CEL 51, DCCN, Monsoon 2020
Lab 2: Basic Network Utilities
Viraj Shah 2018130063 Batch D

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

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

Some Basic command line Networking utilities

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

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

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

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

The syntax in Windows is:

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

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

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

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

EXPERIMENTS WITH PING

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

   Pinging google with 74 byte packet

```
C:\Users\DARSHIL>ping -n 10 -l 74 www.google.com

Pinging www.google.com [172.217.166.68] with 74 bytes of data:
Reply from 172.217.166.68: bytes=68 (sent 74) time=251ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=8ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=4ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=4ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=3ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=6ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=14ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=3ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=4ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 74) time=5ms TTL=119

Ping statistics for 172.217.166.68:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 251ms, Average = 30ms
```

Pinging google with 100 byte packet

```
C:\Users\DARSHIL>ping -n 10 -l 100 www.google.com

Pinging www.google.com [216.58.203.36] with 100 bytes of data:
Reply from 216.58.203.36: bytes=68 (sent 100) time=90ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=3ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=8ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=4ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=63ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=3ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=4ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=4ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=5ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 100) time=6ms TTL=119

Ping statistics for 216.58.203.36:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 90ms, Average = 19ms
```

Pinging google with 500 byte packet

```
C:\Users\DARSHIL>ping -n 10 -l 500 www.google.com

Pinging www.google.com [216.58.203.36] with 500 bytes of data:
Reply from 216.58.203.36: bytes=68 (sent 500) time=5ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=3ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=14ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=8ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=5ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=6ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=6ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=5ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=12ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 500) time=3ms TTL=119

Ping statistics for 216.58.203.36:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 14ms, Average = 6ms
```

Pinging google with 1000 byte packet

```
C:\Users\DARSHIL>ping -n 10 -l 1000 www.google.com

Pinging www.google.com [216.58.203.36] with 1000 bytes of data:
Reply from 216.58.203.36: bytes=68 (sent 1000) time=4ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=3ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=3ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=4ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=7ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=5ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=5ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=8ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=5ms TTL=119
Reply from 216.58.203.36: bytes=68 (sent 1000) time=4ms TTL=119

Ping statistics for 216.58.203.36:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 8ms, Average = 4ms
```

Pinging google with 1400 byte packet

```
C:\Users\DARSHIL>ping -n 10 -l 1400 www.google.com

Pinging www.google.com [172.217.166.68] with 1400 bytes of data:
Reply from 172.217.166.68: bytes=68 (sent 1400) time=4ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=10ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=3ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=3ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=3ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=15ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=23ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=3ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=4ms TTL=119
Reply from 172.217.166.68: bytes=68 (sent 1400) time=5ms TTL=119

Ping statistics for 172.217.166.68:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 23ms, Average = 7ms
```

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. The delays are:

   1. Processing delay – time it takes a router to process the packet header, depends on the processing speed of the switch
   2. Queuing delay – time the packet spends in routing queues depends on the number of packets, size of the packet and bandwidth
   3. 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.
   4. Propagation delay – time for a signal to reach its destination depends on distance and propagation speed.

   So, Yes there is a difference definitely in the average RTT between different hosts which we can see from the above examples

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 delays mentioned above increases there are some which 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).

Pinging www.uw.edu with 64 bytes.

```
C:\Users\DARSHIL>ping -n 10 -l 64 www.uw.edu

Pinging www.washington.edu [128.95.155.198] with 64 bytes of data:
Reply from 128.95.155.198: bytes=64 time=307ms TTL=47
Reply from 128.95.155.198: bytes=64 time=242ms TTL=47
Reply from 128.95.155.198: bytes=64 time=243ms TTL=47
Reply from 128.95.155.198: bytes=64 time=249ms TTL=47
Reply from 128.95.155.198: bytes=64 time=242ms TTL=47
Reply from 128.95.155.198: bytes=64 time=244ms TTL=47
Reply from 128.95.155.198: bytes=64 time=254ms TTL=47
Reply from 128.95.155.198: bytes=64 time=242ms TTL=47
Reply from 128.95.155.198: bytes=64 time=245ms TTL=47
Reply from 128.95.155.198: bytes=64 time=243ms TTL=47

Ping statistics for 128.95.155.198:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 242ms, Maximum = 307ms, Average = 251ms
```

Pinging www.berkeley.edu with 64 bytes.

```
C:\Users\DARSHIL>ping -n 10 -l 64 www.berkeley.edu

Pinging www-production-1113102805.us-west-2.elb.amazonaws.com [52.88.59.144] with 64 bytes of data:
Reply from 52.88.59.144: bytes=64 time=315ms TTL=228
Reply from 52.88.59.144: bytes=64 time=261ms TTL=228
Reply from 52.88.59.144: bytes=64 time=253ms TTL=228
Reply from 52.88.59.144: bytes=64 time=253ms TTL=228
Reply from 52.88.59.144: bytes=64 time=254ms TTL=228
Reply from 52.88.59.144: bytes=64 time=255ms TTL=228
Reply from 52.88.59.144: bytes=64 time=252ms TTL=228
Reply from 52.88.59.144: bytes=64 time=253ms TTL=228
Reply from 52.88.59.144: bytes=64 time=254ms TTL=228
Reply from 52.88.59.144: bytes=64 time=254ms TTL=228

Ping statistics for 52.88.59.144:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 252ms, Maximum = 315ms, Average = 260ms
```

