Name : Jiten Sidhpura
TE COMPS
BATCH D
UID : 2018130051

CEL 51, DCCN, Monsoon 2020
Lab 2: Basic Network Utilities

This lab introduces some basic network monitoring/analysis tools. There are a few exercises along the way. You should write up answers to the ***ping*** and ***traceroute*** exercises and turn them in next lab. (You should try out each tool, whether it is needed for an exercise or not!).

Prerequisite: Basic understanding of command line utilities of Linux Operating system.

Some Basic command line Networking utilities

Start with a few of the most basic command line tools. These commands are available on Unix, including Linux (and the first two, at least, are also for Windows). Some parameters or options might differ on different operating systems. Remember that you can use man <command> to get information about a command and its options.

ping — The command ping <host> sends a series of packets and expects to receieve a response to each packet. When a return packet is received, ping reports the round trip time (the time between sending the packet and receiving the response). Some routers and firewalls block ping requests, so you might get no reponse at all. Ping can be used to check whether a computer is up and running, to measure network delay time, and to check for dropped packets indicating network congestion. Note that <host> can be either a domain name or an IP address. By default, ping will send a packet every second indefinitely; stop it with Control-C

Network latency, specifically round trip time (RTT), can be measured using `ping`, which sends ICMP packets. The syntax for the command in Linux or Mac OS is:

```
ping [-c <count>] [-s <packetsize>] <hostname>
```

The syntax in Windows is:

```
ping [-n <count>] [-l <packetsize>] <hostname>
```

The default number of ICMP packets to send is either infinite (in Linux and Mac OS) or 4 (in Windows). The default packet size is either 64 bytes (in Linux) or 32 bytes (in Windows). You can specify either a hostname (e.g., `spit.ac.in`) or an IP address.

To save the output from ping to a file, include a greater than symbol and a file name at the end of the command. For example:

```
ping -c 10 google.com > ping_c10_s64_google.log
```

EXPERIMENTS WITH PING

1. Ping the any hosts 10 times (i.e., packet count is 10) with a packet size of 64 bytes, 100 bytes, 500 bytes, 1000 bytes, 1400 bytes

PACKET COUNT = 10
HOST : www.geeksforgeeks.org

PACKET SIZE = 32 BYTES:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 10 -s 32  www.geeksforgeeks.org
PING www.geeksforgeeks.org(2405:200:1602::312c:8220 (2405:200:1602::312c:8220)) 32 data bytes
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=1 ttl=61 time=38.2 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=2 ttl=61 time=140 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=3 ttl=61 time=76.9 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=4 ttl=61 time=197 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=5 ttl=61 time=195 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=6 ttl=61 time=53.5 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=7 ttl=61 time=53.3 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=8 ttl=61 time=37.4 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=9 ttl=61 time=69.8 ms
40 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=10 ttl=61 time=130 ms

--- www.geeksforgeeks.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9012ms
rtt min/avg/max/mdev = 37.407/99.307/197.591/58.905 ms
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

PACKET SIZE = 64 BYTES:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 10  www.geeksforgeeks.org
PING www.geeksforgeeks.org(2405:200:1602::312c:8220 (2405:200:1602::312c:8220)) 56 data bytes
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=1 ttl=61 time=42.1 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=2 ttl=61 time=78.0 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=3 ttl=61 time=98.4 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=4 ttl=61 time=397 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=5 ttl=61 time=75.0 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=6 ttl=61 time=116 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=7 ttl=61 time=185 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=8 ttl=61 time=168 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=9 ttl=61 time=173 ms
64 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=10 ttl=61 time=192 ms

--- www.geeksforgeeks.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9010ms
rtt min/avg/max/mdev = 42.179/152.725/397.461/95.581 ms
```

PACKET SIZE = 100 BYTES:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 10 -s 100  www.geeksforgeeks.org
PING www.geeksforgeeks.org(2405:200:1602::312c:8220 (2405:200:1602::312c:8220)) 100 data bytes
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=1 ttl=61 time=78.1 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=2 ttl=61 time=288 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=3 ttl=61 time=114 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=4 ttl=61 time=136 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=5 ttl=61 time=187 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=6 ttl=61 time=107 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=7 ttl=61 time=231 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=8 ttl=61 time=33.7 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=9 ttl=61 time=452 ms
108 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=10 ttl=61 time=67.3 ms

--- www.geeksforgeeks.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9010ms
rtt min/avg/max/mdev = 33.779/169.657/452.302/119.730 ms
```

