Shivangi Kochrekar
2018130020

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 the 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.

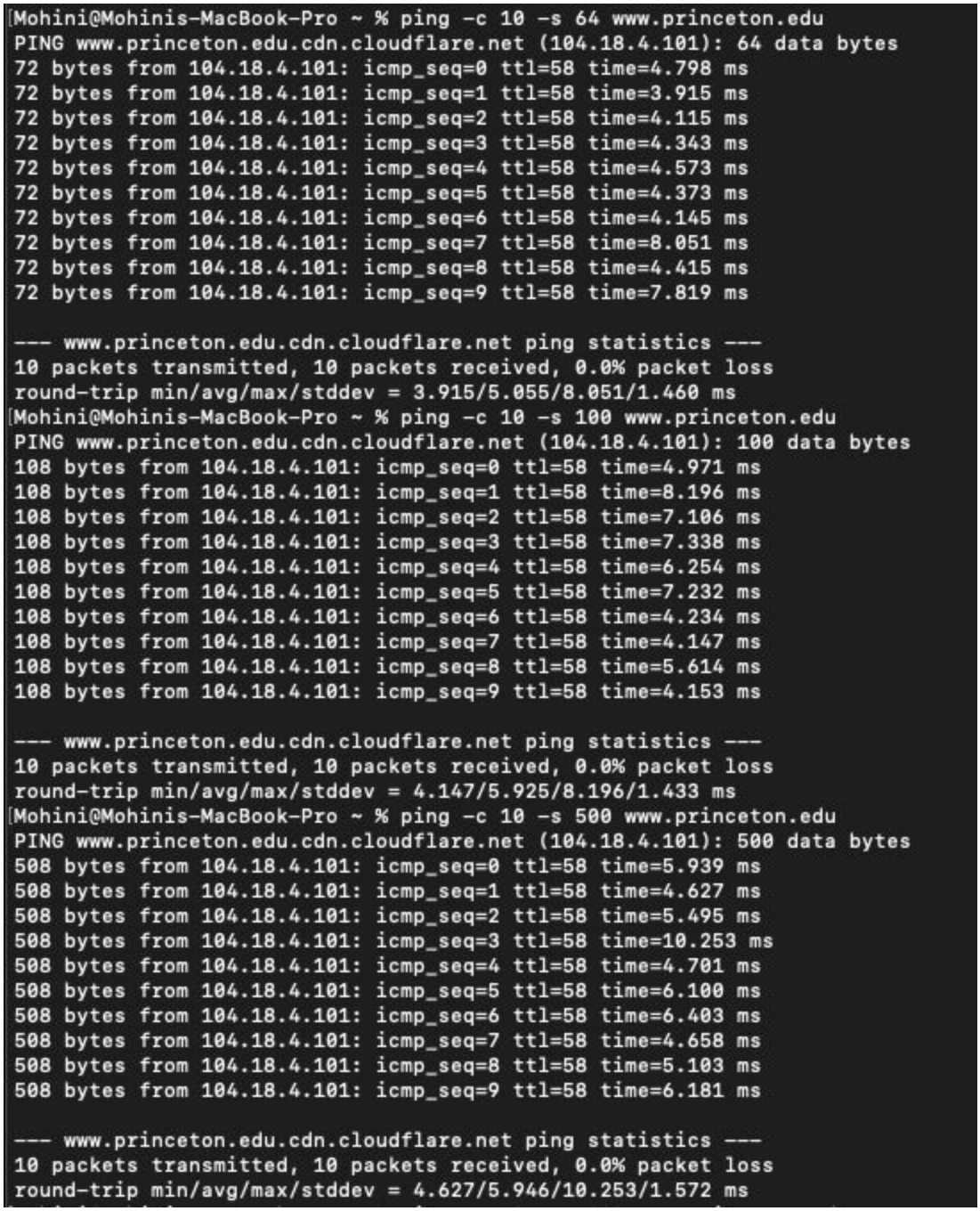
To save the output from `ping` to a file, include a greater than symbol and a file name at the end of the command. For example:

```
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

Result:



```
[Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 1000 www.princeton.edu
PING www.princeton.edu.cdn.cloudflare.net (104.18.4.101): 1000 data bytes
1008 bytes from 104.18.4.101: icmp_seq=0 ttl=58 time=6.166 ms
1008 bytes from 104.18.4.101: icmp_seq=1 ttl=58 time=6.169 ms
1008 bytes from 104.18.4.101: icmp_seq=2 ttl=58 time=5.557 ms
1008 bytes from 104.18.4.101: icmp_seq=3 ttl=58 time=6.393 ms
1008 bytes from 104.18.4.101: icmp_seq=4 ttl=58 time=5.310 ms
1008 bytes from 104.18.4.101: icmp_seq=5 ttl=58 time=5.248 ms
1008 bytes from 104.18.4.101: icmp_seq=6 ttl=58 time=4.985 ms
1008 bytes from 104.18.4.101: icmp_seq=7 ttl=58 time=5.112 ms
1008 bytes from 104.18.4.101: icmp_seq=8 ttl=58 time=10.627 ms
1008 bytes from 104.18.4.101: icmp_seq=9 ttl=58 time=76.318 ms

--- www.princeton.edu.cdn.cloudflare.net ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 4.985/13.188/76.318/21.101 ms
[Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 1400 www.princeton.edu
PING www.princeton.edu.cdn.cloudflare.net (104.18.4.101): 1400 data bytes
1408 bytes from 104.18.4.101: icmp_seq=0 ttl=58 time=12.273 ms
1408 bytes from 104.18.4.101: icmp_seq=1 ttl=58 time=10.470 ms
1408 bytes from 104.18.4.101: icmp_seq=2 ttl=58 time=12.454 ms
1408 bytes from 104.18.4.101: icmp_seq=3 ttl=58 time=7.001 ms
1408 bytes from 104.18.4.101: icmp_seq=4 ttl=58 time=8.162 ms
1408 bytes from 104.18.4.101: icmp_seq=5 ttl=58 time=7.751 ms
1408 bytes from 104.18.4.101: icmp_seq=6 ttl=58 time=13.028 ms
1408 bytes from 104.18.4.101: icmp_seq=7 ttl=58 time=5.415 ms
1408 bytes from 104.18.4.101: icmp_seq=8 ttl=58 time=6.304 ms
1408 bytes from 104.18.4.101: icmp_seq=9 ttl=58 time=7.445 ms

--- www.princeton.edu.cdn.cloudflare.net ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 5.415/9.030/13.028/2.642 ms
```

ping -c -s 64 www.uw.edu

```
[Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 64 www.uw.edu
PING www.washington.edu (128.95.155.134): 64 data bytes
72 bytes from 128.95.155.134: icmp_seq=0 ttl=46 time=332.508 ms
72 bytes from 128.95.155.134: icmp_seq=1 ttl=46 time=252.643 ms
72 bytes from 128.95.155.134: icmp_seq=2 ttl=46 time=252.634 ms
72 bytes from 128.95.155.134: icmp_seq=3 ttl=46 time=265.329 ms
72 bytes from 128.95.155.134: icmp_seq=4 ttl=46 time=262.950 ms
72 bytes from 128.95.155.134: icmp_seq=5 ttl=46 time=252.392 ms
72 bytes from 128.95.155.134: icmp_seq=6 ttl=46 time=262.806 ms
72 bytes from 128.95.155.134: icmp_seq=7 ttl=46 time=258.193 ms
Request timeout for icmp_seq 8
72 bytes from 128.95.155.134: icmp_seq=9 ttl=46 time=258.388 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 9 packets received, 10.0% packet loss
round-trip min/avg/max/stddev = 252.392/266.427/332.508/23.811 ms
```

ping -c 10 -s 64 google.com

