

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 **tracert** 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 receive 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 response 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](http://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
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=1 ttl=61 time=38.2 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=2 ttl=61 time=140 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=3 ttl=61 time=76.9 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=4 ttl=61 time=197 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=5 ttl=61 time=195 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=6 ttl=61 time=53.5 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=7 ttl=61 time=53.3 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=8 ttl=61 time=37.4 ms
10 bytes from 2405:200:1602::312c:8220 (2405:200:1602::312c:8220): icmp_seq=9 ttl=61 time=69.8 ms
10 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
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

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.

### **Propagation 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](http://www.uw.edu), [www.cornell.edu](http://www.cornell.edu), [berkeley.edu](http://berkeley.edu), [www.uchicago.edu](http://www.uchicago.edu), [www.ox.ac.uk](http://www.ox.ac.uk) (England), [www.u-tokyo.ac.jp](http://www.u-tokyo.ac.jp) (Japan).

OTHER HOSTS:

#### 1. [www.uw.edu](http://www.uw.edu)

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 10 www.uw.edu
PING www.washington.edu (128.95.155.135) 56(84) bytes of data:
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=1 ttl=43 time=627 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=2 ttl=43 time=682 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=3 ttl=43 time=912 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=4 ttl=43 time=527 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=5 ttl=43 time=756 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=6 ttl=43 time=576 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=7 ttl=43 time=601 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=8 ttl=43 time=333 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=9 ttl=43 time=385 ms
64 bytes from www2.cac.washington.edu (128.95.155.135): icmp_seq=10 ttl=43 time=879 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9992ms
rtt min/avg/max/mdev = 333.394/628.188/912.510/179.677 ms
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

## 2. [spit.ac.in](http://spit.ac.in)

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ ping -c 5 spit.ac.in
PING spit.ac.in (43.252.193.19) 56(84) bytes of data:

--- spit.ac.in ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4100ms

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

### 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 `nslookup` 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-802eadebcb59: 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
    TX 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:adif: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: telnet <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/>

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ traceroute math.hws.edu
traceroute to math.hws.edu (64.89.144.237), 30 hops max, 60 byte packets
 1 * * *
 2 * * *
 3 10.71.5.19 (10.71.5.19) 69.897 ms 71.603 ms 71.609 ms
 4 192.168.70.217 (192.168.70.217) 69.826 ms 192.168.70.215 (192.168.70.215) 71.465 ms 192.168.70.221 (192.168.70.221) 71.457 ms
 5 192.168.70.218 (192.168.70.218) 69.597 ms 192.168.70.214 (192.168.70.214) 69.540 ms 71.267 ms
 6 * * *
 7 172.25.50.6 (172.25.50.6) 37.008 ms 37.015 ms 40.403 ms
 8 * * *
 9 * * *
10 * * *
11 49.45.4.253 (49.45.4.253) 58.716 ms 103.198.140.174 (103.198.140.174) 66.567 ms 103.198.140.58 (103.198.140.58) 58.555 ms
12 103.198.140.45 (103.198.140.45) 182.129 ms 190.750 ms 191.701 ms
13 103.198.140.56 (103.198.140.56) 181.958 ms 103.198.140.54 (103.198.140.54) 179.439 ms 103.198.140.27 (103.198.140.27) 181.755 ms
14 103.198.140.107 (103.198.140.107) 180.466 ms 181.637 ms 179.230 ms
15 hu0-4-0-1.agr21.lhr01.atlas.cogentco.com (149.14.196.81) 178.870 ms 178.787 ms 103.198.140.45 (103.198.140.45) 194.848 ms
16 hu0-4-0-1.agr21.lhr01.atlas.cogentco.com (149.14.196.81) 194.754 ms 181.045 ms 184.547 ms
17 be3671.ccr51.lhr01.atlas.cogentco.com (130.117.48.137) 176.307 ms be3672.ccr52.lhr01.atlas.cogentco.com (130.117.48.145) 167.462 ms 160.448 ms
18 be3487.ccr41.lon13.atlas.cogentco.com (154.54.60.5) 158.439 ms be2868.ccr21.lon01.atlas.cogentco.com (154.54.57.154) 167.291 ms be3488.ccr42.lon13.atlas.cogentco.com (154.54.60.13) 162.686 ms
19 ae-6.edge7.London1.Level3.net (4.68.62.5) 170.040 ms be2869.ccr22.lon01.atlas.cogentco.com (154.54.57.162) 168.927 ms 167.422 ms
20 ae-6.edge7.London1.Level3.net (4.68.62.5) 160.080 ms ae-228-3604.edge3.London15.Level3.net (4.69.167.102) 157.865 ms ae-226-3602.edge3.London15.Level3.net (4.69.167.94) 154.001 ms
21 ae-116-3502.edge3.London15.Level3.net (4.69.167.78) 161.329 ms ae-227-3603.edge3.London15.Level3.net (4.69.167.98) 153.303 ms 157.139 ms
22 ae-118-3504.edge3.London15.Level3.net (4.69.167.86) 157.036 ms 163.515 ms ae4.ar8.lon15.Level3.net (4.68.111.254) 172.102 ms
23 ae4.ar8.lon15.Level3.net (4.68.111.254) 174.774 ms 176.644 ms 167.245 ms
24 66-195-65-170.static.ctl.one (66.195.65.170) 312.489 ms roc1-ar5-xe-11-0-0-0.us.twtelecom.net (35.248.1.162) 304.548 ms 66-195-65-170.static.ctl.one (66.195.65.170) 304.476 ms
25 66-195-65-170.static.ctl.one (66.195.65.170) 309.748 ms 313.386 ms 302.696 ms
26 nat.hws.edu (64.89.144.100) 302.067 ms 299.649 ms *
27 * * *
28 * * *
29 * * *
30 * * *
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