PACKET SIZE = 500 BYTES:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 10 -s 500  www.geeksforgeeks.org
PING www.geeksforgeeks.org(2405:200:1602::312c:8220 (2405:200:1602::312c:8220)) 500 data bytes
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=1 ttl=61 time=35.8 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=2 ttl=61 time=65.1 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=3 ttl=61 time=74.8 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=4 ttl=61 time=95.7 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=5 ttl=61 time=88.2 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=6 ttl=61 time=67.3 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=7 ttl=61 time=85.1 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=8 ttl=61 time=106 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=9 ttl=61 time=146 ms
508 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=10 ttl=61 time=154 ms

--- www.geeksforgeeks.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 35.845/91.997/154.234/34.533 ms
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

PACKET SIZE = 1000 BYTES:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 10 -s 1000  www.geeksforgeeks.org
PING www.geeksforgeeks.org(2405:200:1602::312c:8220 (2405:200:1602::312c:8220)) 1000 data bytes
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=1 ttl=61 time=77.1 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=2 ttl=61 time=149 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=3 ttl=61 time=143 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=4 ttl=61 time=192 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=5 ttl=61 time=68.6 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=6 ttl=61 time=75.8 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=7 ttl=61 time=117 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=8 ttl=61 time=138 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=9 ttl=61 time=136 ms
1008 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=10 ttl=61 time=183 ms

--- www.geeksforgeeks.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 68.685/128.336/192.876/41.333 ms
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

PACKET SIZE = 1400 BYTES:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 10 -s 1400  www.geeksforgeeks.org
PING www.geeksforgeeks.org(2405:200:1602::312c:8208 (2405:200:1602::312c:8208)) 1400 data bytes
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=1 ttl=61 time=55.6 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=2 ttl=61 time=33.4 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=3 ttl=61 time=103 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=4 ttl=61 time=89.8 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=5 ttl=61 time=138 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=6 ttl=61 time=66.7 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=7 ttl=61 time=200 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=8 ttl=61 time=77.1 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=9 ttl=61 time=603 ms
1408 bytes from 2405:200:1602::312c:8208 (2405:200:1602::312c:8208): icmp_seq=10 ttl=61 time=82.7 ms

--- www.geeksforgeeks.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9010ms
rtt min/avg/max/mdev = 33.454/145.075/603.847/159.205 ms
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

OBSERVATIONS:

1. EXTRA 8 BYTES ARE SENT WHICH ARE FOR THE ICMP HEADER.
2. TIME FOR PINGING IS NOT SAME FOR ALL THE REQUESTS.
3. TTL (TIME TO LIVE) IS SAME FOR ALL THE REQUESTS.

QUESTIONS ABOUT LATENCY

Now look at the results you gathered and answer the following questions about latency. Store your answers in a file named `ping.txt`.

1. Does the average RTT vary between different hosts? What aspects of latency (transmit, propagation, and queueing delay) might impact this and why?
2. Does the average RTT vary with different packet sizes? What aspects of latency (transmit, propagation, and queueing delay) might impact this and why?

Answers :

1.RTT is duration in milliseconds (ms) it takes for a network request to go from a starting point to a destination and back again to the starting point.

Propogation Delay:

Propagation delay is the amount of time it takes for the head of the signal to travel from the sender to the receiver which depends on the distance between source and host.

Transmission Delay:

Transmission delay is a function of the packet's length and has nothing to do with the distance between the two nodes.