Pinging www.uchicago.edu with 64 bytes.

```
C:\Users\DARSHIL>ping -n 10 -l 64 www.uchicago.edu

Pinging wsee2.elb.uchicago.edu [54.89.29.50] with 64 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 54.89.29.50:
    Packets: Sent = 10, Received = 0, Lost = 10 (100% loss),
```

Pinging www.ox.ac.uk with 64 bytes.

```
C:\Users\DARSHIL>ping -n 10 -l 64 www.ox.ac.uk

Pinging www.ox.ac.uk [151.101.2.133] with 64 bytes of data:
Reply from 151.101.2.133: bytes=64 time=61ms TTL=58
Reply from 151.101.2.133: bytes=64 time=4ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58
Reply from 151.101.2.133: bytes=64 time=3ms TTL=58

Ping statistics for 151.101.2.133:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 61ms, Average = 8ms
```

Pinging www.u-tokyo.ac.jp with 64 bytes.

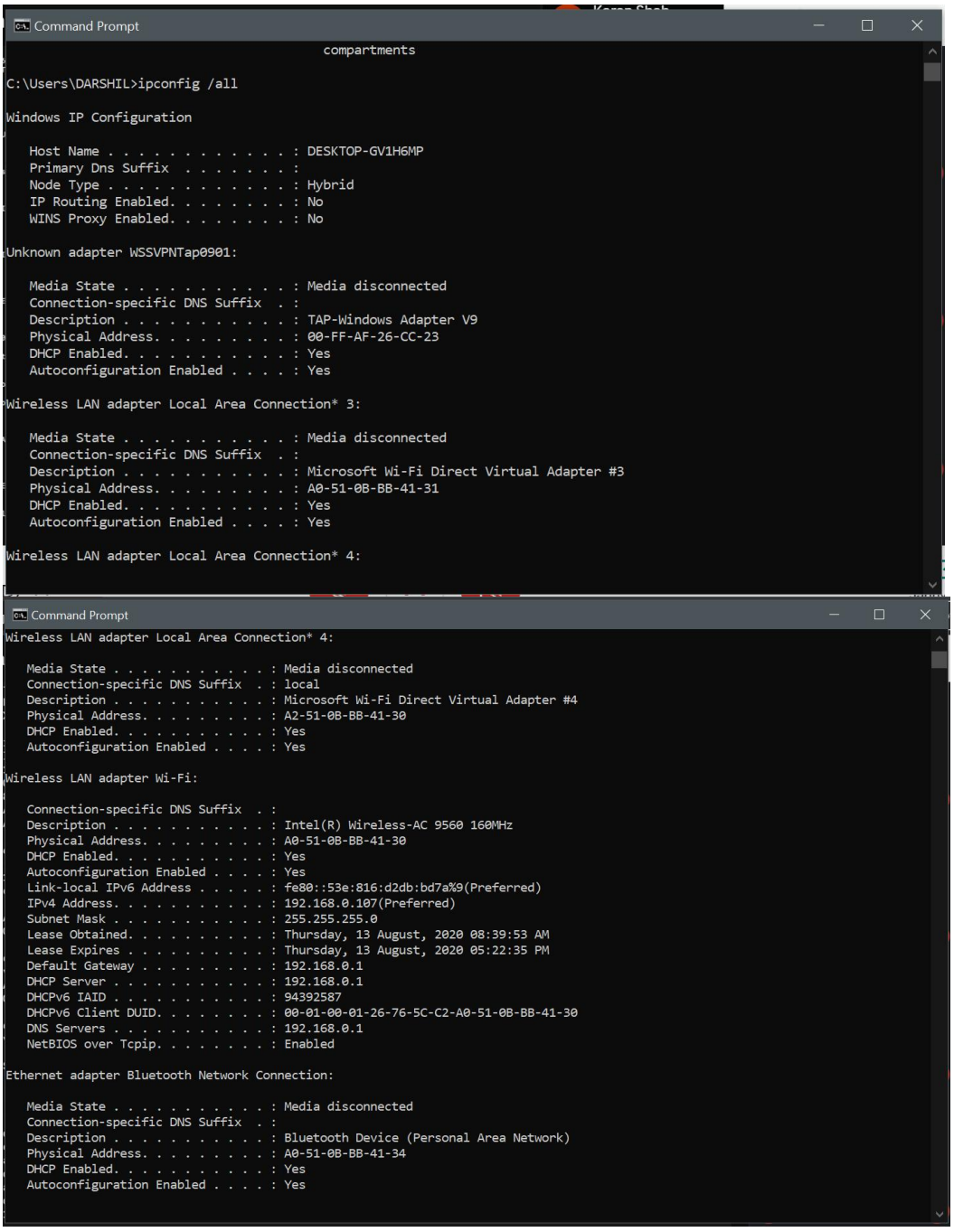
```
C:\Users\DARSHIL>ping -n 10 -l 64 www.u-tokyo.ac.jp

Pinging www.u-tokyo.ac.jp [210.152.243.234] with 64 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 210.152.243.234:
    Packets: Sent = 10, Received = 0, Lost = 10 (100% loss),
```

