

CSP334 : Computer Networks
Lab Assignment No 3
Assignment on traceroute, ping commands

Sahil
2016UCS0008

August 31, 2018

1 Working of *traceroute*:

Brief explanation of its working:

It works by sending packets with TTL values starting from 1 onwards. The packet with TTL 1 reaches the first router, which then responds with ICMP TTL exceeded message, and at the source, its IP address is known from this response packet. Similarly, the packet with TTL 2 reaches the next router, and the process repeats till destination IP is reached or maximum limit of TTL packets has been sent and time out occurs.

1.1 If there were no TTL field in the invocation of the *traceroute* at all:

then we cannot identify the routers in between the source and destination, as the packet would return back only if it reaches the destination or if the TTL value becomes 1 at some point.

1.2 Routers determine whether the TTL value limit has reached:

by decrementing the value of the TTL by 1 as soon as it reaches the router, so whenever a router gets a packet with a TTL value of 1, it knows that the TTL value limit has been reached.

1.3 Intermediate route receiving a packet with ICMP TTL exceeded message:

should not respond again as it already contains TTL exceeded message sent by another router whose IP address should be made available to the source.

1.4 *traceroute* makes use of destination UDP port number which is invalid

so that when it reaches the destination, the connection cannot be established and an error message is returned as the response. Moreover, it helps in distinguishing that the destination has reached since previous responses would be ICMP TTL exceeded, so the INVALID UDP PORT message identifies that we have reached the destination.

1.5 Address of all the routers in between us and destination

is known as the command sends packets with different TTL values starting from 1 to some maximum value, so the one with value of 1 reaches the first router and then router responds with TTL exceeded message, so it sends along its IP

address which is thus known at the source. Similarly, IPA of all the routers in between is known.

1.6 *traceroute* latency

is calculated in terms of the **round trip time**. When the packet is sent from source, a time stamp is put indicating the start time, so after reaching the router which sends a ICMP TTL exceeded message, when the message arrives at the source, the current time is noted, so the difference of the times is the RTT.

2 Problem 2: Executing *traceroute* command:

```
$ traceroute www.yahoo.com
traceroute: Warning: www.yahoo.com has multiple addresses; using 98.137.246.7
traceroute to atsv2-fp-shed.wg1.b.yahoo.com (98.137.246.7), 64 hops max, 52 byte packets
 1  10.10.40.1 (10.10.40.1)  17.867 ms  17.738 ms  11.935 ms
 2  10.10.10.10 (10.10.10.10)  3.698 ms  3.769 ms  16.113 ms
 3  10.119.231.165 (10.119.231.165)  5.281 ms  8.768 ms  2.234 ms
 4  10.148.6.81 (10.148.6.81)  45.037 ms  39.335 ms  39.891 ms
 5  10.255.238.69 (10.255.238.69)  40.761 ms  39.537 ms  39.639 ms
 6  10.255.238.189 (10.255.238.189)  39.238 ms  39.600 ms  40.927 ms
 7  10.152.7.38 (10.152.7.38)  40.594 ms  38.095 ms  38.646 ms
 8  115.248.54.102 (115.248.54.102)  39.698 ms  38.298 ms  *
 9  115.255.239.54 (115.255.239.54)  38.399 ms  36.891 ms  38.183 ms
10  62.216.147.73 (62.216.147.73)  40.203 ms  37.471 ms  47.779 ms
11  xe-9-0-0.0.pjr03.ldn001.Flagtel.com (85.95.27.122)  176.667 ms
    xe-0-0-0.0.pjr03.ldn001.Flagtel.com (85.95.26.238)  180.002 ms
    xe-9-0-0.0.pjr03.ldn001.Flagtel.com (85.95.27.122)  169.759 ms
12  xe-3-3-2.0.cji01.ldn004.Flagtel.com (85.95.27.194)  315.297 ms
    xe-5-2-0.0.cji01.ldn004.Flagtel.com (62.216.128.114)  167.719 ms
    xe-3-3-2.0.cji01.ldn004.Flagtel.com (85.95.27.194)  285.558 ms
13  ge-1-1-0.pat1.the.yahoo.com (195.66.224.129)  189.773 ms  292.140 ms  178.987 ms
14  ae-3.pat1.nyc.yahoo.com (216.115.100.26)  343.407 ms  234.619 ms  278.591 ms
15  ae-7.pat2.dcz.yahoo.com (216.115.96.7)  308.334 ms
    unknown-216-115-110-x.yahoo.com (216.115.110.237)  331.164 ms
    ae-7.pat2.dcz.yahoo.com (216.115.96.7)  339.481 ms
16  184.165.16.49 (184.165.16.49)  276.556 ms
    unknown-216-115-96-x.yahoo.com (216.115.96.2)  302.290 ms
    184.165.16.49 (184.165.16.49)  323.206 ms
17  ae-6.pat1.dnx.yahoo.com (216.115.96.207)  299.176 ms  374.294 ms
    ae-5.pat1.dnx.yahoo.com (216.115.96.34)  305.684 ms
18  ae-8.pat2.gqb.yahoo.com (216.115.96.204)  410.393 ms
    ae-6.pat1.gqb.yahoo.com (216.115.101.195)  325.478 ms
    ae-8.pat2.gqb.yahoo.com (216.115.96.204)  371.466 ms
19  et-19-1-0.msr1.gq1.yahoo.com (66.196.67.99)  402.395 ms
    et-18-1-0.msr1.gq1.yahoo.com (66.196.67.103)  359.900 ms
    et-0-0-0.msr2.gq1.yahoo.com (66.196.67.109)  376.381 ms
20  et-19-1-0.clr1-a-gdc.gq1.yahoo.com (67.195.37.95)  418.717 ms  343.530 ms
    et-19-1-0.clr2-a-gdc.gq1.yahoo.com (67.195.37.99)  410.158 ms
21  et-16-6.bas1-2-flk.gq1.yahoo.com (98.137.120.6)  347.518 ms
    et-16-6.bas2-2-flk.gq1.yahoo.com (98.137.120.14)  398.379 ms  410.508 ms
22  media-router-fp1.prod1.media.vip.gq1.yahoo.com (98.137.246.7)  412.044 ms  388.999 ms  409.106 ms
```

