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

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

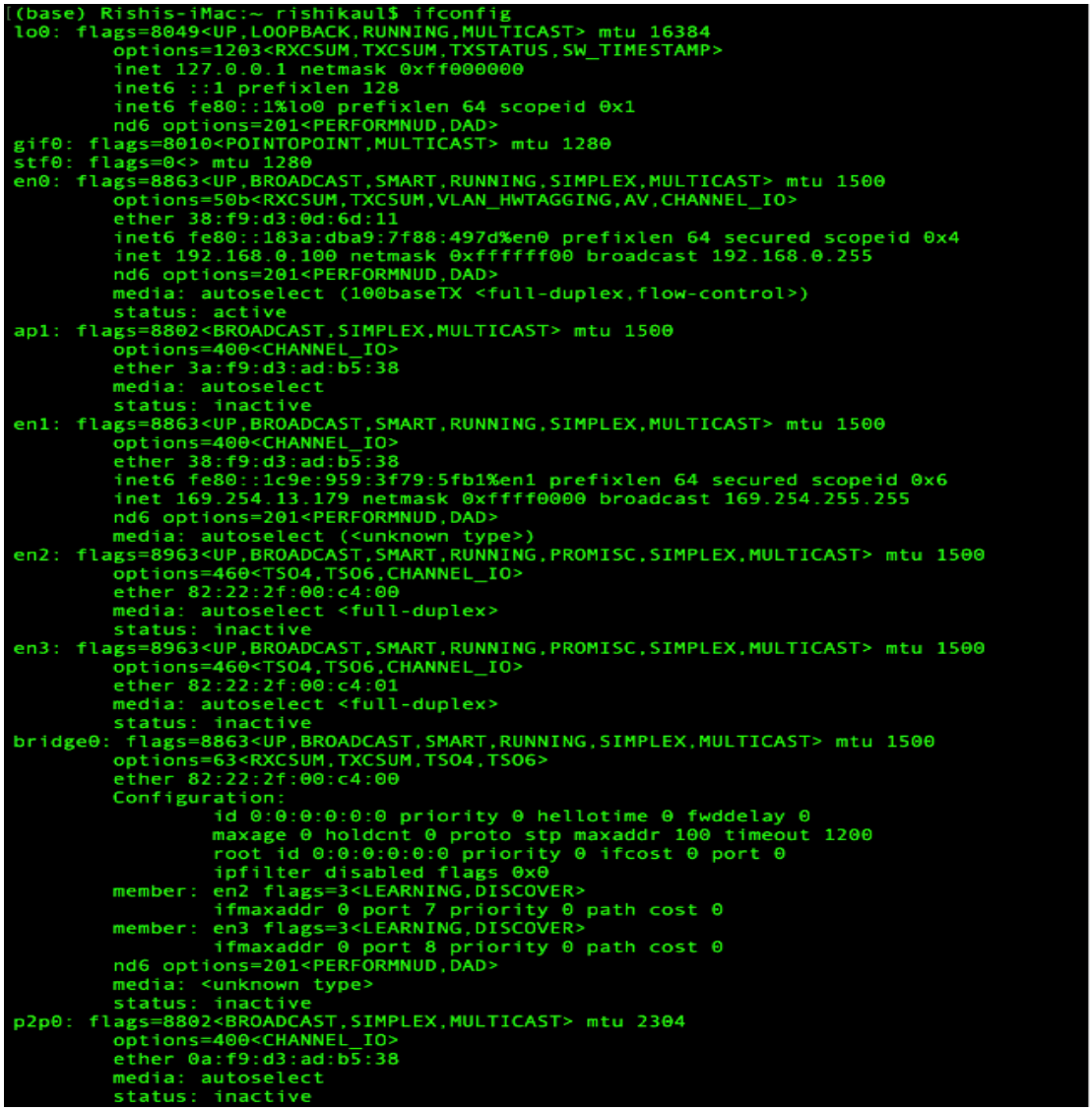
Prerequisite: Basic understanding of command line utilities of Linux Operating system.

Some Basic command line Networking utilities

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

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!!!.)

Experiment 0: Experiment with ifconfig and write details about the information returned.



```
(base) Rishis-iMac:~ rishikaul$ 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:f9:d3:0d:6d:11
        inet6 fe80::183a:dba9:7f88:497d%en0 prefixlen 64 secured scopeid 0x4
        inet 192.168.0.100 netmask 0xffffff00 broadcast 192.168.0.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (100baseTX <full-duplex,flow-control>)
        status: active
ap1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 3a:f9:d3:ad:b5:38
        media: autoselect
        status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 38:f9:d3:ad:b5:38
        inet6 fe80::1c9e:959:3f79:5fb1%en1 prefixlen 64 secured scopeid 0x6
        inet 169.254.13.179 netmask 0xffff0000 broadcast 169.254.255.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (<unknown type>)
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:22:2f:00:c4:00
        media: autoselect <full-duplex>
        status: inactive
en3: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:22:2f:00:c4:01
        media: autoselect <full-duplex>
        status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=63<RXCSUM,TXCSUM,TSO4,TSO6>
        ether 82:22:2f:00:c4:00
        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
        member: en3 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 8 priority 0 path cost 0
        nd6 options=201<PERFORMNUD,DAD>
        media: <unknown type>
        status: inactive
p2p0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 2304
        options=400<CHANNEL_IO>
        ether 0a:f9:d3:ad:b5:38
        media: autoselect
        status: inactive
```

lo is a special virtual network interface called loopback device. Loopback is used mainly for diagnostics and troubleshooting, and to connect to services running on local host.

gif0 is Software Network Interface

stf0 is 6to4 tunnel interface

en0 is a physical interface representing Ethernet network card. It's used for communication with other computers on the network and on the Internet.

