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

Name: Jahnvi Shah
UID: 2018130047

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.

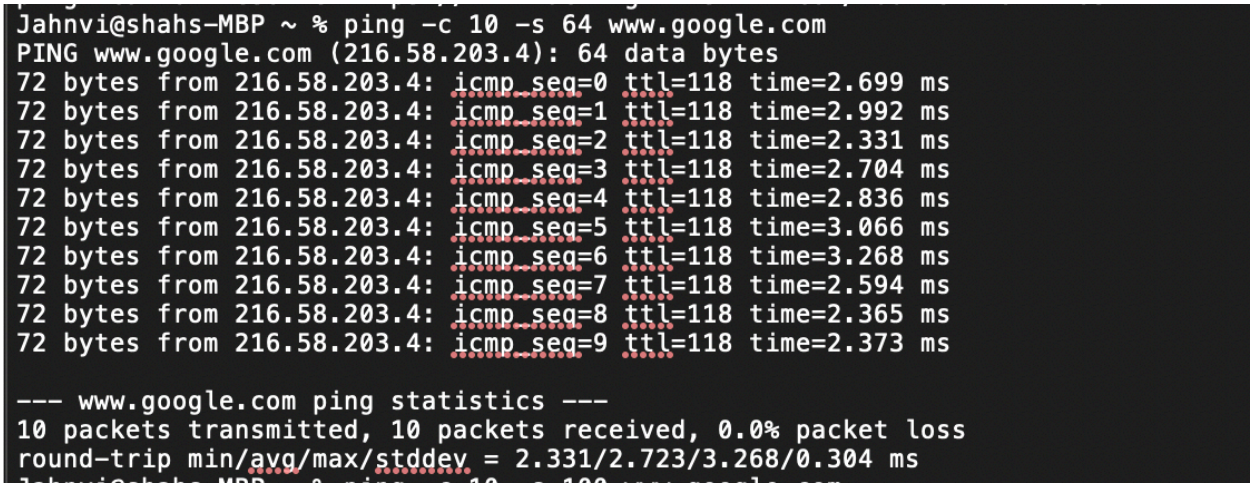
To save the output from `ping` to a file, include a greater than symbol and a file name at the end of the command. For example:

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

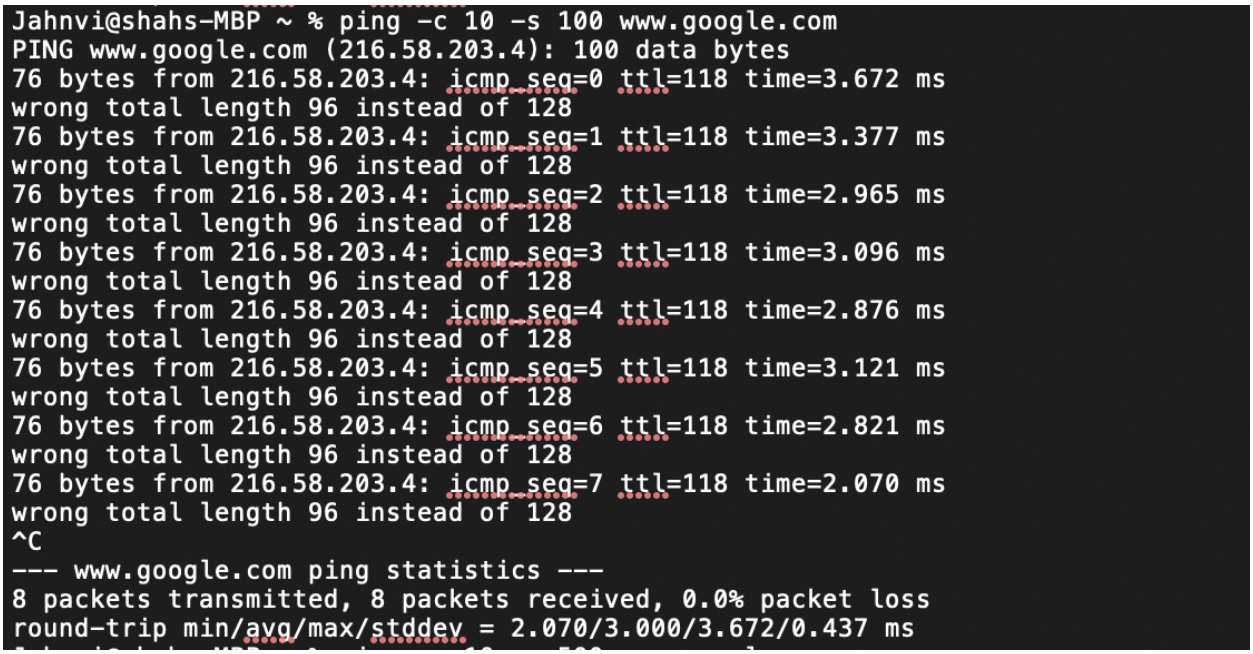
EXPERIMENTS WITH PING

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

- ping -c 10 -s 64 www.google.com



- ping -c 10 -s 100 www.google.com



- ping -c 10 -s 500 www.google.com

```
Jahnvi@shahs-MBP ~ % ping -c 10 -s 500 www.google.com
PING www.google.com (142.250.67.164): 500 data bytes
76 bytes from 142.250.67.164: icmp_seq=0 ttl=118 time=3.202 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=1 ttl=118 time=2.737 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=2 ttl=118 time=2.357 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=3 ttl=118 time=3.108 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=4 ttl=118 time=2.203 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=5 ttl=118 time=2.120 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=6 ttl=118 time=2.657 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=7 ttl=118 time=2.765 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=8 ttl=118 time=2.999 ms
wrong total length 96 instead of 528
76 bytes from 142.250.67.164: icmp_seq=9 ttl=118 time=2.827 ms
wrong total length 96 instead of 528

--- www.google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.120/2.698/3.202/0.351 ms
```

- ping -c 10 -s 64 www.uw.edu

```
Jahnvi@shahs-MBP ~ % ping -c 10 -s 64 www.uw.edu
PING www.washington.edu (128.95.155.135): 64 data bytes
72 bytes from 128.95.155.135: icmp_seq=0 ttl=43 time=306.683 ms
72 bytes from 128.95.155.135: icmp_seq=1 ttl=43 time=398.704 ms
72 bytes from 128.95.155.135: icmp_seq=2 ttl=43 time=452.130 ms
72 bytes from 128.95.155.135: icmp_seq=3 ttl=43 time=369.044 ms
72 bytes from 128.95.155.135: icmp_seq=4 ttl=43 time=290.034 ms
72 bytes from 128.95.155.135: icmp_seq=5 ttl=43 time=517.911 ms
72 bytes from 128.95.155.135: icmp_seq=6 ttl=43 time=245.885 ms
72 bytes from 128.95.155.135: icmp_seq=7 ttl=43 time=360.499 ms
72 bytes from 128.95.155.135: icmp_seq=8 ttl=43 time=283.505 ms
72 bytes from 128.95.155.135: icmp_seq=9 ttl=43 time=511.265 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 245.885/373.566/517.911/90.915 ms
```

