

2. [30 pts] Save a screenshot of *dump* and *pingall* output. Explain what is being shown in the screenshot.

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1813>
<Host h2: h2-eth0:10.0.0.2 pid=1817>
<Host h3: h3-eth0:10.0.0.3 pid=1819>
<Host h4: h4-eth0:10.0.0.4 pid=1821>
<Host h5: h5-eth0:10.0.0.5 pid=1823>
<Host h6: h6-eth0:10.0.0.6 pid=1825>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=1830>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=1833>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None,s3-eth4:None
pid=1836>
<Controller c0: 127.0.0.1:6633 pid=1806>
```

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
```

“dump” command displays information about all devices including host h1 - h6, switch s1 - s3, and controller c0. It lists their interfaces, the IP addresses, and the pids.

“pingall” command does all-pairs “ping” to test connectivity between hosts. For example, host h1 can connect to host h6 through a bunch of hosts h2, h3, h4, h5.

3. [10 pts] Run the *iperf* command as well, and screenshot the output, how fast is the connect?

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h6
*** Results: ['43.8 Gbits/sec', '43.9 Gbits/sec']
```

“iperf” command runs on host h1 and client h6 to analyze the bandwidth between them. The connection between them is around 43 Gbits/sec.

4.a) [20 pts] Run ping from a host to any other host using *hX ping -c 5 hY*. How many *of_packet_in* messages show up? Take a screenshot of your results.

```
mininet> h1 ping -c 5 h6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=1 ttl=64 time=1.78 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=64 time=1.77 ms
64 bytes from 10.0.0.6: icmp_seq=3 ttl=64 time=0.363 ms
64 bytes from 10.0.0.6: icmp_seq=4 ttl=64 time=0.081 ms
64 bytes from 10.0.0.6: icmp_seq=5 ttl=64 time=0.061 ms