Ether is the MAC address which is globally unique.

mtu stands for Maximum Transmission Units is the size of each packet received by the ethernet card. The value of MTU is set to 1500 by default. The loopback device has a higher MTU value than the ethernet device

INET 192.168.0.100 is the ipv4 address

INET6 is the ipv6 address.

bridge0 is a software bridge between other interfaces

p2p0 is a point to point interface for wireless services.

FLAGS:

UP indicates that kernel modules related to the interface have been loaded and interface is activated.

BROADCAST indicates that interface is configured to handle broadcast packets, which is required for obtaining IP address via DHCP.

RUNNING indicates that interface is ready to accept data.

MULTICAST indicates that interface supports multicasting.

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>
```

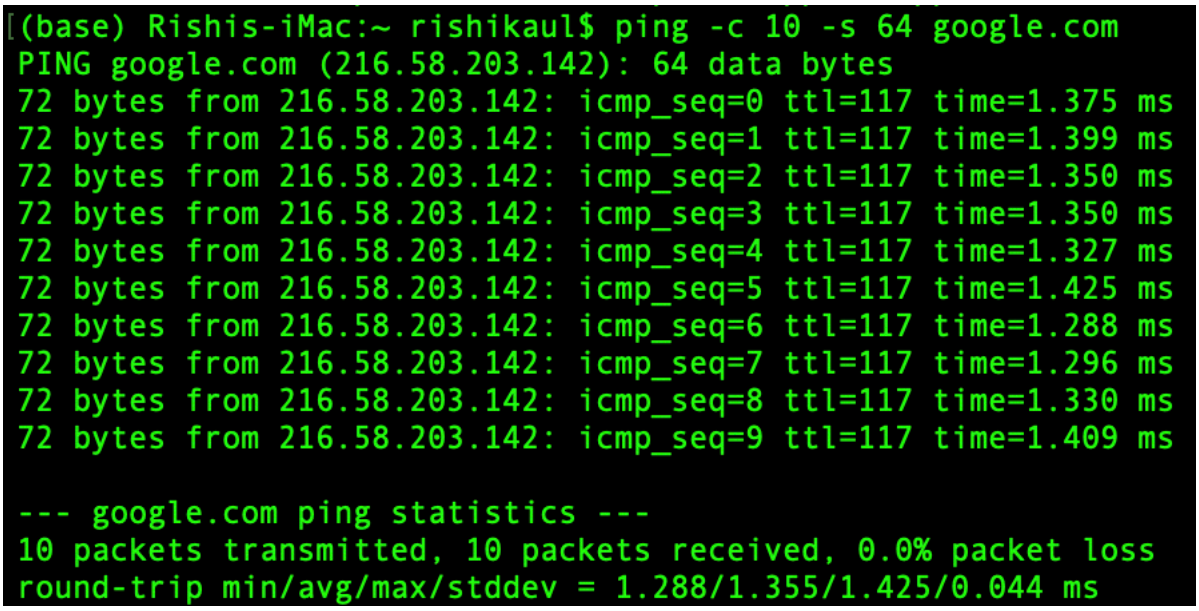
The default number of ICMP packets to send is either infinite (in Linux and Mac OS) or 4 (in Windows). The default packet size is either 64 bytes (in Linux) or 32 bytes (in Windows). You can specify either a hostname (e.g., `spit.ac.in`) or an IP address.

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



```
(base) Rishis-iMac:~ rishikaul$ ping -c 10 -s 100 google.com
PING google.com (216.58.203.142): 100 data bytes
76 bytes from 216.58.203.142: icmp_seq=0 ttl=117 time=1.288 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=1 ttl=117 time=1.302 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=2 ttl=117 time=1.423 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=3 ttl=117 time=1.455 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=4 ttl=117 time=1.511 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=5 ttl=117 time=1.316 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=6 ttl=117 time=1.195 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=7 ttl=117 time=1.454 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=8 ttl=117 time=1.336 ms
wrong total length 96 instead of 128
76 bytes from 216.58.203.142: icmp_seq=9 ttl=117 time=1.421 ms
wrong total length 96 instead of 128

--- google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.195/1.370/1.511/0.093 ms
```

```
(base) Rishis-iMac:~ rishikaul$ ping -c 10 -s 500 google.com
PING google.com (216.58.203.142): 500 data bytes
76 bytes from 216.58.203.142: icmp_seq=0 ttl=117 time=1.415 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=1 ttl=117 time=1.676 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=2 ttl=117 time=2.020 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=3 ttl=117 time=1.621 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=4 ttl=117 time=1.627 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=5 ttl=117 time=1.601 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=6 ttl=117 time=1.384 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=7 ttl=117 time=1.896 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=8 ttl=117 time=1.493 ms
wrong total length 96 instead of 528
76 bytes from 216.58.203.142: icmp_seq=9 ttl=117 time=1.418 ms
wrong total length 96 instead of 528

--- google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.384/1.615/2.020/0.199 ms
```

```
(base) Rishis-iMac:~ rishikaul$ ping -c 10 -s 1400 google.com
PING google.com (216.58.203.142): 1400 data bytes
76 bytes from 216.58.203.142: icmp_seq=0 ttl=117 time=1.974 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=1 ttl=117 time=1.782 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=2 ttl=117 time=1.623 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=3 ttl=117 time=1.797 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=4 ttl=117 time=1.599 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=5 ttl=117 time=1.854 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=6 ttl=117 time=1.603 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=7 ttl=117 time=1.681 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=8 ttl=117 time=1.675 ms
wrong total length 96 instead of 1428
76 bytes from 216.58.203.142: icmp_seq=9 ttl=117 time=1.679 ms
wrong total length 96 instead of 1428

