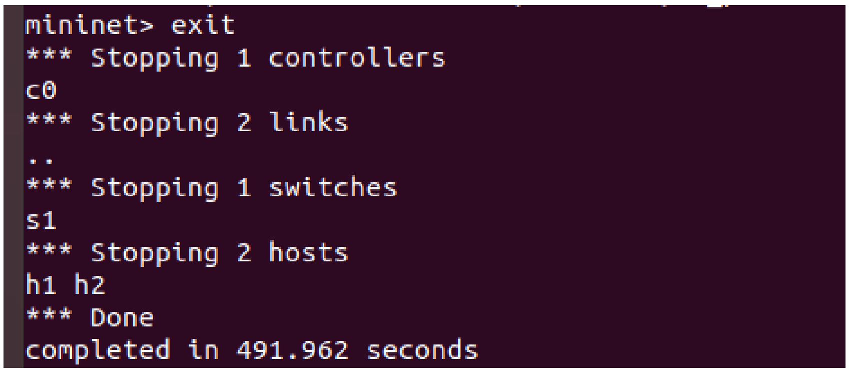


```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
2024-08-27T16:59:31Z|00001|vconn|ERR|unix:/var/run/openvswitch/s1.snoop: received OpenFlow version 0x05 != expected 01
2024-08-27T16:59:31Z|00002|vconn|ERR|unix:/var/run/openvswitch/s1.snoop: received OpenFlow version 0x05 != expected 01
2024-08-27T16:59:35Z|00003|vconn|ERR|unix:/var/run/openvswitch/s1.snoop: received OpenFlow version 0x05 != expected 01
2024-08-27T16:59:35Z|00004|vconn|ERR|unix:/var/run/openvswitch/s1.snoop: received OpenFlow version 0x05 != expected 01
2024-08-27T16:59:52Z|00005|vconn|ERR|unix:/var/run/openvswitch/s1.snoop: received OpenFlow version 0x05 != expected 01
 cookie=0x0, duration=17.898s, table=0, n_packets=0, n_bytes=0, 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=output:"s1-eth1"
 cookie=0x0, duration=12.845s, table=0, n_packets=0, n_bytes=0, 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=output:"s1-eth1"
 cookie=0x0, duration=12.838s, 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=2 actions=output:"s1-eth2"
 cookie=0x0, duration=17.895s, table=0, n_packets=5, n_bytes=490, idle_timeout=60, priority=1,icmp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=17.894s, table=0, n_packets=5, n_bytes=490, idle_timeout=60, priority=1,icmp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=6.886s, table=0, n_packets=4, n_bytes=392, idle_timeout=60, priority=1,icmp,in_port="s1-eth2",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=6.882s, table=0, n_packets=4, n_bytes=392, idle_timeout=60, priority=1,icmp,in_port="s1-eth1",vlan_tci=0x0000/0x1fff,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_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=98.013s, table=0, n_packets=26, n_bytes=2012, priority=0 actions=CONTROLLER:128
```

Observations:

The flow table will now show updated packet and byte counts for the flows corresponding to the recent ping activity between h1 and h2, while the default flow entry.

17. Exit from mininet

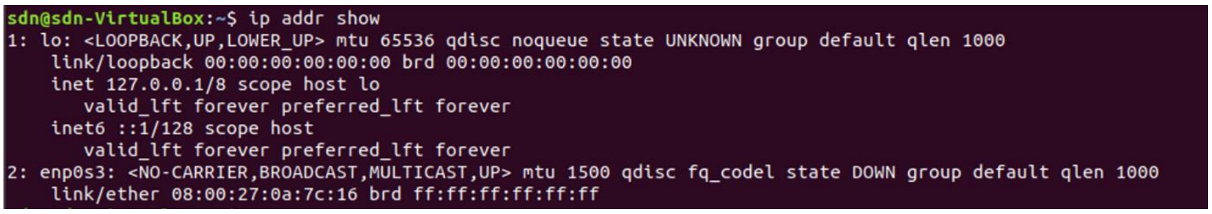


```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 491.962 seconds
```

Observations: Exiting the Network

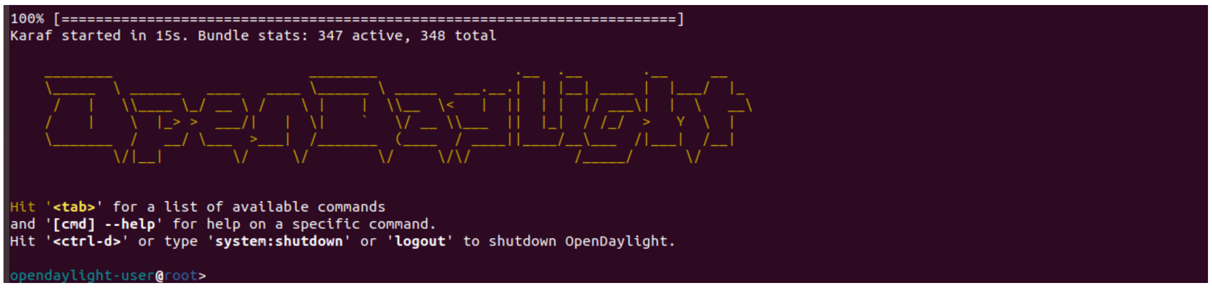
Part 2 Connect Mininet network to Opendaylight controller