2.1 IPA of *www.yahoo.com*

The IPA obtained from traceroute is : 98.137.246.7

2.2 No. of iterations to determine the route is:

22

2.3 IPA of all the machines between source and destination are:

- 10.10.40.1
- 10.10.10.10
- 10.119.231.165
- 10.148.6.81
- 10.255.238.69
- 10.255.238.189
- 10.152.7.38
- 115.248.54.102
- 115.255.239.54
- 62.216.147.73
- 85.95.27.122
- 85.95.27.194
- 195.66.224.129
- 216.115.100.26
- 216.115.96.7
- 184.165.16.49
- 216.115.96.207
- 216.115.96.204
- 66.196.67.99
- 67.195.37.95
- 98.137.120.6
- 98.137.246.7

2.4 Avg. Round trip time of the packet that reached the destination is:

403.38 milli seconds

3 Problem 3: *traceroute* and *tcpdump*

\$ *sudo tcpdump -n -v udp -c 10* command is run in one window and

\$ *traceroute www.yahoo.com* command is run then. The output is shown in the figure.

```
$ sudo tcpdump -n -v udp -c 10
tcpdump: data link type PKTAP
tcpdump: listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
20:17:51.521333 IP (tos 0x0, ttl 1, id 36409, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33435: UDP, length 24
20:17:51.527240 IP (tos 0x0, ttl 1, id 36410, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33436: UDP, length 24
20:17:51.532734 IP (tos 0x0, ttl 1, id 36411, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33437: UDP, length 24
20:17:51.536966 IP (tos 0x0, ttl 2, id 36412, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33438: UDP, length 24
20:17:51.540590 IP (tos 0x0, ttl 2, id 36413, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33439: UDP, length 24
20:17:51.543296 IP (tos 0x0, ttl 2, id 36414, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33440: UDP, length 24
20:17:51.546464 IP (tos 0x0, ttl 3, id 36415, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33441: UDP, length 24
20:17:51.549129 IP (tos 0x0, ttl 3, id 36416, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33442: UDP, length 24
20:17:51.552325 IP (tos 0x0, ttl 3, id 36417, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33443: UDP, length 24
20:17:51.555048 IP (tos 0x0, ttl 4, id 36418, offset 0, flags [none], proto UDP (17), length 52)
  10.10.40.146.36408 > 98.137.246.8.33444: UDP, length 24
10 packets captured
47 packets received by filter
0 packets dropped by kernel
```