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>

Performing nslookup on google.com

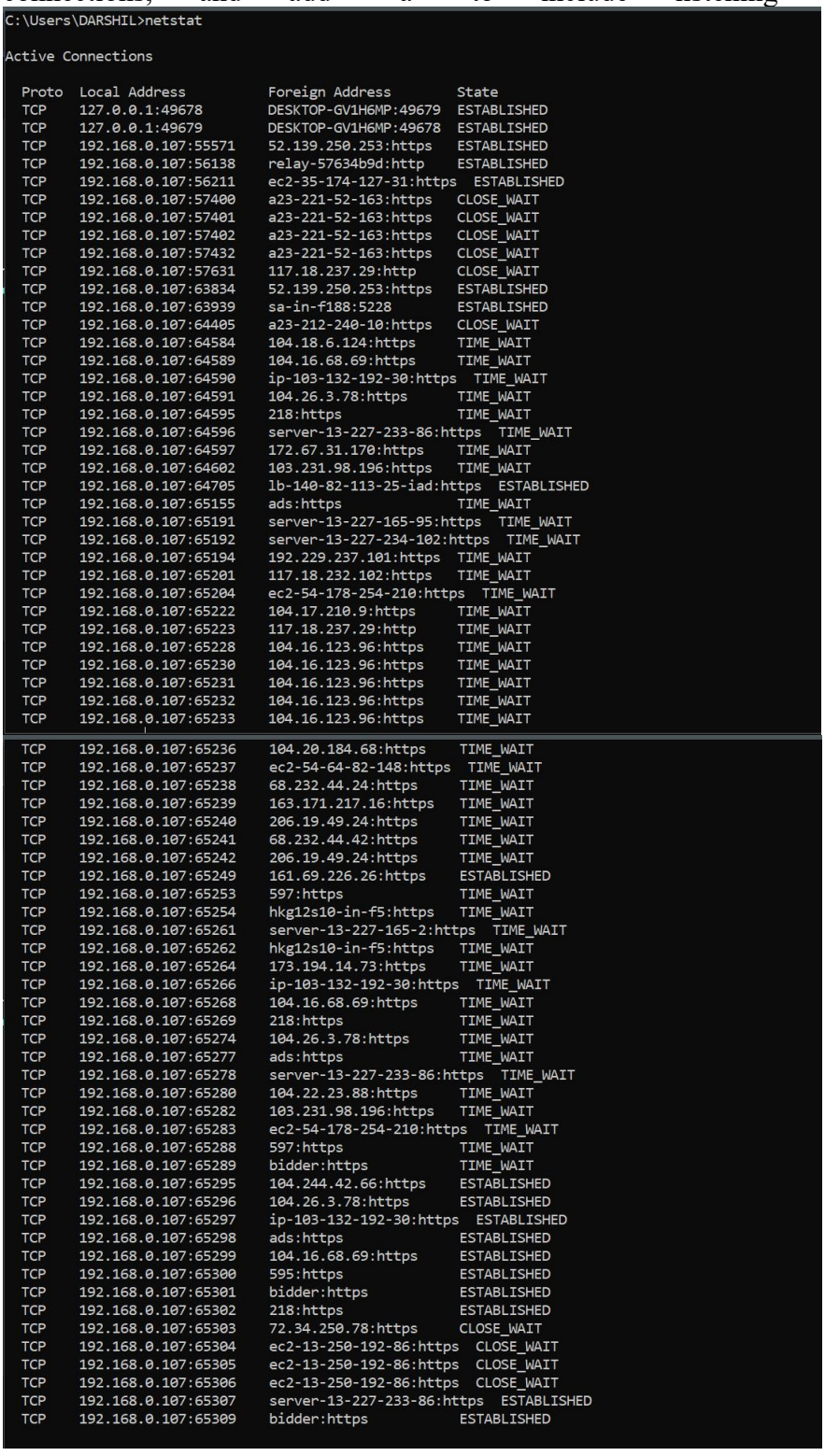
```
C:\Users\DARSHIL>nslookup www.google.com
Server:  UnKnown
Address:  192.168.0.1

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4009:811::2004
          142.250.67.132
```

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!!!.)