```
Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 64 google.com
PING google.com (216.58.196.78): 64 data bytes
72 bytes from 216.58.196.78: icmp_seq=0 ttl=119 time=3.739 ms
72 bytes from 216.58.196.78: icmp_seq=1 ttl=119 time=3.206 ms
72 bytes from 216.58.196.78: icmp_seq=2 ttl=119 time=3.360 ms
72 bytes from 216.58.196.78: icmp_seq=3 ttl=119 time=4.557 ms
72 bytes from 216.58.196.78: icmp_seq=4 ttl=119 time=2.987 ms
72 bytes from 216.58.196.78: icmp_seq=5 ttl=119 time=3.156 ms
72 bytes from 216.58.196.78: icmp_seq=6 ttl=119 time=5.835 ms
72 bytes from 216.58.196.78: icmp_seq=7 ttl=119 time=6.251 ms
72 bytes from 216.58.196.78: icmp_seq=8 ttl=119 time=5.115 ms
72 bytes from 216.58.196.78: icmp_seq=9 ttl=119 time=3.246 ms

--- google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.987/4.145/6.251/1.149 ms
```

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

```
Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 100 www.uw.edu
PING www.washington.edu (128.95.155.198): 100 data bytes
108 bytes from 128.95.155.198: icmp_seq=0 ttl=48 time=244.083 ms
108 bytes from 128.95.155.198: icmp_seq=1 ttl=48 time=244.853 ms
108 bytes from 128.95.155.198: icmp_seq=2 ttl=48 time=317.754 ms
108 bytes from 128.95.155.198: icmp_seq=3 ttl=48 time=241.923 ms
108 bytes from 128.95.155.198: icmp_seq=4 ttl=48 time=356.974 ms
108 bytes from 128.95.155.198: icmp_seq=5 ttl=48 time=244.055 ms
108 bytes from 128.95.155.198: icmp_seq=6 ttl=48 time=295.182 ms
108 bytes from 128.95.155.198: icmp_seq=7 ttl=48 time=242.124 ms
108 bytes from 128.95.155.198: icmp_seq=8 ttl=48 time=241.392 ms
108 bytes from 128.95.155.198: icmp_seq=9 ttl=48 time=241.436 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 241.392/266.978/356.974/39.452 ms
```

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

```
Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 500  www.uw.edu
PING www.washington.edu (128.95.155.198): 500 data bytes
508 bytes from 128.95.155.198: icmp_seq=0 ttl=48 time=242.842 ms
508 bytes from 128.95.155.198: icmp_seq=1 ttl=48 time=249.548 ms
508 bytes from 128.95.155.198: icmp_seq=2 ttl=48 time=242.166 ms
508 bytes from 128.95.155.198: icmp_seq=3 ttl=48 time=242.187 ms
508 bytes from 128.95.155.198: icmp_seq=4 ttl=48 time=243.311 ms
508 bytes from 128.95.155.198: icmp_seq=5 ttl=48 time=246.964 ms
508 bytes from 128.95.155.198: icmp_seq=6 ttl=48 time=241.724 ms
508 bytes from 128.95.155.198: icmp_seq=7 ttl=48 time=242.090 ms
508 bytes from 128.95.155.198: icmp_seq=8 ttl=48 time=242.324 ms
508 bytes from 128.95.155.198: icmp_seq=9 ttl=48 time=242.556 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 241.724/243.571/249.548/2.448 ms
```

ping -c 10 -s 500 berkeley.edu

```
[Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 500 berkeley.edu
PING berkeley.edu (35.163.72.93): 500 data bytes
508 bytes from 35.163.72.93: icmp_seq=0 ttl=40 time=244.334 ms
508 bytes from 35.163.72.93: icmp_seq=1 ttl=40 time=244.478 ms
508 bytes from 35.163.72.93: icmp_seq=2 ttl=40 time=248.398 ms
508 bytes from 35.163.72.93: icmp_seq=3 ttl=40 time=243.825 ms
508 bytes from 35.163.72.93: icmp_seq=4 ttl=40 time=249.596 ms
508 bytes from 35.163.72.93: icmp_seq=5 ttl=40 time=244.178 ms
508 bytes from 35.163.72.93: icmp_seq=6 ttl=40 time=245.412 ms
508 bytes from 35.163.72.93: icmp_seq=7 ttl=40 time=244.154 ms
508 bytes from 35.163.72.93: icmp_seq=8 ttl=40 time=333.100 ms
508 bytes from 35.163.72.93: icmp_seq=9 ttl=40 time=244.110 ms

--- berkeley.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 243.825/254.159/333.100/26.382 ms
```

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

```
[Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 1000  www.uw.edu
PING www.washington.edu (128.95.155.134): 1000 data bytes
1008 bytes from 128.95.155.134: icmp_seq=0 ttl=46 time=267.311 ms
1008 bytes from 128.95.155.134: icmp_seq=1 ttl=46 time=270.528 ms
1008 bytes from 128.95.155.134: icmp_seq=2 ttl=46 time=280.764 ms
1008 bytes from 128.95.155.134: icmp_seq=3 ttl=46 time=267.738 ms
1008 bytes from 128.95.155.134: icmp_seq=4 ttl=46 time=255.685 ms
1008 bytes from 128.95.155.134: icmp_seq=5 ttl=46 time=255.182 ms
1008 bytes from 128.95.155.134: icmp_seq=6 ttl=46 time=263.233 ms
1008 bytes from 128.95.155.134: icmp_seq=7 ttl=46 time=262.313 ms
1008 bytes from 128.95.155.134: icmp_seq=8 ttl=46 time=268.929 ms
1008 bytes from 128.95.155.134: icmp_seq=9 ttl=46 time=256.784 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 255.182/264.847/280.764/7.537 ms
```

ping -c 10 -s 1000 www.ox.ac.uk

```
[Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 1000 www.ox.ac.uk
PING www.ox.ac.uk (151.101.66.133): 1000 data bytes
1008 bytes from 151.101.66.133: icmp_seq=0 ttl=58 time=31.190 ms
1008 bytes from 151.101.66.133: icmp_seq=1 ttl=58 time=31.205 ms
1008 bytes from 151.101.66.133: icmp_seq=2 ttl=58 time=29.031 ms
1008 bytes from 151.101.66.133: icmp_seq=3 ttl=58 time=28.738 ms
1008 bytes from 151.101.66.133: icmp_seq=4 ttl=58 time=28.951 ms
1008 bytes from 151.101.66.133: icmp_seq=5 ttl=58 time=32.719 ms
1008 bytes from 151.101.66.133: icmp_seq=6 ttl=58 time=29.624 ms
1008 bytes from 151.101.66.133: icmp_seq=7 ttl=58 time=30.902 ms
1008 bytes from 151.101.66.133: icmp_seq=8 ttl=58 time=31.396 ms
1008 bytes from 151.101.66.133: icmp_seq=9 ttl=58 time=33.162 ms

--- www.ox.ac.uk ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 28.738/30.692/33.162/1.484 ms
```

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