1.Checking IP Address



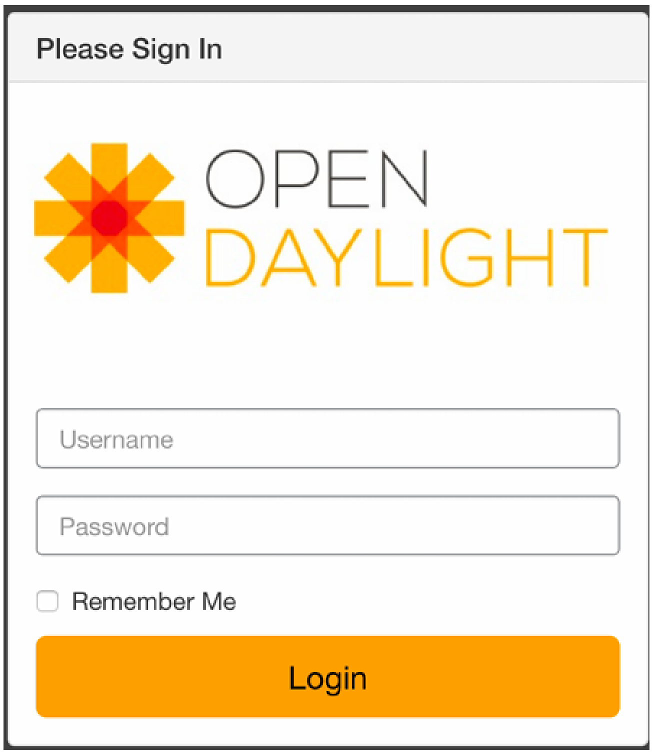
```
sdn@sdn-VirtualBox:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:00:27:0a:7c:16 brd ff:ff:ff:ff:ff:ff
```

2. Running OpenDaylight



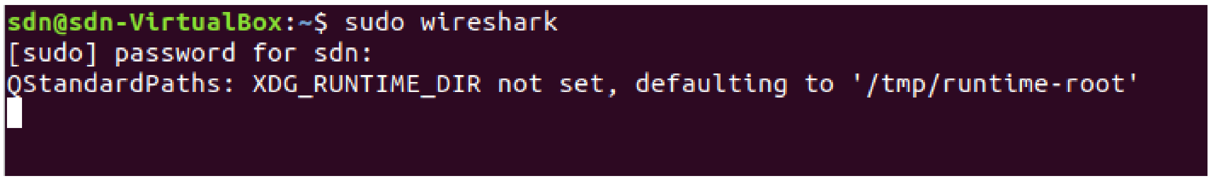
```
100% [========================================================================]
Karaf started in 15s. Bundle stats: 347 active, 348 total

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

opendaylight-user@root>
```

3. Opening up OpenDaylight GUI



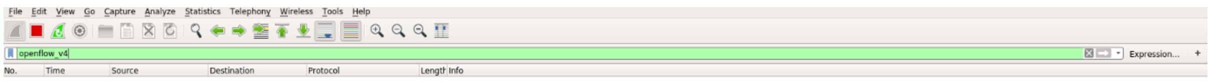
Part 3 Build a simple Mininet network using the OpenDaylight OpenFlow controller

1.Starting Wireshark



```
sdn@sdn-VirtualBox:~$ sudo wireshark
[sudo] password for sdn:
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```

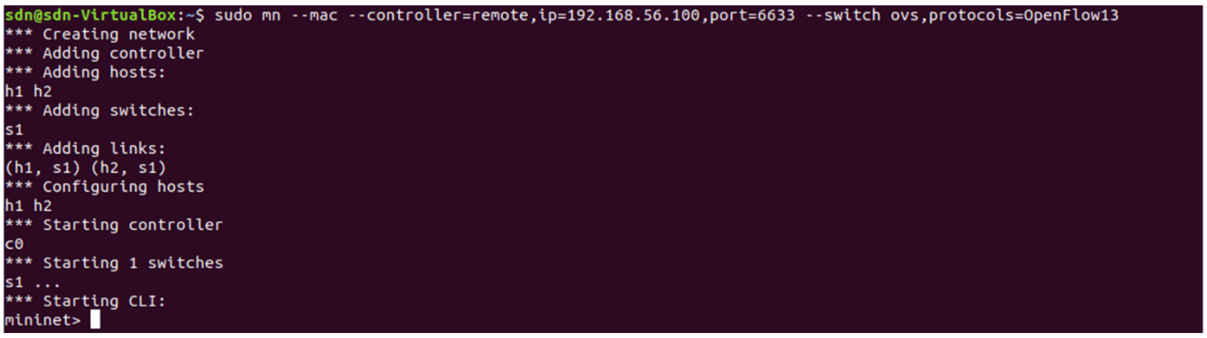
2.Applying Display Filter



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
No. Time Source Destination Protocol Length Info

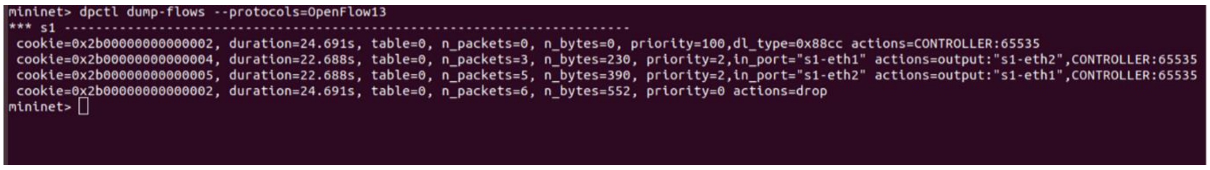
```
openflow_v4
```

3.Creating Simple network



```
sdn@sdn-VirtualBox:~$ sudo mn --mac --controller=remote,ip=192.168.56.100,port=6633 --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:
mininet>
```

4.Check Flow Table