- ping -c 10 -s 100 www.uw.edu

```
Jahnvi@shahs-MBP ~ % ping -c 10 -s 100 www.uw.edu
PING www.washington.edu (128.95.155.197): 100 data bytes
108 bytes from 128.95.155.197: icmp_seq=0 ttl=43 time=304.328 ms
108 bytes from 128.95.155.197: icmp_seq=1 ttl=43 time=376.852 ms
108 bytes from 128.95.155.197: icmp_seq=2 ttl=43 time=453.657 ms
108 bytes from 128.95.155.197: icmp_seq=3 ttl=43 time=398.278 ms
108 bytes from 128.95.155.197: icmp_seq=4 ttl=43 time=295.487 ms
108 bytes from 128.95.155.197: icmp_seq=5 ttl=43 time=239.106 ms
108 bytes from 128.95.155.197: icmp_seq=6 ttl=43 time=440.672 ms
108 bytes from 128.95.155.197: icmp_seq=7 ttl=43 time=358.885 ms
108 bytes from 128.95.155.197: icmp_seq=8 ttl=43 time=281.108 ms
108 bytes from 128.95.155.197: icmp_seq=9 ttl=43 time=508.708 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 239.106/365.708/508.708/81.804 ms
```

- ping -c 10 -s 500 www.uw.edu

```
Jahnvi@shahs-MBP ~ % ping -c 10 -s 500 www.uw.edu
PING www.washington.edu (128.95.155.134): 500 data bytes
508 bytes from 128.95.155.134: icmp_seq=0 ttl=47 time=246.142 ms
508 bytes from 128.95.155.134: icmp_seq=1 ttl=47 time=244.400 ms
508 bytes from 128.95.155.134: icmp_seq=2 ttl=47 time=244.049 ms
Request timeout for icmp_seq 3
508 bytes from 128.95.155.134: icmp_seq=4 ttl=47 time=244.997 ms
508 bytes from 128.95.155.134: icmp_seq=5 ttl=47 time=244.023 ms
508 bytes from 128.95.155.134: icmp_seq=6 ttl=47 time=244.346 ms
508 bytes from 128.95.155.134: icmp_seq=7 ttl=47 time=244.670 ms
508 bytes from 128.95.155.134: icmp_seq=8 ttl=47 time=255.949 ms
508 bytes from 128.95.155.134: icmp_seq=9 ttl=47 time=244.168 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 9 packets received, 10.0% packet loss
round-trip min/avg/max/stddev = 244.023/245.860/255.949/3.620 ms
```

- ping -c 10 -s 1000 www.uw.edu

```
Jahnvi@shahs-MBP ~ % ping -c 10 -s 1000 www.uw.edu
PING www.washington.edu (128.95.155.197): 1000 data bytes
1008 bytes from 128.95.155.197: icmp_seq=0 ttl=43 time=539.683 ms
1008 bytes from 128.95.155.197: icmp_seq=1 ttl=43 time=456.193 ms
1008 bytes from 128.95.155.197: icmp_seq=2 ttl=43 time=378.530 ms
1008 bytes from 128.95.155.197: icmp_seq=3 ttl=43 time=299.342 ms
1008 bytes from 128.95.155.197: icmp_seq=4 ttl=43 time=527.679 ms
1008 bytes from 128.95.155.197: icmp_seq=5 ttl=43 time=449.424 ms
1008 bytes from 128.95.155.197: icmp_seq=6 ttl=43 time=368.439 ms
1008 bytes from 128.95.155.197: icmp_seq=7 ttl=43 time=298.709 ms
1008 bytes from 128.95.155.197: icmp_seq=8 ttl=43 time=513.100 ms
1008 bytes from 128.95.155.197: icmp_seq=9 ttl=43 time=432.685 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 298.709/426.378/539.683/83.764 ms
```

- ping -c 10 -s 1400 www.uw.edu

```
Jahnvi@shahs-MBP ~ % ping -c 10 -s 1400 www.uw.edu
PING www.washington.edu (128.95.155.135): 1400 data bytes
1408 bytes from 128.95.155.135: icmp_seq=0 ttl=43 time=307.341 ms
1408 bytes from 128.95.155.135: icmp_seq=1 ttl=43 time=535.336 ms
1408 bytes from 128.95.155.135: icmp_seq=2 ttl=43 time=452.912 ms
1408 bytes from 128.95.155.135: icmp_seq=3 ttl=43 time=373.482 ms
1408 bytes from 128.95.155.135: icmp_seq=4 ttl=43 time=294.203 ms
1408 bytes from 128.95.155.135: icmp_seq=5 ttl=43 time=523.453 ms
1408 bytes from 128.95.155.135: icmp_seq=6 ttl=43 time=441.472 ms
1408 bytes from 128.95.155.135: icmp_seq=7 ttl=43 time=357.691 ms
1408 bytes from 128.95.155.135: icmp_seq=8 ttl=43 time=278.281 ms
1408 bytes from 128.95.155.135: icmp_seq=9 ttl=43 time=505.862 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 278.281/407.003/535.336/92.671 ms
```

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?

Answer : From the above figures, we can conclude that RTT is dependent on the host on which the 'ping' command is used. Transmission delay is the time taken to put a packet onto a link or simply, the time required to put data bits on the wire/communication medium. It depends on the size of the packet and the bandwidth of the network. Since the hosts are the only parameters changed, there is no transmission delay in the two cases. Propagation delay is the time taken by the first bit to travel from sender to receiver end of the link or simply the time required for bits to reach the destination from the start point. Factors on which propagation delay depends are distance and propagation speed(difference of distance from India between the 2 is around 5000km). So, there exists a propagation delay in the two cases. Queueing delay is the time difference between when the packet arrived at its destination and when the packet data was processed or executed. It depends on the number of packets, size of the packet and bandwidth of the network. Since all the parameters are non-varying in both cases, there is hardly any queueing delay.

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

Answer : From the above images, we can say that the Round Trip Time is impacted due to the difference in the size of the packets. This is because of the Transmission delay and the Queueing delay which depend on the size of the packets.RTT increases with increase in packet size. There would be increased latency for increased packet size due to transmission delay and propagation delay.

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).

From the images shown above, the following observations can be made :

1. The length a signal has to travel correlates with the time taken for a request to reach a server and a response to reach a browser.
2. The medium used to route a signal (e.g., copper wire, fiber optic cables) can impact how quickly a request is received by a server and routed back to a user.
3. Intermediate routers or servers take time to process a signal, increasing RTT. The more hops a signal has to travel through, the higher the RTT.
4. RTT typically increases when a network is congested with high levels of traffic. Conversely, low traffic times can result in decreased RTT.
5. The time taken for a target server to respond to a request depends on its processing capacity, the number of requests being handled and the nature of the request (i.e., how much server-side work is required). A longer server response time increases RTT.

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>

```
Jahnvi@shahs-MBP ~ % nslookup www.uw.edu
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.uw.edu      canonical name = www.washington.edu.
Name:   www.washington.edu
Address: 128.95.155.135
Name:   www.washington.edu
Address: 128.95.155.198
Name:   www.washington.edu
Address: 128.95.155.134
```

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!!!.)

```
Jahnvi@shahs-MBP ~ % ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 38:f9:d3:62:58:27
        inet6 fe80::c66:97fa:ad98:1654%en0 prefixlen 64 secured scopeid 0x4
        inet 192.168.1.26 netmask 0xffffff00 broadcast 192.168.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:b6:7a:c0:a8:01
        media: autoselect <full-duplex>
        status: inactive
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:b6:7a:c0:a8:00
        media: autoselect <full-duplex>
        status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=63<RXCSUM,TXCSUM,TSO4,TSO6>
        ether 82:b6:7a:c0:a8:01
        Configuration:
                id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
                maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
                root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
                ipfilter disabled flags 0x0
        member: en1 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 5 priority 0 path cost 0
        member: en2 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 6 priority 0 path cost 0
        nd6 options=201<PERFORMNUD,DAD>
        media: <unknown type>
        status: inactive
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
        options=400<CHANNEL_IO>
        ether 0a:f9:d3:62:58:27
        media: autoselect
        status: inactive
awdl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
        options=400<CHANNEL_IO>
        ether 3a:f5:e3:51:c5:de
        inet6 fe80::38f5:e3ff:fe51:c5de%awdl0 prefixlen 64 scopeid 0x9
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

