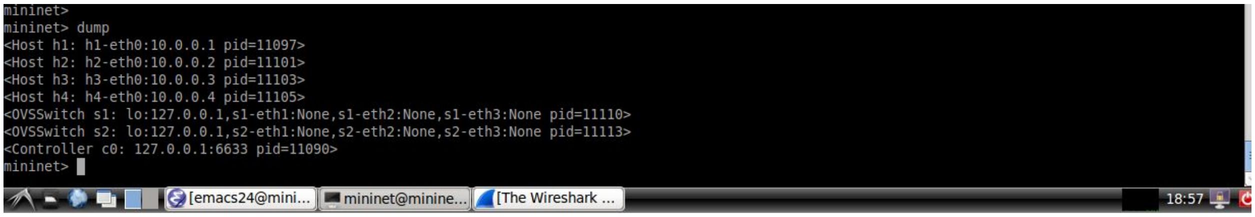
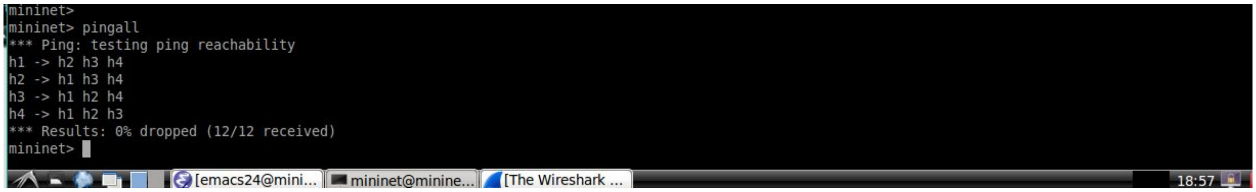
2. Dump command displays the IP addresses and the process IDs of the nodes.
In the screenshot below, after the command dump, it shows all 4 IP addresses of the 4 hosts, switches, and controller, as well as the PID of each.



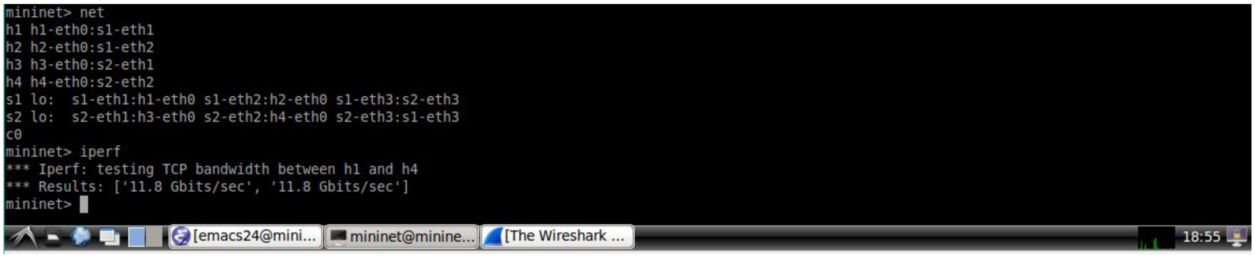
```
mininet>
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=11097>
<Host h2: h2-eth0:10.0.0.2 pid=11101>
<Host h3: h3-eth0:10.0.0.3 pid=11103>
<Host h4: h4-eth0:10.0.0.4 pid=11105>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=11110>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=11113>
<Controller c0: 127.0.0.1:6633 pid=11090>
mininet>
```