```
mininet> dpctl dump-flows --protocols=OpenFlow13
*** s1 ------------------------------------------------------------------------
 cookie=0x2b00000000000002, duration=24.691s, table=0, n_packets=0, n_bytes=0, priority=100,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x2b00000000000004, duration=22.688s, table=0, n_packets=3, n_bytes=230, priority=2,in_port="s1-eth1" actions=output:"s1-eth2",CONTROLLER:65535
 cookie=0x2b00000000000005, duration=22.688s, table=0, n_packets=5, n_bytes=390, priority=2,in_port="s1-eth2" actions=output:"s1-eth1",CONTROLLER:65535
 cookie=0x2b00000000000002, duration=24.691s, table=0, n_packets=6, n_bytes=552, priority=0 actions=drop
mininet>
```

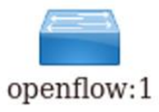
Can you explain what you see ?

LLDP Handling: A high-priority flow captures LLDP packets (EtherType 0x88cc) and sends them to the controller. No LLDP packets have been processed yet.

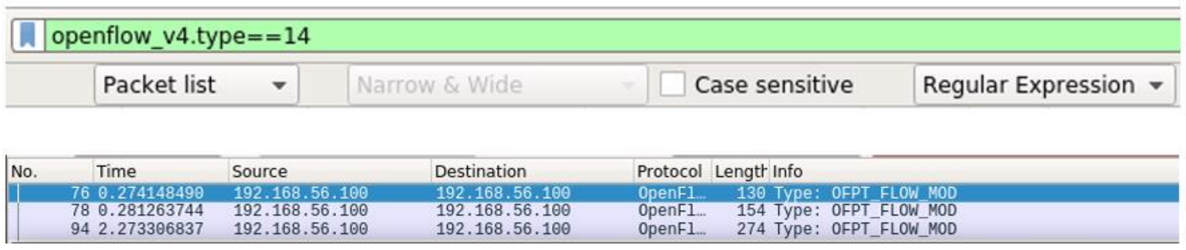
Traffic Forwarding**:** Two low-priority flows handle bidirectional traffic between s1-eth1 and s1-eth2. Packets are forwarded to the opposite port and a copy is sent to the controller. Both flows are active and processing traffic.

Default Drop Rule**:** A catch-all rule with the lowest priority (0) drops any packets that do not match higher-priority rules, ensuring that unintended traffic is not forwarded

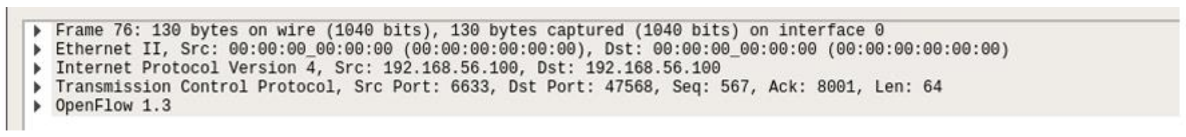
5.View the Topology



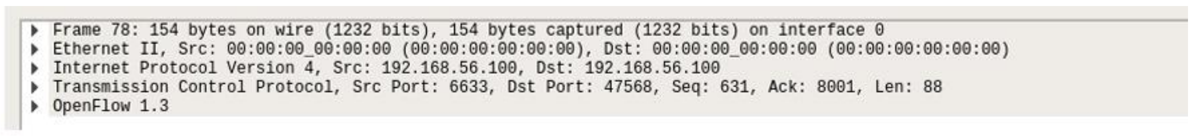
6.Analye OpenFlow messages with Wireshark



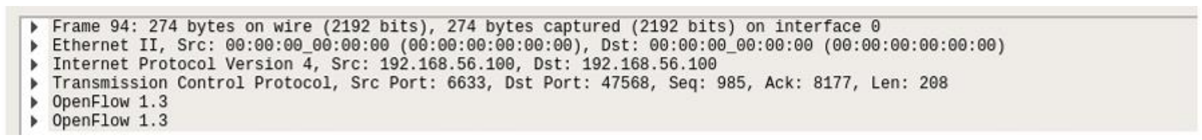
```
No.  Time         Source          Destination     Protocol Length Info
  76 0.274148490  192.168.56.100  192.168.56.100  OpenFl…  130 Type: OFPT_FLOW_MOD
  78 0.281263744  192.168.56.100  192.168.56.100  OpenFl…  154 Type: OFPT_FLOW_MOD
  94 2.273306837  192.168.56.100  192.168.56.100  OpenFl…  274 Type: OFPT_FLOW_MOD
```



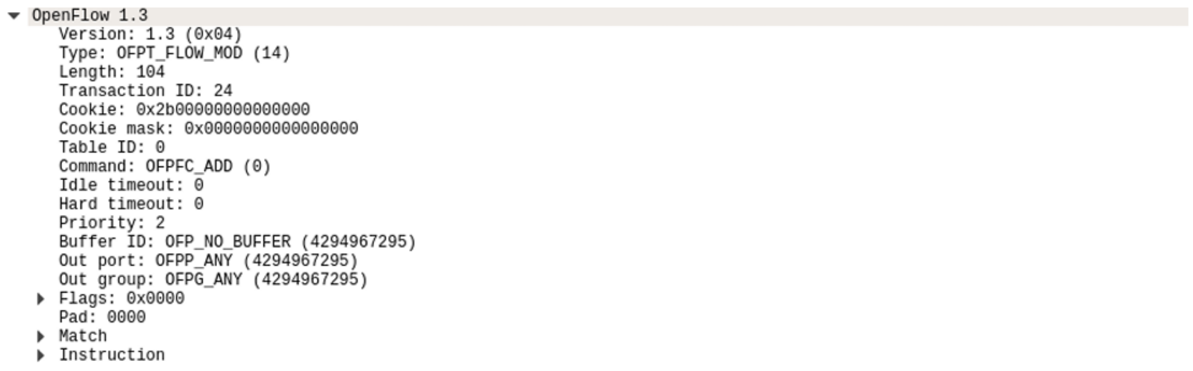
```
▸ Frame 76: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 47568, Seq: 567, Ack: 8001, Len: 64
▸ OpenFlow 1.3
```