```
llw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 3a:f5:e3:51:c5:de
        inet6 fe80::38f5:e3ff:fe51:c5de%llw0 prefixlen 64 scopeid 0xa
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
        inet6 fe80::5cc4:fc4e:689f:15c4%utun0 prefixlen 64 scopeid 0xb
        nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
        inet6 fe80::3aa2:4572:24df:a767%utun1 prefixlen 64 scopeid 0xc
        nd6 options=201<PERFORMNUD,DAD>
Jahnvi@shahs-MBP ~ % ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 38:f9:d3:62:58:27
        inet6 fe80::c66:97fa:ad98:1654%en0 prefixlen 64 secured scopeid 0x4
        inet 192.168.1.26 netmask 0xffffff00 broadcast 192.168.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:b6:7a:c0:a8:01
        media: autoselect <full-duplex>
        status: inactive
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:b6:7a:c0:a8:00
        media: autoselect <full-duplex>
        status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=63<RXCSUM,TXCSUM,TSO4,TSO6>
        ether 82:b6:7a:c0:a8:01
        Configuration:
                id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
                maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
                root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
                ipfilter disabled flags 0x0
        member: en1 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 5 priority 0 path cost 0
        member: en2 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 6 priority 0 path cost 0
        nd6 options=201<PERFORMNUD,DAD>
        media: <unknown type>
        status: inactive
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
        options=400<CHANNEL_IO>
        ether 0a:f9:d3:62:58:27
        media: autoselect
        status: inactive
awdl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
        options=400<CHANNEL_IO>
        ether 3a:f5:e3:51:c5:de
        inet6 fe80::38f5:e3ff:fe51:c5de%awdl0 prefixlen 64 scopeid 0x9
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
llw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 3a:f5:e3:51:c5:de
        inet6 fe80::38f5:e3ff:fe51:c5de%llw0 prefixlen 64 scopeid 0xa
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
        inet6 fe80::5cc4:fc4e:689f:15c4%utun0 prefixlen 64 scopeid 0xb
        nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
        inet6 fe80::3aa2:4572:24df:a767%utun1 prefixlen 64 scopeid 0xc
        nd6 options=201<PERFORMNUD,DAD>
```

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.)

```
Jahnvi@shahs-MBP ~ % netstat -p tcp -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  shahs-mbp.60920        17.188.136.40.5223     ESTABLISHED
tcp4       0      0  shahs-mbp.60918        74.125.24.108.imaps    ESTABLISHED
tcp4       0      0  shahs-mbp.60917        25.224.186.35.bc.https ESTABLISHED
tcp4       0    122  shahs-mbp.60915        17.188.145.38.5223     FIN_WAIT_1
tcp4       0     47  shahs-mbp.60914        17.56.8.133.imaps      FIN_WAIT_1
tcp4       0    122  shahs-mbp.60872        17.188.145.174.5223    FIN_WAIT_1
tcp4       0      0  shahs-mbp.60758        25.224.186.35.bc.https ESTABLISHED
tcp4       0      0  shahs-mbp.59891        whatsapp-cdn-shv.https ESTABLISHED
tcp4       0      0  shahs-mbp.59865        47.224.186.35.bc.https ESTABLISHED
tcp4       0      0  shahs-mbp.59845        238.240.190.35.b.http  ESTABLISHED
tcp4       0      0  *.59843                *.*                    LISTEN
tcp6       0      0  *.59723                *.*                    LISTEN
tcp4       0      0  *.59723                *.*                    LISTEN
tcp4       0      0  *.57621                *.*                    LISTEN
tcp6       0      0  shahs-macbook-pr.black fe80::1d8d:ddd3:.2940  ESTABLISHED
tcp6       0      0  shahs-macbook-pr.1024  fe80::1d8d:ddd3:.1024  ESTABLISHED
tcp4       0      0  localhost.mysql        *.*                    LISTEN
tcp4       0      0  localhost.33060        *.*                    LISTEN
```

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

- has been removed from macOS from Mojave onwards.
-

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:

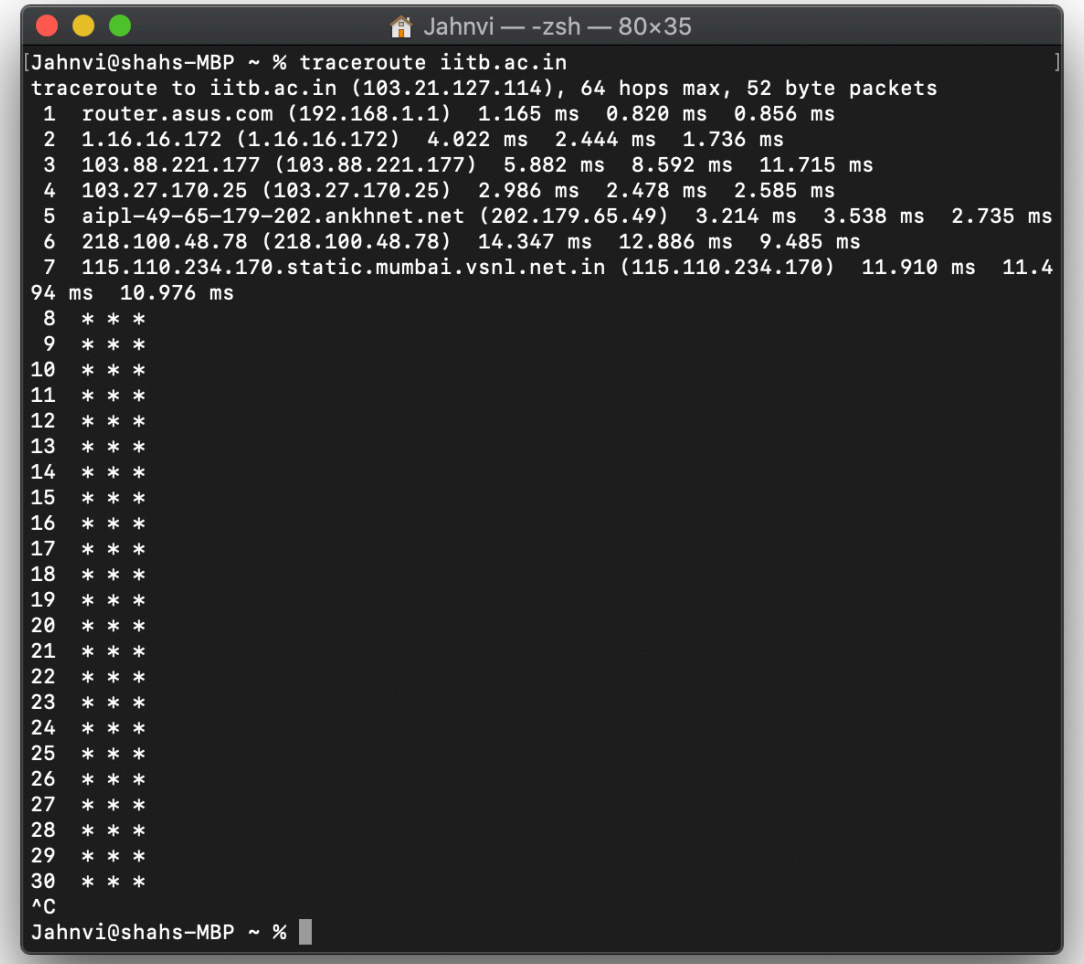
```
tracert <hostname>
```

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

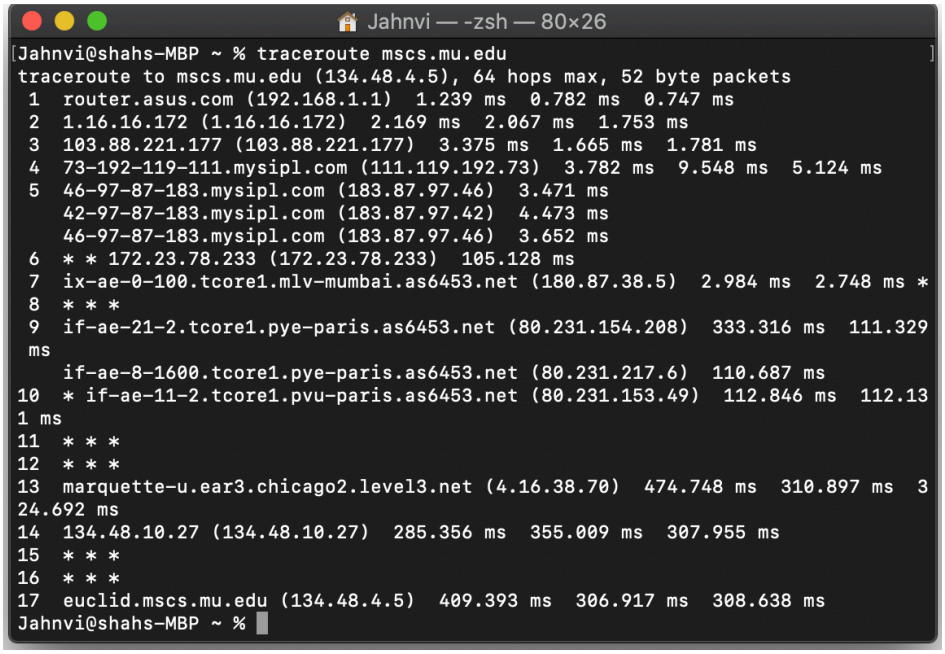
1.2.1 Experiments with Traceroute

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

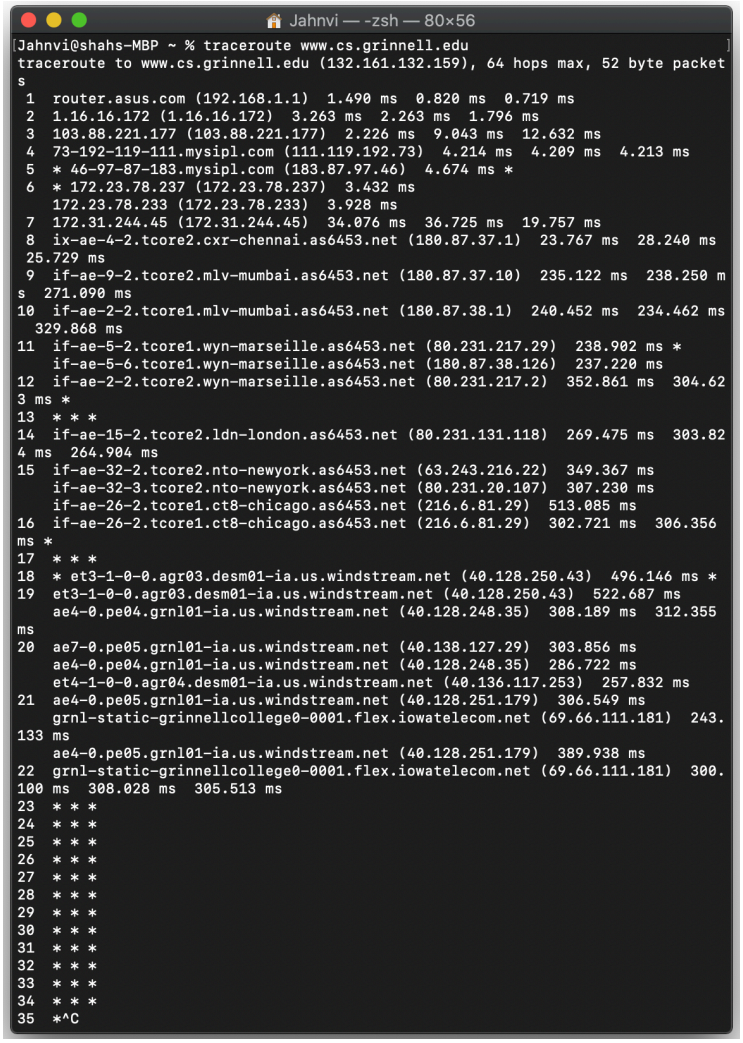
1. ee.iitb.ac.in



2. mscs.mu.edu



3. www.cs.grinnell.edu



