Data Communication and Networks Lab
Experiment 2

Name : Ojasa Chitre
TE Comps
Batch : A
Date : 10th August, 2020

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.

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
```

Ping :

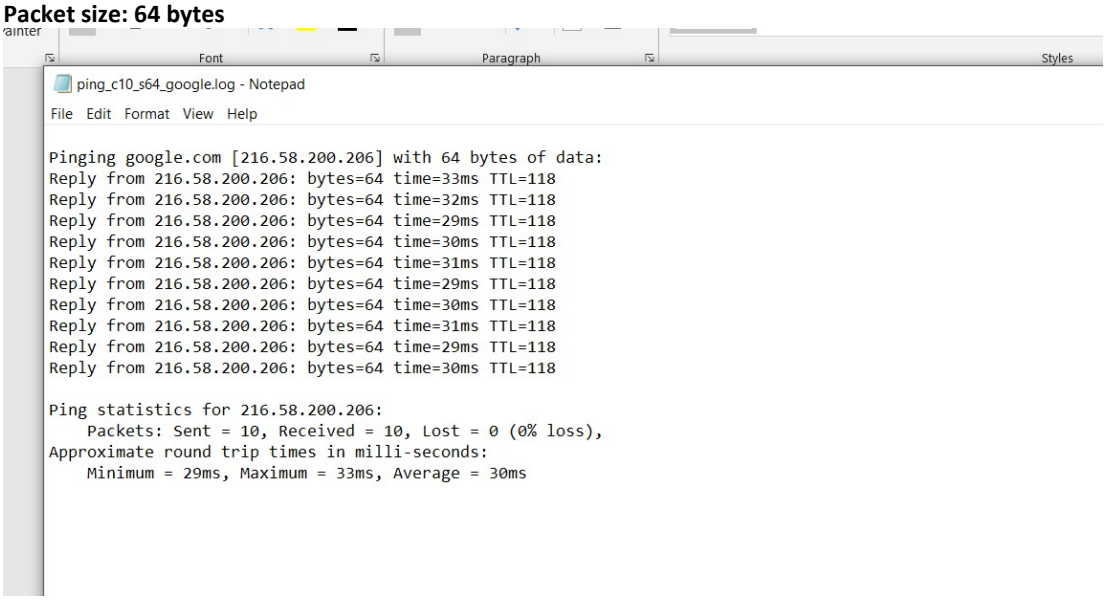
Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network. It is available for virtually all operating systems that have networking capability, including most embedded network administration software.
Ping measures the round-trip time for messages sent from the originating host to a destination computer that are echoed back to the source. The name comes from active sonar terminology that sends a pulse of sound and listens for the echo to detect objects under water.

EXPERIMENTS WITH PING

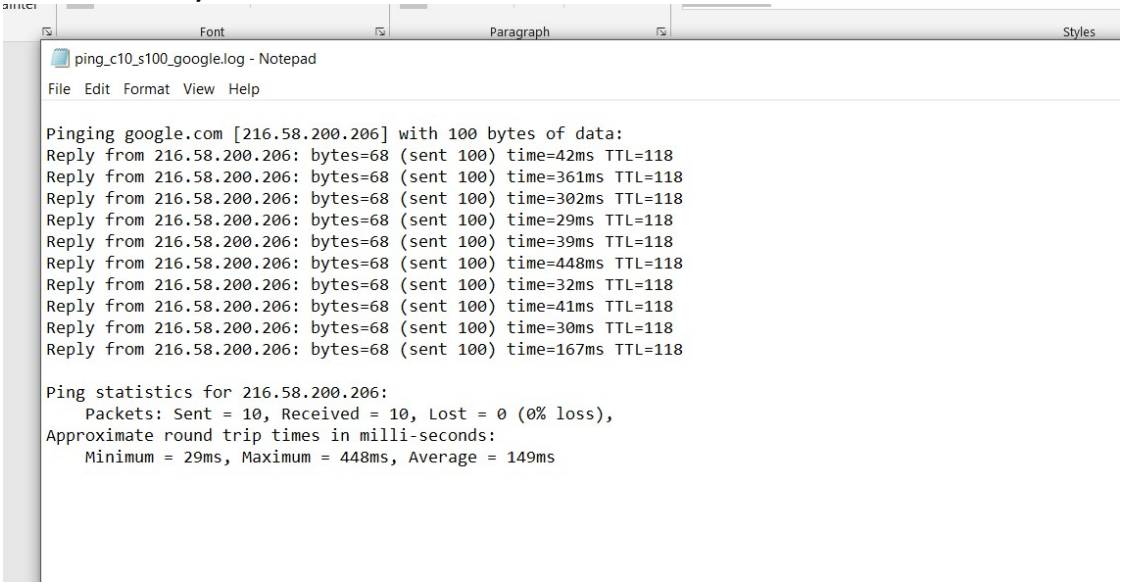
1. Ping the any hosts 10 times (i.e., packet count is 10) with a packet size of 64 bytes, 100 bytes, 500 bytes, 1000 bytes, 1400 bytes

Here are the results of pinging the same site (google.com) with different packet sizes :

1. **Packet size: 64 bytes**



2. Packet size: 100 bytes

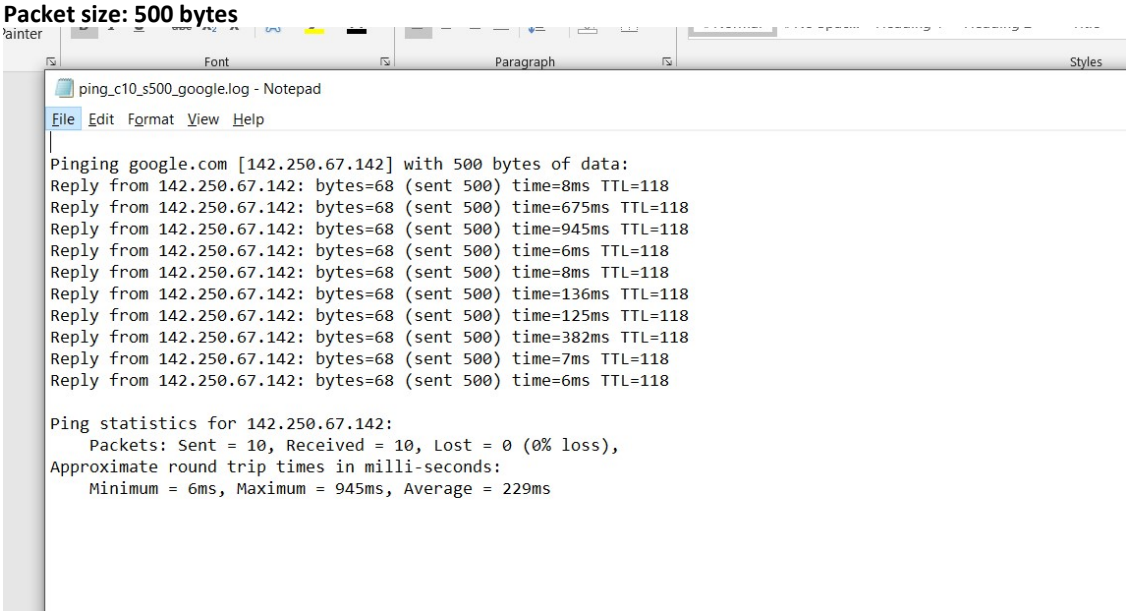


ping_c10_s100_google.log - Notepad

File Edit Format View Help

```
Pinging google.com [216.58.200.206] with 100 bytes of data:
Reply from 216.58.200.206: bytes=68 (sent 100) time=42ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=361ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=302ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=29ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=39ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=448ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=32ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=41ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=30ms TTL=118
Reply from 216.58.200.206: bytes=68 (sent 100) time=167ms TTL=118

