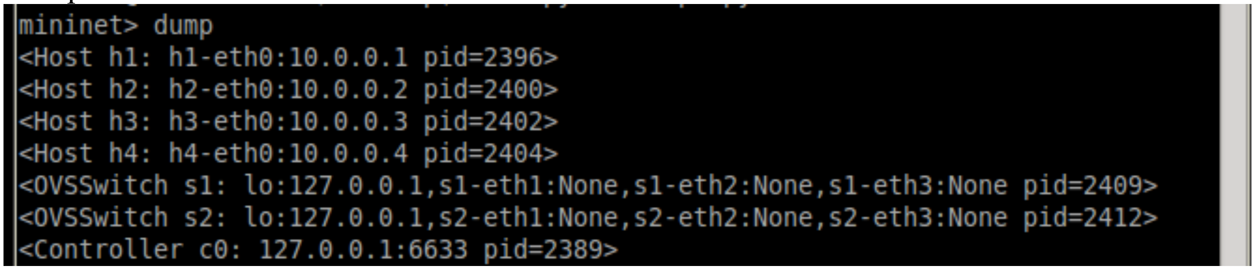
Shawn Chumbar (schumbar)
schumbar@ucsc.edu
Lab 1 Questions

Lab 1

2. Save a screenshot of dump and pingall output. Explain what is being shown in the screenshot.

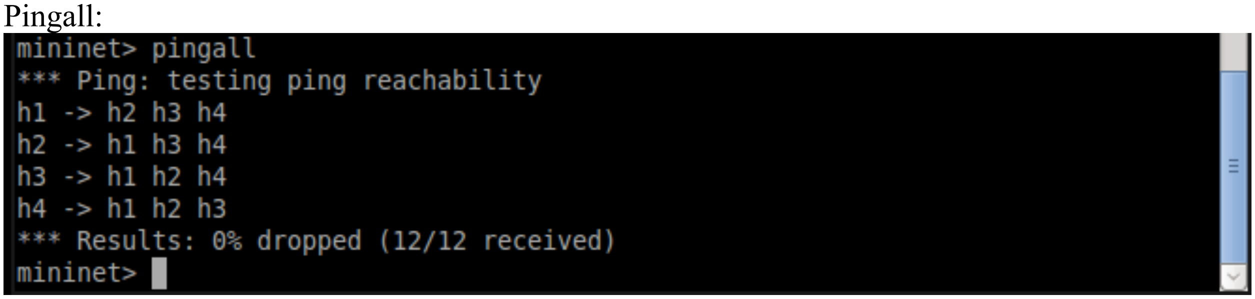
Dump:



```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2396>
<Host h2: h2-eth0:10.0.0.2 pid=2400>
<Host h3: h3-eth0:10.0.0.3 pid=2402>
<Host h4: h4-eth0:10.0.0.4 pid=2404>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=2409>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=2412>
<Controller c0: 127.0.0.1:6633 pid=2389>
```

The *dump* command shows what IP addresses have been assigned to the hosts. We are also given the pids of each process.

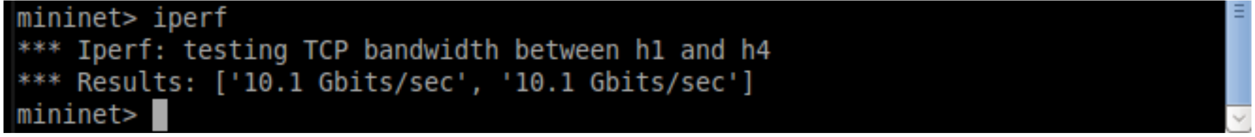
Pingall:



```
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> 
```

The *pingall* command shows us each host trying to connect to all of the other hosts. In this specific case, we can see that there are no packets dropped and that each host is connected to each other.