Queuing Delay:

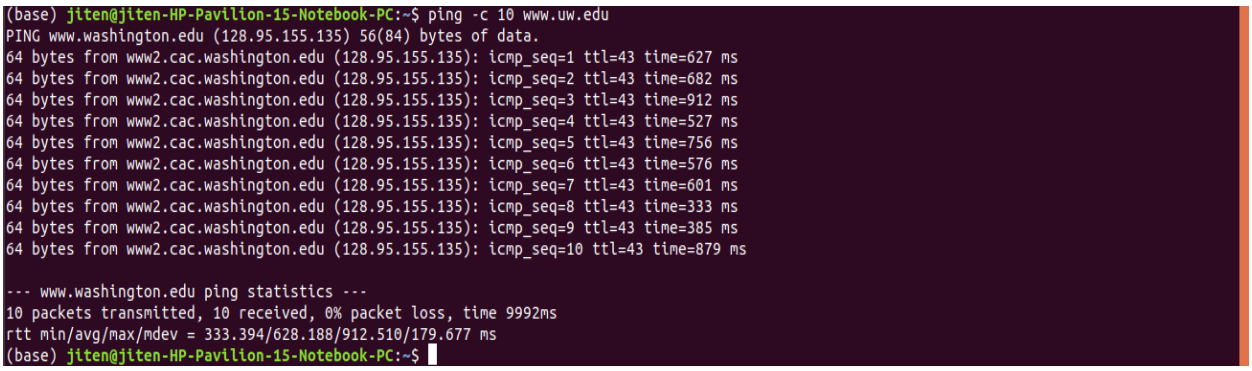
It is the time a packet spends in routing queues depends on the number of packets, size of the packet and bandwidth.

2. Since RTT depends on transmission and queuing delay which depends on packet size. So RTT gets affected by different size of the packets used for communication.

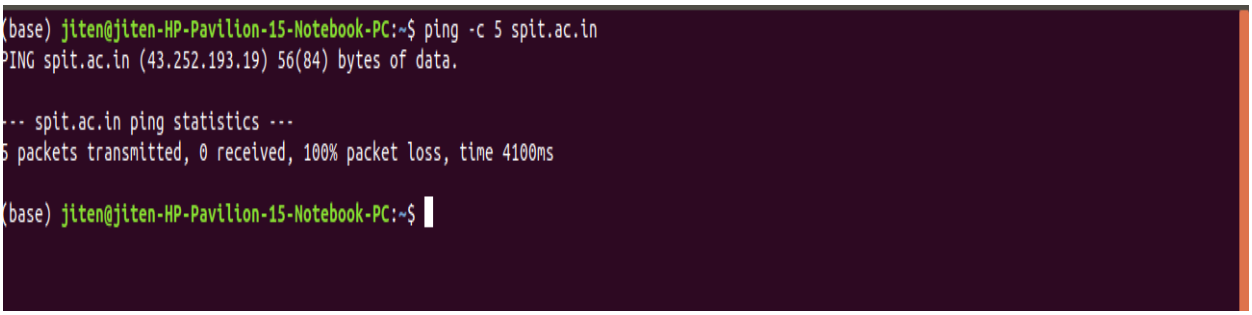
Exercise 1: Experiment with ping to find the round trip times to a variety of destinations. Write up any interesting observations, including in particular how the round trip time compares to the physical distance. Here are few places from who to get replies: www.uw.edu, www.cornell.edu, berkeley.edu, www.uchicago.edu, www.ox.ac.uk (England), www.u-tokyo.ac.jp (Japan).

OTHER HOSTS:

1. www.uw.edu



2. spit.ac.in



Observations:

When host berkeley.edu is pinged with 100 bytes packets, response is received from an IP address 128.95.155.198 The average RTT is almost 0.710 seconds and loss of 10% in packet information. Average RTT depends upon Distance, Transmission medium , Number of network hops, Traffic level and the time taken by server for responding a request.