--- google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.599/1.727/1.974/0.116 ms
```

```
(base) Rishis-iMac:~ rishikaul$ ping -c 10 -s 1000 google.com
PING google.com (216.58.203.142): 1000 data bytes
76 bytes from 216.58.203.142: icmp_seq=0 ttl=117 time=1.666 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=1 ttl=117 time=1.660 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=2 ttl=117 time=2.019 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=3 ttl=117 time=1.641 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=4 ttl=117 time=1.512 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=5 ttl=117 time=1.602 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=6 ttl=117 time=1.522 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=7 ttl=117 time=1.566 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=8 ttl=117 time=1.619 ms
wrong total length 96 instead of 1028
76 bytes from 216.58.203.142: icmp_seq=9 ttl=117 time=1.607 ms
wrong total length 96 instead of 1028

--- google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.512/1.641/2.019/0.135 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?

Round-trip time (RTT) is the duration in milliseconds (ms) it takes for a network request to go from a starting point to a destination and back again to the starting point. RTT is an important metric in determining the health of a connection on a local network or the larger Internet, and is commonly utilized by network administrators to diagnose the speed and reliability of network connections.

Delay may differ slightly, depending on the location of the specific pair of communicating endpoints. Engineers usually report both the maximum and average delay, and they divide the delay into several parts:

- Processing delay – time it takes a router to process the packet header, depends on the processing speed of the switch
- Queuing delay – time the packet spends in routing queues depends on the number of packets, size of the packet and bandwidth
- Transmission delay – time it takes to push the packet's bits onto the link depends on size of the packet and the bandwidth of the network.
- Propagation delay – time for a signal to reach its destination depends on distance and propagation speed.

A certain minimum level of delay is experienced by signals due to the time it takes to transmit a packet serially through a link. This delay is extended by more variable levels of delay due to network congestion. IP network delays can range from a few milliseconds to several hundred milliseconds.

So yes , Average RTT does vary between different hosts due to queuing delay as we can see in above example the average RTT was calculated for google.com and yahoo.com differs .This can mostly be due to propagation Delay as it depends on distance and due to Queuing delay as the packet may be in queue

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

Yes, the average RTT increases with packet size as Queuing delay and Transmission delay increases as they both to rely on size of packets eventually increasing the average RTT

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).

```
PING uw.edu (128.95.155.197): 64 data bytes
72 bytes from 128.95.155.197: icmp_seq=0 ttl=47 time=262.816 ms
72 bytes from 128.95.155.197: icmp_seq=1 ttl=47 time=262.685 ms
72 bytes from 128.95.155.197: icmp_seq=2 ttl=47 time=262.829 ms
72 bytes from 128.95.155.197: icmp_seq=3 ttl=47 time=262.691 ms
72 bytes from 128.95.155.197: icmp_seq=4 ttl=47 time=262.793 ms
72 bytes from 128.95.155.197: icmp_seq=5 ttl=47 time=262.606 ms
72 bytes from 128.95.155.197: icmp_seq=6 ttl=47 time=262.611 ms
72 bytes from 128.95.155.197: icmp_seq=7 ttl=47 time=262.570 ms
72 bytes from 128.95.155.197: icmp_seq=8 ttl=47 time=262.605 ms
72 bytes from 128.95.155.197: icmp_seq=9 ttl=47 time=262.563 ms

--- uw.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 262.563/262.677/262.829/0.098 ms
```

```
PING cornell.edu (128.253.173.248): 64 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
Request timeout for icmp_seq 6
Request timeout for icmp_seq 7
Request timeout for icmp_seq 8

--- cornell.edu ping statistics ---
10 packets transmitted, 0 packets received, 100.0% packet loss
```

```
PING berkeley.edu (35.163.72.93): 64 data bytes
72 bytes from 35.163.72.93: icmp_seq=0 ttl=37 time=258.004 ms
72 bytes from 35.163.72.93: icmp_seq=1 ttl=37 time=258.113 ms
72 bytes from 35.163.72.93: icmp_seq=2 ttl=37 time=258.138 ms
72 bytes from 35.163.72.93: icmp_seq=3 ttl=37 time=258.001 ms
72 bytes from 35.163.72.93: icmp_seq=4 ttl=37 time=258.169 ms
72 bytes from 35.163.72.93: icmp_seq=5 ttl=37 time=257.970 ms
72 bytes from 35.163.72.93: icmp_seq=6 ttl=37 time=257.917 ms
72 bytes from 35.163.72.93: icmp_seq=7 ttl=37 time=257.979 ms
72 bytes from 35.163.72.93: icmp_seq=8 ttl=37 time=257.867 ms
72 bytes from 35.163.72.93: icmp_seq=9 ttl=37 time=257.832 ms

--- berkeley.edu ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 257.832/257.999/258.169/0.107 ms
```

```
PING www.u-tokyo.ac.jp (210.152.243.234): 64 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
Request timeout for icmp_seq 6
Request timeout for icmp_seq 7
Request timeout for icmp_seq 8

--- www.u-tokyo.ac.jp ping statistics ---
10 packets transmitted, 0 packets received, 100.0% packet loss
```

```
PING uchicago.edu (34.200.129.209): 64 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
Request timeout for icmp_seq 6
Request timeout for icmp_seq 7
Request timeout for icmp_seq 8

--- uchicago.edu ping statistics ---
10 packets transmitted, 0 packets received, 100.0% packet loss
```

Interesting Observations:

The average RTT varies for different destinations for the same packet size.

The reasons for this are:

Nature of the transmission medium - the way in which connections are made affects how fast the connection moves; connections made over optical fiber will behave differently than connections made over copper. Likewise, a connection made over a wireless frequency will behave differently than that of a satellite communication.