```
Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 1400  www.uw.edu
PING www.washington.edu (128.95.155.134): 1400 data bytes
1408 bytes from 128.95.155.134: icmp_seq=0 ttl=46 time=254.093 ms
1408 bytes from 128.95.155.134: icmp_seq=1 ttl=46 time=254.604 ms
1408 bytes from 128.95.155.134: icmp_seq=2 ttl=46 time=255.294 ms
1408 bytes from 128.95.155.134: icmp_seq=3 ttl=46 time=260.368 ms
1408 bytes from 128.95.155.134: icmp_seq=4 ttl=46 time=271.496 ms
1408 bytes from 128.95.155.134: icmp_seq=5 ttl=46 time=256.955 ms
1408 bytes from 128.95.155.134: icmp_seq=6 ttl=46 time=254.982 ms
1408 bytes from 128.95.155.134: icmp_seq=7 ttl=46 time=254.099 ms
1408 bytes from 128.95.155.134: icmp_seq=8 ttl=46 time=255.586 ms
1408 bytes from 128.95.155.134: icmp_seq=9 ttl=46 time=255.116 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 254.093/257.259/271.496/5.059 ms
```

ping -c 10 -s 1400 www.mozilla.org

```
Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 1400 www.mozilla.org
PING www.mozilla.org.cdn.cloudflare.net (104.18.164.34): 1400 data bytes
1408 bytes from 104.18.164.34: icmp_seq=0 ttl=58 time=8.163 ms
1408 bytes from 104.18.164.34: icmp_seq=1 ttl=58 time=9.131 ms
1408 bytes from 104.18.164.34: icmp_seq=2 ttl=58 time=5.739 ms
1408 bytes from 104.18.164.34: icmp_seq=3 ttl=58 time=5.980 ms
1408 bytes from 104.18.164.34: icmp_seq=4 ttl=58 time=16.165 ms
1408 bytes from 104.18.164.34: icmp_seq=5 ttl=58 time=10.225 ms
1408 bytes from 104.18.164.34: icmp_seq=6 ttl=58 time=6.060 ms
1408 bytes from 104.18.164.34: icmp_seq=7 ttl=58 time=6.006 ms
1408 bytes from 104.18.164.34: icmp_seq=8 ttl=58 time=6.759 ms
1408 bytes from 104.18.164.34: icmp_seq=9 ttl=58 time=9.492 ms

--- www.mozilla.org.cdn.cloudflare.net ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 5.739/8.372/16.165/3.037 ms
```

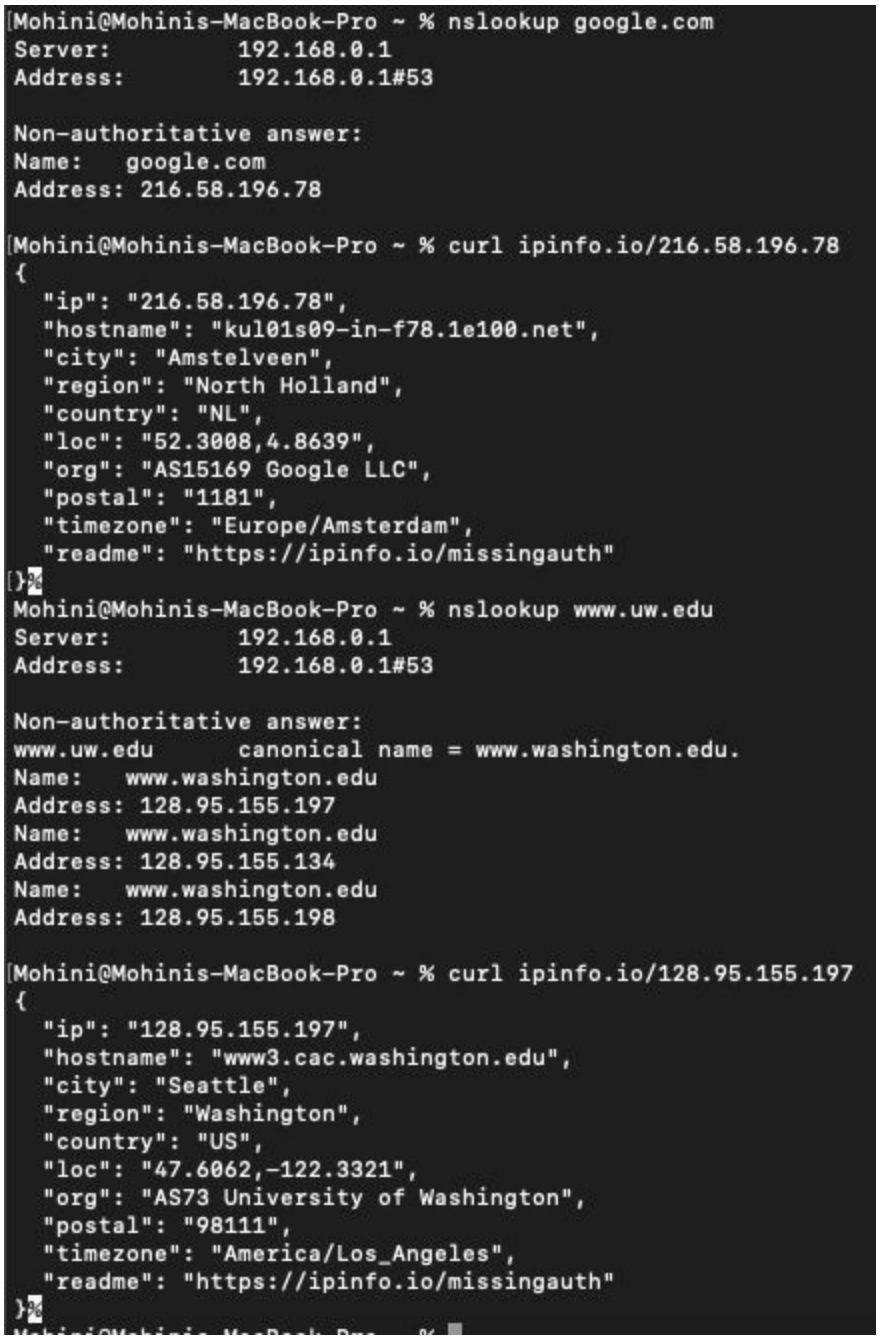
Questions About Latency

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?

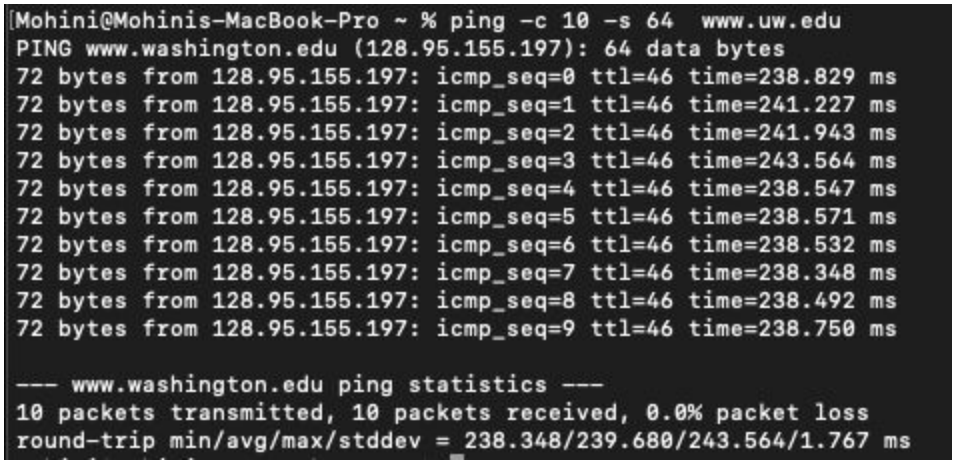
```
Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 64 google.com
PING google.com (216.58.196.78): 64 data bytes
72 bytes from 216.58.196.78: icmp_seq=0 ttl=119 time=3.739 ms
72 bytes from 216.58.196.78: icmp_seq=1 ttl=119 time=3.206 ms
72 bytes from 216.58.196.78: icmp_seq=2 ttl=119 time=3.360 ms
72 bytes from 216.58.196.78: icmp_seq=3 ttl=119 time=4.557 ms
72 bytes from 216.58.196.78: icmp_seq=4 ttl=119 time=2.987 ms
72 bytes from 216.58.196.78: icmp_seq=5 ttl=119 time=3.156 ms
72 bytes from 216.58.196.78: icmp_seq=6 ttl=119 time=5.835 ms
72 bytes from 216.58.196.78: icmp_seq=7 ttl=119 time=6.251 ms
72 bytes from 216.58.196.78: icmp_seq=8 ttl=119 time=5.115 ms
72 bytes from 216.58.196.78: icmp_seq=9 ttl=119 time=3.246 ms