### 2. [www.hws.edu](http://www.hws.edu)

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ traceroute www.hws.edu
traceroute to www.hws.edu (64.89.145.159), 30 hops max, 60 byte packets
 1 * * *
 2 * * *
 3 10.71.5.19 (10.71.5.19) 50.210 ms 10.71.5.3 (10.71.5.3) 50.332 ms 50.586 ms
 4 192.168.70.217 (192.168.70.217) 50.134 ms 192.168.70.221 (192.168.70.221) 50.421 ms 53.114 ms
 5 192.168.70.216 (192.168.70.216) 50.050 ms 192.168.70.218 (192.168.70.218) 107.379 ms 107.381 ms
 6 * * *
 7 172.25.50.6 (172.25.50.6) 42.905 ms 43.472 ms 42.770 ms
 8 * * *
 9 * * *
10 * * *
11 103.198.140.174 (103.198.140.174) 84.864 ms 49.45.4.253 (49.45.4.253) 81.387 ms 103.198.140.176 (103.198.140.176) 81.326 ms
12 103.198.140.45 (103.198.140.45) 194.572 ms 202.881 ms 204.977 ms
13 103.198.140.56 (103.198.140.56) 194.359 ms 159.185 ms 159.054 ms
14 103.198.140.45 (103.198.140.45) 159.332 ms 103.198.140.107 (103.198.140.107) 159.372 ms 178.588 ms
15 hu0-4-0-1.agr21.lhr01.atlas.cogentco.com (149.14.196.81) 176.580 ms 103.198.140.45 (103.198.140.45) 185.137 ms 178.457 ms
16 hu0-4-0-1.agr21.lhr01.atlas.cogentco.com (149.14.196.81) 173.363 ms 173.012 ms 169.360 ms
17 be3487.ccr41.lon13.atlas.cogentco.com (154.54.60.5) 189.056 ms 180.396 ms be3488.ccr42.lon13.atlas.cogentco.com (154.54.60.13) 178.295 ms
18 be3487.ccr41.lon13.atlas.cogentco.com (154.54.60.5) 166.658 ms be3488.ccr42.lon13.atlas.cogentco.com (154.54.60.13) 303.273 ms be2870.ccr42.lon01.atlas.cogentco.com (154.54.58.174) 303.208 ms
19 be2868.ccr21.lon01.atlas.cogentco.com (154.54.57.154) 303.204 ms * ae-6.edge7.London1.Level3.net (4.68.62.5) 286.537 ms
20 * * ae-227-3603.edge3.London15.Level3.net (4.69.167.98) 283.946 ms
21 ae-226-3602.edge3.London15.Level3.net (4.69.167.94) 277.533 ms ae-227-3603.edge3.London15.Level3.net (4.69.167.98) 181.077 ms ae-118-3504.edge3.London15.Level3.net (4.69.167.86) 161.578 ms
22 ae-227-3603.edge3.London15.Level3.net (4.69.167.98) 187.392 ms ae-115-3501.edge3.London15.Level3.net (4.69.167.74) 180.199 ms ae-117-3503.edge3.London15.Level3.net (4.69.167.82) 172.490 ms
23 ae4.ar8.lon15.Level3.net (4.68.111.254) 148.685 ms 153.742 ms 154.001 ms
24 66-195-65-170.static.ctl.one (66.195.65.170) 307.285 ms 308.395 ms roc1-ar5-xe-11-0-0-us.twtelecom.net (35.248.1.162) 296.569 ms
25 nat.hws.edu (64.89.144.100) 306.552 ms 306.858 ms 66-195-65-170.static.ctl.one (66.195.65.170) 287.808 ms
26 nat.hws.edu (64.89.144.100) 287.108 ms 297.048 ms *
27 * * *
28 * * *
29 * * *
30 * * *
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