Physical Distance: Physical distance between two hosts is defined as the length of the great circle arc connecting their locations on the surface of the Earth. The sum of per-hop distances would not significantly diverge from the actual distance between hosts. However, several studies have shown that the sum of per-hop distances may greatly exceed the great circle distance between two end hosts. The only solution to this is to get the content closer to the users.

Server response time – the amount of time it takes a server to process and respond to a request is a potential bottleneck in network latency. When a server is overwhelmed with requests, such as during a DDoS attack, its ability to respond efficiently can be inhibited, resulting in increased 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>

netstat — The netstat command gives information about network connections. I often use netstat -t -n which lists currently open TCP connections (that's the "-t" option) by IP address rather than domain name (that's the "-n" option). Add the option "-l" (lower case ell) to list listening sockets, that is sockets that have been opened by server programs to wait for connection requests from clients: netstat -t -n -l. (On Mac, use netstat -p tcp to list tcp connections, and add "-a" to include listening sockets in the list.)

telnet — Telnet is an old program for remote login. It's not used so much for that any more, since it has no security features. But basically, all it does is open a connection to a server and allow server and client to send lines of plain text to each other. It can be used to check that it's possible to connect to a server and, if the server communicates in plain text, even to interact with the server by hand. Since the Web uses a plain text protocol, you can use telnet to connect to a web client and play the part of the web browser. I will suggest that you to do this with your own web server when you write it, but you might want to try it now. When you use telnet in this way, you need to specify both the host and the port number to which you want to connect: telent <host> <port>. For example, to connect to the web server on www.spit.ac.in: telnet spit.ac.in 80

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

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

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

```
traceroute <hostname>
```

The syntax in Windows is:

```
tracert <hostname>
```

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

1.2.1 Experiments with Traceroute

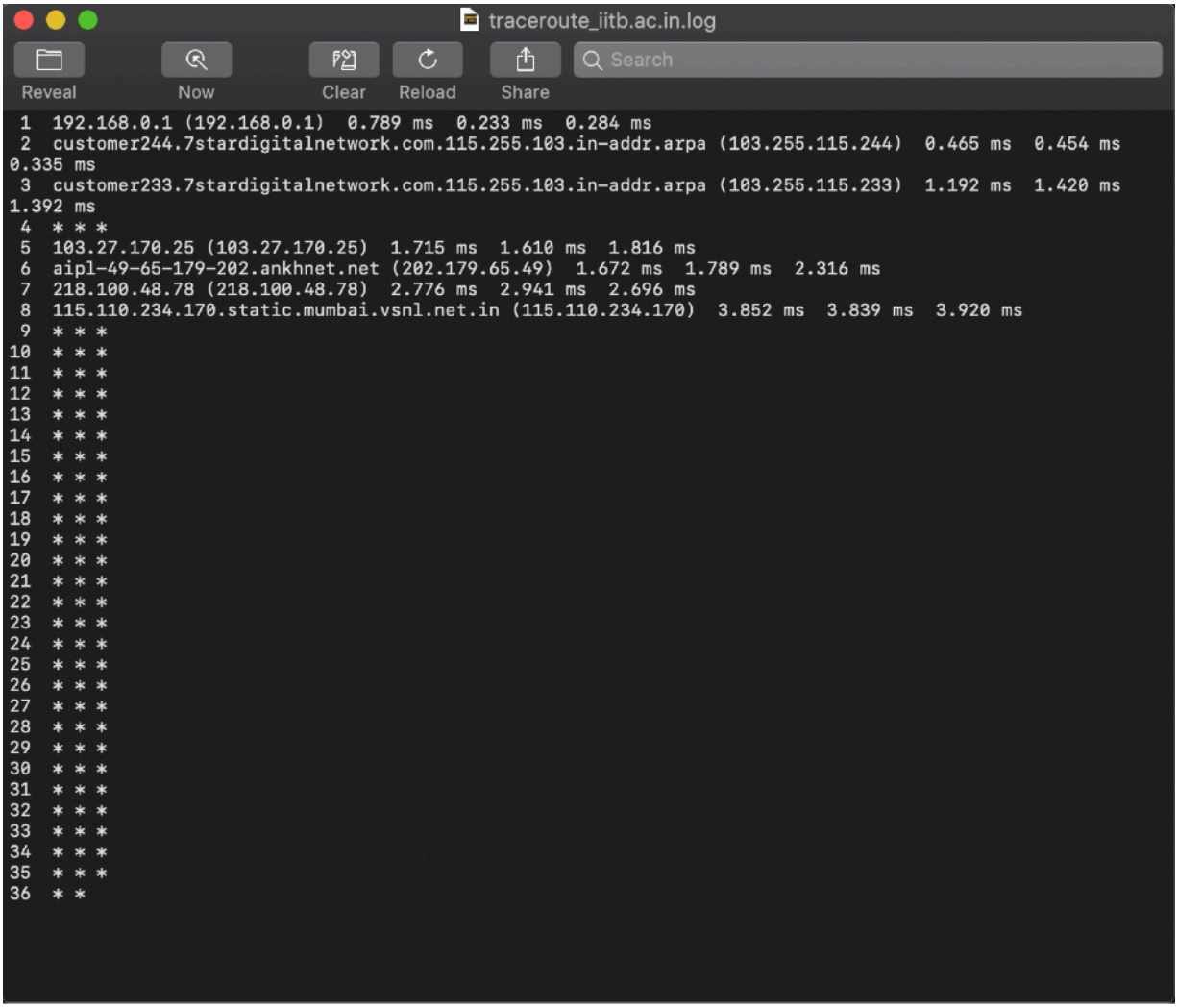
From **your machine** traceroute to the following hosts:

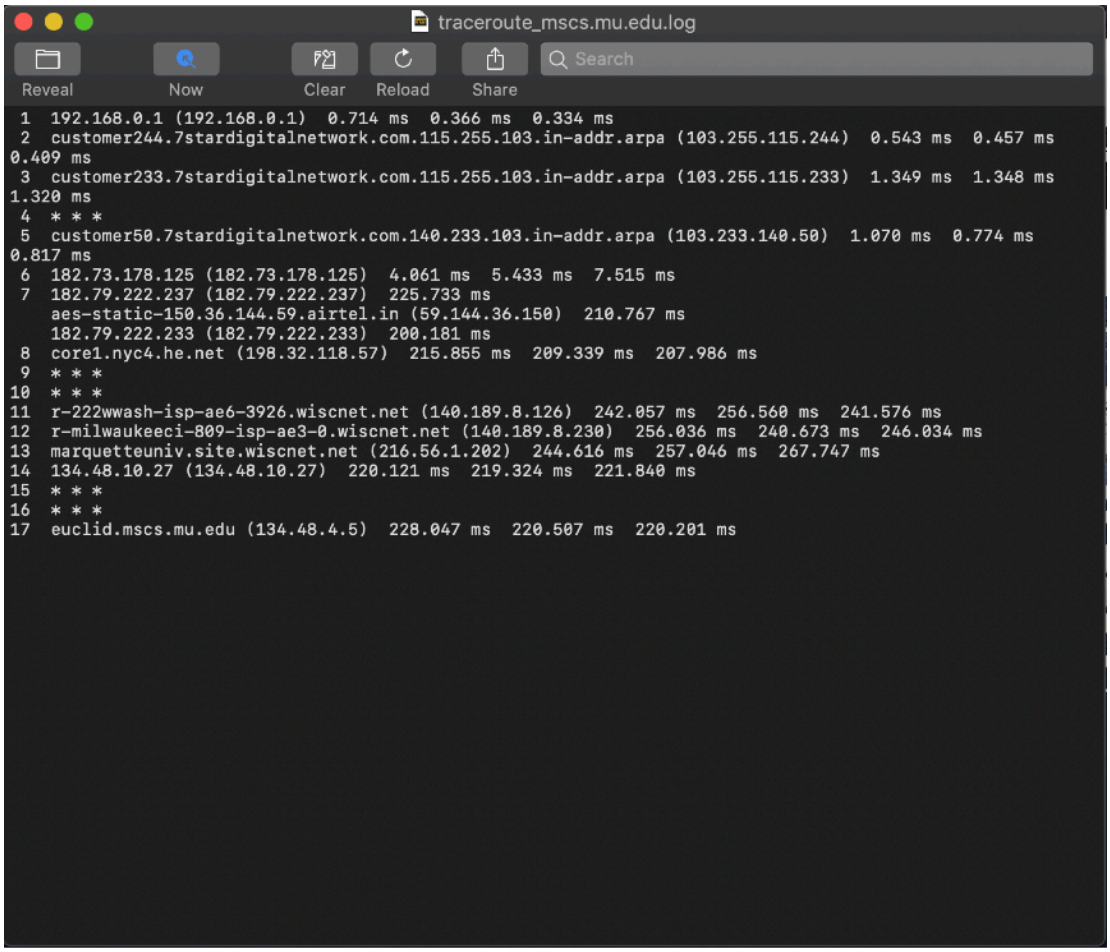
1. ee.iitb.ac.in
2. mscs.mu.edu

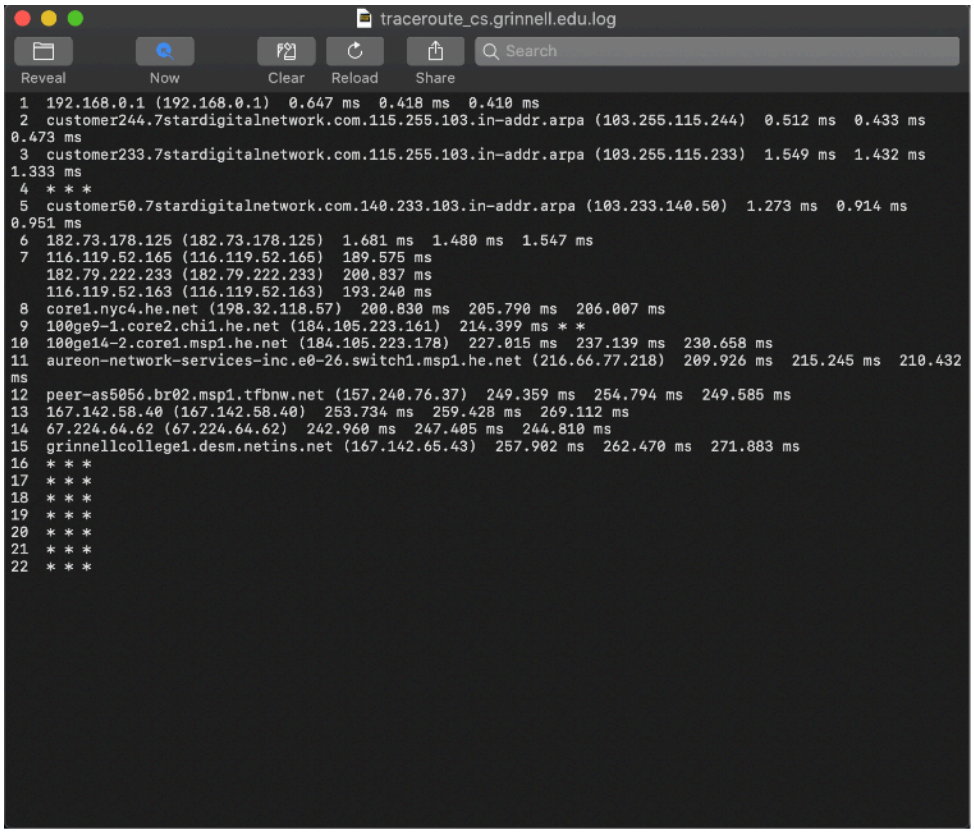
3. www.cs.grinnell.edu
4. csail.mit.edu
5. cs.stanford.edu
6. cs.manchester.ac.uk