--- google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.987/4.145/6.251/1.149 ms
```

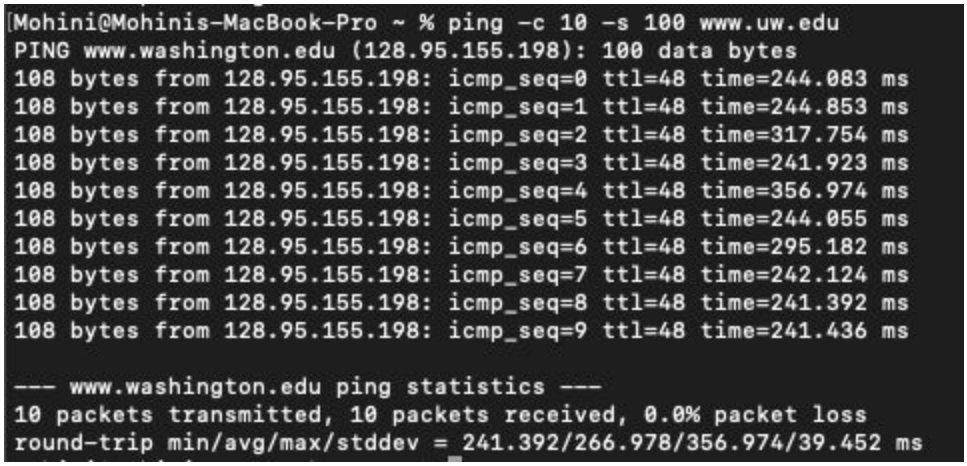
```
Mohini@Mohinis-MacBook-Pro ~ % ping -c 10 -s 64  www.uw.edu
PING www.washington.edu (128.95.155.197): 64 data bytes
72 bytes from 128.95.155.197: icmp_seq=0 ttl=46 time=238.829 ms
72 bytes from 128.95.155.197: icmp_seq=1 ttl=46 time=241.227 ms
72 bytes from 128.95.155.197: icmp_seq=2 ttl=46 time=241.943 ms
72 bytes from 128.95.155.197: icmp_seq=3 ttl=46 time=243.564 ms
72 bytes from 128.95.155.197: icmp_seq=4 ttl=46 time=238.547 ms
72 bytes from 128.95.155.197: icmp_seq=5 ttl=46 time=238.571 ms
72 bytes from 128.95.155.197: icmp_seq=6 ttl=46 time=238.532 ms
72 bytes from 128.95.155.197: icmp_seq=7 ttl=46 time=238.348 ms
72 bytes from 128.95.155.197: icmp_seq=8 ttl=46 time=238.492 ms
72 bytes from 128.95.155.197: icmp_seq=9 ttl=46 time=238.750 ms

--- www.washington.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 238.348/239.680/243.564/1.767 ms
```



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?





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 a 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 nslookup by adding the server name or IP address to the command: nslookup <host> <server>

```
[Mohini@Mohinis-MacBook-Pro ~ % nslookup spit.ac.in
Server:         192.168.0.1
Address:        192.168.0.1#53

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

[Mohini@Mohinis-MacBook-Pro ~ % nslookup google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.199.142
```

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!!!.)

```
Mohini@Mohinis-MacBook-Pro ~ % ifconfig
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=50b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV,CHANNEL_IO>
        ether 38:c9:86:58:f0:9b
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (none)
        status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 24:f0:94:e7:d6:f0
        inet6 fe80::87c:8ef6:e691:42c8%en1 prefixlen 64 secured scopeid 0x5
        inet 192.168.0.101 netmask 0xffffff00 broadcast 192.168.0.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
ham0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1404
        ether 7a:79:19:64:fd:e7
        inet 25.100.253.231 netmask 0xff000000 broadcast 25.255.255.255
        inet6 fe80::7879:19ff:fe64:fde7%ham0 prefixlen 64 scopeid 0x6
        inet6 2620:9b::1964:fde7 prefixlen 96
        nd6 options=201<PERFORMNUD,DAD>
        open (pid 161)
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:0a:7b:00:4f:00
        media: autoselect <full-duplex>
        status: inactive
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
        lladdr 08:74:02:ff:fe:ec:01:3c
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect <full-duplex>
        status: inactive
```

```
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=63<RXCSUM,TXCSUM,TSO4,TSO6>
        ether 82:0a:7b:00:4f:00
        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: en2 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 7 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 06:f0:94:e7:d6:f0
        media: autoselect
        status: inactive
awdl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
        options=400<CHANNEL_IO>
        ether be:35:7f:43:43:31
        inet6 fe80::bc35:7fff:fe43:4331%awdl0 prefixlen 64 scopeid 0xb
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
        inet6 fe80::8ec6:f766:45db:c434%utun0 prefixlen 64 scopeid 0xc
        nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
        inet6 fe80::1d23:9f8d:46c5:cca6%utun1 prefixlen 64 scopeid 0xd
        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.)

```
[Mohini@Mohinis-MacBook-Pro ~ % netstat -p tcp -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.0.101.54951    17.188.142.171.5223    ESTABLISHED
tcp4       0    122  192.168.0.101.54949    17.188.135.40.5223     FIN_WAIT_1
tcp4       0      0  192.168.0.101.54948    whatsapp-cdn-shv.https ESTABLISHED
tcp4       0      0  192.168.0.101.53948    ec2-52-33-90-79..https ESTABLISHED
tcp4       0      0  192.168.0.101.53758    sa-in-f188.1e100.5228  ESTABLISHED
tcp4       0      0  localhost.15292        *.*                    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 the 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 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

-> 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:

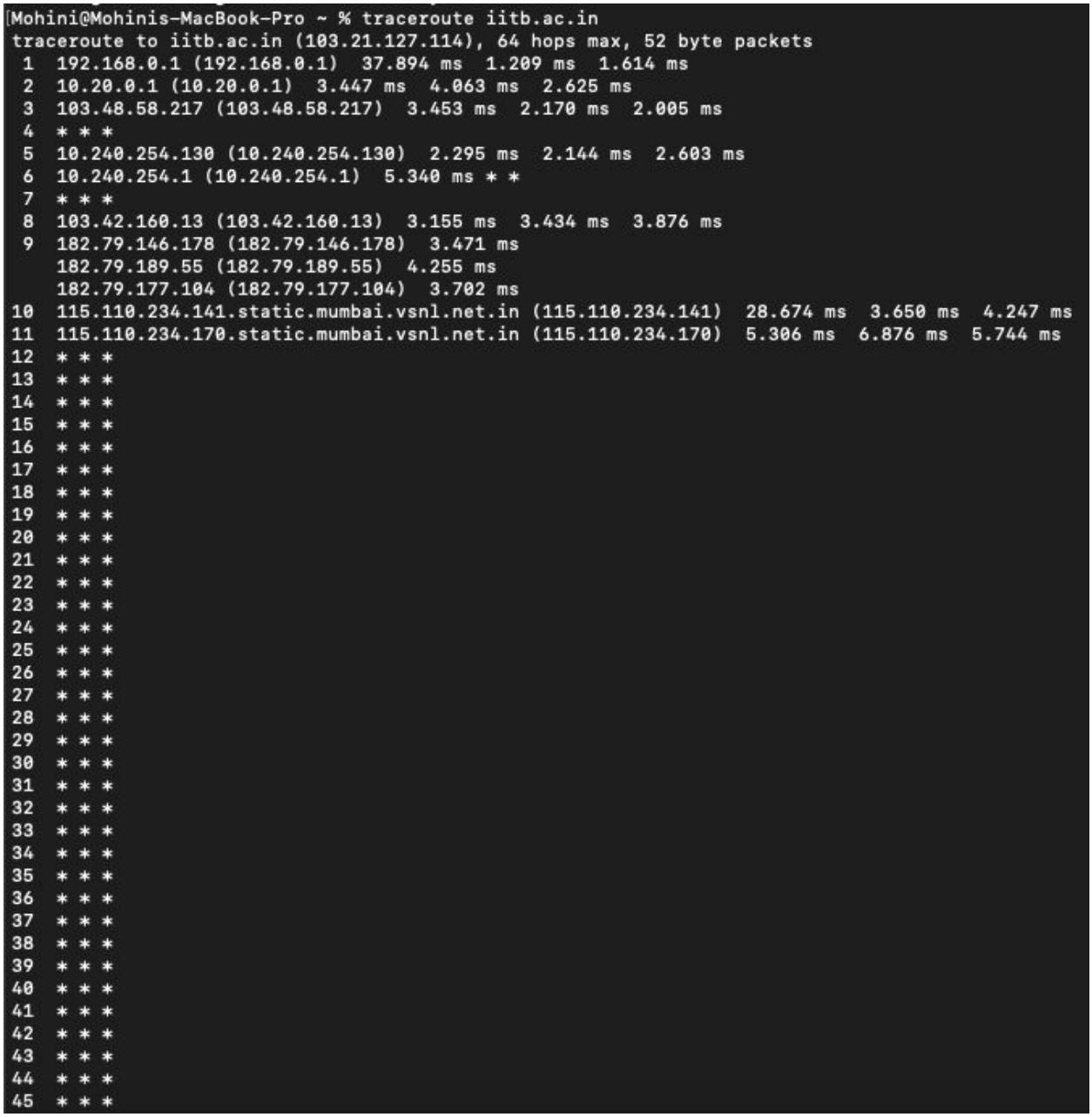
```
tracert <hostname>
```

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. iitb.ac.in