```
Jahnvi@shahs-MBP ~ % traceroute csail.mit.edu
traceroute to csail.mit.edu (128.30.2.109), 64 hops max, 52 byte packets
 1  router.asus.com (192.168.1.1)  1.423 ms  0.923 ms  0.815 ms
 2  1.16.16.172 (1.16.16.172)  2.360 ms  1.889 ms  2.339 ms
 3  103.88.221.177 (103.88.221.177)  7.923 ms  10.070 ms  2.052 ms
 4  73-192-119-111.mysipl.com (111.119.192.73)  78.826 ms  4.279 ms  32.441 ms
 5  * 38-97-87-183.mysipl.com (183.87.97.38)  4.468 ms *
 6  172.23.78.237 (172.23.78.237)  4.371 ms
    172.23.78.233 (172.23.78.233)  2.836 ms
    172.23.78.237 (172.23.78.237)  3.151 ms
 7  ix-ae-0-100.tcore1.mlv-mumbai.as6453.net (180.87.38.5)  3.478 ms  4.443 ms
9.578 ms
 8  if-ae-29-8.tcore1.wyn-marseille.as6453.net (80.231.217.110)  252.253 ms * *
 9  if-ae-2-2.tcore2.wyn-marseille.as6453.net (80.231.217.2)  454.510 ms  311.75
2 ms  292.393 ms
10  if-ae-9-2.tcore2.l78-london.as6453.net (80.231.200.14)  304.357 ms *  201.12
3 ms
11  if-ae-15-2.tcore2.ldn-london.as6453.net (80.231.131.118)  310.808 ms
    if-ae-8-49.tcore2.nto-newyork.as6453.net (216.6.81.34)  306.567 ms  303.127
ms
12  if-ae-2-2.tcore1.n0v-newyork.as6453.net (216.6.90.21)  309.024 ms  204.512 m
s
    if-ae-12-2.tcore1.n75-newyork.as6453.net (66.110.96.5)  311.419 ms
13  66.110.96.150 (66.110.96.150)  199.978 ms
    66.110.96.142 (66.110.96.142)  309.496 ms
    66.110.96.130 (66.110.96.130)  486.520 ms
14  66.110.96.142 (66.110.96.142)  360.666 ms
    66.110.96.138 (66.110.96.138)  404.560 ms
    66.110.96.146 (66.110.96.146)  260.808 ms
15  be-1302-cs03.newyork.ny.ibone.comcast.net (96.110.38.41)  497.416 ms  473.94
5 ms  301.919 ms
16  96.110.42.10 (96.110.42.10)  298.079 ms
    be-1302-cs03.newyork.ny.ibone.comcast.net (96.110.38.41)  502.407 ms  305.84
9 ms
17  ae0-0-eg-bstpmall74w.boston.ma.boston.comcast.net (68.86.238.34)  307.435 ms
  309.731 ms  305.588 ms
18  96.110.42.10 (96.110.42.10)  306.800 ms
    50-201-57-174-static.hfc.comcastbusiness.net (50.201.57.174)  305.417 ms  30
7.074 ms
19  dmz-rtr-1-external-rtr-3.mit.edu (18.0.161.13)  306.727 ms
    50-201-57-174-static.hfc.comcastbusiness.net (50.201.57.174)  295.666 ms
    ae0-0-eg-bstpmall74w.boston.ma.boston.comcast.net (68.86.238.34)  308.549 ms
20  50-201-57-174-static.hfc.comcastbusiness.net (50.201.57.174)  306.366 ms  30
4.838 ms
    dmz-rtr-1-external-rtr-3.mit.edu (18.0.161.13)  307.260 ms
21  dmz-rtr-1-external-rtr-3.mit.edu (18.0.161.13)  306.914 ms
    dmz-rtr-2-dmz-rtr-1-2.mit.edu (18.0.162.6)  197.753 ms
    mitnet.core-1-ext.csail.mit.edu (18.4.7.65)  410.103 ms
22  * dmz-rtr-2-dmz-rtr-1-2.mit.edu (18.0.162.6)  513.090 ms
    dmz-rtr-2-dmz-rtr-1-1.mit.edu (18.0.161.6)  304.057 ms
23  * * bdr.core-1.csail.mit.edu (128.30.0.246)  438.449 ms
24  * * *
25  bdr.core-1.csail.mit.edu (128.30.0.246)  209.267 ms  458.013 ms *
26  * * *
27  * * *
28  * * *
29  * * *
```

4. csail.mit.edu.
5. cs.stanford.edu