Store the output of each traceroute command in a separate file named `traceroute_HOSTNAME.log`, replacing `HOSTNAME` with the hostname for end-host you pinged

(e.g., `traceroute_ee.iitb.ac.in.log`).




traceroute_mscs.mu.edu.log
Reveal
Now
Clear
Reload
Share
Search
1 192.168.0.1 (192.168.0.1) 0.714 ms 0.366 ms 0.334 ms
2 customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244) 0.543 ms 0.457 ms
0.409 ms
3 customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233) 1.349 ms 1.348 ms
1.320 ms
4 * * *
5 customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50) 1.070 ms 0.774 ms
0.817 ms
6 182.73.178.125 (182.73.178.125) 4.061 ms 5.433 ms 7.515 ms
7 182.79.222.237 (182.79.222.237) 225.733 ms
aes-static-150.36.144.59.airtel.in (59.144.36.150) 210.767 ms
182.79.222.233 (182.79.222.233) 200.181 ms
8 core1.nyc4.he.net (198.32.118.57) 215.855 ms 209.339 ms 207.986 ms
9 * * *
10 * * *
11 r-222wwash-isp-ae6-3926.wiscnet.net (140.189.8.126) 242.057 ms 256.560 ms 241.576 ms
12 r-milwaukeeci-809-isp-ae3-0.wiscnet.net (140.189.8.230) 256.036 ms 240.673 ms 246.034 ms
13 marquetteuniv.site.wiscnet.net (216.56.1.202) 244.616 ms 257.046 ms 267.747 ms
14 134.48.10.27 (134.48.10.27) 220.121 ms 219.324 ms 221.840 ms
15 * * *
16 * * *
17 euclid.mscs.mu.edu (134.48.4.5) 228.047 ms 220.507 ms 220.201 ms


traceroute_cs.grinnell.edu.log
Reveal
Now
Clear
Reload
Share
Search
1 192.168.0.1 (192.168.0.1) 0.647 ms 0.418 ms 0.410 ms
2 customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244) 0.512 ms 0.433 ms
0.473 ms
3 customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233) 1.549 ms 1.432 ms
1.333 ms
4 * * *
5 customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50) 1.273 ms 0.914 ms
0.951 ms
6 182.73.178.125 (182.73.178.125) 1.681 ms 1.480 ms 1.547 ms
7 116.119.52.165 (116.119.52.165) 189.575 ms
182.79.222.233 (182.79.222.233) 200.837 ms
116.119.52.163 (116.119.52.163) 193.240 ms
8 core1.nyc4.he.net (198.32.118.57) 200.830 ms 205.790 ms 206.007 ms
9 100ge9-1.core2.chi1.he.net (184.105.223.161) 214.399 ms * *
10 100ge14-2.core1.msp1.he.net (184.105.223.178) 227.015 ms 237.139 ms 230.658 ms
11 aureon-network-services-inc.e0-26.switch1.msp1.he.net (216.66.77.218) 209.926 ms 215.245 ms 210.432
ms
12 peer-as5056.br02.msp1.tfbnw.net (157.240.76.37) 249.359 ms 254.794 ms 249.585 ms
13 167.142.58.40 (167.142.58.40) 253.734 ms 259.428 ms 269.112 ms
14 67.224.64.62 (67.224.64.62) 242.960 ms 247.405 ms 244.810 ms
15 grinnellcollege1.desm.netins.net (167.142.65.43) 257.902 ms 262.470 ms 271.883 ms
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *

traceroute_cs.manchester.ac.uk.log

Reveal Now Clear Reload Share Search

```
 1  192.168.0.1 (192.168.0.1)  0.789 ms  0.316 ms  0.365 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.503 ms  0.413 ms  0.512 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.710 ms  1.884 ms  1.966 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.254 ms  0.929 ms  0.915 ms
 6  182.73.178.125 (182.73.178.125)  1.488 ms  1.590 ms  1.503 ms
 7  182.79.154.0 (182.79.154.0)  134.977 ms
    182.79.134.223 (182.79.134.223)  141.208 ms
    182.79.146.216 (182.79.146.216)  131.452 ms
 8  ldn-b4-link.telia.net (62.115.162.232)  132.380 ms  134.894 ms  132.401 ms
 9  jisc-ic-345131-ldn-b4.c.telia.net (62.115.175.131)  135.344 ms  131.838 ms  130.825 ms
10  ae24.londhx-sbr1.ja.net (146.97.35.197)  131.900 ms  131.810 ms  130.987 ms
11  ae29.londpg-sbr2.ja.net (146.97.33.2)  132.163 ms  132.337 ms  132.465 ms
12  ae31.erdiss-sbr2.ja.net (146.97.33.22)  137.695 ms  136.692 ms  137.706 ms
13  ae29.manckh-sbr2.ja.net (146.97.33.42)  133.670 ms  133.702 ms  152.437 ms
14  ae23.mancrh-rbr1.ja.net (146.97.38.42)  140.485 ms  140.531 ms  140.507 ms
15  * universityofmanchester.ja.net (146.97.169.2)  139.386 ms *
16  130.88.249.194 (130.88.249.194)  140.509 ms  140.379 ms  140.357 ms
17  * * *
18  * * *
19  eps.its.man.ac.uk (130.88.101.49)  140.721 ms  138.059 ms  138.064 ms
```