nslookup — The command nslookup <host> will do a DNS query to find and report the IP address (or addresses) for a domain name or the domain name corresponding to an IP address. To do this, it contacts a "DNS server." Default DNS servers are part of a computer's network configuration. (For a static IP address in Linux, they are configured in the file /etc/network/interfaces that you encountered in the last lab.) You can specify a different DNS server to be used by nslokup by adding the server name or IP address to the command: nslookup <host> <server>

ifconfig — You used ifconfig in the previous lab. When used with no parameters, ifconfig reports some information about the computer's network interfaces. This usually includes lo which stands for localhost; it can be used for communication between programs running on the same computer. Linux often has an interface named eth0, which is the first ethernet card. The information is different on Mac OS and Linux, but includes the IP or "inet" address and ethernet or "hardware" address for an ethernet card. On Linux, you get the number of packets received (RX) and sent (TX), as well as the number of bytes transmitted and received. (A better place to

monitor network bytes on our Linux computers is in the GUI program System Monitor, if it is installed!!!.)

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ifconfig
br-802eadefcb59: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.19.0.1  netmask 255.255.0.0  broadcast 172.19.255.255
        ether 02:42:08:f9:59:b9  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

br-82b0296f0257: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.18.0.1  netmask 255.255.0.0  broadcast 172.18.255.255
        ether 02:42:b9:9b:99:57  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:f6:2f:a7:aa  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 6c:c2:17:74:87:43  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4082  bytes 406397 (406.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
```

```
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:f6:2f:a7:aa  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 6c:c2:17:74:87:43  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4082  bytes 406397 (406.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4082  bytes 406397 (406.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.219  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 2405:204:280:39de:ad1f:f598:fe1f:7689  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::924:750e:735c:57d3  prefixlen 64  scopeid 0x20<link>
        inet6 2405:204:280:39de:c6ba:6205:32ff:7aa9  prefixlen 64  scopeid 0x0<global>
        ether 9c:ad:97:c7:ab:17  txqueuelen 1000  (Ethernet)
        RX packets 221996  bytes 78087813 (78.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 34368
        TX packets 31862  bytes 4643074 (4.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 18

(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

netstat — The netstat command gives information about network connections. I often use netstat -t -n which lists currently open TCP connections (that's the "-t" option) by IP address rather than domain name (that's the "-n" option). Add the option "-l" (lower case ell) to list listening sockets, that is sockets that have been opened by server programs to wait for connection requests from clients: netstat -t -n -l. (On Mac, use netstat -p tcp to list tcp connections, and add "-a" to include listening sockets in the list.)**telnet** — Telnet is an old program for remote login. It's not used so much for that any more, since it has no security features. But basically, all it does is open a connection to a server and allow server and client to send lines of plain text to each other. It can be used to check that it's possible to connect to a server and, if the server communicates in plain text, even to interact with the server by hand. Since the Web uses a plain text protocol, you can use telnet to connect to a web client and play the part of the web browser. I will suggest that you to do this with your own web server when you write it, but you might want to try it now. When you use telnet in this way, you need to specify both the host and the port number to which you want to connect: telent <host> <port>. For example, to connect to the web server on www.spit.ac.in: telnet spit.ac.in 80

traceroute — Traceroute is discussed in man utility. The command traceroute <host> will show routers encountered by packets on their way from your computer to a specified <host>. For each n = 1, 2, 3,..., traceroute sends a packet with "time-to-live" (ttl) equal to n. Every time a router forwards a packet, it decreases the ttl of the packet by one. If the ttl drops to zero, the router discards the packet and sends an error message back to the sender of the packet. (Again, as with ping, the packets might be blocked or might not even be sent, so that the error messages will never be received.) The sender gets the identity of the router from the source of the error message. Traceroute will send packets until n reaches some set upper bound or until a packet actually gets through to the destination. It actually does this three times for each n. In this way, it identifies routers that are one step, two steps, three steps, ... away from the source computer. A packet for which no response is received is indicated in the output as a *.

Traceroute is installed on the computers. If was not installed in your virtual server last week, but you can install it with the command sudo apt-get install traceroute

The path taken through a network, can be measured using `traceroute`. The syntax for the command in Linux is:

```
traceroute <hostname>
```

The syntax in Windows is:

```
tracert <hostname>
```

You can specify either a hostname (e.g., `cs.iitb.ac.in`) or an IP address (e.g., `128.105.2.6`).