3.1 Packets sent in one iteration:

traceroute sends 3 packets in 1 iteration. This is proved from the above figure, as highlighted, there are 3 packets for each TTL value starting from 1.

```

$ traceroute www.yahoo.com
traceroute: Warning: www.yahoo.com has multiple addresses; using 98.137.246.8
traceroute to atsv2-fp-shed.wg1.b.yahoo.com (98.137.246.8), 64 hops max, 52 byte packets
 1  10.10.40.1 (10.10.40.1)  5.365 ms  5.450 ms  4.225 ms
 2  10.10.10.10 (10.10.10.10)  3.078 ms  2.677 ms  3.163 ms
 3  10.119.231.165 (10.119.231.165)  2.102 ms  3.098 ms  2.744 ms
 4  10.148.6.81 (10.148.6.81)  38.540 ms  37.816 ms  51.573 ms
 5  10.255.238.69 (10.255.238.69)  38.289 ms  42.177 ms  38.533 ms
 6  10.255.238.189 (10.255.238.189)  38.448 ms  41.802 ms  39.582 ms
 7  10.152.7.38 (10.152.7.38)  38.512 ms  38.485 ms  43.003 ms
 8  115.248.54.102 (115.248.54.102)  43.364 ms  40.171 ms  40.779 ms
 9  * * *
10  62.216.147.73 (62.216.147.73)  40.306 ms  38.086 ms  38.171 ms
11  xe-0-1-1.0.pjr03.ldn001.flagtel.com (85.95.26.233)  173.416 ms
    xe-9-0-0.0.pjr03.ldn001.flagtel.com (85.95.27.122)  173.569 ms
    xe-0-0-0.0.pjr03.ldn001.flagtel.com (85.95.26.238)  167.471 ms
12  xe-5-2-0.0.cji01.ldn004.flagtel.com (62.216.128.114)  168.483 ms
    xe-3-3-2.0.cji01.ldn004.flagtel.com (85.95.27.194)  200.213 ms  205.452 ms
13  ge-1-1-0.pat1.the.yahoo.com (195.66.224.129)  169.024 ms  168.683 ms  175.339 ms
14  ae-3.pat1.nyc.yahoo.com (216.115.100.26)  290.623 ms  275.686 ms  340.415 ms
15  unknown-216-115-110-x.yahoo.com (216.115.110.237)  320.492 ms  289.977 ms
    ae-7.pat1.dce.yahoo.com (216.115.104.120)  307.088 ms
16  ae-6.pat1.che.yahoo.com (216.115.96.81)  306.845 ms  306.957 ms