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.)



```
C:\Users\DARSHIL>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49678        DESKTOP-GV1H6MP:49679  ESTABLISHED
  TCP    127.0.0.1:49679        DESKTOP-GV1H6MP:49678  ESTABLISHED
  TCP    192.168.0.107:55571    52.139.250.253:https   ESTABLISHED
  TCP    192.168.0.107:56138    relay-57634b9d:http    ESTABLISHED
  TCP    192.168.0.107:56211    ec2-35-174-127-31:https  ESTABLISHED
  TCP    192.168.0.107:57400    a23-221-52-163:https   CLOSE_WAIT
  TCP    192.168.0.107:57401    a23-221-52-163:https   CLOSE_WAIT
  TCP    192.168.0.107:57402    a23-221-52-163:https   CLOSE_WAIT
  TCP    192.168.0.107:57432    a23-221-52-163:https   CLOSE_WAIT
  TCP    192.168.0.107:57631    117.18.237.29:http     CLOSE_WAIT
  TCP    192.168.0.107:63834    52.139.250.253:https   ESTABLISHED
  TCP    192.168.0.107:63939    sa-in-f188:5228        ESTABLISHED
  TCP    192.168.0.107:64405    a23-212-240-10:https   CLOSE_WAIT
  TCP    192.168.0.107:64584    104.18.6.124:https     TIME_WAIT
  TCP    192.168.0.107:64589    104.16.68.69:https     TIME_WAIT
  TCP    192.168.0.107:64590    ip-103-132-192-30:https  TIME_WAIT
  TCP    192.168.0.107:64591    104.26.3.78:https      TIME_WAIT
  TCP    192.168.0.107:64595    218:https              TIME_WAIT
  TCP    192.168.0.107:64596    server-13-227-233-86:https  TIME_WAIT
  TCP    192.168.0.107:64597    172.67.31.170:https    TIME_WAIT
  TCP    192.168.0.107:64602    103.231.98.196:https   TIME_WAIT
  TCP    192.168.0.107:64705    lb-140-82-113-25-iad:https  ESTABLISHED
  TCP    192.168.0.107:65155    ads:https              TIME_WAIT
  TCP    192.168.0.107:65191    server-13-227-165-95:https  TIME_WAIT
  TCP    192.168.0.107:65192    server-13-227-234-102:https  TIME_WAIT
  TCP    192.168.0.107:65194    192.229.237.101:https  TIME_WAIT
  TCP    192.168.0.107:65201    117.18.232.102:https   TIME_WAIT
  TCP    192.168.0.107:65204    ec2-54-178-254-210:https  TIME_WAIT
  TCP    192.168.0.107:65222    104.17.210.9:https     TIME_WAIT
  TCP    192.168.0.107:65223    117.18.237.29:http     TIME_WAIT
  TCP    192.168.0.107:65228    104.16.123.96:https    TIME_WAIT
  TCP    192.168.0.107:65230    104.16.123.96:https    TIME_WAIT
  TCP    192.168.0.107:65231    104.16.123.96:https    TIME_WAIT
  TCP    192.168.0.107:65232    104.16.123.96:https    TIME_WAIT
  TCP    192.168.0.107:65233    104.16.123.96:https    TIME_WAIT
  TCP    192.168.0.107:65236    104.20.184.68:https    TIME_WAIT
  TCP    192.168.0.107:65237    ec2-54-64-82-148:https  TIME_WAIT
  TCP    192.168.0.107:65238    68.232.44.24:https     TIME_WAIT
  TCP    192.168.0.107:65239    163.171.217.16:https   TIME_WAIT
  TCP    192.168.0.107:65240    206.19.49.24:https     TIME_WAIT
  TCP    192.168.0.107:65241    68.232.44.42:https     TIME_WAIT
  TCP    192.168.0.107:65242    206.19.49.24:https     TIME_WAIT
  TCP    192.168.0.107:65249    161.69.226.26:https    ESTABLISHED
  TCP    192.168.0.107:65253    597:https              TIME_WAIT
  TCP    192.168.0.107:65254    hkg12s10-in-f5:https   TIME_WAIT
  TCP    192.168.0.107:65261    server-13-227-165-2:https  TIME_WAIT
  TCP    192.168.0.107:65262    hkg12s10-in-f5:https   TIME_WAIT
  TCP    192.168.0.107:65264    173.194.14.73:https    TIME_WAIT
  TCP    192.168.0.107:65266    ip-103-132-192-30:https  TIME_WAIT
  TCP    192.168.0.107:65268    104.16.68.69:https     TIME_WAIT
  TCP    192.168.0.107:65269    218:https              TIME_WAIT
  TCP    192.168.0.107:65274    104.26.3.78:https      TIME_WAIT
  TCP    192.168.0.107:65277    ads:https              TIME_WAIT
  TCP    192.168.0.107:65278    server-13-227-233-86:https  TIME_WAIT
  TCP    192.168.0.107:65280    104.22.23.88:https     TIME_WAIT
  TCP    192.168.0.107:65282    103.231.98.196:https   TIME_WAIT
  TCP    192.168.0.107:65283    ec2-54-178-254-210:https  TIME_WAIT
  TCP    192.168.0.107:65288    597:https              TIME_WAIT
  TCP    192.168.0.107:65289    bidder:https           TIME_WAIT
  TCP    192.168.0.107:65295    104.244.42.66:https    ESTABLISHED
  TCP    192.168.0.107:65296    104.26.3.78:https      ESTABLISHED
  TCP    192.168.0.107:65297    ip-103-132-192-30:https  ESTABLISHED
  TCP    192.168.0.107:65298    ads:https              ESTABLISHED
  TCP    192.168.0.107:65299    104.16.68.69:https     ESTABLISHED
  TCP    192.168.0.107:65300    595:https              ESTABLISHED
  TCP    192.168.0.107:65301    bidder:https           ESTABLISHED
  TCP    192.168.0.107:65302    218:https              ESTABLISHED
  TCP    192.168.0.107:65303    72.34.250.78:https     CLOSE_WAIT
  TCP    192.168.0.107:65304    ec2-13-250-192-86:https  CLOSE_WAIT
  TCP    192.168.0.107:65305    ec2-13-250-192-86:https  CLOSE_WAIT
  TCP    192.168.0.107:65306    ec2-13-250-192-86:https  CLOSE_WAIT
  TCP    192.168.0.107:65307    server-13-227-233-86:https  ESTABLISHED
  TCP    192.168.0.107:65309    bidder:https           ESTABLISHED
```