Using the command pingall tests the connection by making each host in the network ping every other host in the network. We see that h1 pings h2 and h2 pings h1. We also see h3 pings h4 and vice versa. Also none of the connections dropped.



```
mininet>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

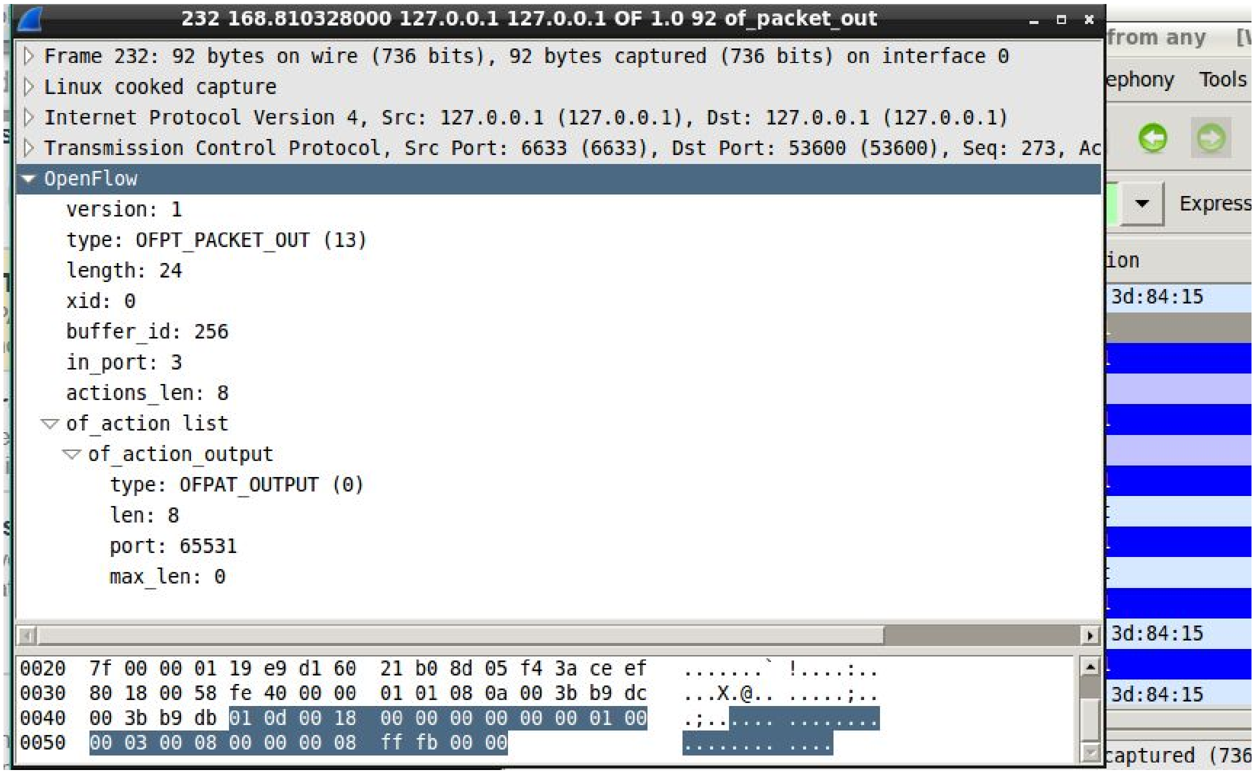
3.



```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s2-eth1
h4 h4-eth0:s2-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:s2-eth3
s2 lo:  s2-eth1:h3-eth0 s2-eth2:h4-eth0 s2-eth3:s1-eth3
c0
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['11.8 Gbits/sec', '11.8 Gbits/sec']
mininet>
```

The output speed is 11.8 Gbits/sec

4a.

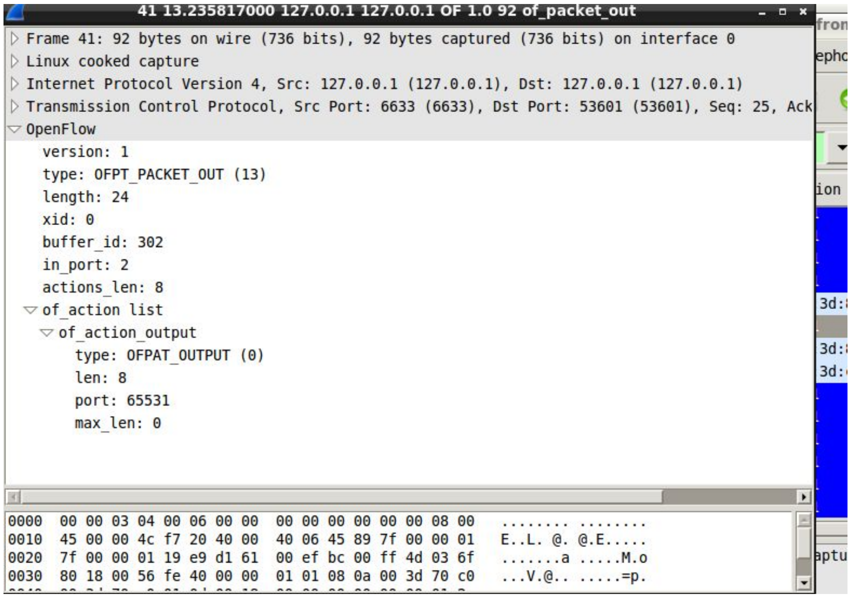


```
232 168.810328000 127.0.0.1 127.0.0.1 OF 1.0 92 of_packet_out
▷ Frame 232: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 53600 (53600), Seq: 273, Ac
▼ OpenFlow
    version: 1
    type: OFPT_PACKET_OUT (13)
    length: 24
    xid: 0
    buffer_id: 256
    in_port: 3
    actions_len: 8
  ▽ of_action list
    ▽ of_action_output
        type: OFPAT_OUTPUT (0)
        len: 8
        port: 65531
        max_len: 0