1.2.1 EXPERIMENTS WITH TRACEROUTE

From **your machine** traceroute to the following hosts:

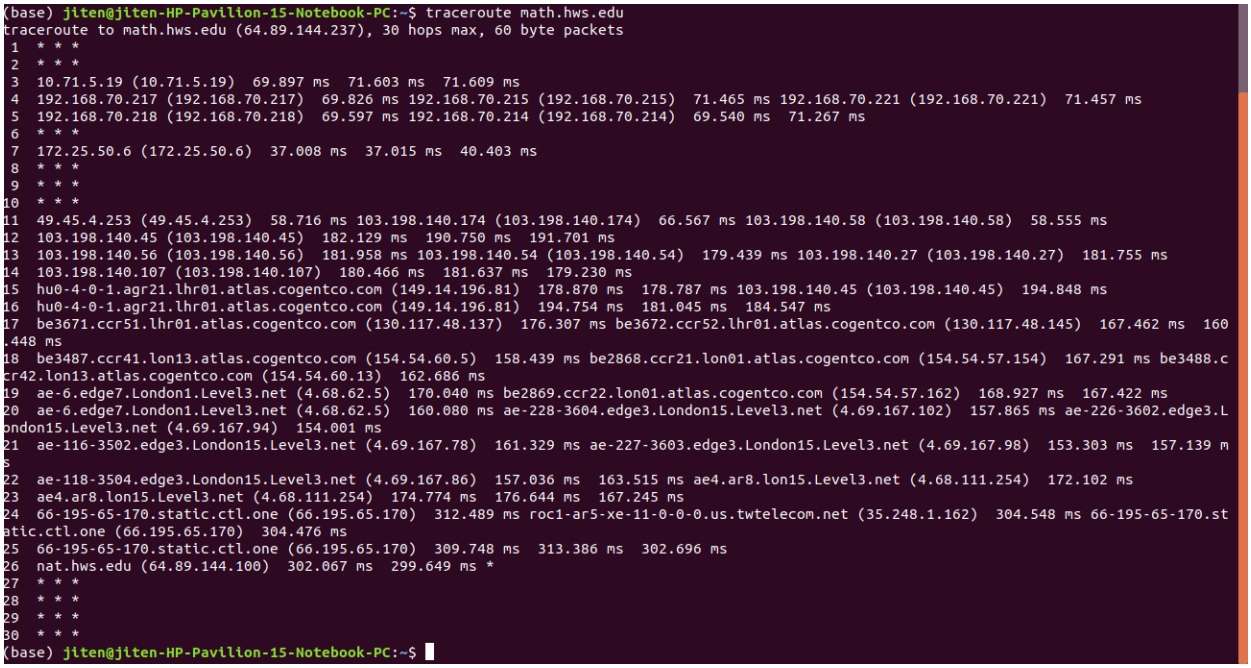
1. ee.iitb.ac.in

2. mscs.mu.edu
3. www.cs.grinnell.edu
4. csail.mit.edu
5. cs.stanford.edu
6. cs.manchester.ac.uk