```
▸ Frame 78: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 47568, Seq: 631, Ack: 8001, Len: 88
▸ OpenFlow 1.3
```



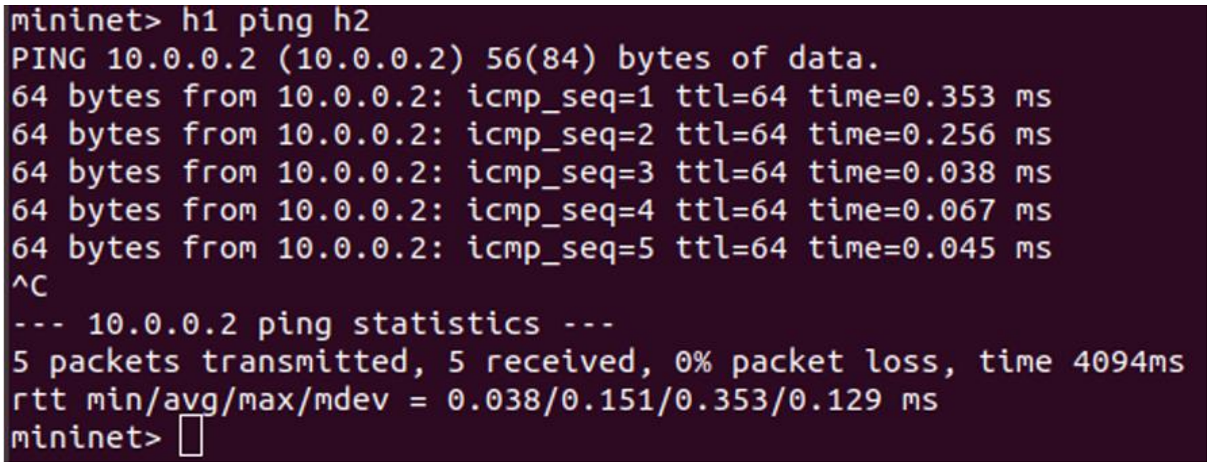
```
▸ Frame 94: 274 bytes on wire (2192 bits), 274 bytes captured (2192 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 47568, Seq: 985, Ack: 8177, Len: 208
▸ OpenFlow 1.3
▸ OpenFlow 1.3
```



```
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_FLOW_MOD (14)
    Length: 104
    Transaction ID: 24
    Cookie: 0x2b00000000000000
    Cookie mask: 0x0000000000000000
    Table ID: 0
    Command: OFPFC_ADD (0)
    Idle timeout: 0
    Hard timeout: 0
    Priority: 2
    Buffer ID: OFP_NO_BUFFER (4294967295)
    Out port: OFPP_ANY (4294967295)
    Out group: OFPG_ANY (4294967295)
  ▸ Flags: 0x0000
    Pad: 0000
  ▸ Match
  ▸ Instruction
```

7.Restart Wireshark

8. Pinging between hosts



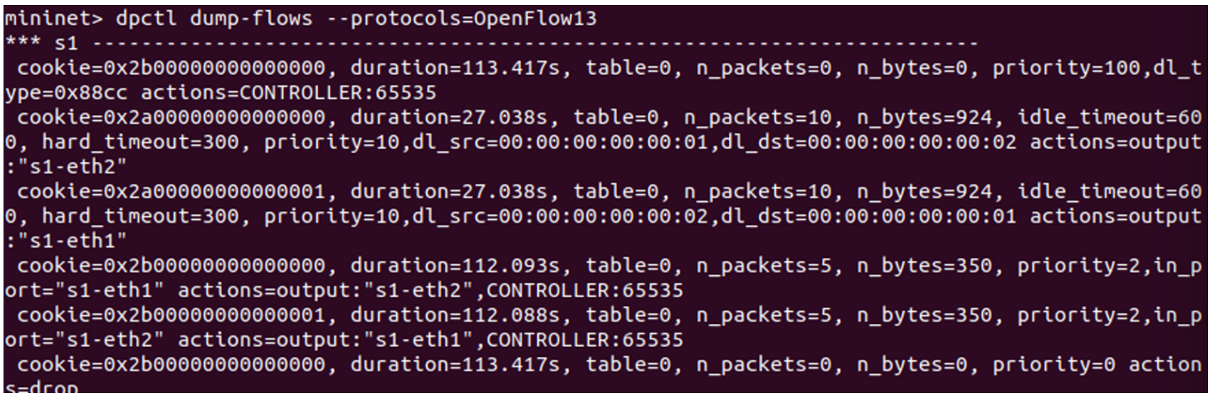
```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.353 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.256 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.045 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4094ms
rtt min/avg/max/mdev = 0.038/0.151/0.353/0.129 ms
mininet> 
```

Observations:

The successful ping between h1 and h2 confirms that both hosts are properly configured with IP addresses in the same subnet, and there is a reliable network connection within the Mininet environment. This setup allows them to communicate without any connectivity issues.

- **Network Setup**: The correct configuration of network interfaces and links in Mininet ensures smooth communication between the hosts.
- **Connection Details**: h1 and h2 are connected to the same switch, s1, with h1 on port eth0 and h2 on port eth1. This configuration allows direct Layer 2 communication between the two hosts.
- **OpenDaylight Controller**: The network is managed by an OpenDaylight controller (c0), which likely set up the switch (s1) to forward packets between h1 and h2. The necessary flow entries were installed in s1 to permit traffic between the hosts.
- **No Misconfigurations**: The accurate configuration of IP addresses, routes, and OpenFlow rules ensures that ICMP Echo Requests (pings) from h1 reach h2, and h2 can send ICMP Echo Replies back to h1, resulting in 0% packet loss.

9. Flow table After pinging