Ping statistics for 216.58.200.206:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 29ms, Maximum = 448ms, Average = 149ms
```

3. Packet size: 500 bytes

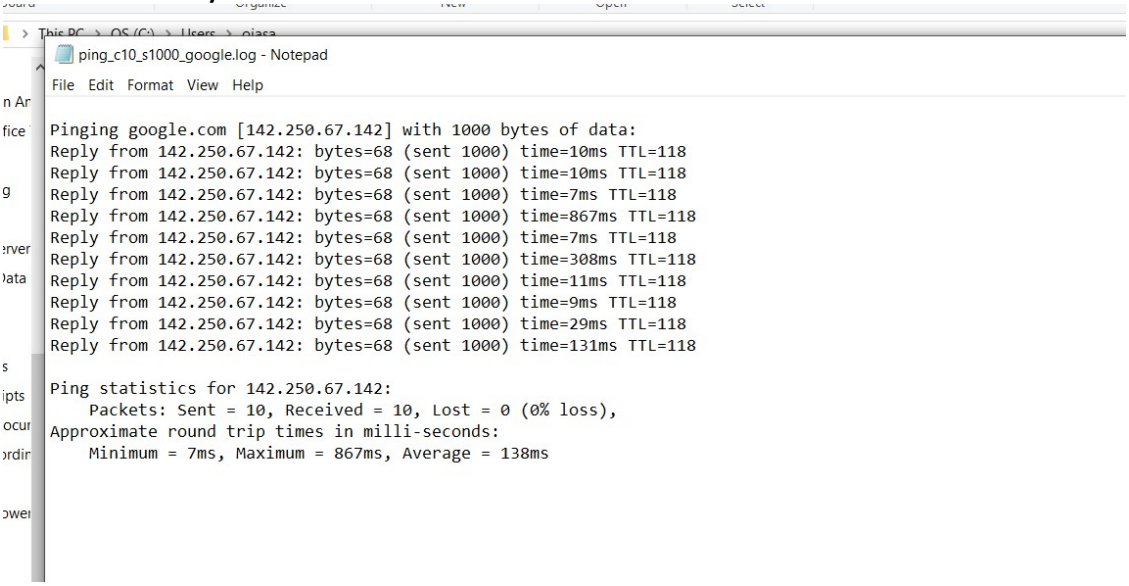


ping_c10_s500_google.log - Notepad

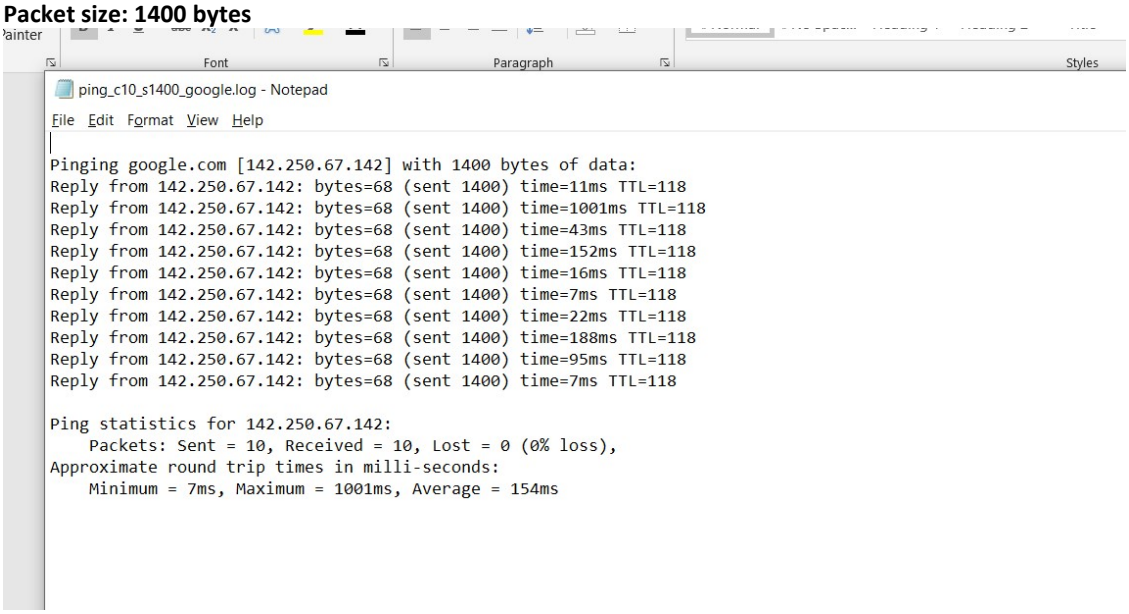
File Edit Format View Help

```
Pinging google.com [142.250.67.142] with 500 bytes of data:
Reply from 142.250.67.142: bytes=68 (sent 500) time=8ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=675ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=945ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=6ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=8ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=136ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=125ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=382ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=7ms TTL=118
Reply from 142.250.67.142: bytes=68 (sent 500) time=6ms TTL=118