```
[Mohini@Mohinis-MacBook-Pro ~ % traceroute iitb.ac.in
traceroute to iitb.ac.in (103.21.127.114), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  37.894 ms  1.209 ms  1.614 ms
 2  10.20.0.1 (10.20.0.1)  3.447 ms  4.063 ms  2.625 ms
 3  103.48.58.217 (103.48.58.217)  3.453 ms  2.170 ms  2.005 ms
 4  * * *
 5  10.240.254.130 (10.240.254.130)  2.295 ms  2.144 ms  2.603 ms
 6  10.240.254.1 (10.240.254.1)  5.340 ms * *
 7  * * *
 8  103.42.160.13 (103.42.160.13)  3.155 ms  3.434 ms  3.876 ms
 9  182.79.146.178 (182.79.146.178)  3.471 ms
    182.79.189.55 (182.79.189.55)  4.255 ms
    182.79.177.104 (182.79.177.104)  3.702 ms
10  115.110.234.141.static.mumbai.vsnl.net.in (115.110.234.141)  28.674 ms  3.650 ms  4.247 ms
11  115.110.234.170.static.mumbai.vsnl.net.in (115.110.234.170)  5.306 ms  6.876 ms  5.744 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
31  * * *
32  * * *
33  * * *
34  * * *
35  * * *
36  * * *
37  * * *
38  * * *
39  * * *
40  * * *
41  * * *
42  * * *
43  * * *
44  * * *
45  * * *
```

```
45  * * *
46  * * *
47  * * *
48  * * *
49  * * *
50  * * *
51  * * *
52  * * *
53  * * *
54  * * *
55  * * *
56  * * *
57  * * *
58  * * *
59  * * *
60  * * *
61  * * *
62  * * *
63  * * *
64  * * *
Mohini@Mohinis-MacBook-Pro ~ %
```

2. mscs.mu.edu

```
Mohini@Mohinis-MacBook-Pro ~ % traceroute -m 20 mscs.mu.edu
traceroute to mscs.mu.edu (134.48.4.5), 20 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  4.244 ms  1.017 ms  0.997 ms
 2  10.20.0.1 (10.20.0.1)  2.890 ms  4.192 ms  3.155 ms
 3  103.48.58.217 (103.48.58.217)  3.014 ms  3.740 ms  2.239 ms
 4  10.241.1.6 (10.241.1.6)  5.064 ms  2.828 ms  3.999 ms
 5  10.240.254.130 (10.240.254.130)  2.029 ms  1.866 ms  3.305 ms
 6  10.240.254.1 (10.240.254.1)  3.046 ms  2.732 ms  4.997 ms
 7  * * *
 8  103.42.160.13 (103.42.160.13)  5.953 ms  5.652 ms  3.091 ms
 9  182.79.222.233 (182.79.222.233)  198.572 ms  207.209 ms
    182.79.222.237 (182.79.222.237)  250.037 ms
10  core1.nyc4.he.net (198.32.118.57)  194.650 ms  312.015 ms  212.019 ms
11  100ge2-1.core2.chi1.he.net (184.104.193.173)  204.650 ms  298.440 ms  216.475 ms
12  * * *
13  r-222wwash-isp-ae6-3926.wiscnet.net (140.189.8.126)  254.150 ms  291.599 ms  307.600 ms
14  r-milwaukeeci-809-isp-ae3-0.wiscnet.net (140.189.8.230)  307.022 ms  259.240 ms  353.156 ms
15  marquetteuniv.site.wiscnet.net (216.56.1.202)  239.592 ms  378.758 ms  303.599 ms
16  134.48.10.26 (134.48.10.26)  255.461 ms  272.539 ms  340.806 ms
17  * * *
18  * * *
19  euclid.mscs.mu.edu (134.48.4.5)  265.651 ms  255.769 ms  298.695 ms
```

3. www.cs.grinnell.edu

```
Mohini@Mohinis-MacBook-Pro ~ % traceroute -m 20 www.cs.grinnell.edu
traceroute to www.cs.grinnell.edu (132.161.132.159), 20 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  4.322 ms  1.389 ms  1.913 ms
 2  10.20.0.1 (10.20.0.1)  5.528 ms  3.146 ms  4.531 ms
 3  103.48.58.217 (103.48.58.217)  2.165 ms  2.001 ms  5.927 ms
 4  10.241.1.6 (10.241.1.6)  6.504 ms  2.794 ms  3.395 ms
 5  10.240.254.130 (10.240.254.130)  2.047 ms  1.962 ms  4.466 ms
 6  10.240.254.1 (10.240.254.1)  2.793 ms  2.921 ms  2.761 ms
 7  10.241.1.1 (10.241.1.1)  2.717 ms  3.111 ms  3.746 ms
 8  103.42.160.13 (103.42.160.13)  2.722 ms  17.276 ms  3.009 ms
 9  182.79.222.233 (182.79.222.233)  201.655 ms  199.062 ms
    aes-static-150.36.144.59.airtel.in (59.144.36.150)  200.453 ms
10  core1.nyc4.he.net (198.32.118.57)  205.059 ms  206.432 ms  206.213 ms
11  100ge9-1.core2.chi1.he.net (184.105.223.161)  331.541 ms  512.009 ms
    100ge2-1.core2.chi1.he.net (184.104.193.173)  206.835 ms
12  100ge14-2.core1.msp1.he.net (184.105.223.178)  217.930 ms  211.210 ms  219.068 ms
13  * 216.66.77.218 (216.66.77.218)  386.485 ms *
14  peer-as5056.br02.msp1.tfbnw.net (157.240.76.37)  322.983 ms  254.054 ms  247.291 ms
15  167.142.58.40 (167.142.58.40)  317.184 ms  715.993 ms  409.808 ms
16  67.224.64.62 (67.224.64.62)  307.987 ms  305.689 ms  409.019 ms
17  grinnellcollege1.desm.netins.net (167.142.65.43)  409.610 ms  715.912 ms  307.305 ms
18  * * *
19  * * *
20  * * *
```

4. csail.mit.edu