```
mininet> dpctl dump-flows --protocols=OpenFlow13
*** s1 ------------------------------------------------------------------------
 cookie=0x2b00000000000000, duration=113.417s, table=0, n_packets=0, n_bytes=0, priority=100,dl_t
ype=0x88cc actions=CONTROLLER:65535
 cookie=0x2a00000000000000, duration=27.038s, table=0, n_packets=10, n_bytes=924, idle_timeout=60
0, hard_timeout=300, priority=10,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output
:"s1-eth2"
 cookie=0x2a00000000000001, duration=27.038s, table=0, n_packets=10, n_bytes=924, idle_timeout=60
0, hard_timeout=300, priority=10,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output
:"s1-eth1"
 cookie=0x2b00000000000000, duration=112.093s, table=0, n_packets=5, n_bytes=350, priority=2,in_p
ort="s1-eth1" actions=output:"s1-eth2",CONTROLLER:65535
 cookie=0x2b00000000000001, duration=112.088s, table=0, n_packets=5, n_bytes=350, priority=2,in_p
ort="s1-eth2" actions=output:"s1-eth1",CONTROLLER:65535
 cookie=0x2b00000000000000, duration=113.417s, table=0, n_packets=0, n_bytes=0, priority=0 action
s=drop
```

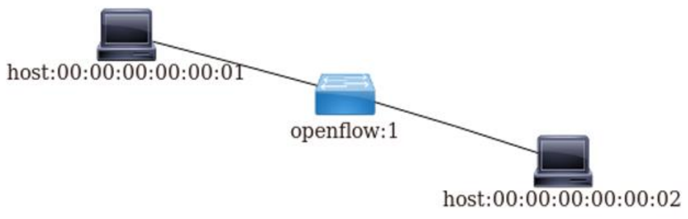
- **LLDP Packet Handling:** The first flow entry is designed to manage LLDP (Link Layer Discovery Protocol) packets, which are used to discover network topology. This flow has a high priority of 100 and matches packets with a `dl_type` of `0x88cc`, indicating LLDP traffic. These packets are forwarded to the controller (`CONTROLLER:65535`). Currently, no packets have matched this rule (`n_packets=0`), suggesting that there hasn't been any recent LLDP activity.

- **MAC Address-Specific Flows:** The second and third flow entries manage traffic between two specific MAC addresses: `00:00:00:00:00:01` and `00:00:00:00:00:02`. The first rule forwards traffic from `00:00:00:00:00:01` to `00:00:00:00:00:02` via `s1-eth2`, while the second handles traffic in the opposite direction, sending it out through `s1-eth1`. Both rules have processed 10 packets and 924 bytes so far. They also have idle and hard timeouts set at 600 and 300 seconds, respectively, meaning they will be removed if not used within those timeframes.

- **General Traffic Forwarding and Controller Interaction:** The fourth and fifth flow entries have lower priorities (priority 2) compared to the MAC-specific flows. These rules match packets based on their input ports (`s1-eth1` and `s1-eth2`) and forward them to the opposite port (`s1-eth2` and `s1-eth1`). Additionally, they send copies of these packets to the controller, which suggests these flows might be used for traffic monitoring or learning.

- **Default Drop Rule:** The sixth flow entry is the default rule with the lowest priority (priority 0), which drops any packets that do not match the higher-priority flows. So far, no packets have been dropped (`n_packets=0`), indicating that all traffic is being successfully matched and handled by the previous rules.

10. View the Topology

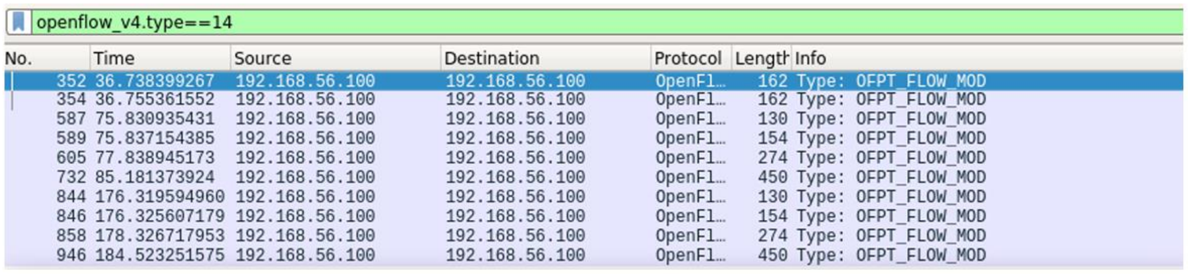


Observations:

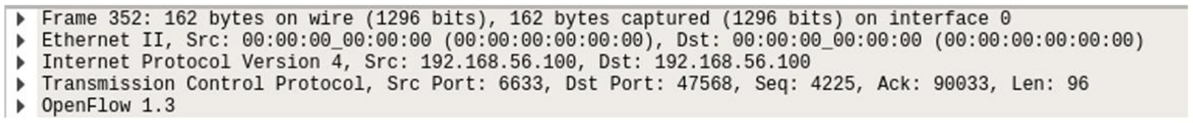
Has anything changed ?

Yes. Previously only the controller was shown. Now the controller and the two hosts are also shown.

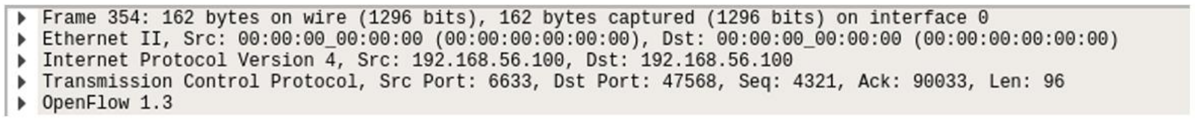
11. Analyze OpenFlow messages with Wireshark



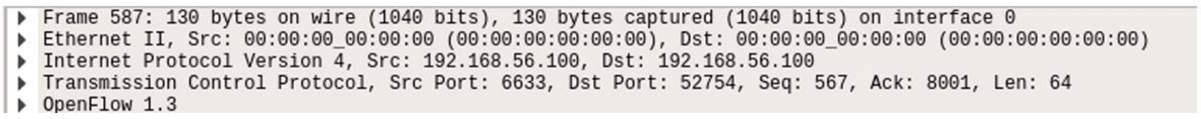
```
openflow_v4.type==14