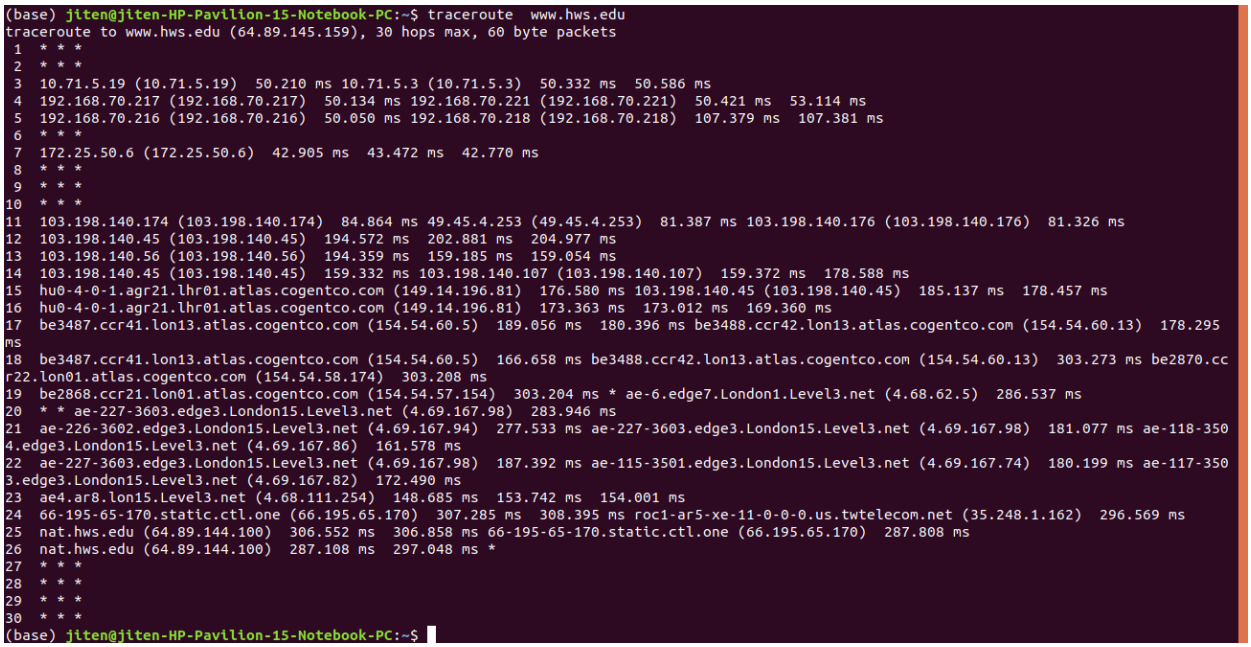
Store the output of each traceroute command in a separate file named `traceroute_HOSTNAME.log`, replacing `HOSTNAME` with the hostname for end-host you pinged (e.g., `traceroute_ee.iitb.ac.in.log`).

Exercise 2: (Very short.) Use traceroute to trace the route from your computer to math.hws.edu and to www.hws.edu. Explain the difference in the results.

1. **http://math.hws.edu/**



2. www.hws.edu



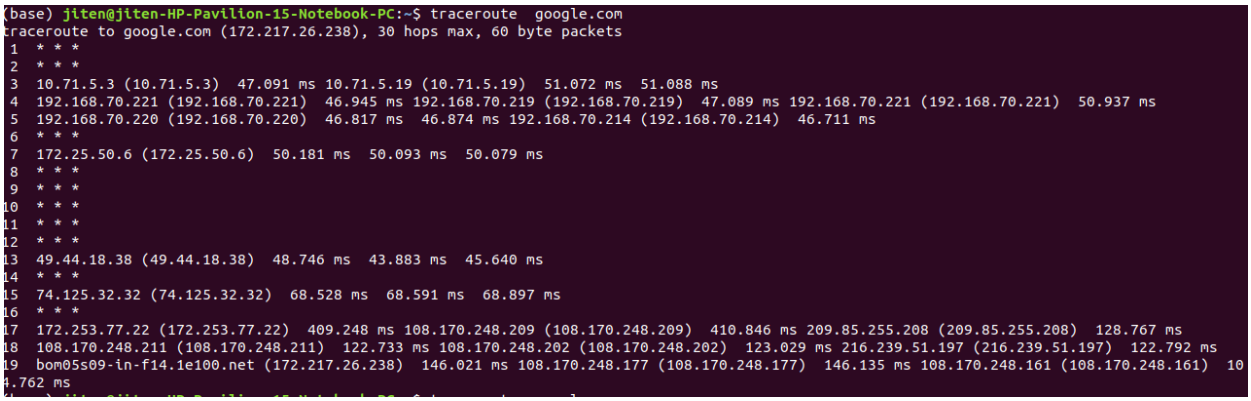
Observations:

Final address of the network is **64.89.144.100** in both cases**.**

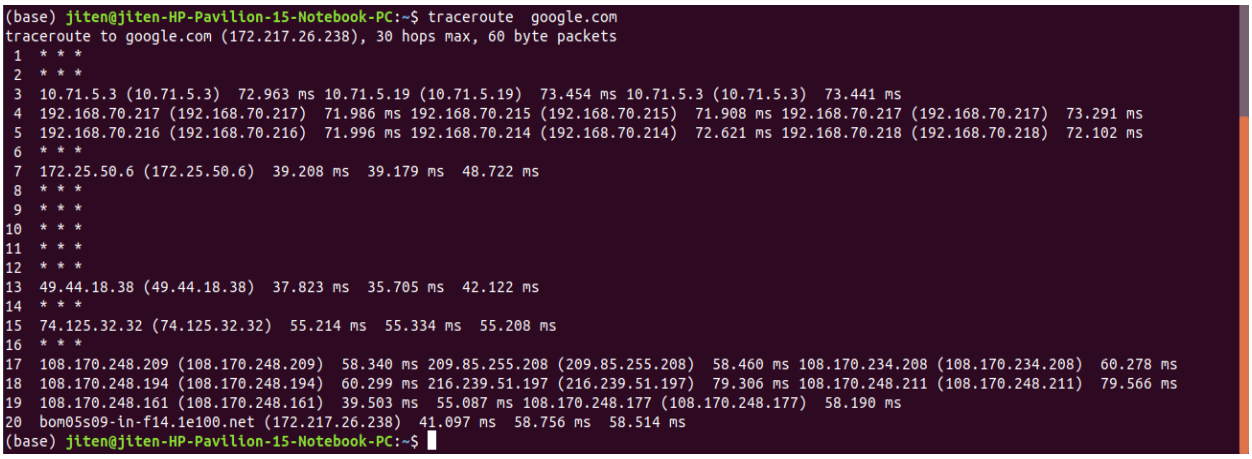
Exercise 3: Two packets sent from the same source to the same destination do not necessarily follow the same path through the net. Experiment with some sources that are fairly far away. Can you find cases where packets sent to the same destination follow different paths? How likely does it seem to be? What about when the packets are sent at very different times? Save some of the outputs from traceroute. (You can copy them from the Terminal window by highlighting and right-clicking, then paste into a text editor.) Come back sometime next week, try the same destinations again, and compare the results with the results from today. Report your observations.

Host : google.com

1 week earlier:



TODAY:



Observations:

Even If both source and destination are same route taken by the packet changes after some time.

QUESTIONS ABOUT PATHS

Now look at the results you gathered and answer the following questions about the paths taken by your packets. Store your answers in a file named `traceroute.txt`.

1. Is any part of the path common for all hosts you tracerouted?

   Yes, the address of the source is same.

2. Is there a relationship between the number of nodes that show up in the traceroute and the location of the host? If so, what is this relationship?

   No, there is no relationship between them. No of hops depends upon physical interface.