```

3.2 Round trip time:

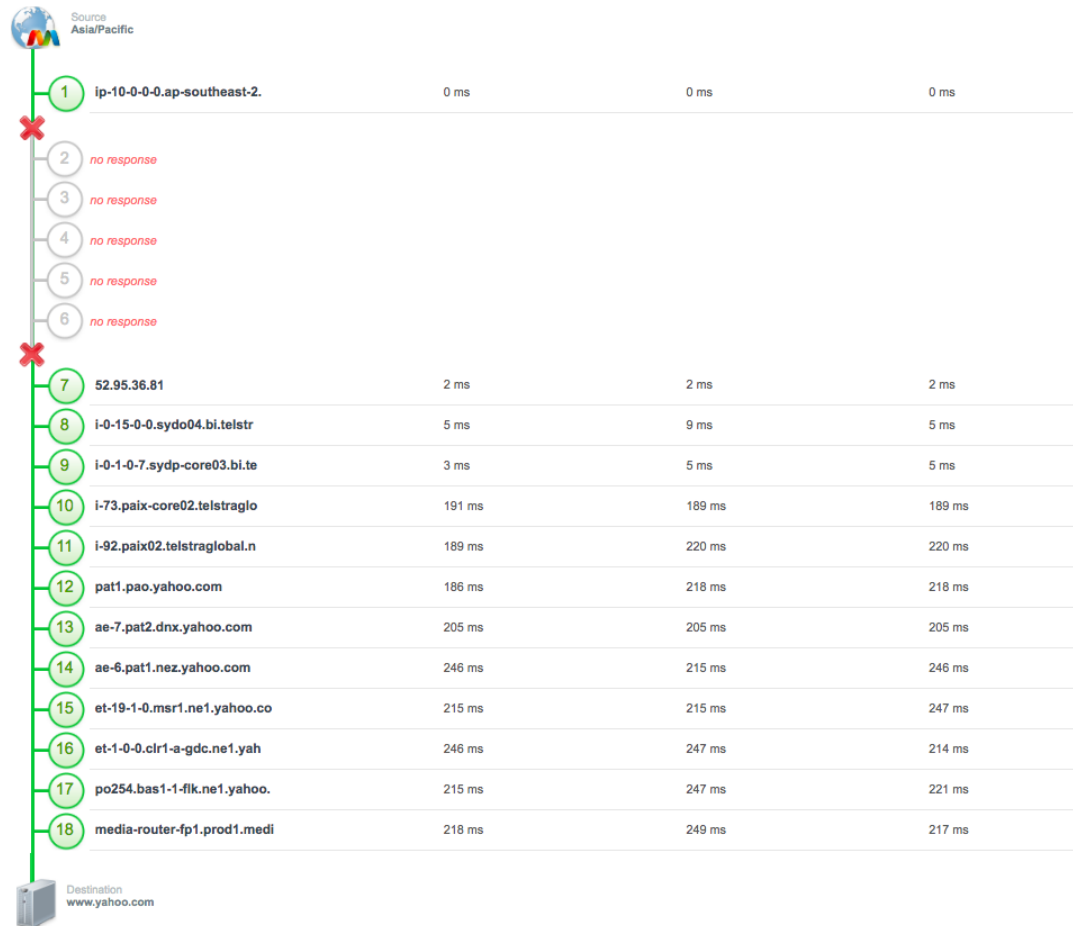
For one specific iteration (consider 14th), individual RTT's of the three probes are: 290.623ms, 275.686ms, 340.415ms. The avg. RTT is: 302.24ms. Comparing it with the individual RTT's, there is a lot of deviation. This is because the RTT depends on the forward path as well as the reverse path, thus, there may be some delays in the reverse path in the 3rd packet, as its RTT is lot higher than the other 2.

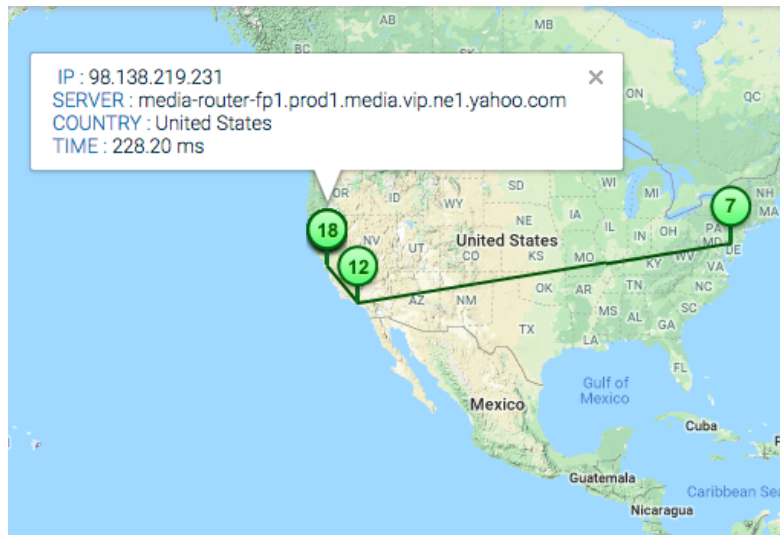
3.3 Port numbers used:

```
1.yahoo.com. (89)
20:40:40.778905 IP (tos 0x0, ttl 118, id 9649, offset 0, flags [none], proto UDP
    8.8.8.8.53 > 10.10.40.146.54657: [udp sum ok] 45290 q: PTR? 6.120.137.98.in-d
1.yahoo.com. (89)
20:40:40.779500 IP (tos 0x0, ttl 21, id 36604, offset 0, flags [none], proto UDP
    10.10.40.146.36542 > 98.137.246.8.33496: [udp sum ok] UDP, length 24
20:40:41.184773 IP (tos 0x0, ttl 21, id 36605, offset 0, flags [none], proto UDP
    10.10.40.146.36542 > 98.137.246.8.33497: [udp sum ok] UDP, length 24
20:40:41.608074 IP (tos 0x0, ttl 255, id 31483, offset 0, flags [none], proto UDP
    10.10.40.146.57292 > 8.8.8.8.53: [udp sum ok] 5467+ PTR? 14.120.137.98.in-add
20:40:41.905170 IP (tos 0x0, ttl 118, id 11783, offset 0, flags [none], proto UDP
    8.8.8.8.53 > 10.10.40.146.57292: [udp sum ok] 5467 q: PTR? 14.120.137.98.in-d
q1.yahoo.com. (90)
20:40:41.905839 IP (tos 0x0, ttl 22, id 36606, offset 0, flags [none], proto UDP
    10.10.40.146.36542 > 98.137.246.8.33498: [udp sum ok] UDP, length 24
20:40:42.314984 IP (tos 0x0, ttl 22, id 36607, offset 0, flags [none], proto UDP
    10.10.40.146.36542 > 98.137.246.8.33499: [udp sum ok] UDP, length 24
20:40:42.720804 IP (tos 0x0, ttl 22, id 36608, offset 0, flags [none], proto UDP
    10.10.40.146.36542 > 98.137.246.8.33500: [udp sum ok] UDP, length 24
^C
103 packets captured
884 packets received by filter
0 packets dropped by kernel
```