No.   Time           Source          Destination     Protocol Lengtł Info
  352 36.738399267   192.168.56.100  192.168.56.100  OpenFl…  162 Type: OFPT_FLOW_MOD
  354 36.755361552   192.168.56.100  192.168.56.100  OpenFl…  162 Type: OFPT_FLOW_MOD
  587 75.830935431   192.168.56.100  192.168.56.100  OpenFl…  130 Type: OFPT_FLOW_MOD
  589 75.837154385   192.168.56.100  192.168.56.100  OpenFl…  154 Type: OFPT_FLOW_MOD
  605 77.838945173   192.168.56.100  192.168.56.100  OpenFl…  274 Type: OFPT_FLOW_MOD
  732 85.181373924   192.168.56.100  192.168.56.100  OpenFl…  450 Type: OFPT_FLOW_MOD
  844 176.319594960  192.168.56.100  192.168.56.100  OpenFl…  130 Type: OFPT_FLOW_MOD
  846 176.325607179  192.168.56.100  192.168.56.100  OpenFl…  154 Type: OFPT_FLOW_MOD
  858 178.326717953  192.168.56.100  192.168.56.100  OpenFl…  274 Type: OFPT_FLOW_MOD
  946 184.523251575  192.168.56.100  192.168.56.100  OpenFl…  450 Type: OFPT_FLOW_MOD
```



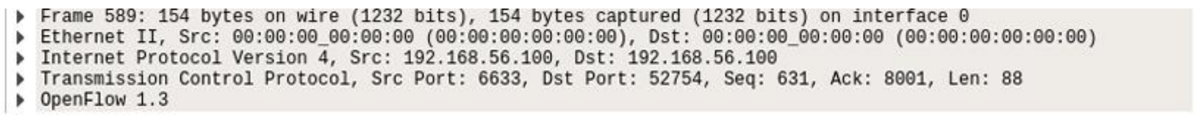
```
▶ Frame 352: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 47568, Seq: 4225, Ack: 90033, Len: 96
▶ OpenFlow 1.3
```



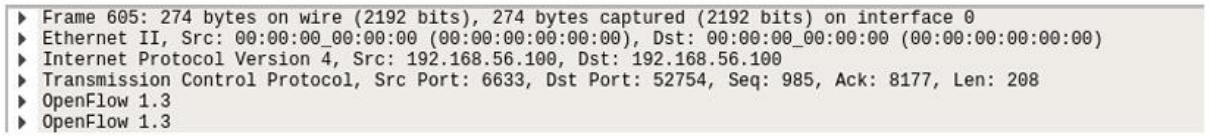
```
▶ Frame 354: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 47568, Seq: 4321, Ack: 90033, Len: 96
▶ OpenFlow 1.3
```



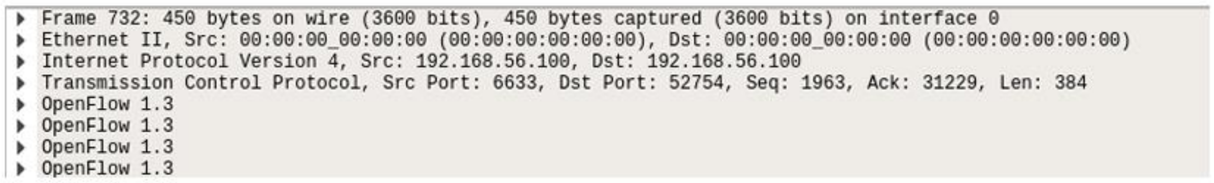
```
▶ Frame 587: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 52754, Seq: 567, Ack: 8001, Len: 64
▶ OpenFlow 1.3
```



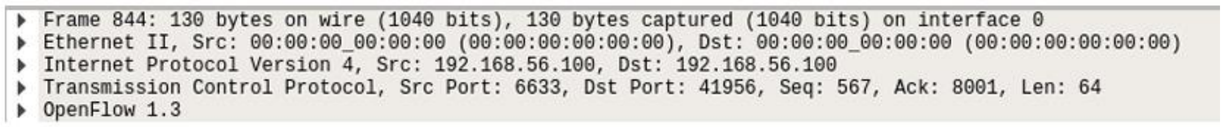
```
▶ Frame 589: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 52754, Seq: 631, Ack: 8001, Len: 88
▶ OpenFlow 1.3
```



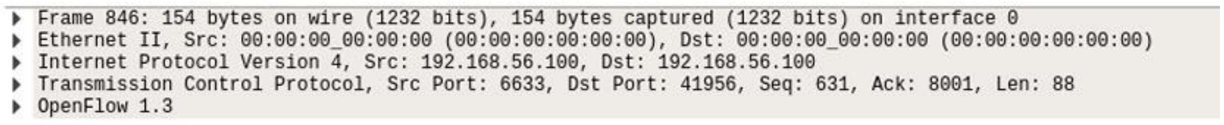
```
▶ Frame 605: 274 bytes on wire (2192 bits), 274 bytes captured (2192 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 52754, Seq: 985, Ack: 8177, Len: 208
▶ OpenFlow 1.3
▶ OpenFlow 1.3
```



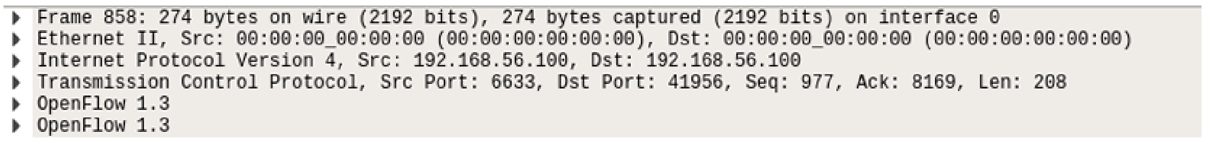
```
▶ Frame 732: 450 bytes on wire (3600 bits), 450 bytes captured (3600 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 52754, Seq: 1963, Ack: 31229, Len: 384
▶ OpenFlow 1.3
▶ OpenFlow 1.3
▶ OpenFlow 1.3
▶ OpenFlow 1.3
```



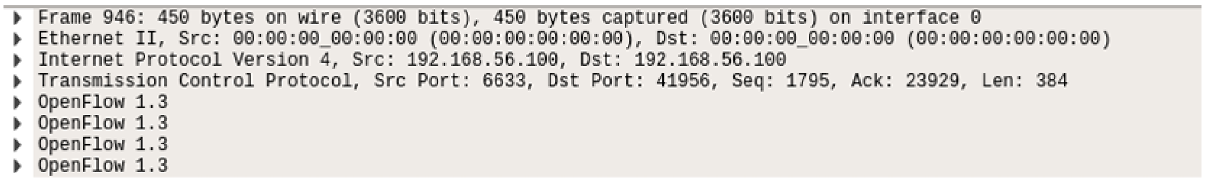
```
▶ Frame 844: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 41956, Seq: 567, Ack: 8001, Len: 64
▶ OpenFlow 1.3
```