traceroute_csail.mit.edu.log

Reveal Now Clear Reload Share Search

```
 1  192.168.0.1 (192.168.0.1)  0.631 ms  0.328 ms  0.339 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.589 ms  0.466 ms  0.553 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.568 ms  1.339 ms  1.327 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.039 ms  1.029 ms  0.919 ms
 6  182.73.178.125 (182.73.178.125)  1.537 ms  1.623 ms  1.586 ms
 7  182.79.255.11 (182.79.255.11)  257.001 ms
    182.79.243.33 (182.79.243.33)  260.872 ms
    182.79.201.102 (182.79.201.102)  235.399 ms
 8  ae58.edge1.losangeles6.level3.net (4.26.0.17)  263.467 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  238.449 ms
    xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  234.300 ms
 9  * * *
10  massachuset.bear1.boston1.level3.net (4.53.48.98)  302.880 ms  302.649 ms  301.356 ms
11  dmz-rtr-1-external-rtr-1.mit.edu (18.0.161.17)  304.863 ms  295.778 ms  301.417 ms
12  dmz-rtr-2-dmz-rtr-1-1.mit.edu (18.0.161.6)  307.175 ms
    dmz-rtr-2-dmz-rtr-1-2.mit.edu (18.0.162.6)  308.791 ms
    dmz-rtr-2-dmz-rtr-1-1.mit.edu (18.0.161.6)  299.056 ms
13  mitnet.core-1-ext.csail.mit.edu (18.4.7.65)  289.822 ms  298.413 ms  294.555 ms
14  * * core-1-ext.bdr.csail.mit.edu (128.30.13.26)  295.978 ms
15  bdr.core-1.csail.mit.edu (128.30.0.246)  312.579 ms  305.461 ms  309.008 ms
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  * * *
46  * * *
47  * * *
48  * * *
49  * * *
50  * * *
51  * * *
52  * * *
```

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.

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30  math.hws.edu
traceroute to math.hws.edu (64.89.144.237), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  4.880 ms  0.268 ms  0.365 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.579 ms  0.528 ms  0.558 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.845 ms  1.375 ms  1.823 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.912 ms  0.962 ms  1.171 ms
 6  182.73.178.125 (182.73.178.125)  1.542 ms  1.550 ms  1.922 ms
 7  182.79.245.81 (182.79.245.81)  234.760 ms
    182.79.243.35 (182.79.243.35)  220.671 ms
    182.79.247.94 (182.79.247.94)  220.986 ms
 8  xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  230.611 ms
    xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  246.739 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  230.759 ms
 9  * * *
10  * * *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  249.036 ms  248.882 ms  256.465 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  263.665 ms  251.154 ms  252.032 ms
13  64.89.144.100 (64.89.144.100)  257.683 ms  259.659 ms  264.156 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30 www.hws.edu
traceroute to www.hws.edu (64.89.145.159), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  0.833 ms  0.549 ms  0.538 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.537 ms  0.513 ms  0.427 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.401 ms  1.631 ms  1.393 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.186 ms  1.052 ms  1.245 ms
 6  182.73.178.125 (182.73.178.125)  1.633 ms  1.654 ms  1.561 ms
 7  182.79.234.217 (182.79.234.217)  226.716 ms
    182.79.243.35 (182.79.243.35)  221.806 ms
    182.79.243.31 (182.79.243.31)  233.940 ms
 8  ae58.edge1.losangeles6.level3.net (4.26.0.17)  237.713 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  231.872 ms  230.667 ms
 9  * * *
10  gblx-level3-400g.losangeles1.level3.net (4.68.73.189)  247.193 ms  234.882 ms *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  255.871 ms  250.035 ms  245.653 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  258.513 ms  251.061 ms  254.247 ms
13  64.89.144.100 (64.89.144.100)  248.724 ms  264.281 ms  252.086 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

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.