3. Run the iperf command as well, and screenshot the output, how fast is the connect?

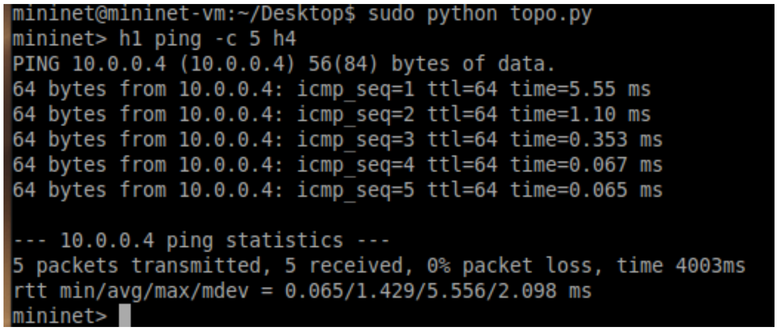


```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['10.1 Gbits/sec', '10.1 Gbits/sec']
mininet> 
```

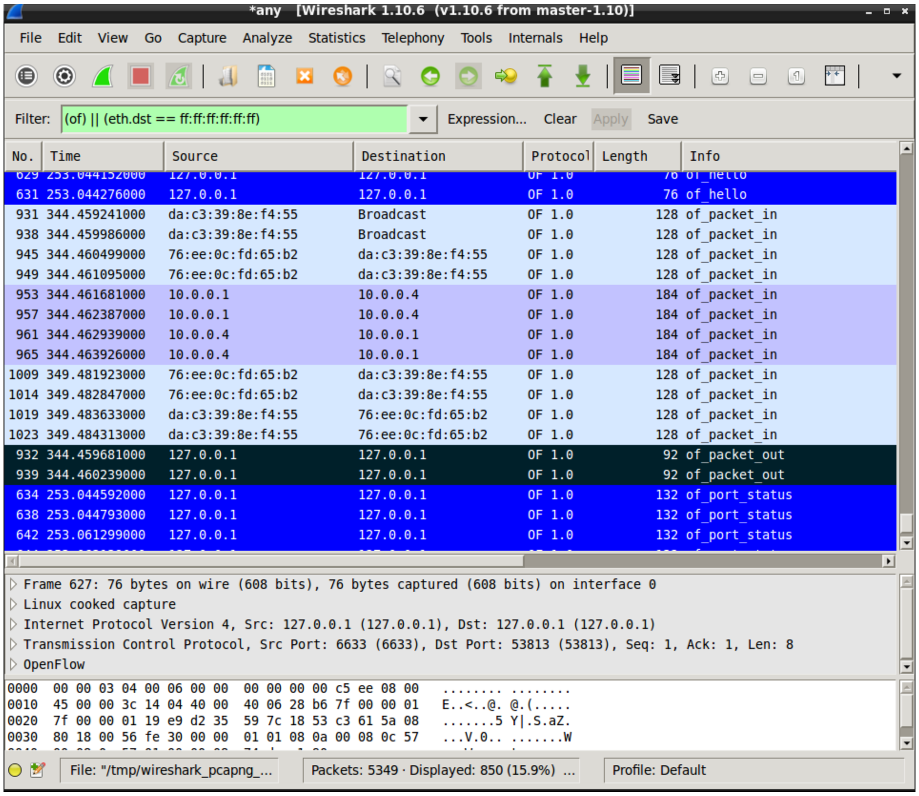
According to the output of the *iperf* command, the bandwidth between h1 and h4 has about 10.1 Gbits/sec as the throughput. This means that the connection is 10.1 Gbits/sec fast.

4. Run wireshark, and using the display filter, filter for “of”. Note: When you run wireshark you should do so as “sudo wireshark”. When you choose an interface to capture on, you should select “any”.