```
▶ Frame 846: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 41956, Seq: 631, Ack: 8001, Len: 88
▶ OpenFlow 1.3
```

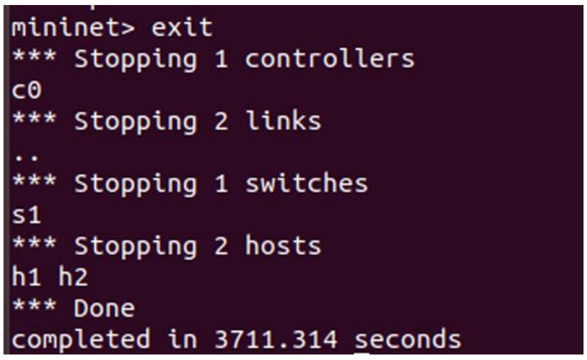


```
▶ Frame 858: 274 bytes on wire (2192 bits), 274 bytes captured (2192 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 41956, Seq: 977, Ack: 8169, Len: 208
▶ OpenFlow 1.3
▶ OpenFlow 1.3
```



```
▶ Frame 946: 450 bytes on wire (3600 bits), 450 bytes captured (3600 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 192.168.56.100, Dst: 192.168.56.100
▶ Transmission Control Protocol, Src Port: 6633, Dst Port: 41956, Seq: 1795, Ack: 23929, Len: 384
▶ OpenFlow 1.3
▶ OpenFlow 1.3
▶ OpenFlow 1.3
▶ OpenFlow 1.3
```

12.Exit from mininet

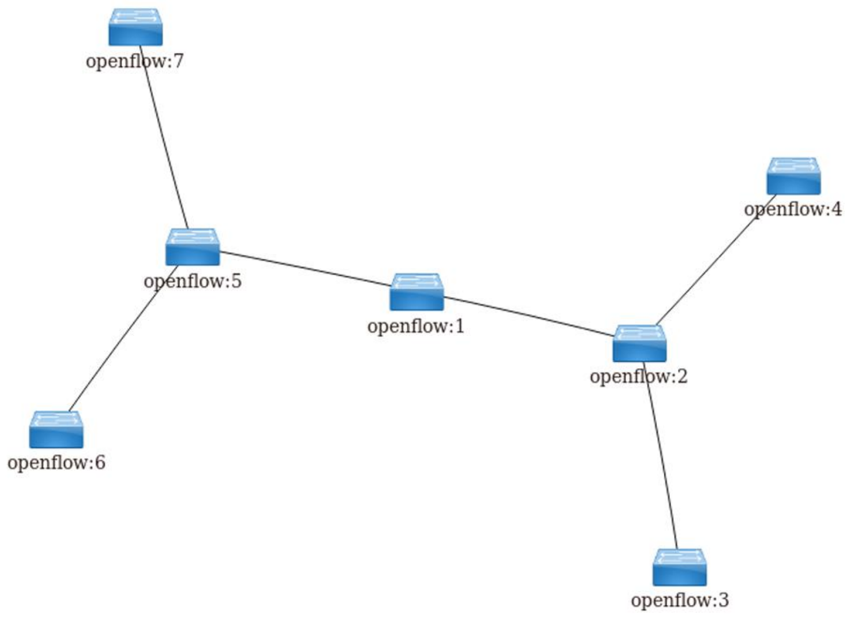