Ping statistics for 142.250.67.142:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 945ms, Average = 229ms
```

4. **Packet size: 1000 bytes**



5. **Packet size: 1400 bytes**



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`.

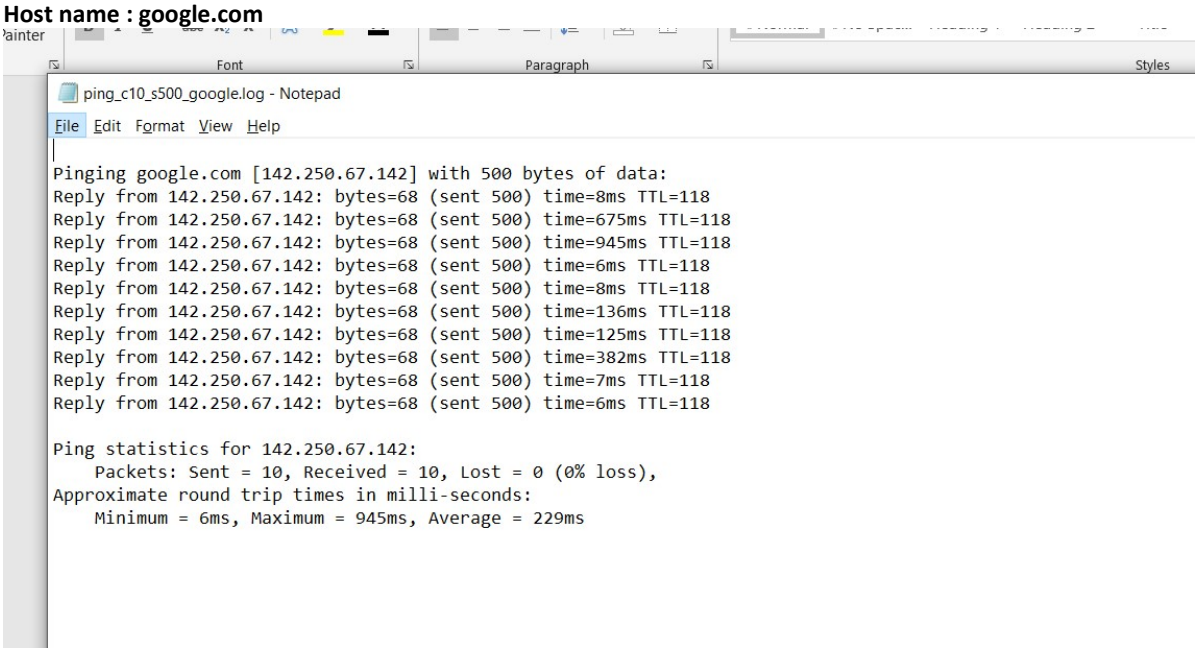
1. Does the average RTT vary between different hosts? What aspects of latency (transmit, propagation, and queueing delay) might impact this and why?

   A. Based on the data given above we can see that the average RTT increases with packet size. It is intuitive since more data has to be transported via the network which is bound to take more time.

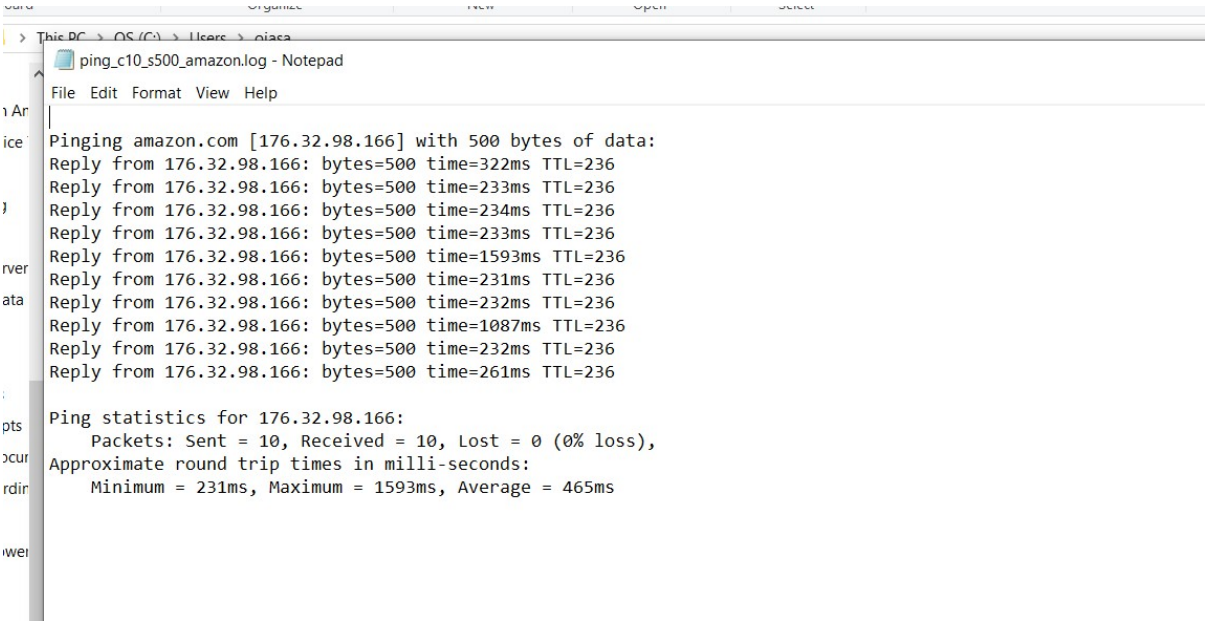
If we consider the aspects of latency they are :

1. Transmission delay depends on the medium of transmission. In this case since all transmissions were made around the same time it is safe to assume that it hasn't changed. Hence it isn't the transmission medium that has affected the RTT.
2. Propagation is the amount of time taken by a packet to travel from source to destination. This may have increased since the packet size has increased.
3. Queueing delay is the time that a packet or a job waits in a queue until it is executed. This term is most often used in reference to routers. When packets arrive at a router, they have to be processed and transmitted. A router can only process one packet at a time. If packets arrive faster than the router can process them (such as in a burst transmission) the router puts them into the queue (also called the buffer) until it can get around to transmitting them. This may have affected the RTT we observed.

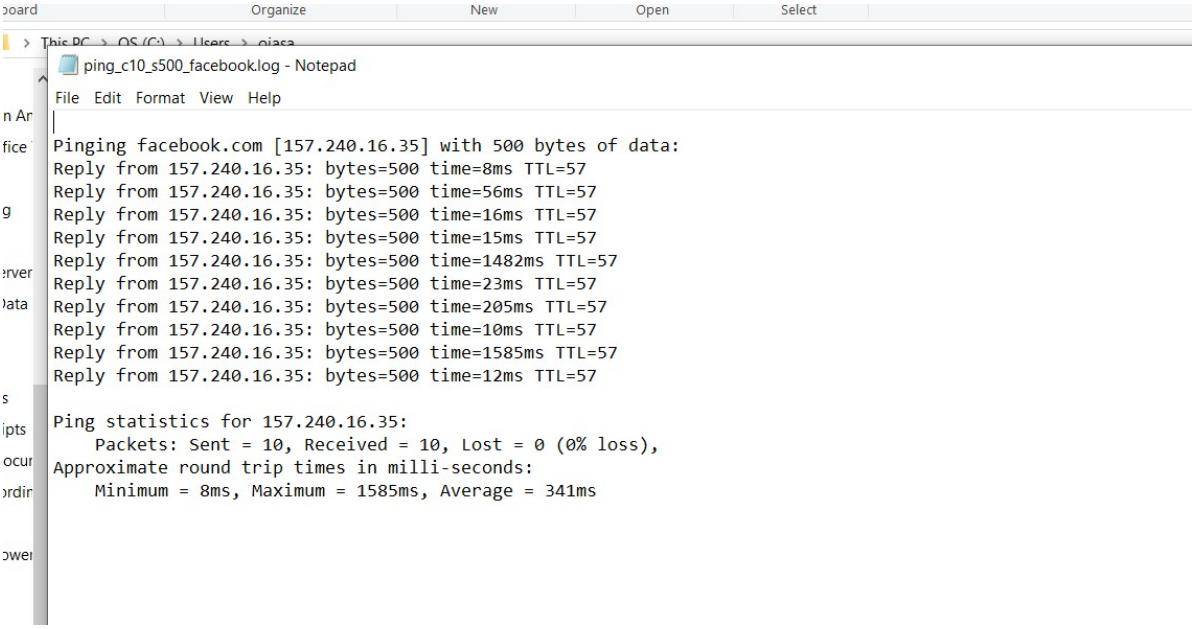
1. **Host name : google.com**



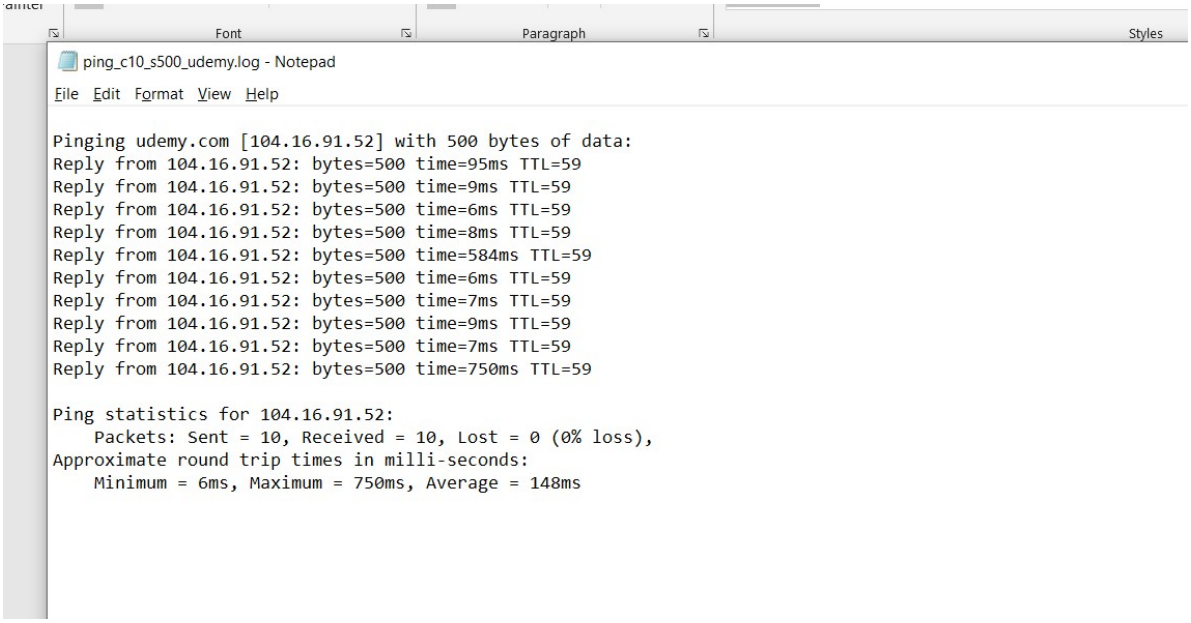