a. 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@mininet-vm:~/Desktop$ sudo python topo.py
mininet> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=5.55 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=1.10 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.353 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.065 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 0.065/1.429/5.556/2.098 ms
mininet>
```



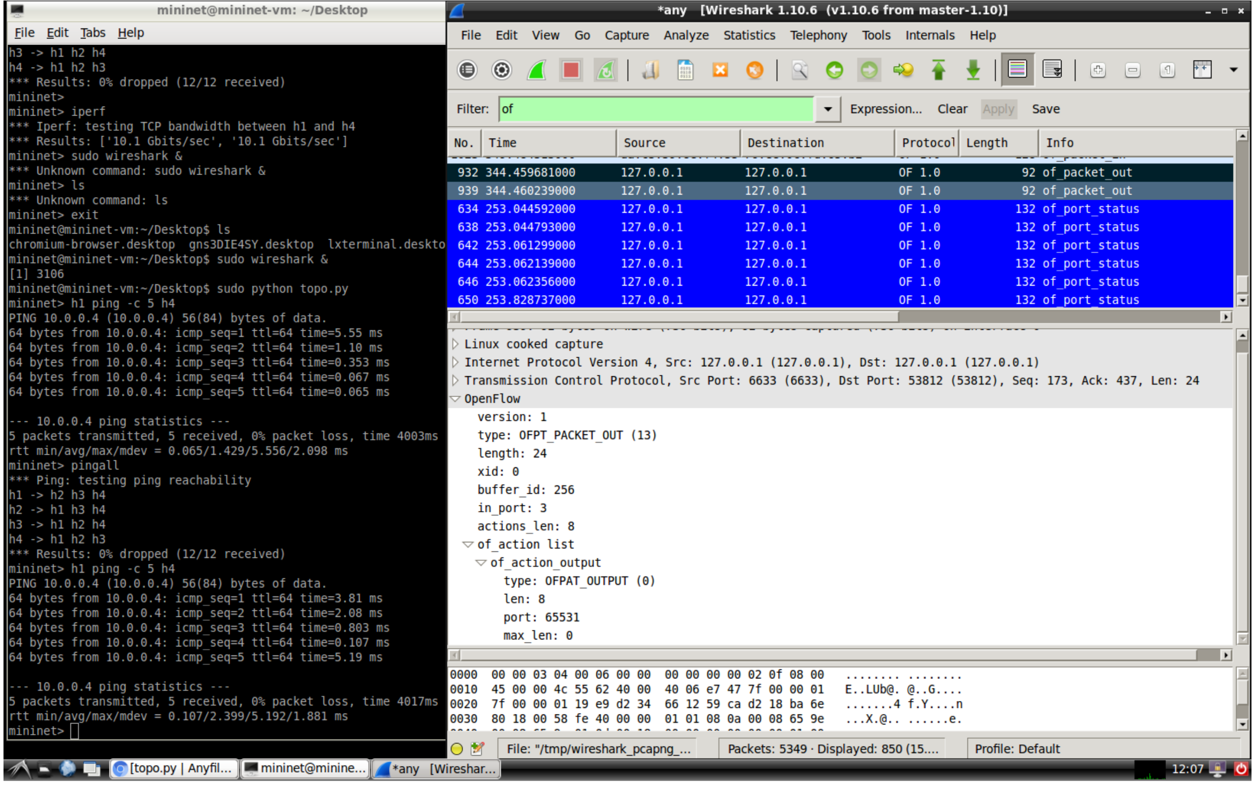
*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: (of) || (eth.dst == ff:ff:ff:ff:ff:ff)