```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 3711.314 seconds
```

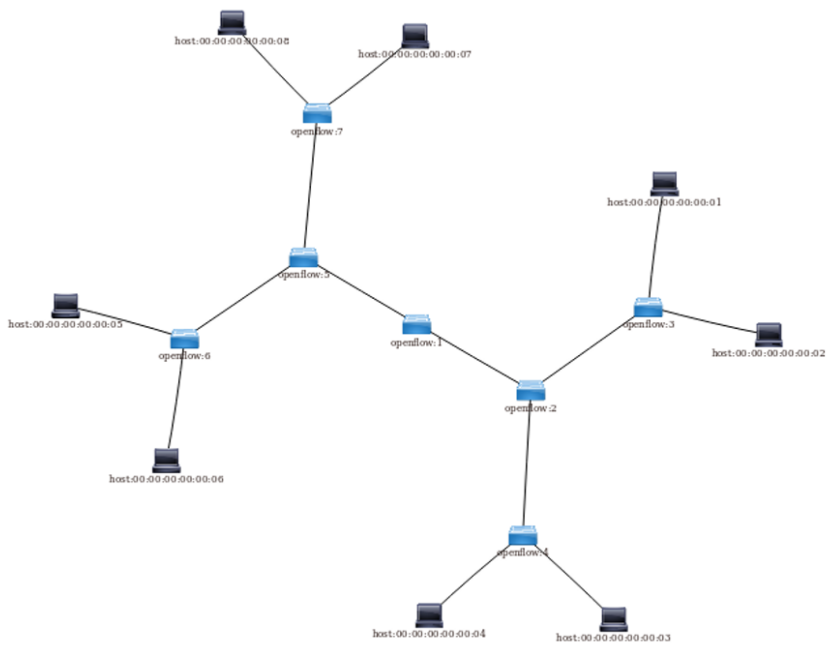
13. Analyzing Other topologies

13.1 Tree Topology

Before Pinging

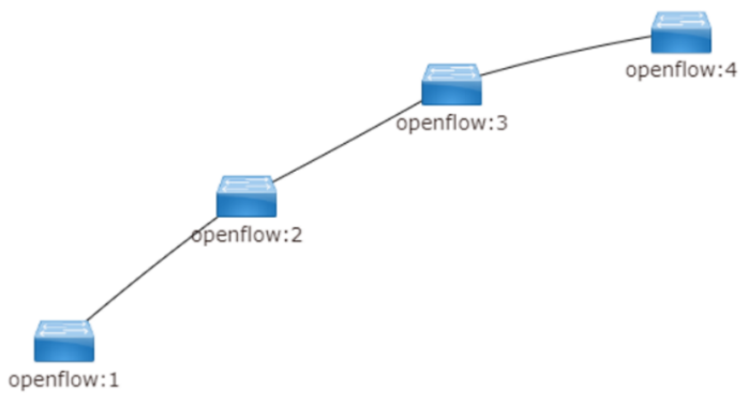


After Pinging

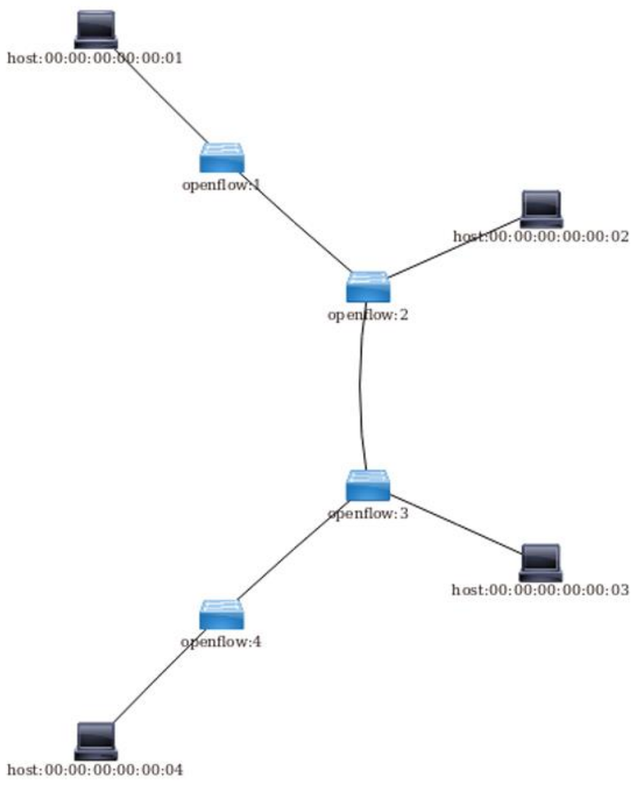


13.2 Linear Topology

Before Pinging



After Pinging


host:00:00:00:00:00:01
openflow:1
host:00:00:00:00:00:02
openflow:2
openflow:3
host:00:00:00:00:00:03
openflow:4
host:00:00:00:00:00:04