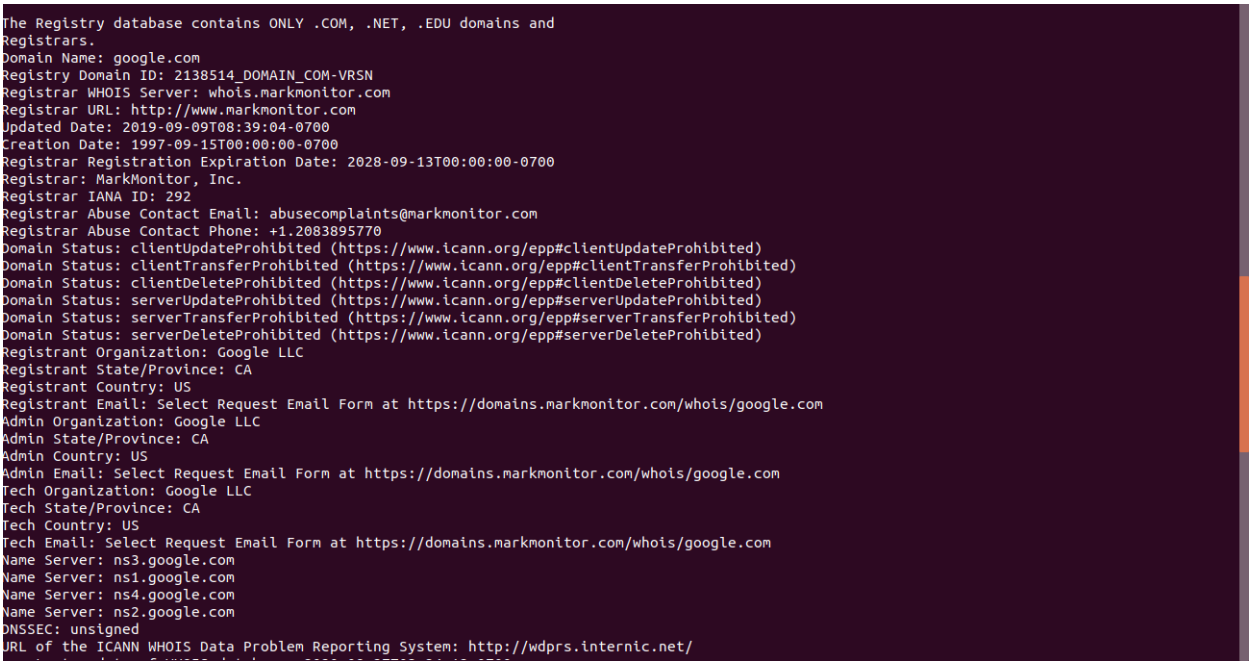
3. Is there a relationship between the number of nodes that show up in the traceroute and latency of the host (from your ping results above)? Does the same relationship hold for all hosts?

   Yes, as number of nodes increases then latency also increases.

Whois — The *whois* command can give detailed information about domain names and IP addresses. If it is not installed on the computers then install it with command sudo apt-get install whois in. *Whois* can tell you what organization owns or is responsible for the name or address and where to contact them. It often includes a list of domain name servers for the organization.

When using *whois* to look up a domain name, use the simple two-part network name, not an individual computer name (for example, *whois spit.ac.in*).

Exercise 4: (Short.) Use *whois* to investigate a well-known web site such as google.com or amazon.com, and write a couple of sentences about what you find out.



Exercise 5: (Should be short.) Because of NAT, the domain name *spit.ac.in* has a different IP address outside of SPIT than it does on campus. Using information in this lab and working on a home computer, find the outside IP address for spit.ac.in. Explain how you did it.

Geolocation — A geolocation service tries to tell, approximately, where a given IP address is located physically. They can't be completely accurate—but they probably get at least the country right most of the time.

This geolocation program is not installed on our computers, but you can access one on the command line using the *curl* command, which can send HTTP requests and display the response. The following command uses *curl* to contact a public web service that will look up an IP address for you: curl ipinfo.io/<IP-address>.

For a specific example: curl ipinfo.io/129.64.99.200

1. nslookup command:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ nslookup spit.ac.in
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   spit.ac.in
Address: 43.252.193.19
```

2. curl ipinfo.io/129.64.99.200

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ curl ipinfo.io/129.64.99.200
{
  "ip": "129.64.99.200",
  "hostname": "websrv-prod.unet.brandeis.edu",
  "city": "Waltham",
  "region": "Massachusetts",
  "country": "US",
  "loc": "42.3765,-71.2356",
  "org": "AS10561 Brandeis University",
  "postal": "02453",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

(As you can see, you get back more than just the location.)

Exercise 6: Find a few IP addresses that are connected to the web server on spit.ac.in right now, and determine where those IP addresses are located. (I'm expecting that there will be several; if not, try again in a few minutes or sometime later.) Find one that is far from Geneva, NY. Explain how you did it.

Conclusion:

I implemented various commands used in computer networking in a Linux based system such as ping, traceroute and etc. This practical also helped me to understand the path taken by a packet from source to destination.