```
Jahnvi@shahs-MBP ~ % traceroute cs.stanford.edu
traceroute to cs.stanford.edu (171.64.64.64), 64 hops max, 52 byte packets
 1  router.asus.com (192.168.1.1)  1.161 ms  0.908 ms  0.837 ms
 2  1.16.16.172 (1.16.16.172)  2.263 ms  1.980 ms  1.691 ms
 3  103.88.221.177 (103.88.221.177)  1.954 ms  2.657 ms  1.456 ms
 4  73-192-119-111.mysipl.com (111.119.192.73)  3.792 ms  3.543 ms  4.450 ms
 5  38-97-87-183.mysipl.com (183.87.97.38)  3.271 ms  3.489 ms
    46-97-87-183.mysipl.com (183.87.97.46)  3.543 ms
 6  172.23.78.237 (172.23.78.237)  3.096 ms
    172.23.78.233 (172.23.78.233)  3.148 ms
    172.23.78.237 (172.23.78.237)  3.177 ms
 7  172.31.244.45 (172.31.244.45)  24.553 ms  20.323 ms  25.093 ms
 8  ix-ae-4-2.tcore2.cxr-chennai.as6453.net (180.87.37.1)  28.980 ms  19.998 ms
 19.825 ms
 9  if-ae-10-4.tcore2.svw-singapore.as6453.net (180.87.67.16)  301.003 ms  303.3
67 ms  310.745 ms
10  if-ae-7-2.tcore2.lvw-losangeles.as6453.net (180.87.15.26)  305.167 ms  292.7
14 ms  307.024 ms
11  if-ae-2-2.tcore1.lvw-losangeles.as6453.net (66.110.59.1)  307.465 ms  304.31
5 ms  307.124 ms
12  las-b24-link.telia.net (80.239.128.214)  307.173 ms  304.098 ms  307.322 ms
13  * palo-b24-link.telia.net (62.115.119.90)  525.038 ms  304.723 ms
14  palo-b1-link.telia.net (62.115.122.169)  308.423 ms  273.789 ms  305.268 ms
15  hurricane-ic-308019-palo-b1.c.telia.net (80.239.167.174)  419.685 ms  499.11
8 ms  306.630 ms
16  stanford-university.100gigabitethernet5-1.core1.pao1.he.net (184.105.177.238
)  306.275 ms  249.713 ms  361.725 ms
17  csee-west-rtr-vl3.sunet (171.66.255.140)  309.290 ms  304.811 ms  257.284 ms
18  cs.stanford.edu (171.64.64.64)  354.245 ms  304.288 ms  308.950 ms
```

6. cs.manchester.ac.uk

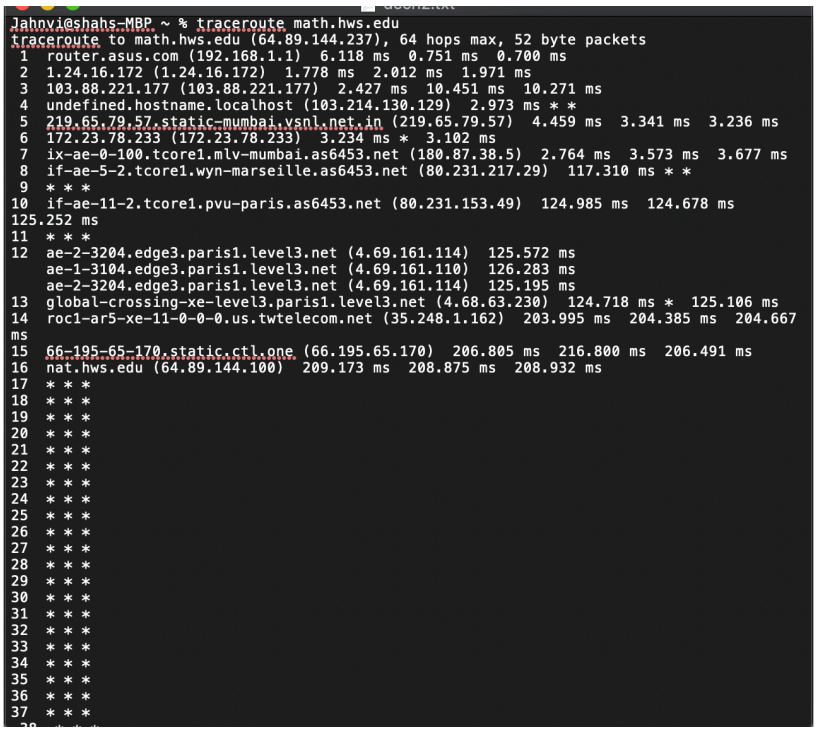
```
[Jahnvi@shahs-MBP ~ % traceroute cs.manchester.ac.uk
traceroute to cs.manchester.ac.uk (130.88.101.49), 64 hops max, 52 byte packets
 1  router.asus.com (192.168.1.1)  1.412 ms  0.829 ms  0.771 ms
 2  1.16.16.172 (1.16.16.172)  1.930 ms  1.877 ms  1.944 ms
 3  103.88.221.177 (103.88.221.177)  2.052 ms  2.103 ms  1.858 ms
 4  73-192-119-111.mysipl.com (111.119.192.73)  2.977 ms  3.241 ms  4.134 ms
 5  38-97-87-183.mysipl.com (183.87.97.38)  3.255 ms
    46-97-87-183.mysipl.com (183.87.97.46)  2.697 ms  3.838 ms
 6  172.23.78.237 (172.23.78.237)  4.825 ms *
    172.23.78.233 (172.23.78.233)  4.417 ms
 7  ix-ae-0-100.tcore1.mlv-mumbai.as6453.net (180.87.38.5)  3.061 ms  3.167 ms
3.749 ms
 8  if-ae-5-2.tcore1.wyn-marseille.as6453.net (80.231.217.29)  111.124 ms
    if-ae-5-6.tcore1.wyn-marseille.as6453.net (180.87.38.126)  124.003 ms
    if-ae-5-2.tcore1.wyn-marseille.as6453.net (80.231.217.29)  110.524 ms
 9  if-ae-8-1600.tcore1.pye-paris.as6453.net (80.231.217.6)  110.631 ms
    if-ae-21-2.tcore1.pye-paris.as6453.net (80.231.154.208)  110.755 ms  110.883
 ms
10  if-ae-11-2.tcore1.pvu-paris.as6453.net (80.231.153.49)  110.912 ms  109.935
ms  110.382 ms
11  * * *
12  * ae-1-9.bear1.manchesteruk1.level3.net (4.69.167.38)  335.062 ms  303.054 m
s
13  janet.bear1.manchester1.level3.net (212.187.174.238)  307.194 ms  333.279 ms
  291.206 ms
14  ae22.manckh-sbr2.ja.net (146.97.35.189)  308.806 ms  609.337 ms  309.509 ms
15  ae23.mancrh-rbr1.ja.net (146.97.38.42)  614.767 ms  307.011 ms  307.675 ms
16  * universityofmanchester.ja.net (146.97.169.2)  221.865 ms *
17  130.88.249.194 (130.88.249.194)  132.239 ms  304.273 ms  130.869 ms
18  * * *
19  * * *
20  eps.its.man.ac.uk (130.88.101.49)  132.427 ms  307.268 ms  307.184 ms
Jahnvi@shahs-MBP ~ %
```

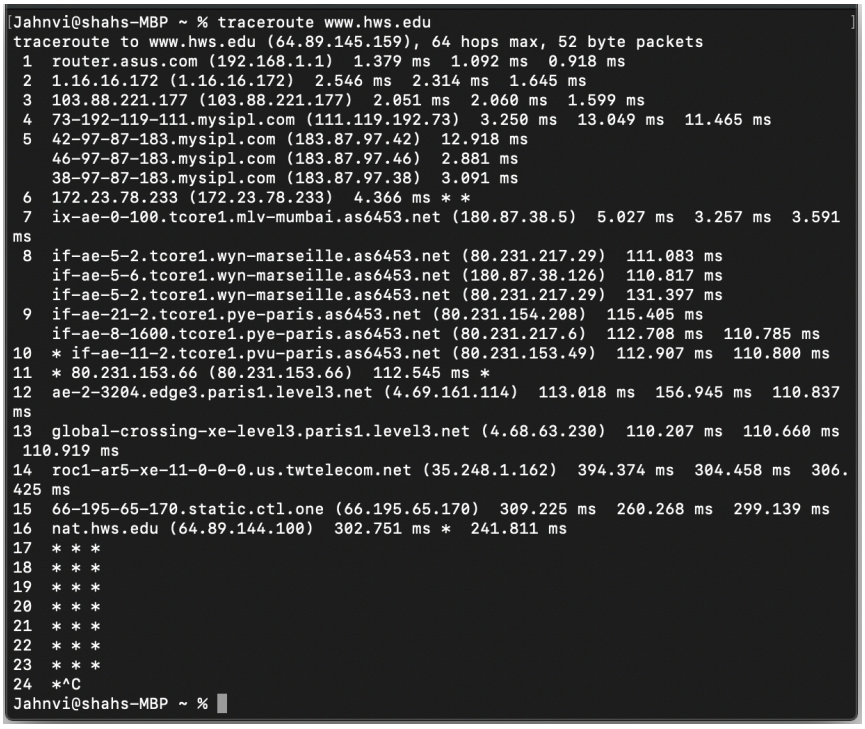
Store the output of each traceroute command in a separate file named `traceroute_HOSTNAME.log`, replacing `HOSTNAME` with the hostname for end-host you pinged (e.g., `traceroute_ee.iitb.ac.in.log`).

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.

From the above images, the first row shows that the process of route tracing has started as the last column shows the Default Gateway of the user. The next three rows in both the cases are similar as the route is being traced starting from the ISP (Internet service provider) of the user. The next few rows, after which the tracing reaches the common IP address of 66.195.65.170 and then nat.hws.edu [64.89.144.100], clearly show that the route is completely different after crossing the ISP for both the cases. A domain name might have multiple IP addresses associated. If this is the case, multiple traces may access two or more IP addresses. This will yield trace paths that differ from one another, even if the origin and destinations are the same.
Domains may also use multiple servers for its subdomains. Tracing the path to the base domain might result in a completely different path when tracing to the subdomain. A URL with the www prefix is technically a subdomain, so it's possible that traces to example.com and www.example.com follow two very different paths. Many domains use separate hosting for

email. If you try to trace the domain, you'll get data for the website server, not the email server. This concept is popularly known as Caveats [1].





<u>Exercise 3:</u> 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.