2. **Host name : amazon.com**



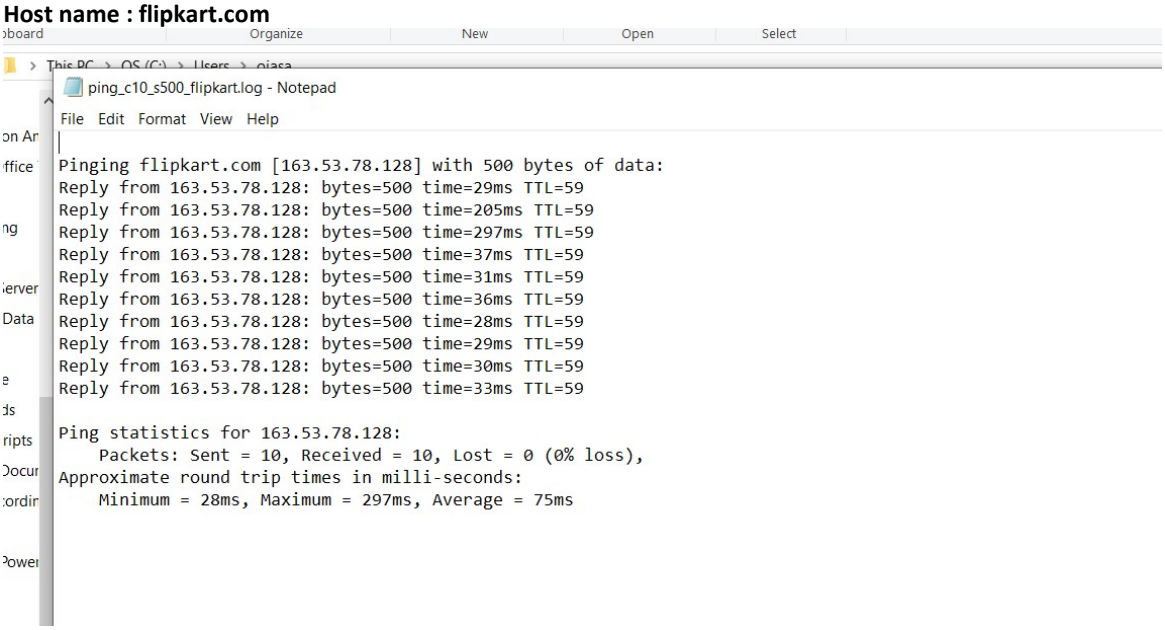
3. Host name : facebook.com



4. Host name : udemy.com



5. **Host name : flipkart.com**



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

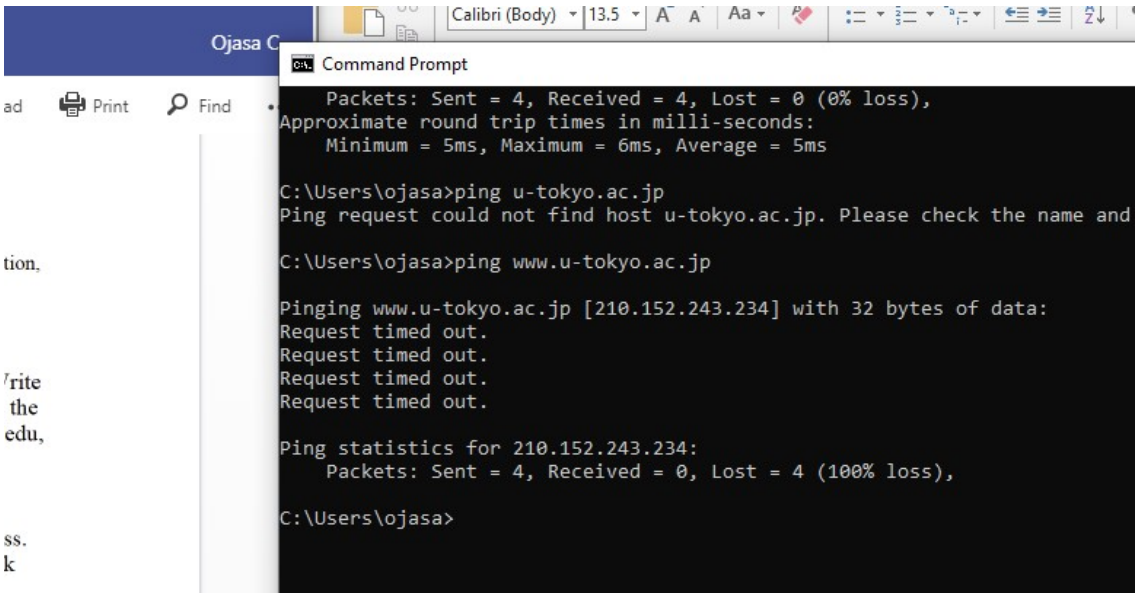
A. The average RTT is very different for different hosts.

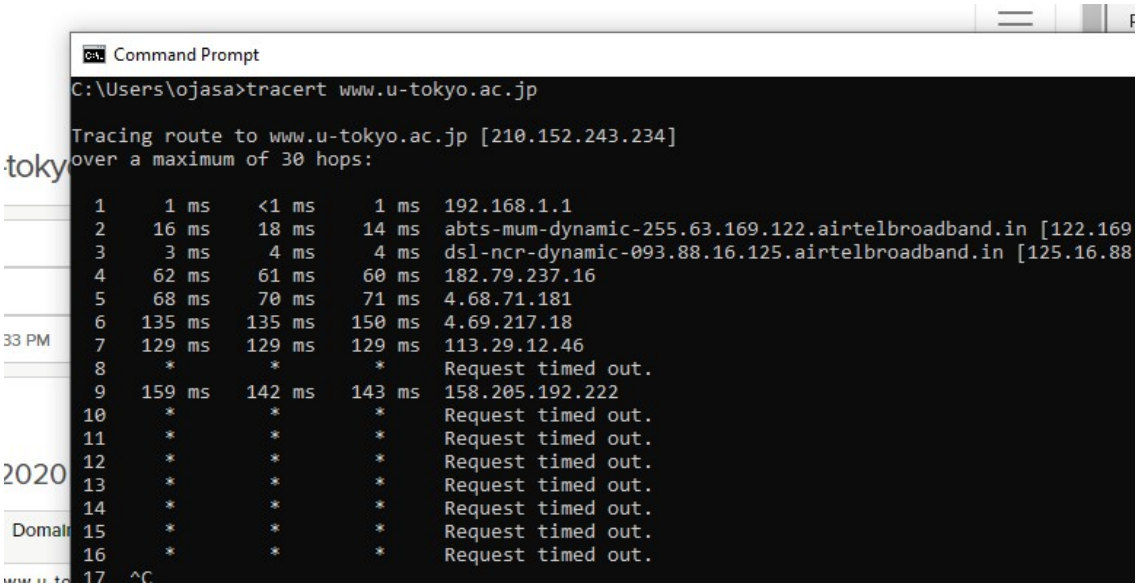
Considering the aspects of latency :

- Transmit depends on the medium of transmission. In this case since the hosts are different it may have caused the difference in transmission rate hence the RTT.
- Propagation is the amount of time taken by a packet to travel from source to destination. Since the hosts are at different geographical locations it may have contributed to the difference in the RTT.
- Queueing delay is the time that a packet or a job waits in a queue until it is executed. Now depending on the type of algorithm the host is using this might also be the reason for the delay . All the different hosts could also have different lengths of queues this may have contributed to the 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).

(some of the website weren't able to be pinged)

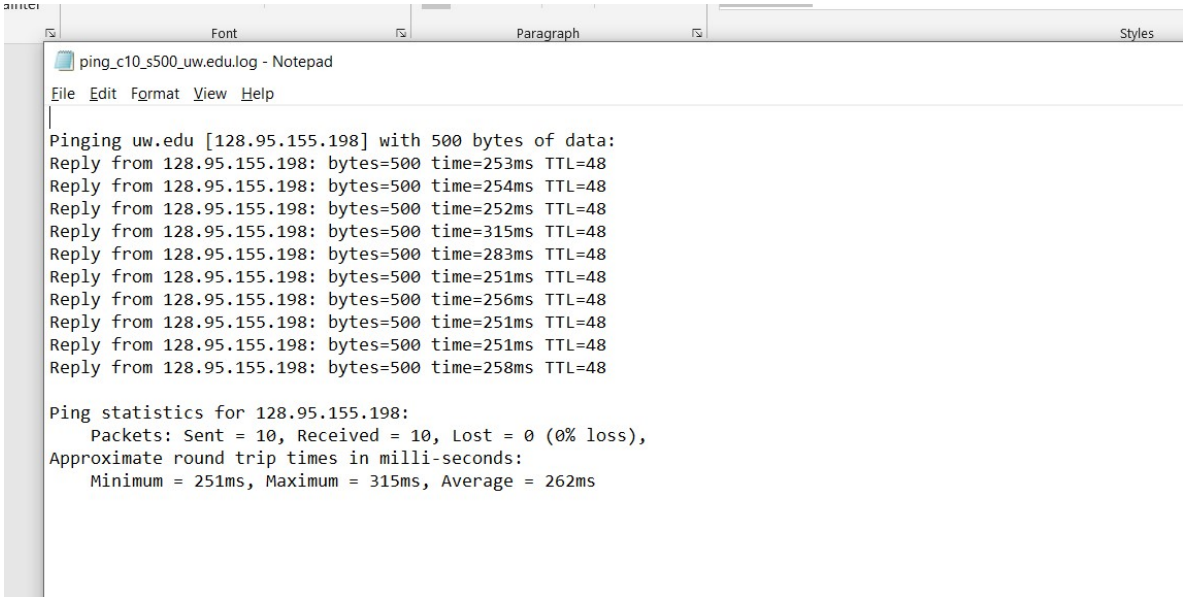




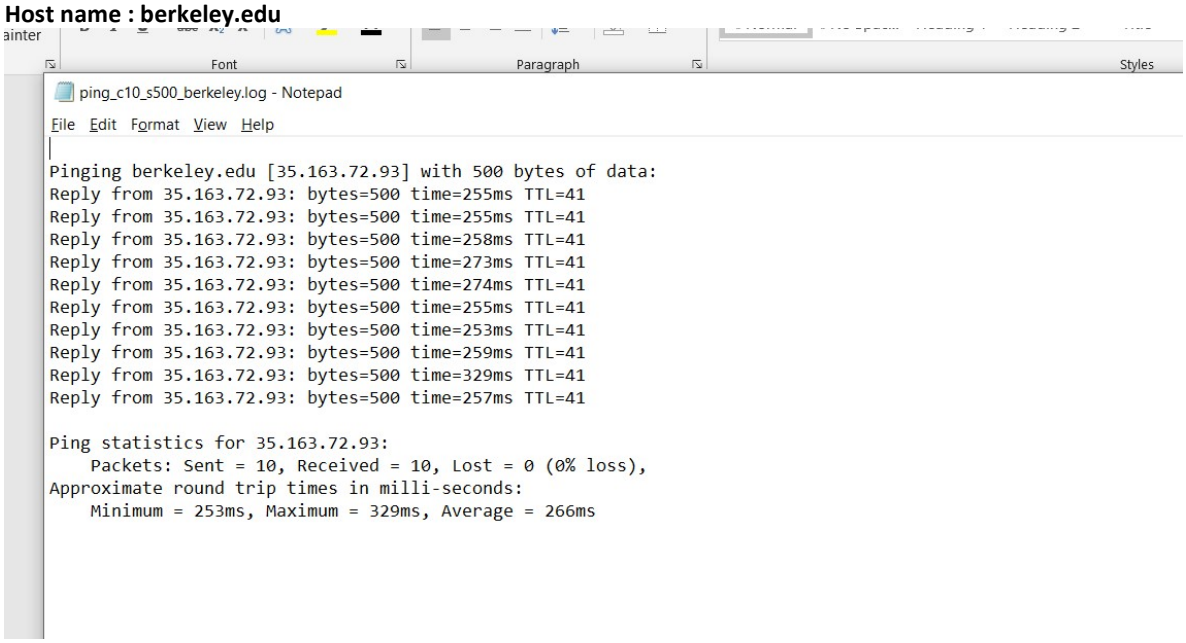
On doing tracert I realized that after a few hops it starts to take a lot of time to access the intermediate servers. And after a point the requests started to timeout. In case of tracert if 2 packets are lost consecutively then the connection is lost.

(hence I have changed Tokyo university to Tokyo institute of Technology)

Host name : www.uw.edu



Host name : berkeley.edu

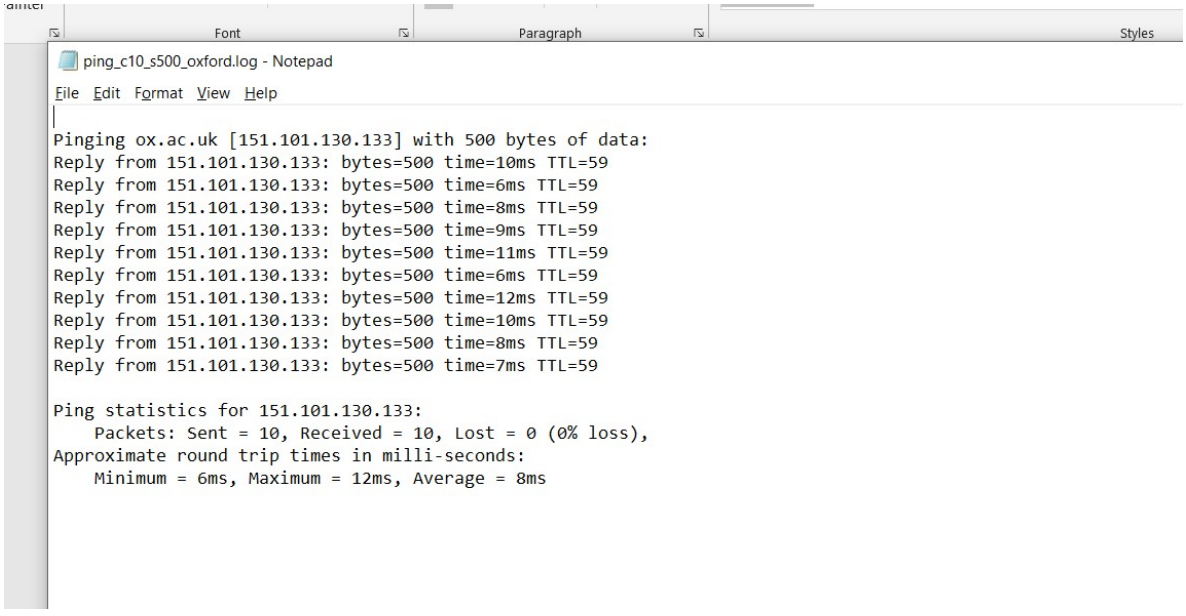


ping_c10_s500_berkeley.log - Notepad

File Edit Format View Help