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>
```

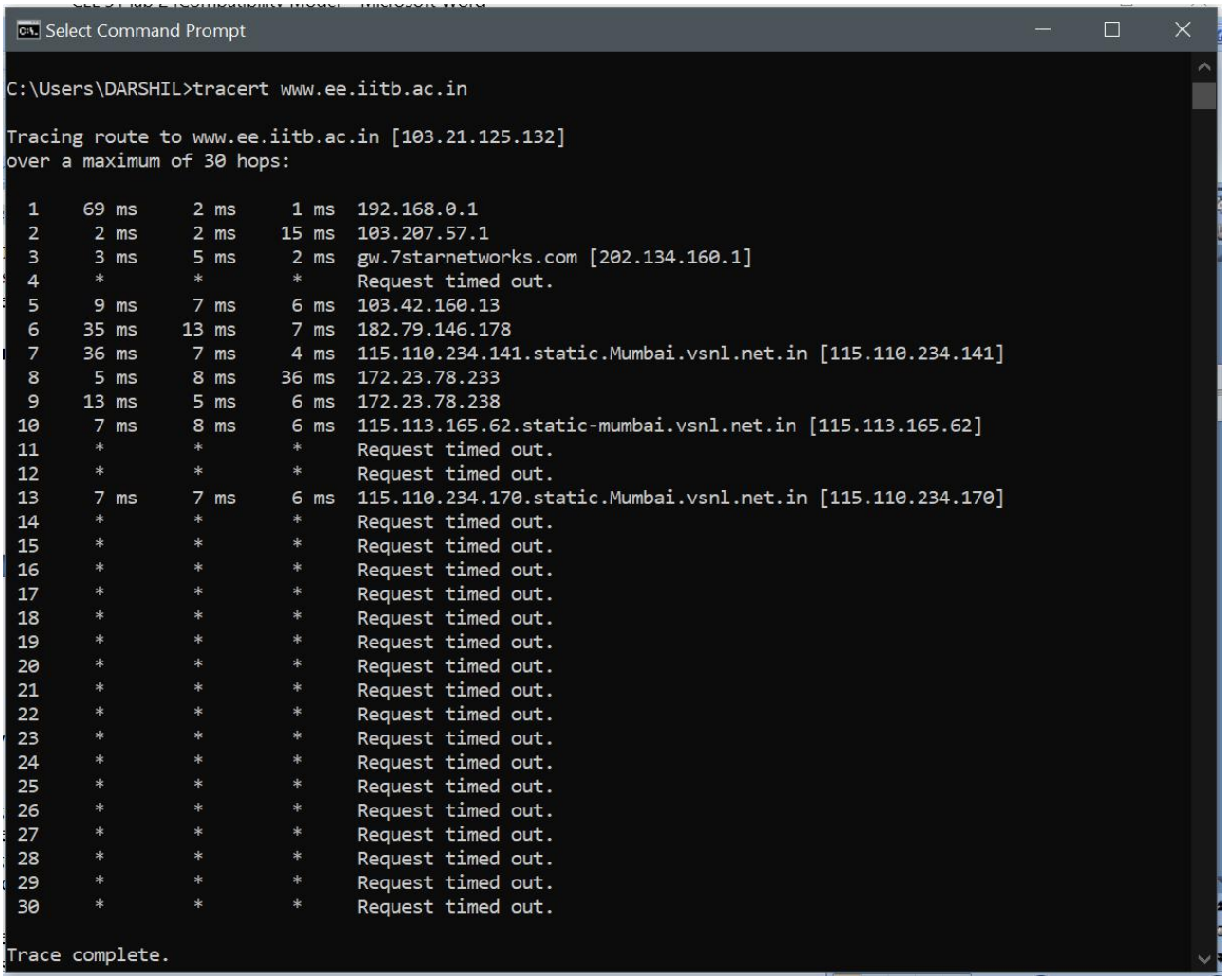
You can specify either a hostname (e.g., cs.iitb.ac.in) or an IP address (e.g., 128.105.2.6).

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

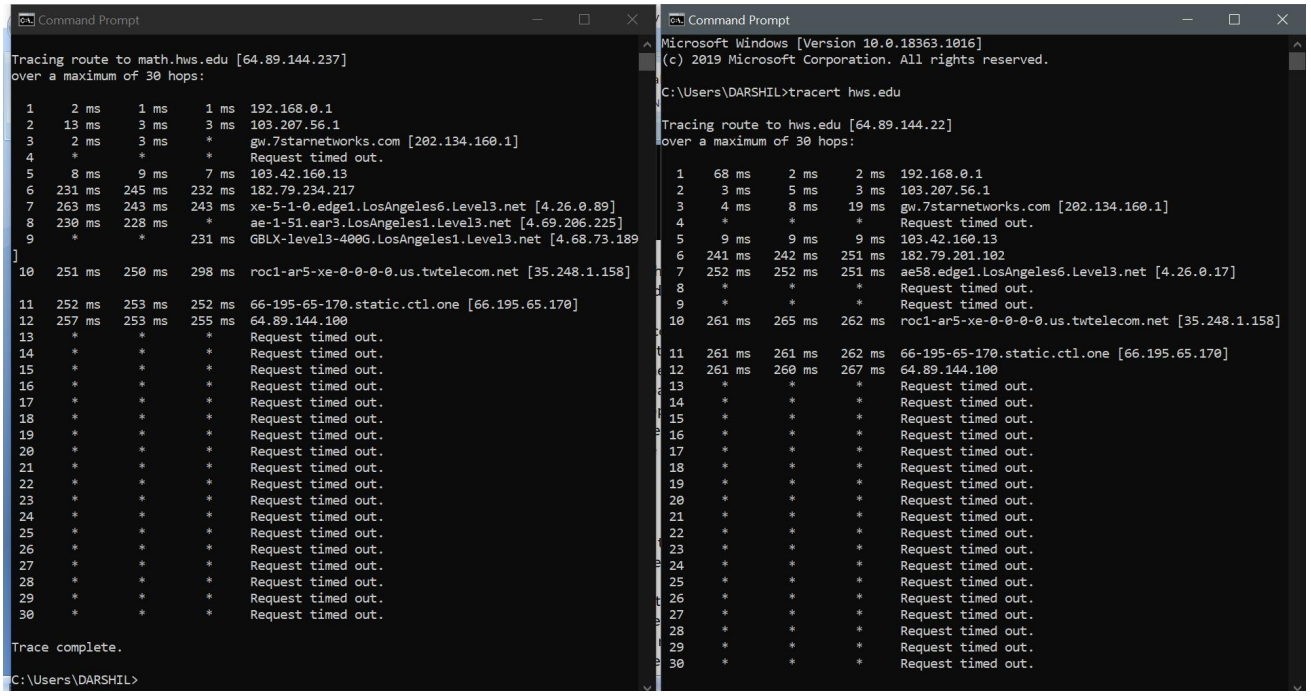
Store the output of each traceroute command in a separate file named traceroute_HOSTNAME.log, replacing HOSTNAME with the hostname for end-host you pinged


Select Command Prompt
C:\Users\DARSHIL>tracert www.ee.iitb.ac.in
Tracing route to www.ee.iitb.ac.in [103.21.125.132]
over a maximum of 30 hops:
1 69 ms 2 ms 1 ms 192.168.0.1
2 2 ms 2 ms 15 ms 103.207.57.1
3 3 ms 5 ms 2 ms gw.7starnetworks.com [202.134.160.1]
4 * * * Request timed out.
5 9 ms 7 ms 6 ms 103.42.160.13
6 35 ms 13 ms 7 ms 182.79.146.178
7 36 ms 7 ms 4 ms 115.110.234.141.static.Mumbai.vsnl.net.in [115.110.234.141]
8 5 ms 8 ms 36 ms 172.23.78.233
9 13 ms 5 ms 6 ms 172.23.78.238
10 7 ms 8 ms 6 ms 115.113.165.62.static-mumbai.vsnl.net.in [115.113.165.62]
11 * * * Request timed out.
12 * * * Request timed out.
13 7 ms 7 ms 6 ms 115.110.234.170.static.Mumbai.vsnl.net.in [115.110.234.170]
14 * * * Request timed out.
15 * * * Request timed out.
16 * * * Request timed out.
17 * * * Request timed out.
18 * * * Request timed out.
19 * * * Request timed out.
20 * * * Request timed out.
21 * * * Request timed out.
22 * * * Request timed out.
23 * * * Request timed out.
24 * * * Request timed out.
25 * * * Request timed out.
26 * * * Request timed out.
27 * * * Request timed out.
28 * * * Request timed out.
29 * * * Request timed out.
30 * * * Request timed out.
Trace complete.