```
No.  Time          Source             Destination        Protocol Length Info
 631 253.044276000 127.0.0.1          127.0.0.1          OF 1.0      76 of_hello
 931 344.459241000 da:c3:39:8e:f4:55  Broadcast          OF 1.0     128 of_packet_in
 938 344.459986000 da:c3:39:8e:f4:55  Broadcast          OF 1.0     128 of_packet_in
 945 344.460499000 76:ee:0c:fd:65:b2  da:c3:39:8e:f4:55  OF 1.0     128 of_packet_in
 949 344.461095000 76:ee:0c:fd:65:b2  da:c3:39:8e:f4:55  OF 1.0     128 of_packet_in
 953 344.461681000 10.0.0.1           10.0.0.4           OF 1.0     184 of_packet_in
 957 344.462387000 10.0.0.1           10.0.0.4           OF 1.0     184 of_packet_in
 961 344.462939000 10.0.0.4           10.0.0.1           OF 1.0     184 of_packet_in
 965 344.463926000 10.0.0.4           10.0.0.1           OF 1.0     184 of_packet_in
1009 349.481923000 76:ee:0c:fd:65:b2  da:c3:39:8e:f4:55  OF 1.0     128 of_packet_in
1014 349.482847000 76:ee:0c:fd:65:b2  da:c3:39:8e:f4:55  OF 1.0     128 of_packet_in
1019 349.483633000 da:c3:39:8e:f4:55  76:ee:0c:fd:65:b2  OF 1.0     128 of_packet_in
1023 349.484313000 da:c3:39:8e:f4:55  76:ee:0c:fd:65:b2  OF 1.0     128 of_packet_in
 932 344.459681000 127.0.0.1          127.0.0.1          OF 1.0      92 of_packet_out
 939 344.460239000 127.0.0.1          127.0.0.1          OF 1.0      92 of_packet_out
 634 253.044592000 127.0.0.1          127.0.0.1          OF 1.0     132 of_port_status
 638 253.044793000 127.0.0.1          127.0.0.1          OF 1.0     132 of_port_status
 642 253.061299000 127.0.0.1          127.0.0.1          OF 1.0     132 of_port_status

▷ Frame 627: 76 bytes on wire (608 bits), 76 bytes captured (608 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: 53813 (53813), Seq: 1, Ack: 1, Len: 8
▷ OpenFlow

0000  00 00 03 04 00 06 00 00  00 00 00 00 c5 ee 08 00   ........ ........
0010  45 00 00 3c 14 04 40 00  40 06 28 b6 7f 00 00 01   E..<..@. @.(.....
0020  7f 00 00 01 19 e9 d2 35  59 7c 18 53 c3 61 5a 08   .......5 Y|.S.aZ.
0030  80 18 00 56 fe 30 00 00  01 01 08 0a 00 08 0c 57   ...V.0.. .......W

File: "/tmp/wireshark_pcapng_...   Packets: 5349 · Displayed: 850 (15.9%) ...   Profile: Default
```

When I ran <**h1 ping -c 5 h4**>, I found that about 12 messages of type *of_packet_in* showed up. The source and destination IP addresses for the messages of type *of_packet_in* were host1 and host4. For some of the messages, the source was host1, while for others the source was host4. However, the destination IP was the opposite host of the source IP (i.e. if source IP was host1, destination IP would be host4 and vice versa).

b. 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.


mininet@mininet-vm: ~/Desktop
File Edit Tabs Help
*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
File: "/tmp/wireshark_pcapng_...
Packets: 5349 · Displayed: 850 (15....
Profile: Default
[topo.py | Anyfil...
mininet@minine...
*any [Wireshar...
12:07