```
Pinging berkeley.edu [35.163.72.93] with 500 bytes of data:
Reply from 35.163.72.93: bytes=500 time=255ms TTL=41
Reply from 35.163.72.93: bytes=500 time=255ms TTL=41
Reply from 35.163.72.93: bytes=500 time=258ms TTL=41
Reply from 35.163.72.93: bytes=500 time=273ms TTL=41
Reply from 35.163.72.93: bytes=500 time=274ms TTL=41
Reply from 35.163.72.93: bytes=500 time=255ms TTL=41
Reply from 35.163.72.93: bytes=500 time=253ms TTL=41
Reply from 35.163.72.93: bytes=500 time=259ms TTL=41
Reply from 35.163.72.93: bytes=500 time=329ms TTL=41
Reply from 35.163.72.93: bytes=500 time=257ms TTL=41

Ping statistics for 35.163.72.93:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 253ms, Maximum = 329ms, Average = 266ms
```

Host name : www.ox.ac.uk

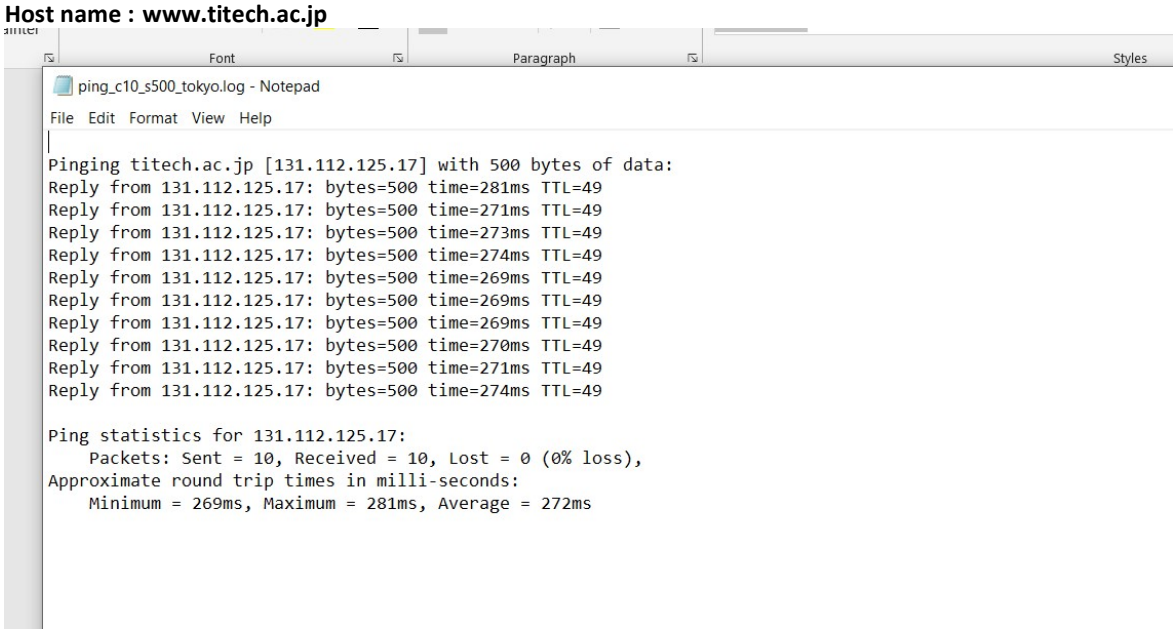


ping_c10_s500_oxford.log - Notepad

File Edit Format View Help

```
Pinging ox.ac.uk [151.101.130.133] with 500 bytes of data:
Reply from 151.101.130.133: bytes=500 time=10ms TTL=59
Reply from 151.101.130.133: bytes=500 time=6ms TTL=59
Reply from 151.101.130.133: bytes=500 time=8ms TTL=59
Reply from 151.101.130.133: bytes=500 time=9ms TTL=59
Reply from 151.101.130.133: bytes=500 time=11ms TTL=59
Reply from 151.101.130.133: bytes=500 time=6ms TTL=59
Reply from 151.101.130.133: bytes=500 time=12ms TTL=59
Reply from 151.101.130.133: bytes=500 time=10ms TTL=59
Reply from 151.101.130.133: bytes=500 time=8ms TTL=59
Reply from 151.101.130.133: bytes=500 time=7ms TTL=59