```
Jahnvi@shahs-MBP ~ % traceroute www.google.com
traceroute to www.google.com (142.250.67.164), 64 hops max, 52 byte packets
 1  router.asus.com (192.168.1.1)  1.328 ms  1.085 ms  0.754 ms
 2  1.24.16.172 (1.24.16.172)  3.770 ms  2.674 ms  2.251 ms
 3  103.88.221.177 (103.88.221.177)  2.407 ms  2.379 ms  1.815 ms
 4  72.14.219.48 (72.14.219.48)  2.603 ms  3.547 ms  2.772 ms
 5  * * *
 6  216.239.47.148 (216.239.47.148)  5.471 ms
    108.170.248.209 (108.170.248.209)  4.388 ms
    74.125.251.132 (74.125.251.132)  5.458 ms
 7  108.170.248.210 (108.170.248.210)  5.596 ms
    108.170.248.218 (108.170.248.218)  4.281 ms
    142.250.227.73 (142.250.227.73)  3.347 ms
 8  bom12s07-in-f4.1e100.net (142.250.67.164)  2.767 ms
    108.170.248.161 (108.170.248.161)  3.106 ms
    bom12s07-in-f4.1e100.net (142.250.67.164)  2.521 ms
```

```
Jahnvi@shahs-MBP ~ %  traceroute www.google.com
traceroute to www.google.com (172.217.174.228), 64 hops max, 52 byte packets
 1  router.asus.com (192.168.1.1)  1.432 ms  1.035 ms  0.886 ms
 2  1.16.16.172 (1.16.16.172)  2.615 ms  1.855 ms  1.570 ms
 3  103.88.221.177 (103.88.221.177)  2.579 ms  2.388 ms  2.337 ms
 4  72.14.219.48 (72.14.219.48)  4.755 ms  2.781 ms  2.816 ms
 5  10.252.244.190 (10.252.244.190)  3.132 ms  3.038 ms  3.728 ms
 6  209.85.244.156 (209.85.244.156)  3.165 ms
    108.170.248.209 (108.170.248.209)  4.028 ms
    172.253.77.20 (172.253.77.20)  3.092 ms
 7  108.170.248.194 (108.170.248.194)  6.556 ms
    216.239.50.167 (216.239.50.167)  7.614 ms  4.509 ms
 8  bom12s03-in-f4.1e100.net (172.217.174.228)  2.491 ms  2.961 ms
    108.170.248.161 (108.170.248.161)  3.385 ms
```

It is observed that when packets are sent to the same destination at very different times, it doesn't necessarily follow the same path.