```
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['10.1 Gbits/sec', '10.1 Gbits/sec']
mininet> sudo wireshark &
*** Unknown command: sudo wireshark &
mininet> ls
*** Unknown command: ls
mininet> exit
mininet@mininet-vm:~/Desktop$ ls
chromium-browser.desktop  gns3DIE4SY.desktop  lxterminal.deskto
mininet@mininet-vm:~/Desktop$ sudo wireshark &
[1] 3106
mininet@mininet-vm:~/Desktop$ sudo python topo.py
mininet> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=5.55 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=1.10 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.353 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.065 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 0.065/1.429/5.556/2.098 ms
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> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=3.81 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=2.08 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.803 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.107 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=5.19 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4017ms
rtt min/avg/max/mdev = 0.107/2.399/5.192/1.881 ms
mininet> 
```

Filter: of

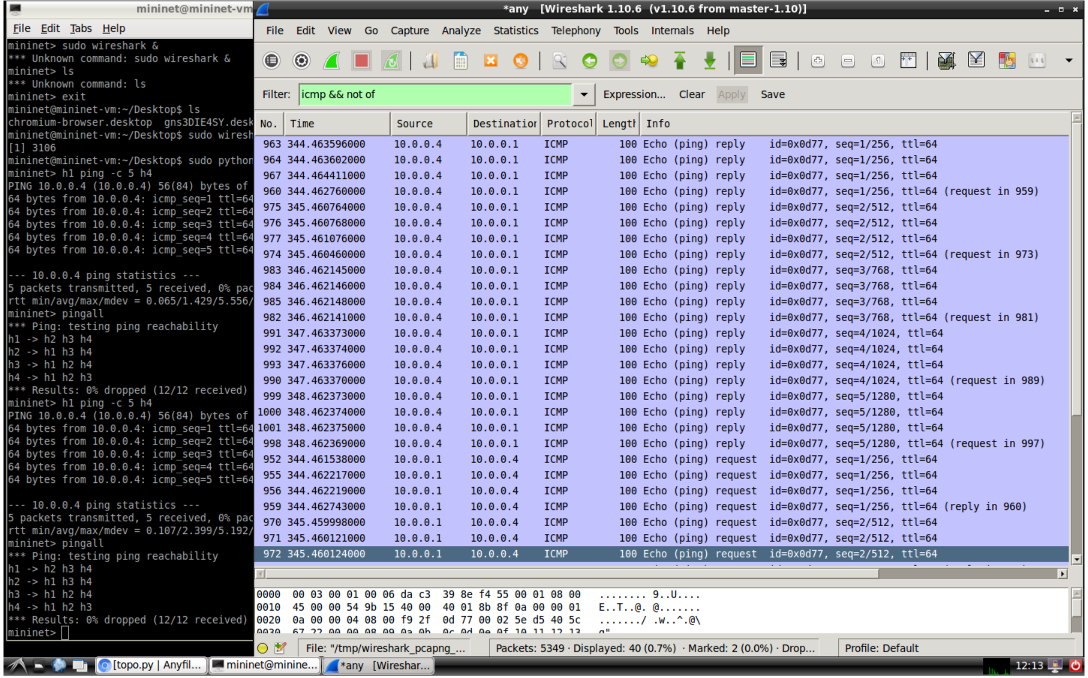
No.	Time	Source	Destination	Protocol	Length	Info
932	344.459681000	127.0.0.1	127.0.0.1	OF 1.0	92	of_packet_out
939	344.460239000	127.0.0.1	127.0.0.1	OF 1.0	92	of_packet_out
634	253.044592000	127.0.0.1	127.0.0.1	OF 1.0	132	of_port_status
638	253.044793000	127.0.0.1	127.0.0.1	OF 1.0	132	of_port_status
642	253.061299000	127.0.0.1	127.0.0.1	OF 1.0	132	of_port_status
644	253.062139000	127.0.0.1	127.0.0.1	OF 1.0	132	of_port_status
646	253.062356000	127.0.0.1	127.0.0.1	OF 1.0	132	of_port_status
650	253.828737000	127.0.0.1	127.0.0.1	OF 1.0	132	of_port_status

```
▷ 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: 53812 (53812), Seq: 173, Ack: 437, Len: 24
▽ 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
```