```
Mohini@Mohinis-MacBook-Pro ~ % traceroute -m 20 csail.mit.edu

traceroute to csail.mit.edu (128.30.2.109), 20 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  1.548 ms  0.984 ms  4.463 ms
 2  10.20.0.1 (10.20.0.1)  2.669 ms  3.871 ms  11.981 ms
 3  103.48.58.217 (103.48.58.217)  4.397 ms  2.212 ms  2.231 ms
 4  10.241.1.6 (10.241.1.6)  2.896 ms  2.827 ms  2.617 ms
 5  10.240.254.130 (10.240.254.130)  4.107 ms  5.747 ms  3.293 ms
 6  10.240.254.1 (10.240.254.1)  2.846 ms  2.967 ms  4.928 ms
 7  * * *
 8  103.42.160.13 (103.42.160.13)  3.020 ms  3.888 ms  3.018 ms
 9  182.79.247.32 (182.79.247.32)  231.896 ms
    182.79.243.25 (182.79.243.25)  265.875 ms
    182.79.243.29 (182.79.243.29)  220.529 ms
10  xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  218.538 ms
    xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  224.153 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  253.290 ms
11  * * *
12  massachuset.bear1.boston1.level3.net (4.53.48.98)  299.746 ms  301.113 ms  323.132 ms
13  dmz-rtr-1-external-rtr-1.mit.edu (18.0.161.17)  304.951 ms  301.450 ms  294.627 ms
14  dmz-rtr-2-dmz-rtr-1-1.mit.edu (18.0.161.6)  305.934 ms  345.944 ms  407.611 ms
15  mitnet.core-1-ext.csail.mit.edu (18.4.7.65)  410.097 ms  408.787 ms  510.470 ms
16  * * *
17  bdr.core-1.csail.mit.edu (128.30.0.246)  300.378 ms  305.789 ms  306.332 ms
18  * * *
19  * * *
20  * * *
```

5. cs.stanford.edu

```
Mohini@Mohinis-MacBook-Pro ~ % traceroute cs.stanford.edu

traceroute to cs.stanford.edu (171.64.64.64), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  27.894 ms  1.118 ms  0.905 ms
 2  10.20.0.1 (10.20.0.1)  11.372 ms  5.750 ms  3.243 ms
 3  103.48.58.217 (103.48.58.217)  2.950 ms  4.124 ms  2.201 ms
 4  10.241.1.6 (10.241.1.6)  2.691 ms  3.193 ms  3.521 ms
 5  10.240.254.130 (10.240.254.130)  3.702 ms  2.317 ms  2.526 ms
 6  10.240.254.1 (10.240.254.1)  2.649 ms  6.290 ms  3.454 ms
 7  10.241.1.1 (10.241.1.1)  2.418 ms  2.883 ms  2.088 ms
 8  103.42.160.13 (103.42.160.13)  2.915 ms  4.876 ms  2.819 ms
 9  182.79.222.233 (182.79.222.233)  221.593 ms
    aes-static-150.36.144.59.airtel.in (59.144.36.150)  199.684 ms  200.286 ms
10  core1.nyc4.he.net (198.32.118.57)  194.398 ms  199.901 ms  201.079 ms
11  100ge8-1.core1.sjc2.he.net (184.105.81.218)  249.405 ms  251.457 ms  250.866 ms
12  100ge1-1.core1.pao1.he.net (72.52.92.158)  235.047 ms  315.684 ms
    10ge4-5.core1.pao1.he.net (72.52.92.69)  259.553 ms
13  stanford-university.100gigabitethernet5-1.core1.pao1.he.net (184.105.177.238)  321.444 ms  320.870 ms  307.021 ms
14  csee-west-rtr-vl3.sunet (171.66.255.140)  252.214 ms  250.220 ms  244.790 ms
15  cs.stanford.edu (171.64.64.64)  242.795 ms  250.948 ms  244.064 ms
```

6. cs.manchester.ac.uk

```
Mohini@Mohinis-MacBook-Pro ~ % traceroute cs.manchester.ac.uk
traceroute to cs.manchester.ac.uk (130.88.101.49), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  3.347 ms  1.429 ms  0.785 ms
 2  10.20.0.1 (10.20.0.1)  4.603 ms  10.061 ms  2.386 ms
 3  103.48.58.217 (103.48.58.217)  3.246 ms  2.576 ms  2.147 ms
 4  10.241.1.6 (10.241.1.6)  2.807 ms  2.812 ms  3.674 ms
 5  10.240.254.130 (10.240.254.130)  1.850 ms  1.811 ms  2.107 ms
 6  10.240.254.1 (10.240.254.1)  6.227 ms  3.800 ms  3.695 ms
 7  * * 10.241.1.1 (10.241.1.1)  11.302 ms
 8  103.42.160.13 (103.42.160.13)  2.846 ms  2.925 ms  3.384 ms
 9  182.79.154.0 (182.79.154.0)  139.455 ms
    182.79.146.216 (182.79.146.216)  174.409 ms  176.027 ms
10  ldn-b4-link.telia.net (62.115.162.232)  137.115 ms *  144.506 ms
11  jisc-ic-345131-ldn-b4.c.telia.net (62.115.175.131)  135.500 ms  135.340 ms  136.681 ms
12  ae24.londhx-sbr1.ja.net (146.97.35.197)  133.963 ms  132.058 ms  135.372 ms
13  ae29.londpg-sbr2.ja.net (146.97.33.2)  132.428 ms  133.909 ms  133.124 ms
14  ae31.erdiss-sbr2.ja.net (146.97.33.22)  139.455 ms  140.291 ms  139.652 ms
15  ae29.manckh-sbr2.ja.net (146.97.33.42)  142.029 ms  145.748 ms  138.166 ms
16  ae23.mancrh-rbr1.ja.net (146.97.38.42)  140.173 ms  144.386 ms  138.212 ms
17  * * universityofmanchester.ja.net (146.97.169.2)  150.757 ms
18  130.88.249.194 (130.88.249.194)  139.044 ms  138.987 ms  142.290 ms
19  * * *
20  gw-jh.its.manchester.ac.uk (130.88.250.32)  145.422 ms  142.979 ms  142.599 ms
21  eps.its.man.ac.uk (130.88.101.49)  142.411 ms  146.125 ms  143.112 ms
```

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.