For each iteration, it uses different port numbers for the destination. As in the above figure, for TTL 21, port numbers used are: 33496, 33497 whereas for TTL 22, port numbers 33498, 33499, 33500 are used. This might be because simultaneously packets might come to the destination at the same port otherwise.

4 Problem 4: Visual *traceroute*





- Source IP address: 52.95.36.81
- Destination IP address: 98.138.219.231

5 Problem 5: Knowing whether firewall has come in the way

We can know that a firewall has come into way and stopped the packet, if we are unable to reach to the destination while using the traceroute command. As seen in the figure below, * * * starts appearing in the output.

```
$ traceroute iitd.ac.in
traceroute to iitd.ac.in (103.27.9.20), 64 hops max, 52 byte packets
 1 10.10.40.1 (10.10.40.1) 8.761 ms 13.106 ms 6.134 ms
 2 10.10.10.10 (10.10.10.10) 3.048 ms 5.459 ms 3.153 ms
 3 10.119.231.165 (10.119.231.165) 9.556 ms 21.178 ms 3.969 ms
 4 10.148.6.81 (10.148.6.81) 16.857 ms 28.995 ms 28.846 ms
 5 10.255.238.69 (10.255.238.69) 17.463 ms 24.383 ms 15.387 ms
 6 10.255.237.1 (10.255.237.1) 20.213 ms 17.716 ms
  10.255.238.117 (10.255.238.117) 20.394 ms
 7 10.1.200.142 (10.1.200.142) 21.834 ms 16.142 ms 16.667 ms
 8 10.119.233.65 (10.119.233.65) 102.626 ms 14.159 ms 21.517 ms
 9 10.119.233.66 (10.119.233.66) 22.336 ms 47.070 ms 28.297 ms
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
^C
```

But, when the same *traceroute* command is used with **-I** option, i.e. sending ICMP requests only, we get the complete output:

```
$ traceroute -I iitd.ac.in
traceroute to iitd.ac.in (103.27.9.20), 64 hops max, 72 byte packets
 1  10.10.40.1 (10.10.40.1)  6.738 ms  5.342 ms  5.317 ms
 2  10.10.10.10 (10.10.10.10)  5.344 ms  3.783 ms  4.184 ms
 3  10.119.231.165 (10.119.231.165)  4.184 ms  5.661 ms  4.322 ms
 4  10.148.6.81 (10.148.6.81)  18.133 ms  28.048 ms  18.300 ms
 5  10.255.238.69 (10.255.238.69)  16.054 ms  16.144 ms  16.294 ms
 6  10.255.237.1 (10.255.237.1)  18.124 ms  18.997 ms  17.097 ms
 7  10.1.200.142 (10.1.200.142)  16.205 ms  17.843 ms  15.925 ms
 8  10.119.233.65 (10.119.233.65)  29.890 ms  16.658 ms  16.016 ms
 9  10.119.233.66 (10.119.233.66)  15.884 ms  17.154 ms  19.271 ms
10  103.27.9.20 (103.27.9.20)  28.973 ms  21.798 ms  26.860 ms
11  103.27.9.20 (103.27.9.20)  16.463 ms  15.456 ms  18.245 ms
12  103.27.9.20 (103.27.9.20)  16.968 ms  20.281 ms  16.036 ms
```

Thus, the IP address of the firewall is based on the destination IP address as the last few IP addresses are similar in the above output.