```
0000  00 00 03 04 00 06 00 00  00 00 00 00 02 0f 08 00   ........ ........
0010  45 00 00 4c 55 62 40 00  40 06 e7 47 7f 00 00 01   E..LUb@. @..G....
0020  7f 00 00 01 19 e9 d2 34  66 12 59 ca d2 18 ba 6e   .......4 f.Y....n
0030  80 18 00 58 fe 40 00 00  01 01 08 0a 00 08 65 9e   ...X.@.. ......e.
```

The source IP and destination IP for these two entries is host 1.

c. 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.

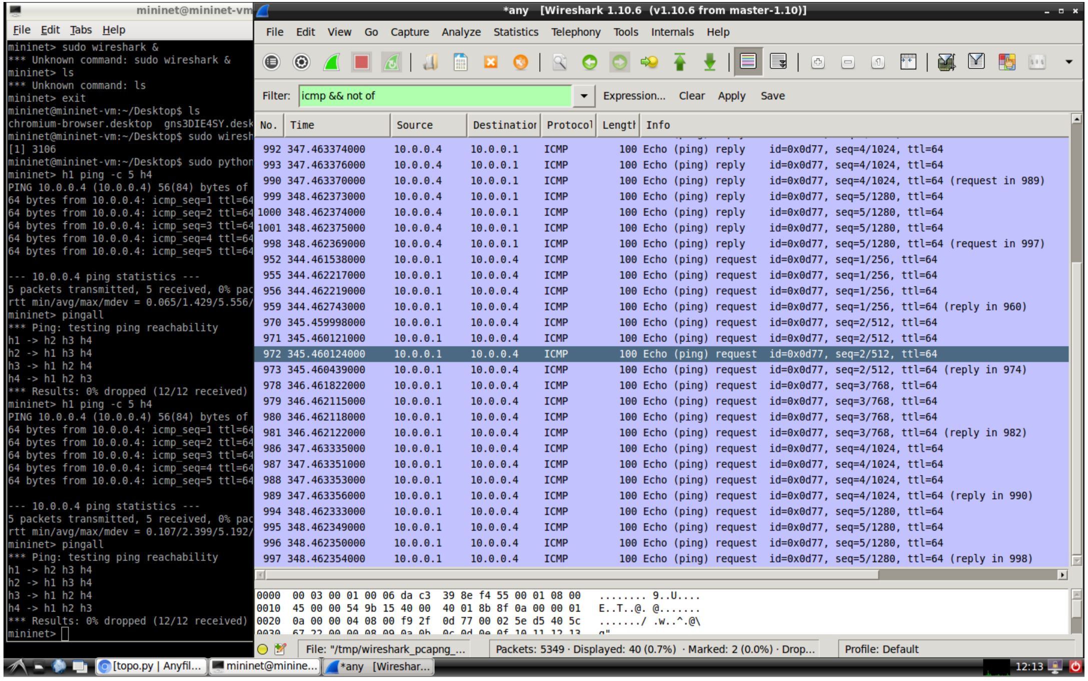

mininet@mininet-vm
File Edit Tabs Help
*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
[topo.py | Anyfil...
mininet@minine...
*any [Wireshar...
12:13

```
mininet> sudo wireshark &
*** Unknown command: sudo wireshark &
mininet> ls
*** Unknown command: ls
mininet> exit
mininet@mininet-vm:~/Desktop$ ls
chromium-browser.desktop  gns3DIE4SY.desk
mininet@mininet-vm:~/Desktop$ sudo wiresh
[1] 3106
mininet@mininet-vm:~/Desktop$ sudo python
mininet> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% pac
rtt min/avg/max/mdev = 0.065/1.429/5.556/
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> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% pac
rtt min/avg/max/mdev = 0.107/2.399/5.192/
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> 
```

Filter: icmp && not of

```
No.  Time           Source      Destinatior Protocol Lengtl Info
 963 344.463596000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=1/256, ttl=64
 964 344.463602000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=1/256, ttl=64
 967 344.464411000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=1/256, ttl=64
 960 344.462760000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=1/256, ttl=64 (request in 959)
 975 345.460764000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=2/512, ttl=64
 976 345.460768000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=2/512, ttl=64
 977 345.461076000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=2/512, ttl=64
 974 345.460460000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=2/512, ttl=64 (request in 973)
 983 346.462145000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=3/768, ttl=64
 984 346.462146000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=3/768, ttl=64
 985 346.462148000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=3/768, ttl=64
 982 346.462141000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=3/768, ttl=64 (request in 981)
 991 347.463373000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=4/1024, ttl=64
 992 347.463374000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=4/1024, ttl=64
 993 347.463376000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=4/1024, ttl=64
 990 347.463370000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=4/1024, ttl=64 (request in 989)
 999 348.462373000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64
1000 348.462374000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64
1001 348.462375000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64
 998 348.462369000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64 (request in 997)
 952 344.461538000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64
 955 344.462217000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64
 956 344.462219000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64
 959 344.462743000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64 (reply in 960)
 970 345.459998000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=2/512, ttl=64
 971 345.460121000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=2/512, ttl=64
 972 345.460124000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=2/512, ttl=64