```
C:\Users\DARSHIL>tracert www.spit.ac.in

Tracing route to www.spit.ac.in [43.252.193.19]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  192.168.0.1
  2     2 ms     2 ms     2 ms  103.207.57.1
  3     3 ms     *        2 ms  gw.7starnetworks.com [202.134.160.1]
  4     2 ms     3 ms     1 ms  10.2.10.34
  5     2 ms     2 ms     2 ms  103.243.114.197
  6     3 ms     2 ms     2 ms  as17625.bom.extreme-ix.net [103.77.108.156]
  7     3 ms     2 ms     2 ms  27.109.1.150
  8     2 ms     9 ms     2 ms  103.205.124.82
  9     3 ms     2 ms     2 ms  43.252.192.230
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24     *        *        *     Request timed out.
 25     *        *        *     Request timed out.
 26     *        *        *     Request timed out.
 27     *        *        *     Request timed out.
 28     *        *        *     Request timed out.
 29     *        *        *     Request timed out.
 30     *        *        *     Request timed out.

Trace complete.
```

```
C:\Users\DARSHIL>tracert www.mscs.mu.edu

Tracing route to turing.mscs.mu.edu [134.48.4.34]
over a maximum of 30 hops:

  1   101 ms    13 ms    63 ms  192.168.0.1
  2    12 ms     6 ms     5 ms  103.207.57.1
  3    24 ms     5 ms     8 ms  gw.7starnetworks.com [202.134.160.1]
  4     *        *        *     Request timed out.
  5     7 ms     9 ms    10 ms  103.42.160.13
  6   194 ms   189 ms   188 ms  116.119.52.163
  7   202 ms   204 ms   206 ms  core1.nyc4.he.net [198.32.118.57]
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10   257 ms   245 ms   301 ms  r-222wwash-isp-ae6-3926.wiscnet.net [140.189.8.126]
 11   244 ms   245 ms   242 ms  r-milwaukeeci-809-isp-ae3-0.wiscnet.net [140.189.8.230]
 12   251 ms   253 ms   245 ms  MarquetteUniv.site.wiscnet.net [216.56.1.202]
 13   247 ms   247 ms   249 ms  134.48.10.27
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24     *        *        *     Request timed out.
 25     *        *        *     Request timed out.
 26     *        *        *     Request timed out.
 27     *        *        *     Request timed out.
 28     *        *        *     Request timed out.
 29     *        *        *     Request timed out.
 30     *        *        *     Request timed out.