6 Problem 6: Last IP address in *traceroute*

If a firewall has not obstructed the packet sent, the last IP address appearing in the *traceroute* must indicate the IP address of the destination as *traceroute* command terminates when it receives UDP Port INVALID response from the destination IP. In the above figure shown for problem 5, clearly the IPA of destination matches the last IPA.

7 Problem 7: Usages of the *ping* program

```
$ ping www.iitd.ac.in
PING www.iitd.ac.in (103.27.9.20): 56 data bytes
64 bytes from 103.27.9.20: icmp_seq=0 ttl=53 time=22.740 ms
64 bytes from 103.27.9.20: icmp_seq=1 ttl=53 time=56.193 ms
64 bytes from 103.27.9.20: icmp_seq=2 ttl=53 time=15.031 ms
64 bytes from 103.27.9.20: icmp_seq=3 ttl=53 time=15.642 ms
^C
--- www.iitd.ac.in ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 15.031/27.401/56.193/16.897 ms
sahilbansal at Bluecp0cod3r in ~
$ ping www.iitd.ac.in
ping: cannot resolve www.iitd.ac.in: Unknown host
```

- Ping is mainly used to check whether a machine is connected to the network. It also displays the round trip times. As in the figure shown above, the second ping command is unable to resolve the host as the machine is not connected to the network.

```

sahilbansal at Bluecp0cod3r in ~
$ ping -D -s 1500 -c 1 www.google.com
PING www.google.com (172.217.166.228): 1500 data bytes
ping: sendto: Message too long
^C
--- www.google.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
sahilbansal at Bluecp0cod3r in ~
$ ping -D -s 1472 -c 1 www.google.com
PING www.google.com (172.217.166.228): 1472 data bytes
72 bytes from 172.217.166.228: icmp_seq=0 ttl=52 time=15.672 ms
wrong total length 92 instead of 1500

--- www.google.com ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 15.672/15.672/15.672/0.000 ms
sahilbansal at Bluecp0cod3r in ~
$ ping -D -s 1473 -c 1 www.google.com
PING www.google.com (172.217.166.228): 1473 data bytes
ping: sendto: Message too long
^C
--- www.google.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
sahilbansal at Bluecp0cod3r in ~
$ ping -D -s 1473 -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 1473 data bytes
1481 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.075 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.075/0.075/0.075/0.000 ms
sahilbansal at Bluecp0cod3r in ~
$ ping -D -s 1500 -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 1500 data bytes
1508 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.078 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.078/0.078/0.078/0.000 ms

```

- It can be used to know the maximum frame size of message allowed by the Network Interface Card. As in the figure shown above, the **-D** option is used to set the **don't fragment** bit and **-s** option is used to specify the size of the packet, so we get the reply **Message too long**, if the size exceeds the maximum allowed message size.

We can see from the first 3 commands in above figure, that the maximum size allowed is 1472, this is because after adding the headers at the link layer (8 bytes) and network layer (28 bytes), the size becomes 1500 which is the default maximum allowed size.

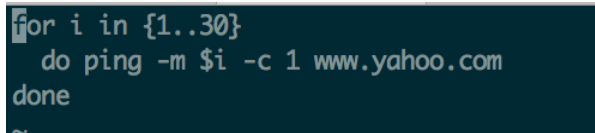
Also, it allows sending any size packet to the localhost since it is the same machine, and also adds only 8 bytes of header (link layer), since there is

no need of network layer in this case.

- It can be used to identify the presence of a firewall as some good firewalls block time stamp requests and source routed packets.

8 Problem 8: Using ping to simulate the working of *traceroute*

A file named **ping_as_traceroute** is created which contains the shell script to use *ping* as *traceroute*. It is displayed below:



```
for i in {1..30}
do ping -m $i -c 1 www.yahoo.com
done
```