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 tracerouting follows a particular path from the user's IP address through the IP addresses of the ISP and then the path really depends on which access point is ready to respond and which access points or routers have firewalls configured for blocking the requests and accordingly, the destination can be reached through different paths at different times.

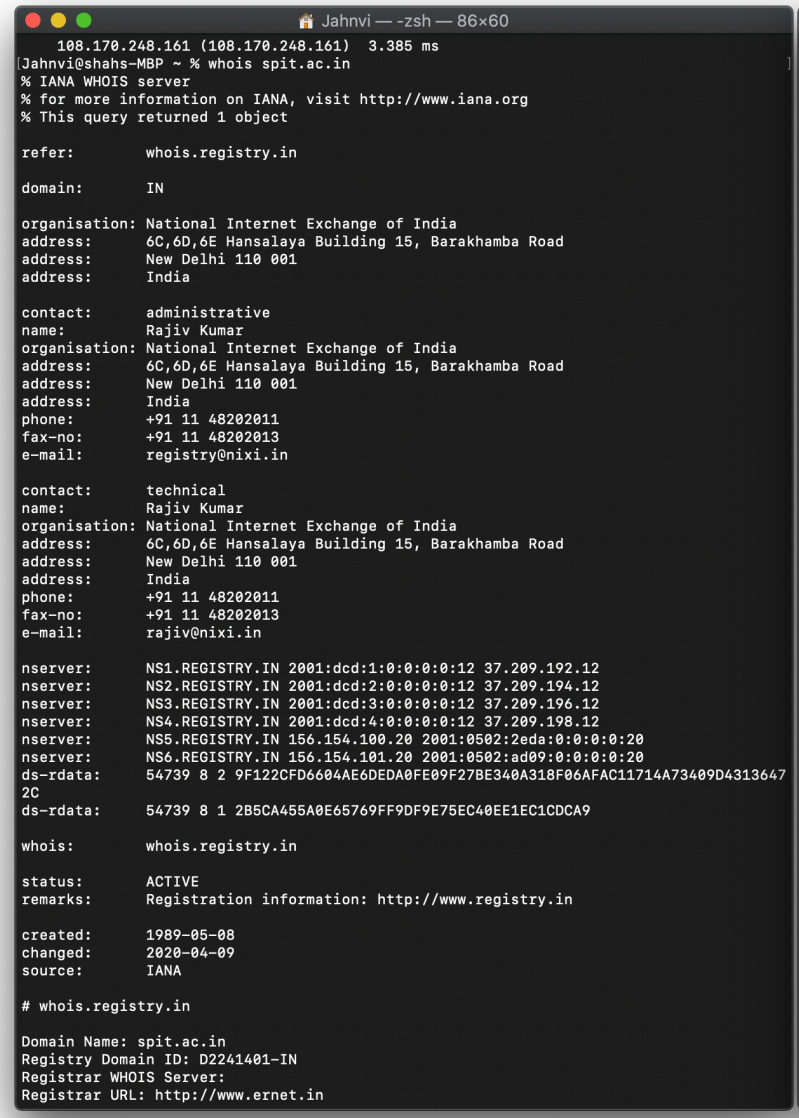
2. Is there a relationship between the number of nodes that show up in the traceroute and the location of the host? If so, what is this relationship?

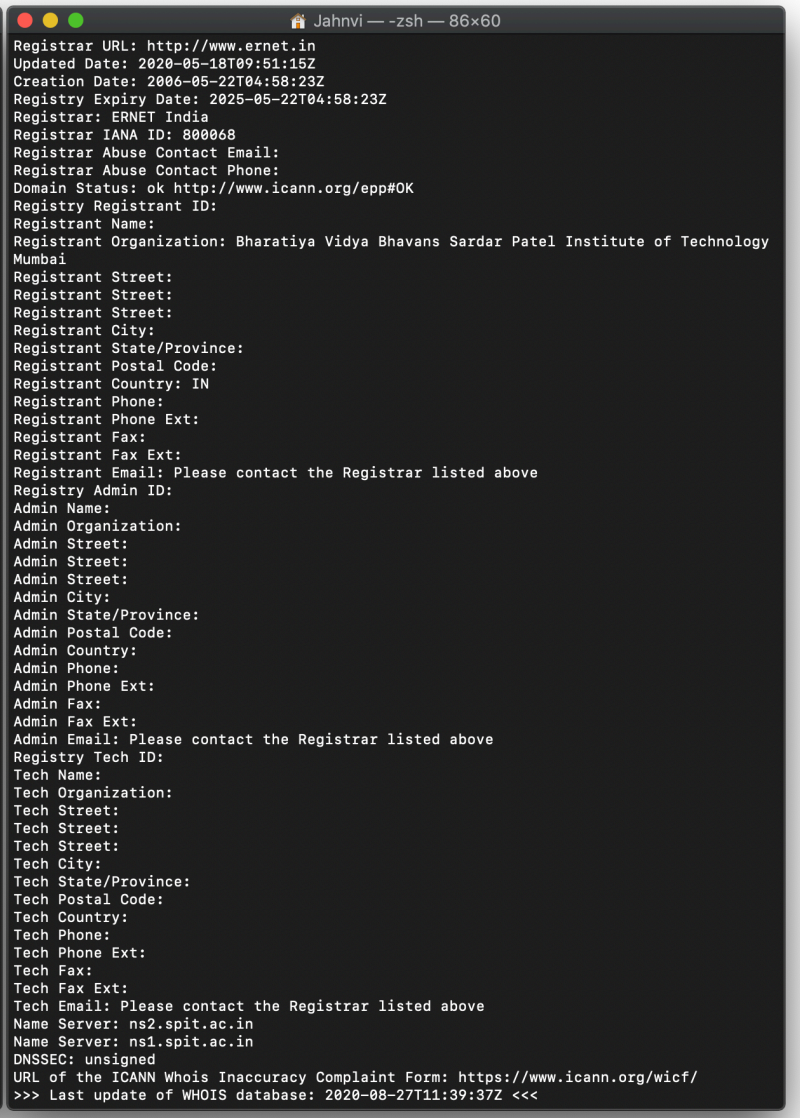
Yes, the number of nodes(number of hops subtract 1) is directly proportional to the distance between the source and destination.

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?

   There is a direct relationship between the number of nodes and the latency of the host. It also depends on the packet size. The amount of latency is largely dependent on how far the visitor is from the server location and how many nodes the signal has to travel through.

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.

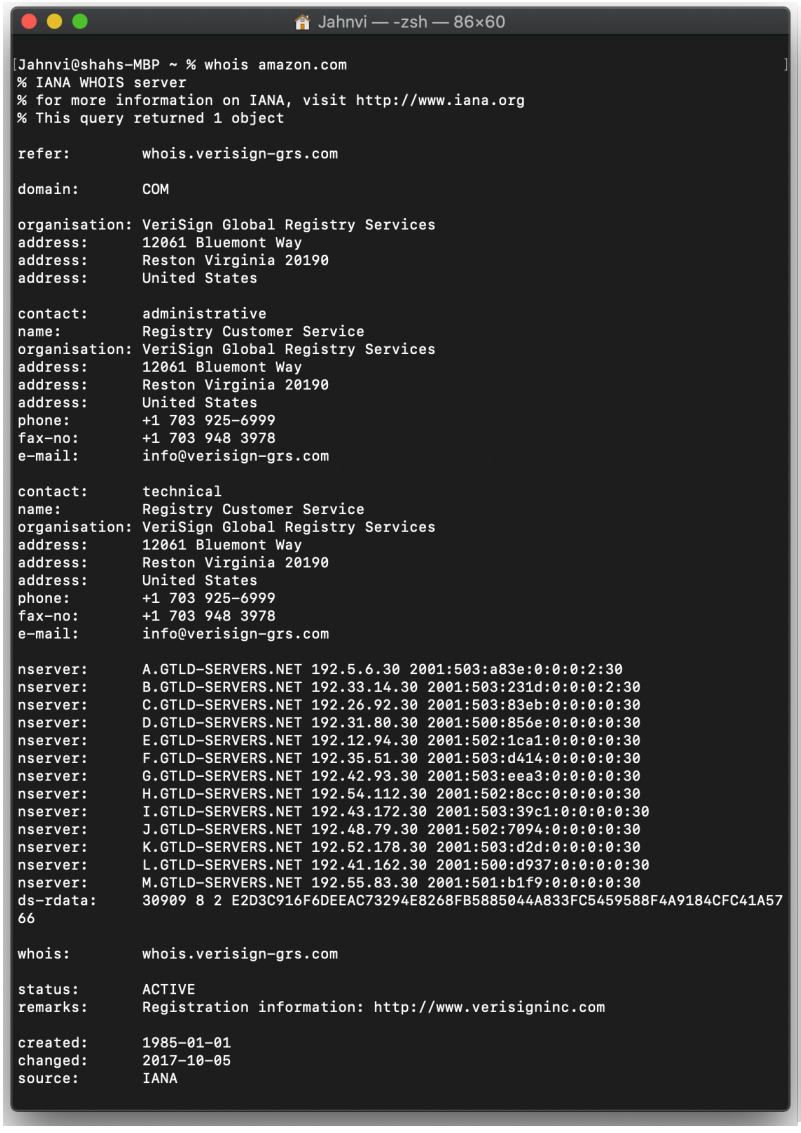


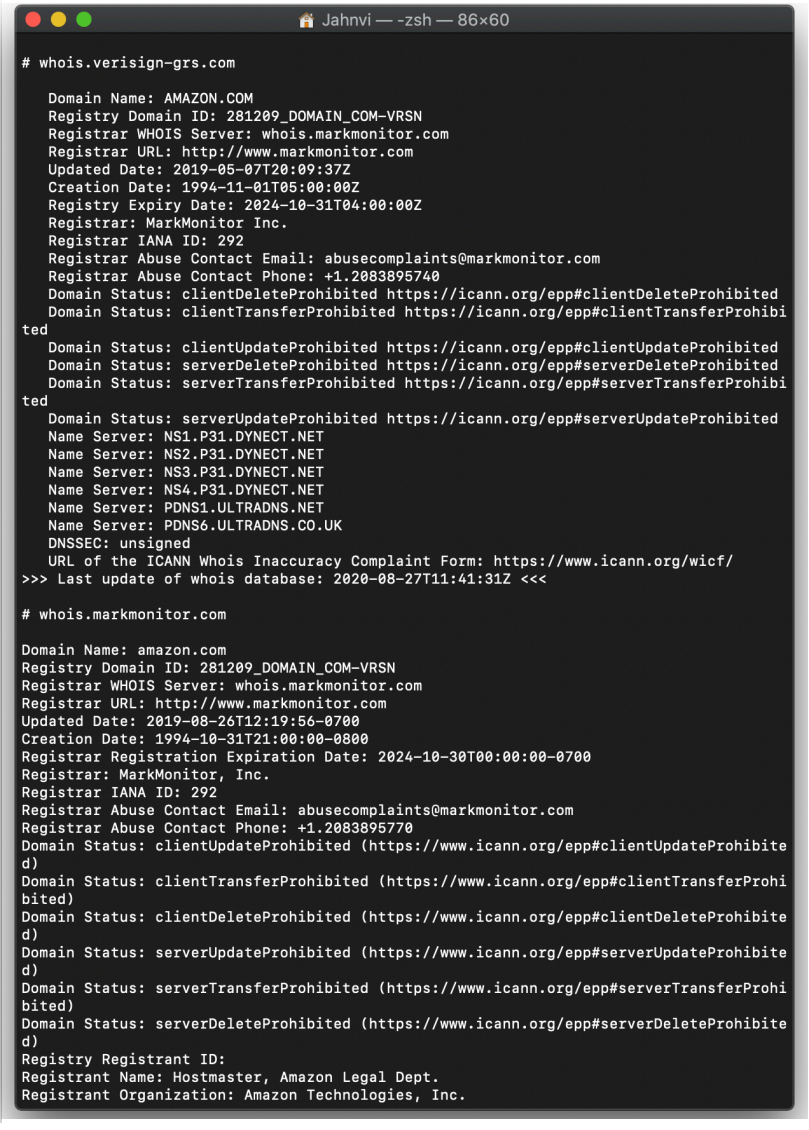


When using *whois* to look up a domain name, use the simple two-part network name, not an individual computer name (for example, *whois* spit.ac.in).

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.

The whois command gives information about the domain name, the Registry Domain ID and some other details such as the details of the Registrar and the Registrant. For example, in case of google.com (domain name), the Registrant Organization is Google LLC, the Registrant State/Province is California and the Registrant Country is the United States. It also provides the domain expiry date.