0020  7f 00 00 01 19 e9 d1 60  21 b0 8d 05 f4 3a ce ef   .......` !....:..
0030  80 18 00 58 fe 40 00 00  01 01 08 0a 00 3b b9 dc   ...X.@.. .....;..
0040  00 3b b9 db 01 0d 00 18  00 00 00 00 00 00 01 00   .;...... ........
0050  00 03 00 08 00 00 00 08  ff fb 00 00               ........ ....
```

After running a ping from one host to another, I can see on wireshark that there are 13 OFPT packet out messages.

4b.

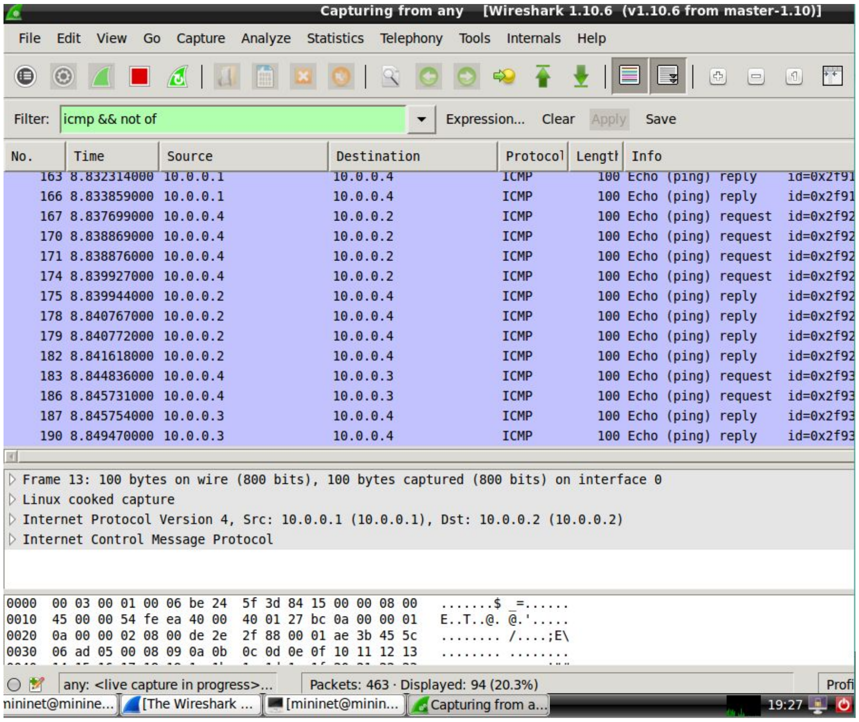


```
41 13.235817000 127.0.0.1 127.0.0.1 OF 1.0 92 of_packet_out
▷ Frame 41: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 53601 (53601), Seq: 25, Ack
▽ OpenFlow
    version: 1
    type: OFPT_PACKET_OUT (13)
    length: 24
    xid: 0
    buffer_id: 302
    in_port: 2
    actions_len: 8
  ▽ of_action list
    ▽ of_action_output
        type: OFPAT_OUTPUT (0)
        len: 8
        port: 65531
        max_len: 0

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ........ ........
0010  45 00 00 4c f7 20 40 00  40 06 45 89 7f 00 00 01   E..L. @. @.E.....
0020  7f 00 00 01 19 e9 d1 61  00 ef bc 00 ff 4d 03 6f   .......a .....M.o
0030  80 18 00 56 fe 40 00 00  01 01 08 0a 00 3d 70 c0   ...V.@.. .....=p.
```

The source IP address is: 127.0.0.1
The Destination IP is: 127.0.0.1

4c.


Capturing from any   [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: icmp && not of

```
No.    Time         Source       Destination   Protocol Length Info
   163 8.832314000 10.0.0.1     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f91
   166 8.833859000 10.0.0.1     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f91
   167 8.837699000 10.0.0.4     10.0.0.2      ICMP     100 Echo (ping) request  id=0x2f92
   170 8.838869000 10.0.0.4     10.0.0.2      ICMP     100 Echo (ping) request  id=0x2f92
   171 8.838876000 10.0.0.4     10.0.0.2      ICMP     100 Echo (ping) request  id=0x2f92
   174 8.839927000 10.0.0.4     10.0.0.2      ICMP     100 Echo (ping) request  id=0x2f92
   175 8.839944000 10.0.0.2     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f92
   178 8.840767000 10.0.0.2     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f92
   179 8.840772000 10.0.0.2     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f92
   182 8.841618000 10.0.0.2     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f92
   183 8.844836000 10.0.0.4     10.0.0.3      ICMP     100 Echo (ping) request  id=0x2f93
   186 8.845731000 10.0.0.4     10.0.0.3      ICMP     100 Echo (ping) request  id=0x2f93
   187 8.845754000 10.0.0.3     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f93
   190 8.849470000 10.0.0.3     10.0.0.4      ICMP     100 Echo (ping) reply    id=0x2f93
```

```
▷ Frame 13: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 10.0.0.2 (10.0.0.2)
▷ Internet Control Message Protocol
```

```
0000  00 03 00 01 00 06 be 24  5f 3d 84 15 00 00 08 00   .......$ _=......
0010  45 00 00 54 fe ea 40 00  40 01 27 bc 0a 00 00 01   E..T..@. @.'.....
0020  0a 00 00 02 08 00 de 2e  2f 88 00 01 ae 3b 45 5c   ........ /....;E\
0030  06 ad 05 00 08 09 0a 0b  0c 0d 0e 0f 10 11 12 13   ........ ........
```

any: <live capture in progress>...   Packets: 463 · Displayed: 94 (20.3%)   Profi

94 entries showed up.These are ping request and replies.