```
traceroute to math.hws.edu (64.89.144.237), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  12.067 ms  3.823 ms  2.292 ms
 2  10.20.0.1 (10.20.0.1)  7.896 ms  5.215 ms  4.890 ms
 3  103.48.58.217 (103.48.58.217)  4.862 ms  3.463 ms  3.468 ms
 4  10.241.1.6 (10.241.1.6)  4.902 ms  4.210 ms  3.954 ms
 5  10.240.254.130 (10.240.254.130)  5.682 ms  3.211 ms  3.184 ms
 6  10.240.254.1 (10.240.254.1)  10.491 ms  4.703 ms  4.713 ms
 7  * * *
 8  103.42.160.13 (103.42.160.13)  5.817 ms  13.802 ms  3.713 ms
 9  182.79.239.78 (182.79.239.78)  331.229 ms
    182.79.247.34 (182.79.247.34)  303.199 ms
    116.119.35.6 (116.119.35.6)  300.580 ms
10  ae58.edge1.losangeles6.level3.net (4.26.0.17)  298.259 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  301.239 ms
    ae58.edge1.losangeles6.level3.net (4.26.0.17)  225.492 ms
11  * * *
12  * * *
13  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  362.965 ms  306.729 ms  307.202 ms
14  66-195-65-170.static.ctl.one (66.195.65.170)  410.576 ms  306.870 ms  291.740 ms
15  64.89.144.100 (64.89.144.100)  306.859 ms  299.833 ms  301.148 ms
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
```

```
[Mohini@Mohinis-MacBook-Pro ~ % traceroute www.hws.edu
traceroute to www.hws.edu (64.89.145.159), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  1.780 ms  1.135 ms  1.107 ms
 2  10.20.0.1 (10.20.0.1)  3.346 ms  20.437 ms  7.605 ms
 3  103.48.58.217 (103.48.58.217)  2.524 ms  2.307 ms  2.779 ms
 4  10.241.1.6 (10.241.1.6)  3.269 ms  7.395 ms  5.236 ms
 5  10.240.254.130 (10.240.254.130)  1.915 ms  1.901 ms  1.784 ms
 6  10.240.254.1 (10.240.254.1)  4.039 ms  5.119 ms  5.705 ms
 7  10.241.1.1 (10.241.1.1)  2.605 ms *^[[B *
 8  103.42.160.13 (103.42.160.13)  2.988 ms  5.356 ms  4.056 ms
 9  182.79.211.194 (182.79.211.194)  237.856 ms
    182.79.222.29 (182.79.222.29)  236.484 ms
    182.79.247.34 (182.79.247.34)  221.267 ms
10  ae58.edge1.losangeles6.level3.net (4.26.0.17)  218.041 ms
    xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  223.344 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  216.268 ms
11  * * *
12  * * *
13  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  329.810 ms  408.166 ms  304.663 ms
14  66-195-65-170.static.ctl.one (66.195.65.170)  329.132 ms  298.091 ms  298.442 ms
15  64.89.144.100 (64.89.144.100)  308.977 ms  310.204 ms  334.300 ms
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
```

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].

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.

```
Mohini@Mohinis-MacBook-Pro ~ % traceroute cs.stanford.edu
traceroute to cs.stanford.edu (171.64.64.64), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  2.658 ms  2.329 ms  2.452 ms
 2  10.20.0.1 (10.20.0.1)  5.156 ms  9.368 ms  5.847 ms
 3  103.48.58.217 (103.48.58.217)  4.771 ms  4.966 ms  3.772 ms
 4  10.241.1.6 (10.241.1.6)  6.114 ms  8.364 ms  5.050 ms
 5  10.240.254.130 (10.240.254.130)  4.657 ms  9.213 ms  4.491 ms
 6  10.240.254.1 (10.240.254.1)  6.660 ms  7.281 ms  5.237 ms
 7  10.241.1.1 (10.241.1.1)  3.692 ms  5.075 ms  5.211 ms
 8  103.42.160.13 (103.42.160.13)  4.823 ms  8.560 ms  5.401 ms
 9  182.79.222.237 (182.79.222.237)  200.044 ms
    aes-static-150.36.144.59.airtel.in (59.144.36.150)  202.831 ms  205.460 ms
10  core1.nyc4.he.net (198.32.118.57)  222.853 ms  215.268 ms  218.127 ms
11  100ge8-1.core1.sjc2.he.net (184.105.81.218)  275.707 ms  256.965 ms  261.571 ms
12  100ge1-1.core1.pao1.he.net (72.52.92.158)  251.740 ms
    10ge4-5.core1.pao1.he.net (72.52.92.69)  258.369 ms
    100ge1-1.core1.pao1.he.net (72.52.92.158)  238.240 ms
13  stanford-university.100gigabitethernet5-1.core1.pao1.he.net (184.105.177.238)  246.615 ms  246.657 ms  252.055 ms
14  csee-west-rtr-vl3.sunet (171.66.255.140)  269.905 ms  254.198 ms  257.598 ms
15  cs.stanford.edu (171.64.64.64)  252.570 ms  251.527 ms  270.380 ms
```

```
Mohini@Mohinis-MacBook-Pro ~ % traceroute cs.stanford.edu
traceroute to cs.stanford.edu (171.64.64.64), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  3.119 ms  46.250 ms  4.505 ms
 2  10.20.0.1 (10.20.0.1)  66.659 ms  4.715 ms  5.222 ms
 3  103.48.58.217 (103.48.58.217)  6.694 ms  4.795 ms  4.713 ms
 4  10.241.1.6 (10.241.1.6)  9.730 ms  7.899 ms  5.490 ms
 5  10.240.254.130 (10.240.254.130)  2.885 ms  2.803 ms  4.928 ms
 6  10.240.254.1 (10.240.254.1)  11.642 ms  4.113 ms  7.575 ms
 7  10.241.1.1 (10.241.1.1)  4.714 ms  3.209 ms  3.992 ms
 8  103.42.160.13 (103.42.160.13)  3.966 ms  3.419 ms  2.977 ms
 9  182.79.222.233 (182.79.222.233)  199.769 ms  198.471 ms
    aes-static-150.36.144.59.airtel.in (59.144.36.150)  204.966 ms
10  core1.nyc4.he.net (198.32.118.57)  193.569 ms  206.730 ms  206.005 ms
11  100ge8-1.core1.sjc2.he.net (184.105.81.218)  272.136 ms  252.693 ms  256.674 ms
12  100ge1-1.core1.pao1.he.net (72.52.92.158)  236.287 ms
    10ge4-5.core1.pao1.he.net (72.52.92.69)  243.664 ms  267.099 ms
13  stanford-university.100gigabitethernet5-1.core1.pao1.he.net (184.105.177.238)  252.406 ms  245.348 ms  268.246 ms
14  csee-west-rtr-vl3.sunet (171.66.255.140)  250.901 ms  257.363 ms  259.427 ms
15  cs.stanford.edu (171.64.64.64)  262.664 ms  251.463 ms  257.113 ms
Mohini@Mohinis-MacBook-Pro ~ %
```

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 traceroute?

   Yes, the tracerouting follows a particular path from the user's IP address through the IP addresses of the ISP and then the path 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 (number of hops minus 1)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*).