--- 10.0.0.6 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.061/0.811/1.780/0.796 ms
```

The screenshot shows the Wireshark 1.10.6 interface. The packet capture filter is set to `of10.packet_in.type == 10`. The packet list shows 11 packets, all of type `of_packet_in`. The packet details pane shows the structure of an `OFPT_PACKET_IN` packet, including fields like `version`, `type`, `length`, `xid`, `buffer_id`, `total_len`, `in_port`, and `reason`. The packet bytes pane shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
50	27.40801400	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
57	27.40850300	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
66	27.40881500	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
67	27.40884700	10.0.0.6	10.0.0.1	OF 1.0	184	of_packet_in
76	27.40934300	10.0.0.6	10.0.0.1	OF 1.0	184	of_packet_in
82	28.41041000	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
87	28.41141200	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
123	32.42364100	86:f1:9a:d4:89:14	8a:4d:78:26:f0:8b	OF 1.0	128	of_packet_in
128	32.42410400	86:f1:9a:d4:89:14	8a:4d:78:26:f0:8b	OF 1.0	128	of_packet_in
133	32.42449100	8a:4d:78:26:f0:8b	86:f1:9a:d4:89:14	OF 1.0	128	of_packet_in
137	32.42477600	8a:4d:78:26:f0:8b	86:f1:9a:d4:89:14	OF 1.0	128	of_packet_in

There are 11 *of_packet_in* messages ping from host1 to host6.

b). [20 pts] What is the source and destination IP addresses for these entries? Find another packet that matches the “of” filter with the OpenFlow typefield set to *OFPT_PACKET_OUT*. What is the source and destination IP address for this entry? Take screenshots showing your results.

The source and destination IP addresses for these entries are displayed below.

No.	Time	Source	Destination	Protocol	Length	Info
69	34.52176700	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
76	34.52246000	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
85	34.52302700	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