The **-m** option is used to send packet with a specific TTL value, and **-c** option is used to send a particular no. of packets. So for all TTL values from 1 to 30, we send a single packet, and in response we get the IPA of all the routers in between the source (our machine) and the destination. This value 30 can be large if we reach the destination in first few iterations only, so it is just a maximum value chosen for convenience, we can adjust it by hit and trial. The output received is displayed in the following figure:

```

$ vi ping_as_traceroute
sahilbansal at Bluecp0cod3r in ~
$ chmod 777 ping_as_traceroute
sahilbansal at Bluecp0cod3r in ~
$ ./ping_as_traceroute
PING atsv2-fp-shed.wg1.b.yahoo.com (106.10.250.10): 56 data bytes
36 bytes from 10.10.40.1: Time to live exceeded
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
 4 5 00 5400 e3b2 0 0000 01 01 3f46 10.10.40.146 106.10.250.10

--- atsv2-fp-shed.wg1.b.yahoo.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
PING atsv2-fp-shed.wg1.b.yahoo.com (106.10.250.10): 56 data bytes
72 bytes from 10.10.10.10: Time to live exceeded
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
 4 5 00 5400 dc89 0 0000 01 01 466f 10.10.40.146 106.10.250.10

--- atsv2-fp-shed.wg1.b.yahoo.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
PING atsv2-fp-shed.wg1.b.yahoo.com (98.137.246.7): 56 data bytes
36 bytes from 10.119.231.165: Time to live exceeded
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
 4 5 00 5400 e53d 0 0000 01 01 493f 10.10.40.146 98.137.246.7

--- atsv2-fp-shed.wg1.b.yahoo.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
PING atsv2-fp-shed.wg1.b.yahoo.com (98.137.246.7): 56 data bytes
152 bytes from 10.148.6.81: Time to live exceeded
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
 4 5 00 5400 db1a 0 0000 01 01 5362 10.10.40.146 98.137.246.7

--- atsv2-fp-shed.wg1.b.yahoo.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
PING atsv2-fp-shed.wg1.b.yahoo.com (98.137.246.7): 56 data bytes
152 bytes from 10.255.238.69: Time to live exceeded
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
 4 5 00 5400 98f1 0 0000 02 01 948b 10.10.40.146 98.137.246.7

--- atsv2-fp-shed.wg1.b.yahoo.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
PING atsv2-fp-shed.wg1.b.yahoo.com (98.137.246.7): 56 data bytes
148 bytes from 10.255.238.189: Time to live exceeded

```

9 Problem 9: Approaches that can be used to do a *ping sweep*

Ping sweep is used to find which range of IP addresses are active on a host. It sends multiple ICMP requests to multiple hosts. If the given address is live, the response is received. The various methods to do ping sweep are:

- **nmap:** nmap is a tool for generally used for port scanning. The **-sP** option can be used with it to do ping sweeping. The range of IP can then be specified. It then works similar to ping my sending ICMP requests and waiting for a response before sending the next packet to the next host.

```

sahilbansal@Bluecp0cod3r:~$ ping -c 1 www.google.com
PING www.google.com (172.217.31.4): 56 data bytes
64 bytes from 172.217.31.4: icmp_seq=0 ttl=52 time=22.279 ms

--- www.google.com ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 22.279/22.279/22.279/0.000 ms
sahilbansal at Bluecp0cod3r in ~
$ nmap -sP 172.217.31.0-10

Starting Nmap 7.40 ( https://nmap.org ) at 2018-08-31 11:12 IST
Nmap scan report for del03s01-in-f0.1e100.net (172.217.31.0)
Host is up (0.014s latency).
Nmap scan report for del03s01-in-f1.1e100.net (172.217.31.1)
Host is up (0.018s latency).
Nmap scan report for del03s01-in-f2.1e100.net (172.217.31.2)
Host is up (0.013s latency).
Nmap scan report for del03s01-in-f3.1e100.net (172.217.31.3)
Host is up (0.013s latency).
Nmap scan report for del03s01-in-f4.1e100.net (172.217.31.4)
Host is up (0.015s latency).
Nmap scan report for del03s01-in-f5.1e100.net (172.217.31.5)
Host is up (0.015s latency).
Nmap scan report for del03s01-in-f6.1e100.net (172.217.31.6)
Host is up (0.014s latency).
Nmap scan report for del03s01-in-f7.1e100.net (172.217.31.7)
Host is up (0.017s latency).
Nmap scan report for del03s01-in-f8.1e100.net (172.217.31.8)
Host is up (0.014s latency).
Nmap scan report for del03s01-in-f9.1e100.net (172.217.31.9)
Host is up (0.014s latency).
Nmap scan report for del03s01-in-f10.1e100.net (172.217.31.10)
Host is up (0.014s latency).
Nmap done: 11 IP addresses (11 hosts up) scanned in 0.22 seconds