```
Mohini@Mohinis-MacBook-Pro ~ % whois spit.ac.in
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:        whois.registry.in

domain:       IN

organisation: National Internet Exchange of India
address:      6C,6D,6E Hansalaya Building 15, Barakhamba Road
address:      New Delhi 110 001
address:      India

contact:      administrative
name:         Rajiv Kumar
organisation: National Internet Exchange of India
address:      6C,6D,6E Hansalaya Building 15, Barakhamba Road
address:      New Delhi 110 001
address:      India
phone:        +91 11 48202011
fax-no:       +91 11 48202013
e-mail:       registry@nixi.in

contact:      technical
name:         Rajiv Kumar
organisation: National Internet Exchange of India
address:      6C,6D,6E Hansalaya Building 15, Barakhamba Road
address:      New Delhi 110 001
address:      India
phone:        +91 11 48202011
fax-no:       +91 11 48202013
e-mail:       rajiv@nixi.in

nserver:      NS1.REGISTRY.IN 2001:dcd:1:0:0:0:0:12 37.209.192.12
nserver:      NS2.REGISTRY.IN 2001:dcd:2:0:0:0:0:12 37.209.194.12
nserver:      NS3.REGISTRY.IN 2001:dcd:3:0:0:0:0:12 37.209.196.12
nserver:      NS4.REGISTRY.IN 2001:dcd:4:0:0:0:0:12 37.209.198.12
nserver:      NS5.REGISTRY.IN 156.154.100.20 2001:0502:2eda:0:0:0:0:20
nserver:      NS6.REGISTRY.IN 156.154.101.20 2001:0502:ad09:0:0:0:0:20
ds-rdata:     54739 8 2 9F122CFD6604AE6DEDA0FE09F27BE340A318F06AFAC11714A73409D43136472C
ds-rdata:     54739 8 1 2B5CA455A0E65769FF9DF9E75EC40EE1EC1CDCA9
```

```
whois:        whois.registry.in

status:       ACTIVE
remarks:      Registration information: http://www.registry.in

created:      1989-05-08
changed:      2020-04-09
source:       IANA

# whois.registry.in

Domain Name: spit.ac.in
Registry Domain ID: D2241401-IN
Registrar WHOIS Server:
Registrar URL: http://www.ernet.in
Updated Date: 2020-05-18T09:51:15Z
Creation Date: 2006-05-22T04:58:23Z
Registry Expiry Date: 2025-05-22T04:58:23Z
Registrar: ERNET India
Registrar IANA ID: 800068
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: ok http://www.icann.org/epp#OK
Registry Registrant ID:
Registrant Name:
Registrant Organization: Bharatiya Vidya Bhavans Sardar Patel Institute of Technology Mumbai
Registrant Street:
Registrant Street:
Registrant Street:
Registrant City:
Registrant State/Province:
Registrant Postal Code:
Registrant Country: IN
Registrant Phone:
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: Please contact the Registrar listed above
Registry Admin ID:
```

```
Admin Name:
Admin Organization:
Admin Street:
Admin Street:
Admin Street:
Admin City:
Admin State/Province:
Admin Postal Code:
Admin Country:
Admin Phone:
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: Please contact the Registrar listed above
Registry Tech ID:
Tech Name:
Tech Organization:
Tech Street:
Tech Street:
Tech Street:
Tech City:
Tech State/Province:
Tech Postal Code:
Tech Country:
Tech Phone:
Tech Phone Ext:
Tech Fax:
Tech Fax Ext:
Tech Email: Please contact the Registrar listed above
Name Server: ns2.spit.ac.in
Name Server: ns1.spit.ac.in
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of WHOIS database: 2020-08-24T21:10:42Z <<<
```

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.

```
[Mohini@Mohinis-MacBook-Pro ~ % whois google.com
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:        whois.verisign-grs.com

domain:       COM

organisation: VeriSign Global Registry Services
address:      12061 Bluemont Way
address:      Reston Virginia 20190
address:      United States

contact:      administrative
name:         Registry Customer Service
organisation: VeriSign Global Registry Services
address:      12061 Bluemont Way
address:      Reston Virginia 20190
address:      United States
phone:        +1 703 925-6999
fax-no:       +1 703 948 3978
e-mail:       info@verisign-grs.com

contact:      technical
name:         Registry Customer Service
organisation: VeriSign Global Registry Services
address:      12061 Bluemont Way
address:      Reston Virginia 20190
address:      United States
phone:        +1 703 925-6999
fax-no:       +1 703 948 3978
e-mail:       info@verisign-grs.com

nserver:      A.GTLD-SERVERS.NET 192.5.6.30 2001:503:a83e:0:0:0:2:30
nserver:      B.GTLD-SERVERS.NET 192.33.14.30 2001:503:231d:0:0:0:2:30
nserver:      C.GTLD-SERVERS.NET 192.26.92.30 2001:503:83eb:0:0:0:0:30
nserver:      D.GTLD-SERVERS.NET 192.31.80.30 2001:500:856e:0:0:0:0:30
nserver:      E.GTLD-SERVERS.NET 192.12.94.30 2001:502:1ca1:0:0:0:0:30
nserver:      F.GTLD-SERVERS.NET 192.35.51.30 2001:503:d414:0:0:0:0:30
nserver:      G.GTLD-SERVERS.NET 192.42.93.30 2001:503:eea3:0:0:0:0:30
nserver:      H.GTLD-SERVERS.NET 192.54.112.30 2001:502:8cc:0:0:0:0:30
nserver:      I.GTLD-SERVERS.NET 192.43.172.30 2001:503:39c1:0:0:0:0:30
nserver:      J.GTLD-SERVERS.NET 192.48.79.30 2001:502:7094:0:0:0:0:30
nserver:      K.GTLD-SERVERS.NET 192.52.178.30 2001:503:d2d:0:0:0:0:30
nserver:      L.GTLD-SERVERS.NET 192.41.162.30 2001:500:d937:0:0:0:0:30
nserver:      M.GTLD-SERVERS.NET 192.55.83.30 2001:501:b1f9:0:0:0:0:30
ds-rdata:     30909 8 2 E2D3C916F6DEEAC73294E8268FB5885044A833FC5459588F4A9184CFC41A5766
```

```
whois:        whois.verisign-grs.com

status:       ACTIVE
remarks:      Registration information: http://www.verisigninc.com

created:      1985-01-01
changed:      2017-10-05
source:       IANA

# whois.verisign-grs.com

   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: 2019-09-09T15:39:04Z
   Creation Date: 1997-09-15T04:00:00Z
   Registry Expiry Date: 2028-09-14T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
   Name Server: NS4.GOOGLE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2020-08-24T21:17:08Z <<<

# whois.markmonitor.com
```

```
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: ns1.google.com
Name Server: ns3.google.com
Name Server: ns2.google.com
Name Server: ns4.google.com
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
>>> Last update of WHOIS database: 2020-08-24T14:09:38-0700 <<<
```

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.

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.

```
Mohini@Mohinis-MacBook-Pro ~ % nslookup spit.ac.in
Server:         192.168.0.1
Address:        192.168.0.1#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.

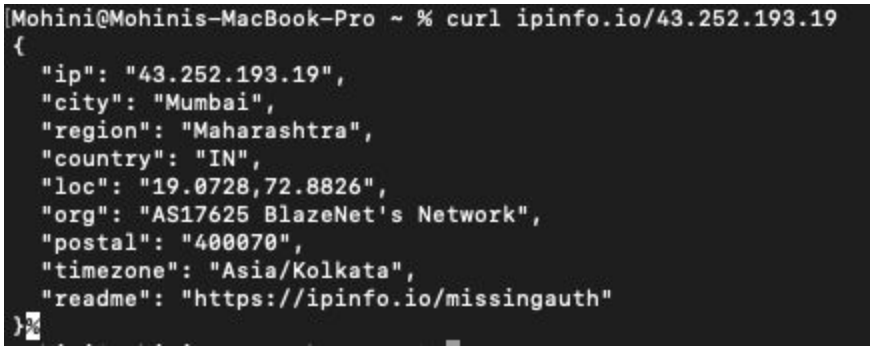
Geolocation — A geolocation service tries to tell, approximately, where a given IP address is located physically. They can't be completely accurate—but they probably get at least the country right most of the time.

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.)

```
Mohini@Mohinis-MacBook-Pro ~ % 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 some basic command line network utilities.
2. Also came to know about Network Latency, RTT and the factors impacting RTT.