86	34.52306300	10.0.0.6	10.0.0.1	OF 1.0	184	of_packet_in
95	34.52363300	10.0.0.6	10.0.0.1	OF 1.0	184	of_packet_in
101	35.52454000	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
106	35.52510400	10.0.0.1	10.0.0.6	OF 1.0	184	of_packet_in
142	39.53178900	52:77:3f:e9:c9:ac	ca:7e:cd:68:00:8f	OF 1.0	128	of_packet_in
147	39.53251000	52:77:3f:e9:c9:ac	ca:7e:cd:68:00:8f	OF 1.0	128	of_packet_in
152	39.53322500	ca:7e:cd:68:00:8f	52:77:3f:e9:c9:ac	OF 1.0	128	of_packet_in
156	39.53368600	ca:7e:cd:68:00:8f	52:77:3f:e9:c9:ac	OF 1.0	128	of_packet_in

Capturing from any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

Filter: `of10.packet_out.type == 13` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
51	27.40835700	127.0.0.1	127.0.0.1	OF 1.0	92	of_packet_out
58	27.40872200	127.0.0.1	127.0.0.1	OF 1.0	92	of_packet_out
68	27.40902000	127.0.0.1	127.0.0.1	OF 1.0	92	of_packet_out

Frame 68: 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: 37825 (37825), Seq: 49, Ack: 165, Len: 24

OpenFlow

version: 1

type: OFPT_PACKET_OUT (13)

length: 24

xid: 0

buffer_id: 334

in_port: 3

actions len: 8

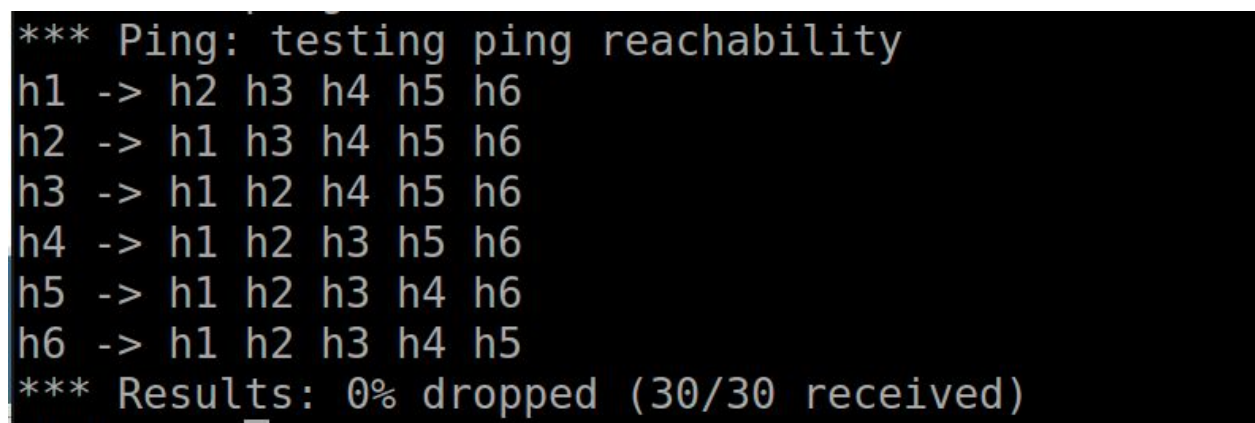
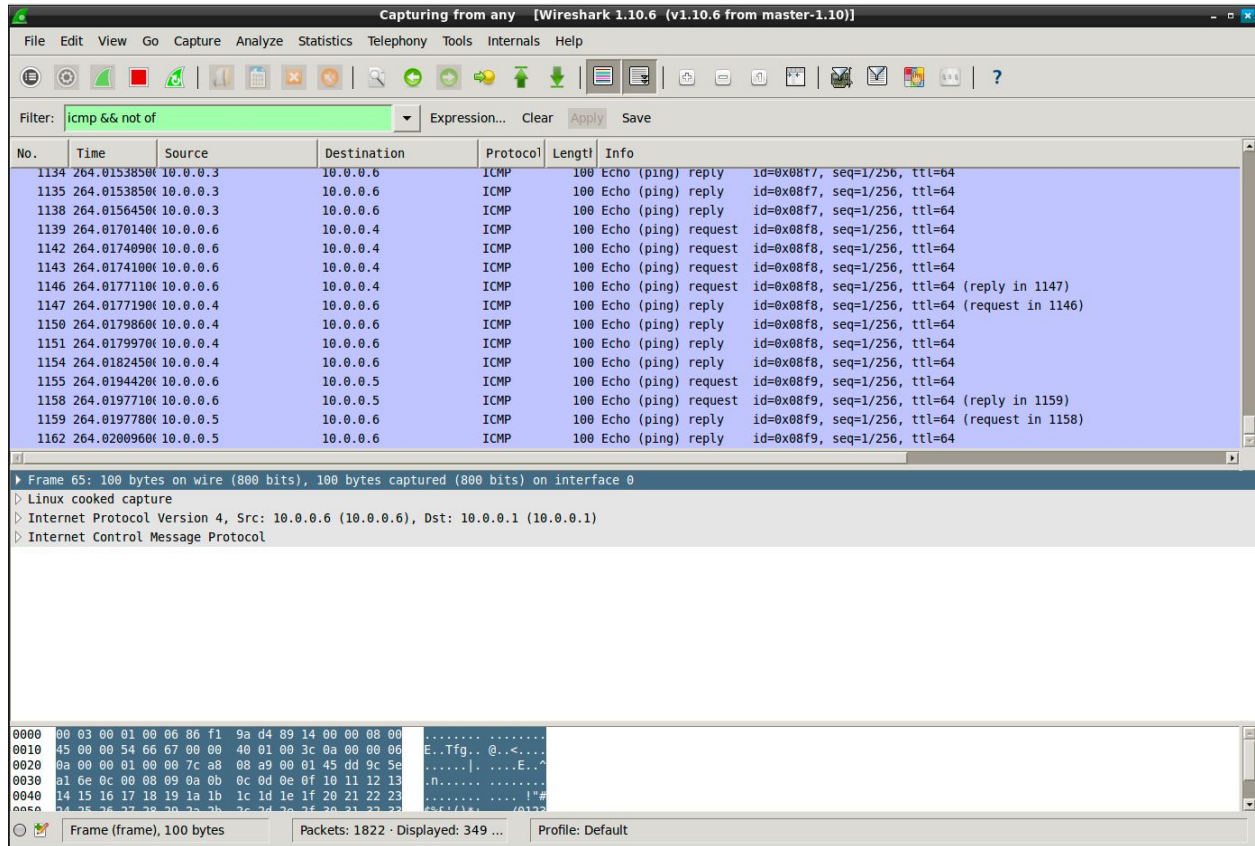
of_action list

0000 00 00 03 04 00 06 00 00 00 00 00 00 08 00
 0010 45 00 00 4c 8b f1 40 00 40 06 b8 b8 7f 00 00 01 E..L..@. @.....
 0020 7f 00 00 01 19 e9 93 c1 18 fd 53 0b 37 52 f0 a7S.7R..
 0030 80 18 00 58 fe 40 00 00 01 01 00 0a 00 00 62 0e ...X.@.b..
 0040 00 00 62 0e 01 0d 00 18 00 00 00 00 00 01 4e ..b.....N
 0050 00 03 00 00 00 00 00 00 ff ff 00 00

Packets: 426 · Displayed: 3 (0.7...)

There are 3 entries with the source ip address 127.0.0.1 to the destination ip address 127.0.0.1

c). [20 pts] Replace the display filter for “of” to “icmp && not of”. Run *pingall* again, how many entries are generated in wireshark? What types of icmp entries show up? Take a screenshot of your results.



There are 349 entries generated when running *pingall*. ICMP provides two types of query messages, Echo (ping) request and Echo (ping) reply.