Trace complete.
```

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.

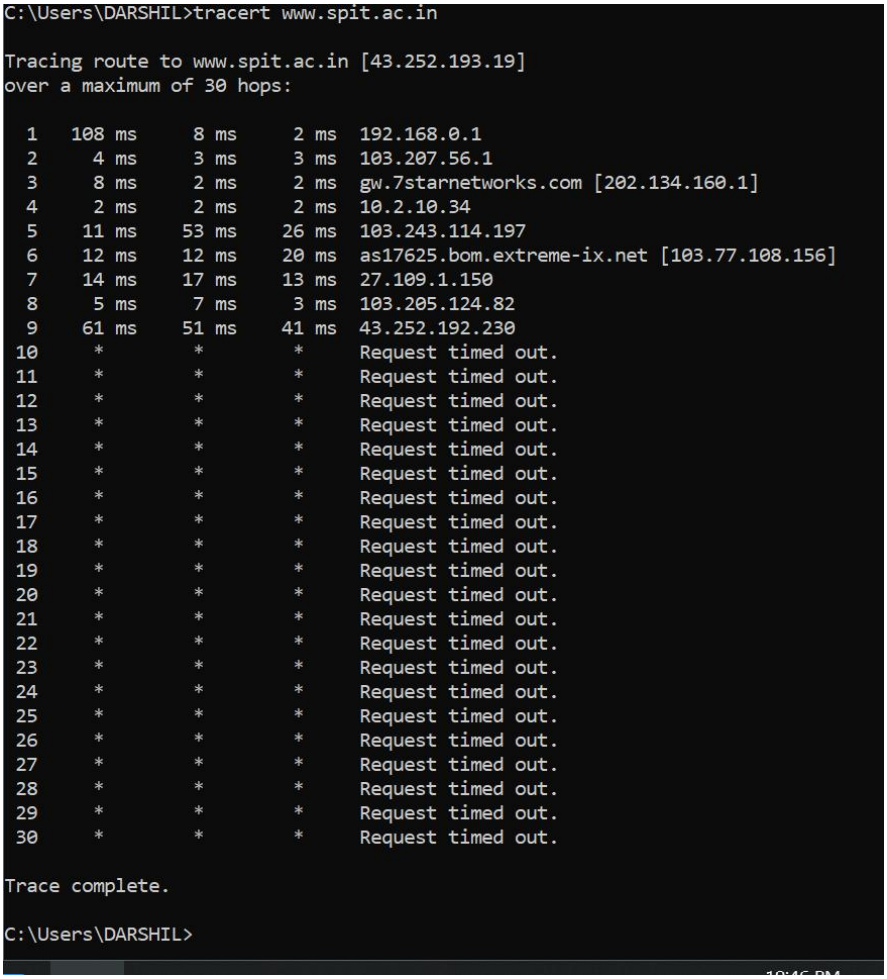


The only difference spotted in the tracert of both the websites is a slight time difference and some paths chosen were also different in the start hops before the status started showing Request timed out appeared after which the response is somewhat same.

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.

```
C:\Users\DARSHIL>tracert www.spit.ac.in

Tracing route to www.spit.ac.in [43.252.193.19]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  192.168.0.1
  2     2 ms     2 ms     2 ms  103.207.57.1
  3     3 ms     *        2 ms  gw.7starnetworks.com [202.134.160.1]
  4     2 ms     3 ms     1 ms  10.2.10.34
  5     2 ms     2 ms     2 ms  103.243.114.197
  6     3 ms     2 ms     2 ms  as17625.bom.extreme-ix.net [103.77.108.156]
  7     3 ms     2 ms     2 ms  27.109.1.150
  8     2 ms     9 ms     2 ms  103.205.124.82
  9     3 ms     2 ms     2 ms  43.252.192.230
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24     *        *        *     Request timed out.
 25     *        *        *     Request timed out.
 26     *        *        *     Request timed out.
 27     *        *        *     Request timed out.
 28     *        *        *     Request timed out.
 29     *        *        *     Request timed out.
 30     *        *        *     Request timed out.

Trace complete.
```

```
C:\Users\DARSHIL>tracert www.spit.ac.in

Tracing route to www.spit.ac.in [43.252.193.19]
over a maximum of 30 hops:

  1    64 ms     2 ms     3 ms  192.168.0.1
  2     2 ms     2 ms     2 ms  103.207.56.1
  3     *        *        *     Request timed out.
  4     2 ms    13 ms     2 ms  10.2.10.34
  5     3 ms     9 ms    17 ms  103.243.114.197
  6     3 ms     2 ms     2 ms  as17625.bom.extreme-ix.net [103.77.108.156]
  7     9 ms     3 ms     3 ms  27.109.1.150
  8     5 ms     4 ms     3 ms  103.205.124.82
  9     3 ms     3 ms     3 ms  43.252.192.230
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24     *        *        *     Request timed out.
 25     *        *        *     Request timed out.
 26     *        *        *     Request timed out.
 27     *        *        *     Request timed out.
 28     *        *        *     Request timed out.
 29     *        *        *     Request timed out.
 30     *        *        *     Request timed out.

Trace complete.
```



From the above results I can conclude that there may be some paths in the route which can have a different output if we try out tracert command on different days.

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, and then the path depends on which access point is ready to respond.

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?

   There is a proportional relation between the number of nodes and the location of the host.

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

Yes there is a direct relationship between the number of nodes and the latency of the host.

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

```
C:\Users\DARSHIL>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"
}
```

(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:
1. https://stackoverflow.com/questions/17868153/propagation-delay-vs-transmission-delay
2. https://www.callstats.io/blog/what-is-round-trip-time-and-how-does-it-relate-to-network-latency
3. https://www.researchgate.net/figure/2-Round-Trip-Time-RTT-versus-Packet-Length-bytes-for-different-Modulation-rates_fig3_274915210