```
Registrant Organization: Amazon Technologies, Inc.
Registrant Street: P.O. Box 8102
Registrant City: Reno
Registrant State/Province: NV
Registrant Postal Code: 89507
Registrant Country: US
Registrant Phone: +1.2062664064
Registrant Phone Ext:
Registrant Fax: +1.2062667010
Registrant Fax Ext:
Registrant Email: hostmaster@amazon.com
Registry Admin ID:
Admin Name: Hostmaster, Amazon Legal Dept.
Admin Organization: Amazon Technologies, Inc.
Admin Street: P.O. Box 8102
Admin City: Reno
Admin State/Province: NV
Admin Postal Code: 89507
Admin Country: US
Admin Phone: +1.2062664064
Admin Phone Ext:
Admin Fax: +1.2062667010
Admin Fax Ext:
Admin Email: hostmaster@amazon.com
Registry Tech ID:
Tech Name: Hostmaster, Amazon Legal Dept.
Tech Organization: Amazon Technologies, Inc.
Tech Street: P.O. Box 8102
Tech City: Reno
Tech State/Province: NV
Tech Postal Code: 89507
Tech Country: US
Tech Phone: +1.2062664064
Tech Phone Ext:
Tech Fax: +1.2062667010
Tech Fax Ext:
Tech Email: hostmaster@amazon.com
Name Server: ns3.p31.dynect.net
Name Server: ns2.p31.dynect.net
Name Server: ns1.p31.dynect.net
Name Server: pdns1.ultradns.net
Name Server: ns4.p31.dynect.net
Name Server: pdns6.ultradns.co.uk
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
>>> Last update of WHOIS database: 2020-08-27T04:34:01-0700 <<<

Jahnvi@shahs-MBP ~ %
```

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.

```
Jahnvi@shahs-MBP ~ % nslookup spit.ac.in
Server:         192.168.1.1
Address:        192.168.1.1#53

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

nslookup command is a program for querying Internet domain name servers (DNS).
nslookup has two modes, which are interactive and interactive.
Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain.
Non-interactive mode is used to print just the name and requested information for a host or domain.
It is a network administration tool that helps diagnose and resolve DNS related issues.
Hence,with the help of it the outside IP address for spit.ac.in was found out.[2]
Alternatively, ping, fping and so on can be used to find out the IP address.

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

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

```
Jahnvi@shahs-MBP ~ % curl  ipinfo.io/43.252.193.19
{
  "ip": "43.252.193.19",
  "city": "Mumbai",
  "region": "Maharashtra",
  "country": "IN",
  "loc": "19.0728,72.8826",
  "org": "AS17625 BlazeNet's Network",
  "postal": "400070",
  "timezone": "Asia/Kolkata",
  "readme": "https://ipinfo.io/missingauth"
}%
```

```
Jahnvi@shahs-MBP ~ % 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"
}%
```

Reference:

1. https://network-tools.com/trace/
2. https://www.2daygeek.com/linux-command-find-check-domain-ip-address/

3. https://www.cloudflare.com/learning/cdn/glossary/round-trip-time-rtt/

Conclusion:

1. I learned about Network Latency, RTT and the factors that impact RTT
2. Also came to know about some basic command line network utilities.