0000  00 03 00 01 00 06 da c3  39 8e f4 55 00 01 08 00   ........ 9..U....
0010  45 00 00 54 9b 15 40 00  40 01 8b 8f 0a 00 00 01   E..T..@. @.......
0020  0a 00 00 04 08 00 f9 2f  0d 77 00 02 5e d5 40 5c   ......./ .w..^.@\
0030  67 22 00 00 08 09 0a 0b  0c 0d 0e 0f 10 11 12 13   g"
```

File: "/tmp/wireshark_pcapng_... Packets: 5349 · Displayed: 40 (0.7%) · Marked: 2 (0.0%) · Drop... Profile: Default


mininet@mininet-vm
File Edit Tabs Help
*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
[topo.py | Anyfil...
mininet@minine...
*any [Wireshar...
12:13

```
mininet> sudo wireshark &
*** Unknown command: sudo wireshark &
mininet> ls
*** Unknown command: ls
mininet> exit
mininet@mininet-vm:~/Desktop$ ls
chromium-browser.desktop  gns3DIE4SY.desk
mininet@mininet-vm:~/Desktop$ sudo wiresh
[1] 3106
mininet@mininet-vm:~/Desktop$ sudo python
mininet> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% pac
rtt min/avg/max/mdev = 0.065/1.429/5.556/
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> h1 ping -c 5 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% pac
rtt min/avg/max/mdev = 0.107/2.399/5.192/
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> 
```

Filter: icmp && not of

```
No.  Time           Source      Destinatior Protocol Lengtl Info
 992 347.463374000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=4/1024, ttl=64
 993 347.463376000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=4/1024, ttl=64
 990 347.463370000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=4/1024, ttl=64 (request in 989)
 999 348.462373000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64
1000 348.462374000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64
1001 348.462375000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64
 998 348.462369000  10.0.0.4    10.0.0.1    ICMP      100 Echo (ping) reply    id=0x0d77, seq=5/1280, ttl=64 (request in 997)
 952 344.461538000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64
 955 344.462217000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64
 956 344.462219000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64
 959 344.462743000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=1/256, ttl=64 (reply in 960)
 970 345.459998000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=2/512, ttl=64
 971 345.460121000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=2/512, ttl=64
 972 345.460124000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=2/512, ttl=64
 973 345.460439000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=2/512, ttl=64 (reply in 974)
 978 346.461822000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=3/768, ttl=64
 979 346.462115000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=3/768, ttl=64
 980 346.462118000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=3/768, ttl=64
 981 346.462122000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=3/768, ttl=64 (reply in 982)
 986 347.463335000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=4/1024, ttl=64
 987 347.463351000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=4/1024, ttl=64
 988 347.463353000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=4/1024, ttl=64
 989 347.463356000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=4/1024, ttl=64 (reply in 990)
 994 348.462333000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=5/1280, ttl=64
 995 348.462349000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=5/1280, ttl=64
 996 348.462350000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=5/1280, ttl=64
 997 348.462354000  10.0.0.1    10.0.0.4    ICMP      100 Echo (ping) request  id=0x0d77, seq=5/1280, ttl=64 (reply in 998)

0000  00 03 00 01 00 06 da c3  39 8e f4 55 00 01 08 00   ........ 9..U....
0010  45 00 00 54 9b 15 40 00  40 01 8b 8f 0a 00 00 01   E..T..@. @.......
0020  0a 00 00 04 08 00 f9 2f  0d 77 00 02 5e d5 40 5c   ......./ .w..^.@\
0030  67 22 00 00 08 09 0a 0b  0c 0d 0e 0f 10 11 12 13   g"
```

File: "/tmp/wireshark_pcapng_... Packets: 5349 · Displayed: 40 (0.7%) · Marked: 2 (0.0%) · Drop... Profile: Default

After changing the display filter to *icmp && not of,* running the *pingall* command showed 40 entries. There were two different types of ICMP entries that showed up (Echo (ping) reply and Echo (ping) request). These two different types of messages are called type 0 and type 8 ICMP messages respectively. The type 0 message is an Echo reply message and the type 8 message is an Echo request. The two messages are used to verify a valid communication path between the hosts exists.

Sources Used:

http://www.informit.com/articles/article.aspx?p=26557&seqNum=5