```

- **fping:** It works a little differently than ping sending multiple ICMP requests at the same time and not waiting for the responses before sending the next one. As seen in the output below the result for 172.217.167.1 is arrived before 172.217.167.0. This proves it. The **-g** option is used to specify the start and the end IPA for the ping sweep.

```

$ fping -g 172.217.167.0 172.217.167.10
172.217.167.1 is alive
172.217.167.0 is alive
172.217.167.2 is alive
172.217.167.3 is alive
172.217.167.4 is alive
172.217.167.5 is alive
172.217.167.6 is alive
172.217.167.7 is alive
172.217.167.8 is alive
172.217.167.9 is alive
172.217.167.10 is alive

```

- **nmap using TCP:** nmap can also be used to do ping sweep by sending TCP requests instead of ICMP requests. The **-PT** option can be used followed by the port number without any space to specify the port number on which to send the request. As seen in the below output, the host is alive only when sending a request on port 80 since that is the standard HTTP port and we are doing ping on google, so it doesn't respond on other ports.

```
$ nmap -sP -PT33333 172.217.31.0-10

Starting Nmap 7.40 ( https://nmap.org ) at 2018-08-31 11:18 IST
Nmap done: 11 IP addresses (0 hosts up) scanned in 4.02 seconds
sahilbansal at Bluecp0cod3r in ~
$ nmap -sP -PT80 172.217.31.0-10

Starting Nmap 7.40 ( https://nmap.org ) at 2018-08-31 11:18 IST
Nmap scan report for del03s01-in-f0.1e100.net (172.217.31.0)
Host is up (0.021s latency).
Nmap scan report for del03s01-in-f1.1e100.net (172.217.31.1)
Host is up (0.025s latency).
Nmap scan report for del03s01-in-f2.1e100.net (172.217.31.2)
Host is up (0.025s latency).
Nmap scan report for del03s01-in-f3.1e100.net (172.217.31.3)
Host is up (0.025s latency).
Nmap scan report for del03s01-in-f4.1e100.net (172.217.31.4)
Host is up (0.025s latency).
Nmap scan report for del03s01-in-f5.1e100.net (172.217.31.5)
Host is up (0.025s latency).
Nmap scan report for del03s01-in-f6.1e100.net (172.217.31.6)
Host is up (0.014s latency).
Nmap scan report for del03s01-in-f7.1e100.net (172.217.31.7)
Host is up (0.025s latency).
Nmap scan report for del03s01-in-f8.1e100.net (172.217.31.8)
Host is up (0.025s latency).
Nmap scan report for del03s01-in-f9.1e100.net (172.217.31.9)
Host is up (0.024s latency).
Nmap scan report for del03s01-in-f10.1e100.net (172.217.31.10)
Host is up (0.025s latency).
Nmap done: 11 IP addresses (11 hosts up) scanned in 0.17 seconds
sahilbansal at Bluecp0cod3r in ~
$ nmap -sP -PT81 172.217.31.0-10

Starting Nmap 7.40 ( https://nmap.org ) at 2018-08-31 11:18 IST
Nmap done: 11 IP addresses (0 hosts up) scanned in 4.02 seconds
```

- **bash for loop:** We can write a for loop for doing a ping sweep by running ping to different address each time. As in the below output, we ping all IP addresses from 172.217.167.0 – 172.217.167.10.


```
$ for i in {0..10}; do ping -c 1 172.217.167.$i; done
PING 172.217.167.0 (172.217.167.0): 56 data bytes
64 bytes from 172.217.167.0: icmp_seq=0 ttl=52 time=17.608 ms

--- 172.217.167.0 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.608/17.608/17.608/0.000 ms
PING 172.217.167.1 (172.217.167.1): 56 data bytes
64 bytes from 172.217.167.1: icmp_seq=0 ttl=52 time=20.836 ms

--- 172.217.167.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 20.836/20.836/20.836/0.000 ms
PING 172.217.167.2 (172.217.167.2): 56 data bytes
64 bytes from 172.217.167.2: icmp_seq=0 ttl=52 time=35.448 ms

--- 172.217.167.2 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 35.448/35.448/35.448/0.000 ms
PING 172.217.167.3 (172.217.167.3): 56 data bytes
64 bytes from 172.217.167.3: icmp_seq=0 ttl=52 time=14.534 ms
```