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:**

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ traceroute google.com
traceroute to google.com (172.217.26.238), 30 hops max, 60 byte packets
 1 * * *
 2 * * *
 3 10.71.5.3 (10.71.5.3) 47.091 ms 10.71.5.19 (10.71.5.19) 51.072 ms 51.088 ms
 4 192.168.70.221 (192.168.70.221) 46.945 ms 192.168.70.219 (192.168.70.219) 47.089 ms 192.168.70.221 (192.168.70.221) 50.937 ms
 5 192.168.70.220 (192.168.70.220) 46.817 ms 46.874 ms 192.168.70.214 (192.168.70.214) 46.711 ms
 6 * * *
 7 172.25.50.6 (172.25.50.6) 50.181 ms 50.093 ms 50.079 ms
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 49.44.18.38 (49.44.18.38) 48.746 ms 43.883 ms 45.640 ms
14 * * *
15 74.125.32.32 (74.125.32.32) 68.528 ms 68.591 ms 68.897 ms
16 * * *
17 172.253.77.22 (172.253.77.22) 409.248 ms 108.170.248.209 (108.170.248.209) 410.846 ms 209.85.255.208 (209.85.255.208) 128.767 ms
18 108.170.248.211 (108.170.248.211) 122.733 ms 108.170.248.202 (108.170.248.202) 123.029 ms 216.239.51.197 (216.239.51.197) 122.792 ms
19 bom05s09-in-f14.1e100.net (172.217.26.238) 146.021 ms 108.170.248.177 (108.170.248.177) 146.135 ms 108.170.248.161 (108.170.248.161) 104.762 ms
```

## TODAY:

```
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$ traceroute google.com
traceroute to google.com (172.217.26.238), 30 hops max, 60 byte packets
 1 * * *
 2 * * *
 3 10.71.5.3 (10.71.5.3) 72.963 ms 10.71.5.19 (10.71.5.19) 73.454 ms 10.71.5.3 (10.71.5.3) 73.441 ms
 4 192.168.70.217 (192.168.70.217) 71.986 ms 192.168.70.215 (192.168.70.215) 71.908 ms 192.168.70.217 (192.168.70.217) 73.291 ms
 5 192.168.70.216 (192.168.70.216) 71.996 ms 192.168.70.214 (192.168.70.214) 72.621 ms 192.168.70.218 (192.168.70.218) 72.102 ms
 6 * * *
 7 172.25.50.6 (172.25.50.6) 39.208 ms 39.179 ms 48.722 ms
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 49.44.18.38 (49.44.18.38) 37.823 ms 35.705 ms 42.122 ms
14 * * *
15 74.125.32.32 (74.125.32.32) 55.214 ms 55.334 ms 55.208 ms
16 * * *
17 108.170.248.209 (108.170.248.209) 58.340 ms 209.85.255.208 (209.85.255.208) 58.460 ms 108.170.234.208 (108.170.234.208) 60.278 ms
18 108.170.248.194 (108.170.248.194) 60.299 ms 216.239.51.197 (216.239.51.197) 79.306 ms 108.170.248.211 (108.170.248.211) 79.566 ms
19 108.170.248.161 (108.170.248.161) 39.503 ms 55.087 ms 108.170.248.177 (108.170.248.177) 58.190 ms
20 bom05s09-in-f14.1e100.net (172.217.26.238) 41.097 ms 58.756 ms 58.514 ms
(base) jiten@jiten-HP-Pavilion-15-Notebook-PC:~$
```

## 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`. *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.

```
The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.
Domain Name: google.com
Registry Domain ID: 2138514 DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T08:39:04-0700
Creation Date: 1997-09-15T00:00:00-0700
Registrar Registration Expiration Date: 2028-09-13T00:00:00-0700
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895770
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
Domain Status: serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Domain Status: serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
Domain Status: serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
Registrant Organization: Google LLC
Registrant State/Province: CA
Registrant Country: US
Registrant Email: Select Request Email Form at https://domains.markmonitor.com/whois/google.com
Admin Organization: Google LLC
Admin State/Province: CA
Admin Country: US
Admin Email: Select Request Email Form at https://domains.markmonitor.com/whois/google.com
Tech Organization: Google LLC
Tech State/Province: CA
Tech Country: US
Tech Email: Select Request Email Form at https://domains.markmonitor.com/whois/google.com
Name Server: ns3.google.com
Name Server: ns1.google.com
Name Server: ns4.google.com
Name Server: ns2.google.com
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
```

**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.