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30 www.hws.edu
traceroute to www.hws.edu (64.89.145.159), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  0.812 ms  0.311 ms  0.269 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.633 ms  0.393 ms  0.326 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.687 ms  1.380 ms  1.398 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.319 ms  1.080 ms  1.009 ms
 6  182.73.178.125 (182.73.178.125)  1.525 ms  1.750 ms  1.615 ms
 7  182.79.245.81 (182.79.245.81)  222.751 ms
    182.79.255.9 (182.79.255.9)  242.492 ms
    116.119.35.8 (116.119.35.8)  238.984 ms
 8  xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  234.418 ms
    ae58.edge1.losangeles6.level3.net (4.26.0.17)  231.846 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  243.423 ms
 9  ae-2-52.ear3.losangeles1.level3.net (4.69.207.49)  237.384 ms  238.443 ms *
10  * * *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  244.552 ms  254.687 ms  249.019 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  263.669 ms  258.138 ms  251.977 ms
13  64.89.144.100 (64.89.144.100)  258.454 ms  259.370 ms  252.725 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30  math.hws.edu
traceroute to math.hws.edu (64.89.144.237), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  0.982 ms  0.242 ms  0.337 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.499 ms  0.314 ms  0.456 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.297 ms  1.268 ms  1.282 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.326 ms  1.178 ms  0.914 ms
 6  182.73.178.125 (182.73.178.125)  1.583 ms  1.612 ms  1.746 ms
 7  182.79.152.225 (182.79.152.225)  234.330 ms
    182.79.245.145 (182.79.245.145)  227.764 ms
    182.79.201.106 (182.79.201.106)  260.857 ms
 8  ae58.edge1.losangeles6.level3.net (4.26.0.17)  265.358 ms
    xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  234.508 ms
    ae58.edge1.losangeles6.level3.net (4.26.0.17)  238.544 ms
 9  * * *
10  * * *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  249.188 ms  261.227 ms  250.049 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  259.551 ms  258.933 ms  253.396 ms
13  64.89.144.100 (64.89.144.100)  264.243 ms  251.955 ms  252.296 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30  math.hws.edu
traceroute to math.hws.edu (64.89.144.237), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  4.880 ms  0.268 ms  0.365 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.579 ms  0.528 ms  0.558 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.845 ms  1.375 ms  1.823 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.912 ms  0.962 ms  1.171 ms
 6  182.73.178.125 (182.73.178.125)  1.542 ms  1.550 ms  1.922 ms
 7  182.79.245.81 (182.79.245.81)  234.760 ms
    182.79.243.35 (182.79.243.35)  220.671 ms
    182.79.247.94 (182.79.247.94)  220.986 ms
 8  xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  230.611 ms
    xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  246.739 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  230.759 ms
 9  * * *
10  * * *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  249.036 ms  248.882 ms  256.465 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  263.665 ms  251.154 ms  252.032 ms
13  64.89.144.100 (64.89.144.100)  257.683 ms  259.659 ms  264.156 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30 www.hws.edu
traceroute to www.hws.edu (64.89.145.159), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  0.833 ms  0.549 ms  0.538 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.537 ms  0.513 ms  0.427 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.401 ms  1.631 ms  1.393 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.186 ms  1.052 ms  1.245 ms
 6  182.73.178.125 (182.73.178.125)  1.633 ms  1.654 ms  1.561 ms
 7  182.79.234.217 (182.79.234.217)  226.716 ms
    182.79.243.35 (182.79.243.35)  221.806 ms
    182.79.243.31 (182.79.243.31)  233.940 ms
 8  ae58.edge1.losangeles6.level3.net (4.26.0.17)  237.713 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  231.872 ms  230.667 ms
 9  * * *
10  gblx-level3-400g.losangeles1.level3.net (4.68.73.189)  247.193 ms  234.882 ms *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  255.871 ms  250.035 ms  245.653 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  258.513 ms  251.061 ms  254.247 ms
13  64.89.144.100 (64.89.144.100)  248.724 ms  264.281 ms  252.086 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30 www.hws.edu
traceroute to www.hws.edu (64.89.145.159), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  0.812 ms  0.311 ms  0.269 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.633 ms  0.393 ms  0.326 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.687 ms  1.380 ms  1.398 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.319 ms  1.080 ms  1.009 ms
 6  182.73.178.125 (182.73.178.125)  1.525 ms  1.750 ms  1.615 ms
 7  182.79.245.81 (182.79.245.81)  222.751 ms
    182.79.255.9 (182.79.255.9)  242.492 ms
    116.119.35.8 (116.119.35.8)  238.984 ms
 8  xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  234.418 ms
    ae58.edge1.losangeles6.level3.net (4.26.0.17)  231.846 ms
    xe-9-1-0.edge1.losangeles6.level3.net (4.26.0.61)  243.423 ms
 9  ae-2-52.ear3.losangeles1.level3.net (4.69.207.49)  237.384 ms  238.443 ms *
10  * * *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  244.552 ms  254.687 ms  249.019 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  263.669 ms  258.138 ms  251.977 ms
13  64.89.144.100 (64.89.144.100)  258.454 ms  259.370 ms  252.725 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

```
(base) Rishis-iMac:Desktop rishikaul$ traceroute -m30  math.hws.edu
traceroute to math.hws.edu (64.89.144.237), 30 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  0.982 ms  0.242 ms  0.337 ms
 2  customer244.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.244)  0.499 ms  0.314 ms  0.456 ms
 3  customer233.7stardigitalnetwork.com.115.255.103.in-addr.arpa (103.255.115.233)  1.297 ms  1.268 ms  1.282 ms
 4  * * *
 5  customer50.7stardigitalnetwork.com.140.233.103.in-addr.arpa (103.233.140.50)  1.326 ms  1.178 ms  0.914 ms
 6  182.73.178.125 (182.73.178.125)  1.583 ms  1.612 ms  1.746 ms
 7  182.79.152.225 (182.79.152.225)  234.330 ms
    182.79.245.145 (182.79.245.145)  227.764 ms
    182.79.201.106 (182.79.201.106)  260.857 ms
 8  ae58.edge1.losangeles6.level3.net (4.26.0.17)  265.358 ms
    xe-5-1-0.edge1.losangeles6.level3.net (4.26.0.89)  234.508 ms
    ae58.edge1.losangeles6.level3.net (4.26.0.17)  238.544 ms
 9  * * *
10  * * *
11  roc1-ar5-xe-0-0-0-0.us.twtelecom.net (35.248.1.158)  249.188 ms  261.227 ms  250.049 ms
12  66-195-65-170.static.ctl.one (66.195.65.170)  259.551 ms  258.933 ms  253.396 ms
13  64.89.144.100 (64.89.144.100)  264.243 ms  251.955 ms  252.296 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

From the above experiments, I can conclude that for the same source and same destination, the packets sent at different times have different RTT's and take different paths through the net.

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 path to my ISP is always the same.

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

   Yes there is, larger the distance larger is the number of nodes.

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?

Whois — The *whois* command can give detailed information about domain names and IP addresses. If it is not installed on the computers then install it with command sudo apt-get install whois in. *Whois* can tell you what organization owns or is responsible for the name or

address and where to contact them. It often includes a list of domain name servers for the organization.

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

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

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

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

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

curl ipinfo.io/129.64.99.200

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

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

References : StackOverflow, Wikipedia, StackPath