Ping statistics for 151.101.130.133:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 12ms, Average = 8ms
```

Host name : www.titech.ac.jp



The above shown images show the pinging time for various websites world wide. For www.uw.edu,

berkeley.edu, titech.ac.jp have similar pinging time. Although they are at different location and distances all over the world. Surprisingly the pinging time for was extremely low for ox.ac.uk .

Also no matter where the server might be it seems there is no loss of packets.

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).

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>

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\ojasa>netstat -t -n

Active Connections

Proto	Local Address	Foreign Address	State	Offload State
TCP	127.0.0.1:49727	127.0.0.1:49728	ESTABLISHED	InHost
TCP	127.0.0.1:49728	127.0.0.1:49727	ESTABLISHED	InHost
TCP	127.0.0.1:49814	127.0.0.1:49815	ESTABLISHED	InHost
TCP	127.0.0.1:49815	127.0.0.1:49814	ESTABLISHED	InHost
TCP	192.168.1.93:8085	192.168.1.1:33370	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33371	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33372	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33373	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33374	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33375	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33376	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33377	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33378	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33379	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33380	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33381	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33382	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33383	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33384	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33385	TIME_WAIT	InHost
TCP	192.168.1.93:8085	192.168.1.1:33386	TIME_WAIT	InHost
TCP	192.168.1.93:49677	5.45.58.216:80	ESTABLISHED	InHost
TCP	192.168.1.93:49697	40.90.189.152:443	ESTABLISHED	InHost
TCP	192.168.1.93:49698	5.62.54.63:443	ESTABLISHED	InHost
TCP	192.168.1.93:50021	74.125.24.188:5228	ESTABLISHED	InHost
TCP	192.168.1.93:50050	104.120.79.78:80	TIME_WAIT	InHost
TCP	192.168.1.93:50081	117.18.232.200:443	ESTABLISHED	InHost

TCP	192.168.1.93:50102	40.90.22.191:443	TIME_WAIT	InHost
TCP	192.168.1.93:50103	40.81.30.101:443	TIME_WAIT	InHost
TCP	192.168.1.93:50118	216.58.196.78:443	TIME_WAIT	InHost
TCP	192.168.1.93:50119	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50120	216.58.196.197:443	TIME_WAIT	InHost
TCP	192.168.1.93:50123	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50124	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50139	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50140	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50141	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50142	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50147	117.18.237.29:80	TIME_WAIT	InHost
TCP	192.168.1.93:50152	52.43.91.27:443	TIME_WAIT	InHost
TCP	192.168.1.93:50153	117.18.237.29:80	TIME_WAIT	InHost
TCP	192.168.1.93:50154	117.18.237.29:80	TIME_WAIT	InHost
TCP	192.168.1.93:50155	74.208.255.134:443	TIME_WAIT	InHost
TCP	192.168.1.93:50156	74.208.255.134:443	TIME_WAIT	InHost
TCP	192.168.1.93:50160	213.165.66.58:443	TIME_WAIT	InHost
TCP	192.168.1.93:50162	195.20.251.98:443	TIME_WAIT	InHost
TCP	192.168.1.93:50164	195.20.250.196:443	TIME_WAIT	InHost
TCP	192.168.1.93:50165	195.20.250.196:443	TIME_WAIT	InHost
TCP	192.168.1.93:50169	104.16.126.175:443	TIME_WAIT	InHost
TCP	192.168.1.93:50170	217.160.86.75:443	TIME_WAIT	InHost
TCP	192.168.1.93:50171	217.160.86.75:443	TIME_WAIT	InHost
TCP	192.168.1.93:50172	217.160.86.75:443	TIME_WAIT	InHost
TCP	192.168.1.93:50173	217.160.86.75:443	TIME_WAIT	InHost
TCP	192.168.1.93:50178	161.69.226.72:443	TIME_WAIT	InHost
TCP	192.168.1.93:50179	31.13.79.35:443	TIME_WAIT	InHost
TCP	192.168.1.93:50180	144.2.1.5:443	TIME_WAIT	InHost
TCP	192.168.1.93:50183	204.79.197.200:443	ESTABLISHED	InHost
TCP	192.168.1.93:50186	23.57.14.10:443	ESTABLISHED	InHost
TCP	192.168.1.93:50187	23.57.14.10:443	ESTABLISHED	InHost
TCP	192.168.1.93:50188	162.125.81.1:443	ESTABLISHED	InHost
TCP	192.168.1.93:50189	104.215.155.1:443	ESTABLISHED	InHost
TCP	192.168.1.93:50192	192.124.249.23:80	ESTABLISHED	InHost
TCP	192.168.1.93:50193	168.62.200.169:443	TIME_WAIT	InHost
TCP	192.168.1.93:50194	162.125.35.135:443	ESTABLISHED	InHost
TCP	192.168.1.93:50196	104.215.155.1:443	ESTABLISHED	InHost
TCP	192.168.1.93:50197	13.227.130.89:443	ESTABLISHED	InHost
TCP	192.168.1.93:50199	54.149.80.27:443	ESTABLISHED	InHost
TCP	192.168.1.93:50200	161.69.226.73:443	TIME_WAIT	InHost
TCP	192.168.1.93:50202	23.57.14.10:443	ESTABLISHED	InHost
TCP	192.168.1.93:50203	23.57.14.10:443	ESTABLISHED	InHost
TCP	192.168.1.93:50204	23.57.14.10:443	ESTABLISHED	InHost
TCP	192.168.1.93:50205	184.25.109.122:443	CLOSE_WAIT	InHost
TCP	192.168.1.93:50206	23.215.205.169:80	ESTABLISHED	InHost

TCP	192.168.1.93:50207	23.215.205.169:80	ESTABLISHED	InHost
TCP	192.168.1.93:50208	117.18.237.29:80	ESTABLISHED	InHost
TCP	192.168.1.93:50209	40.90.22.183:443	ESTABLISHED	InHost
TCP	192.168.1.93:50210	23.215.205.169:80	ESTABLISHED	InHost
TCP	192.168.1.93:50211	23.215.205.169:80	ESTABLISHED	InHost
TCP	192.168.1.93:50212	23.215.205.169:80	ESTABLISHED	InHost
TCP	192.168.1.93:50213	23.215.205.169:80	ESTABLISHED	InHost
TCP	192.168.1.93:50216	23.57.14.10:443	ESTABLISHED	InHost
TCP	192.168.1.93:50219	37.228.108.133:443	TIME_WAIT	InHost
TCP	192.168.1.93:50220	37.228.108.133:443	TIME_WAIT	InHost
TCP	192.168.1.93:50222	74.208.255.134:443	TIME_WAIT	InHost
TCP	192.168.1.93:50223	74.208.255.134:443	TIME_WAIT	InHost
TCP	192.168.1.93:50224	161.69.45.107:443	ESTABLISHED	InHost
TCP	192.168.1.93:50226	52.43.91.27:443	TIME_WAIT	InHost
TCP	192.168.1.93:50227	172.217.166.163:443	TIME_WAIT	InHost
TCP	192.168.1.93:50228	104.215.155.1:443	TIME_WAIT	InHost

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

2. Traceroute (tracert) :

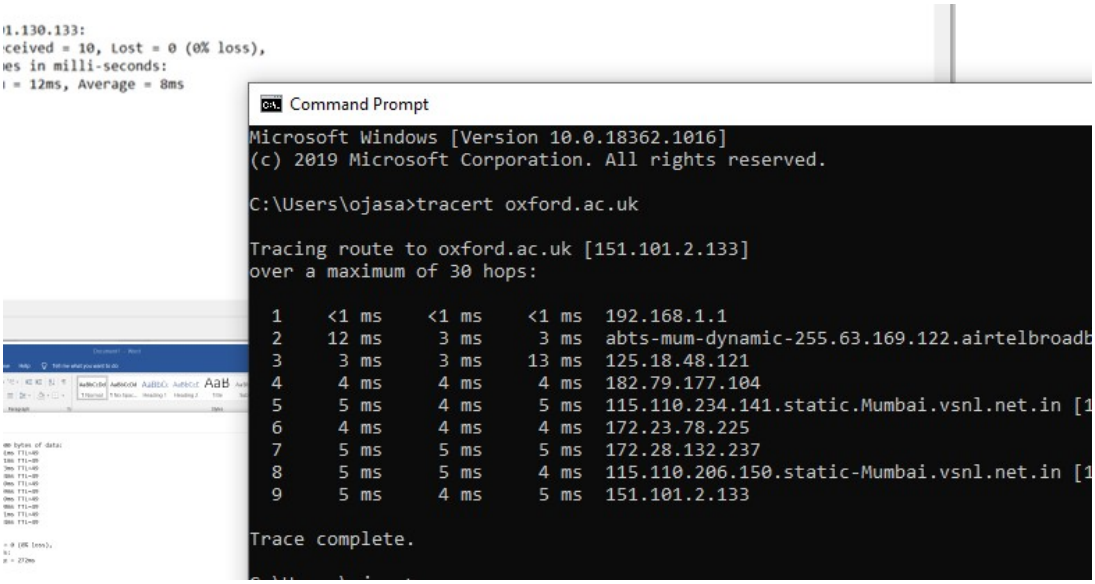
In computing, traceroute and tracert are computer network diagnostic commands for displaying possible routes (paths) and measuring transit delays of packets across an Internet Protocol (IP) network. The history of the route is recorded as the round-trip times of the packets received from each successive host (remote node) in the route (path); the sum of the mean times in each hop is a measure of the total time spent to establish the connection. Traceroute proceeds unless all (usually three) sent packets are lost more than twice; then the connection is lost and the route cannot be evaluated. Ping, on the other hand, only computes the final round-trip times from the destination point.

TTL - Time to live

Time to live (TTL) or hop limit is a mechanism that limits the lifespan or lifetime of data in a computer or network. TTL may be implemented as a counter or timestamp attached to or embedded in the data. Once the prescribed event count or timespan has elapsed, data is discarded or revalidated.

How to read a Traceroute

Once the traceroute is run, it generates the report as it goes along the route. Below is a sample traceroute:



As you can see, there are several rows divided into columns on the report. Each row represents a "hop" along the route. Think of it as a check-in point where the signal gets its next set of directions. Each row is divided into five columns. A sample row is below:
Let's break this particular hop down into its parts.

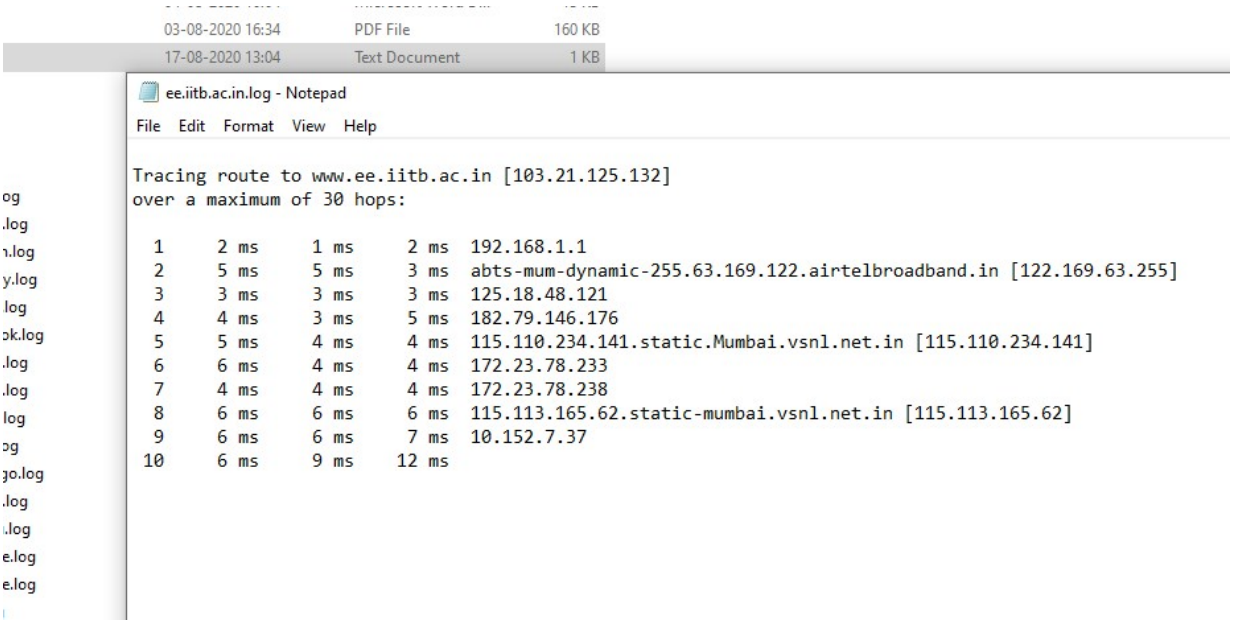
Hop #	RTT 1	RTT 2	RTT 3	Name/IP Address
9	5 ms	4 ms	5 ms	151.101.2.133

Hop Number – This is the first column and is simply the number of the hop along the route. In this case, it is the tenth hop.

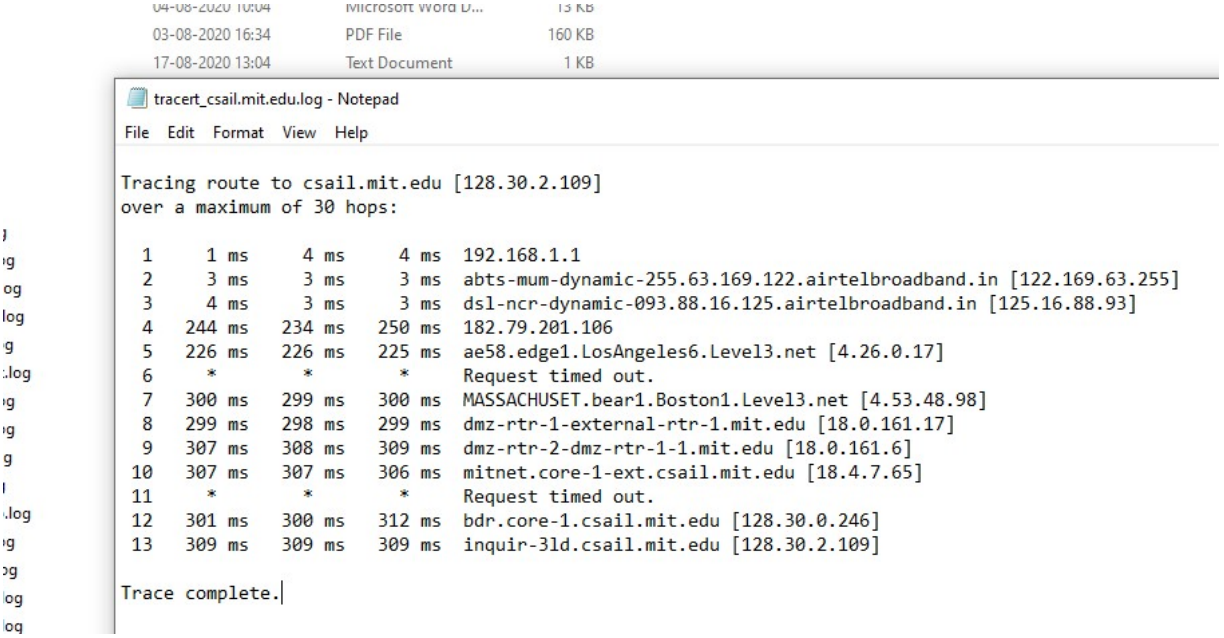
RTT Columns – The next three columns display the round trip time (RTT) for your packet to reach that point and return to your computer. This is listed in milliseconds. There are three columns because the traceroute sends three separate signal packets. This is to display consistency, or a lack thereof, in the route.

Domain/IP column – The last column has the IP address of the router. If it is available, the domain name will also be listed.

1. Trace for ee.iitb.ac.in



2. Trace for csail.mit.edu



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.

tracert_math.hws.edu.log - Notepad

File Edit Format View Help

```
Tracing route to math.hws.edu [64.89.144.237]
over a maximum of 30 hops:

  1     4 ms    56 ms   102 ms  192.168.1.1
  2    88 ms    55 ms     *     abts-mum-dynamic-255.63.169.122.airtelbroadband.in [122.169.63.255]
  3    36 ms     7 ms     6 ms  dsl-ncr-dynamic-093.88.16.125.airtelbroadband.in [125.16.88.93]
  4   231 ms   232 ms   237 ms  182.79.247.92
  5   349 ms   329 ms   246 ms  xe-9-1-0.edge1.LosAngeles6.Level3.net [4.26.0.61]
  6     *      333 ms   273 ms  ae-1-51.ear3.LosAngeles1.Level3.net [4.69.206.225]
  7     *        *        *     Request timed out.
  8   404 ms   319 ms   290 ms  roc1-ar5-xe-0-0-0-0.us.twtelecom.net [35.248.1.158]
  9   319 ms   323 ms   315 ms  66-195-65-170.static.ctl.one [66.195.65.170]
 10   438 ms   353 ms   357 ms  64.89.144.100
 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.
```

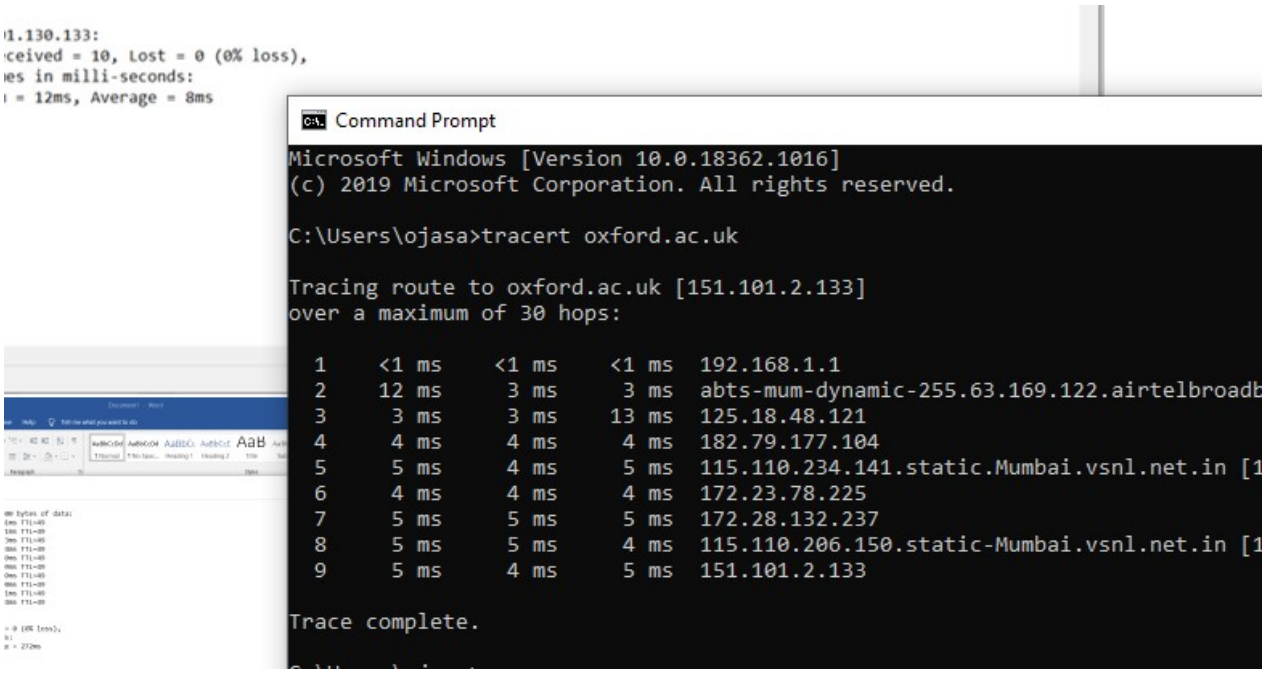
tracert_www.hws.edu.log - Notepad

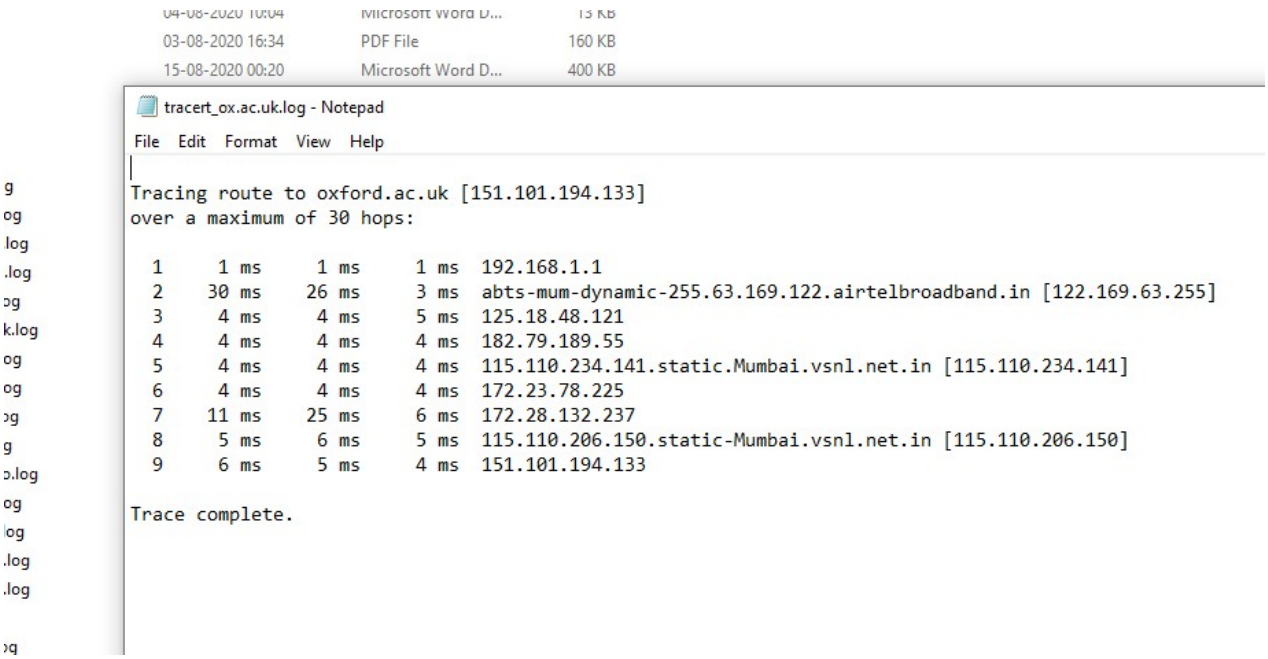
File Edit Format View Help

```
Tracing route to www.hws.edu [64.89.145.159]
over a maximum of 30 hops:

  1   102 ms     8 ms     2 ms  192.168.1.1
  2     7 ms    19 ms    27 ms  abts-mum-dynamic-255.63.169.122.airtelbroadband.in [122.169.63.255]
  3    15 ms    24 ms     5 ms  dsl-ncr-dynamic-093.88.16.125.airtelbroadband.in [125.16.88.93]
  4   247 ms   241 ms   239 ms  182.79.222.25
  5   231 ms   231 ms   243 ms  xe-5-1-0.edge1.LosAngeles6.Level3.net [4.26.0.89]
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8   335 ms   303 ms   301 ms  roc1-ar5-xe-0-0-0-0.us.twtelecom.net [35.248.1.158]
  9   357 ms   314 ms   315 ms  66-195-65-170.static.ctl.one [66.195.65.170]
 10   336 ms   316 ms   316 ms  64.89.144.100
 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.
```

As far as the trace has happened the hops seem to be similar.

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.





15-08-2020 00:20 Microsoft Word D... 400 KB

The above routes were the routes for the same host on different days. We can clearly see that the request was routed through the exact same path.The last two rotes are from the same time they are also similar. We can see that not just the path but the RTT is also pretty similar.

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?

   The first two hops are the same. This is probably because one of them is the router my device is connected to. That will always be the second hop as long as I am connected to the same broadband. As far has the first hop is concerned the reason is that the IP address is a non-routable IP address. A non-routable IP address, also known as a private IP address, is not assigned to any one organization and does not need to be assigned by an Internet Service Provider.

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?

   In wired networks, the hop count refers to the number of intermediate network devices through which data must pass between source and destination.[1] Hop count is a rough measure of distance between two hosts. A hop count of n means that n network devices separate the source host from the destination host.[2]

On a layer 3 network such as Internet Protocol (IP), each router along the data path constitutes a hop. By itself, this metric is, however, not useful for determining the optimum network path, as it does not take into consideration the speed, load, reliability, or latency of any particular hop, but merely the total count. Nevertheless, some routing protocols, such as Routing Information Protocol (RIP), use hop count as their sole metric.

Each time a router receives a packet, it modifies the packet, decrementing the time to live (TTL). The router discards any packets received with a zero TTL value. This prevents packets from endlessly bouncing around the network in the event of routing errors. Routers are capable of managing hop counts, but other types of network devices (e.g. Ethernet hubs and bridges) are not.

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?

   As seen above the IP doesn't take the latency into consideration. Although the latency for a particular host is affected by the IP. As far as I understand since the same protocol is used everywhere this should apply to all hosts.

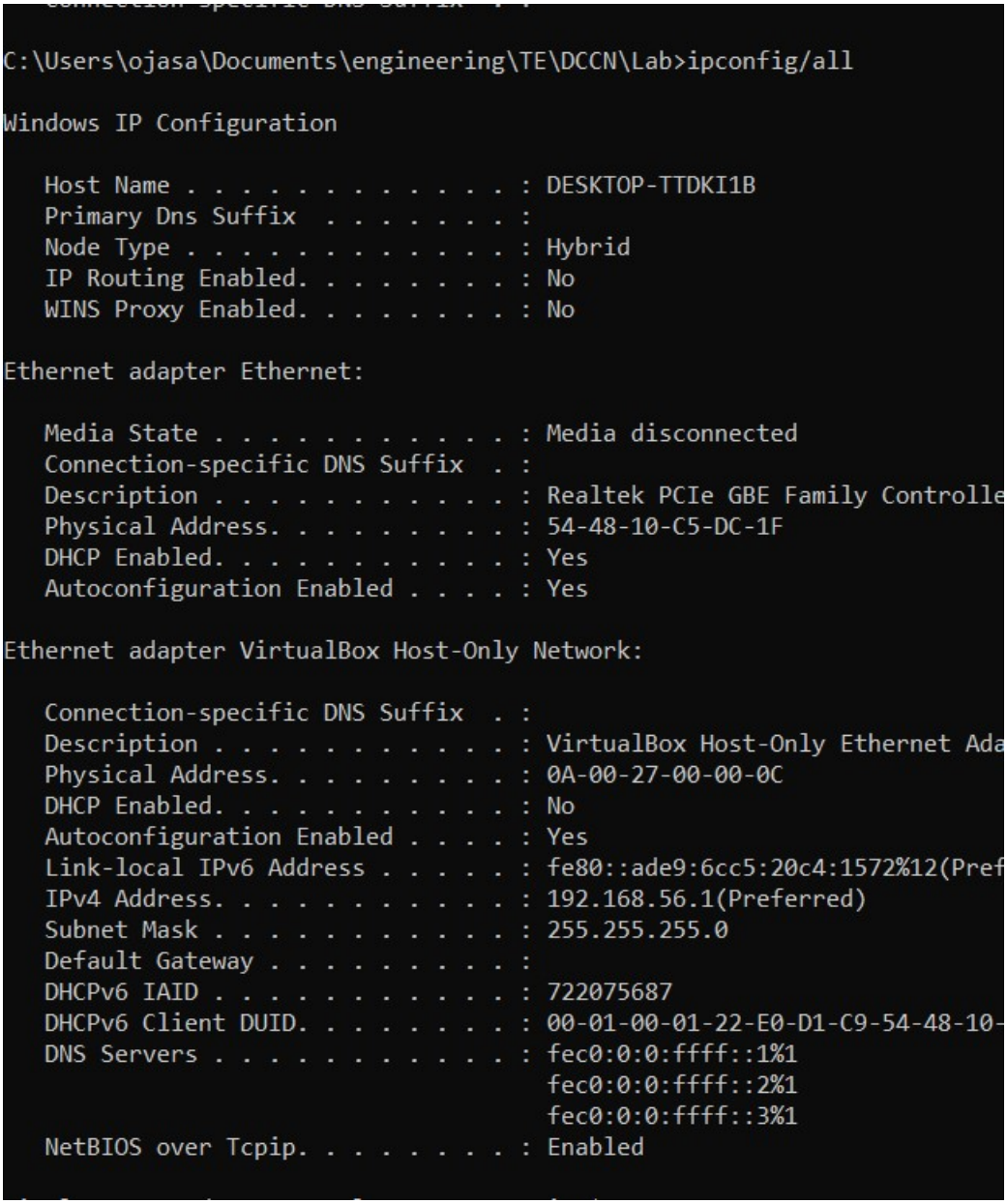
   Just an interesting observation that while using my mobile hotspot it will use the IPv6 by default:

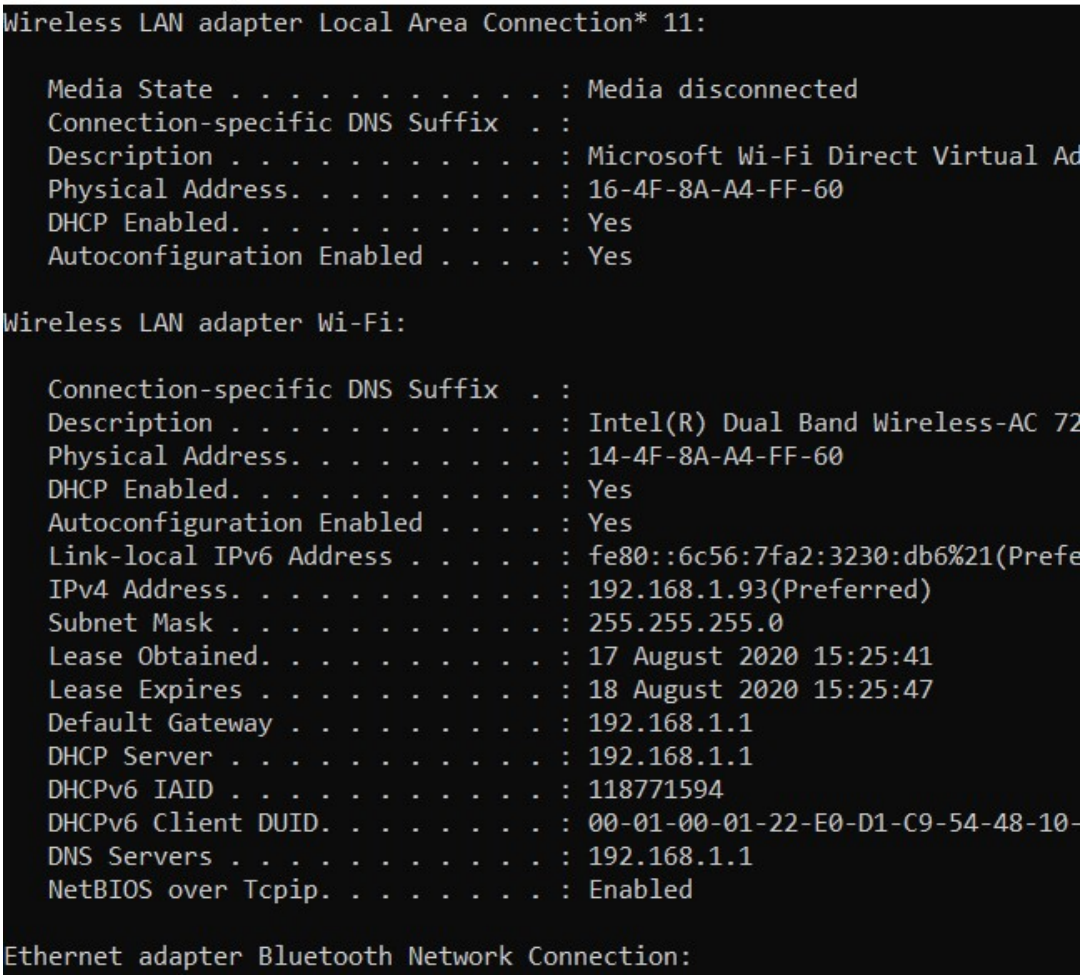
```
  9    32 ms    36 ms    34 ms  2001:4860:0:115e::3
 10    55 ms    51 ms    58 ms  2001:4860:0:115b::1
 11    55 ms    53 ms    66 ms  2001:4860:0:1::2205
 12    28 ms    41 ms    46 ms  bom05s10-in-x0e.1e100.net [2404:6800:400

Trace complete.

C:\Users\ojasa\Documents\engineering\TE\DCCN\Lab>tracert google.com

Tracing route to google.com [2404:6800:4009:803::200e]
over a maximum of 30 hops:

  1     7 ms     2 ms     2 ms  2402:3a80:1642:ef84::8
```

Ipconfig :

In computing, ipconfig (internet protocol configuration) is a console application of some operating systems that displays all current TCP/IP network configuration values and refresh Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings.

I have used ipconfig/all it displays the full TCP/IP configuration for all adapters. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connection


C:\Users\ojasa\Documents\engineering\TE\DCCN\Lab>ipconfig/all
Windows IP Configuration
Host Name . . . . . : DESKTOP-TTDKI1B
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
Ethernet adapter Ethernet:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Realtek PCIe GBE Family Controlle
Physical Address. . . . . : 54-48-10-C5-DC-1F
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Ethernet adapter VirtualBox Host-Only Network:
Connection-specific DNS Suffix . :
Description . . . . . : VirtualBox Host-Only Ethernet Ada
Physical Address. . . . . : 0A-00-27-00-00-0C
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::ade9:6cc5:20c4:1572%12(Pref
IPv4 Address. . . . . : 192.168.56.1(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 722075687
DHCPv6 Client DUID. . . . . : 00-01-00-01-22-E0-D1-C9-54-48-10-
DNS Servers . . . . . : fec0:0:0:ffff::1%1
fec0:0:0:ffff::2%1
fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled



Host name : This is the name of the computer, as seen by Internet Protocol
Primary DNS Suffix : Most small LANs don't have a DNS server setup that is why it is blank
The Node Type : tells us how this computer identifies the address of another computer on the LAN.

The Physical Address is the MAC address for this network card. If this is the Vendor Assigned address, it is unique for this device. All Vendor Assigned addresses are unique, for every device in the world. If this is a User Defined address, it was set using tools provided by the vendor. For NT compliant network hardware, this was likely the device properties wizard, accessed from Local Area Connection Properties in Network Connections.

The IP Address for each computer must be unique. Taking the IP Address and the Subnet Mask, and subnetting the IP address, we see that this subnet is 192.168.1.0/24, and the Host Address is 50. On any LAN segment, all hosts (computers) must have the same subnet, and all computers must have a different host address.

While the Subnet and Host addresses together determine which computers on a LAN can communicate, the Default Gateway determines if the computer can communicate with any hosts outside the subnet. The Default Gateway must be the IP address of another host, on that same

subnet, that also connects outside the LAN. With no default gateway value, or with an invalid IP address here, your computer won't have access outside the LAN.

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*).

Whois :

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.

Whois is a widely used Internet record listing that identifies who owns a domain and how to get in contact with them. The Internet Corporation for Assigned Names and Numbers (ICANN) regulates domain name registration and ownership. Whois records have proven to be extremely useful and have developed into an essential resource for maintaining the integrity of the domain name registration and website ownership process.

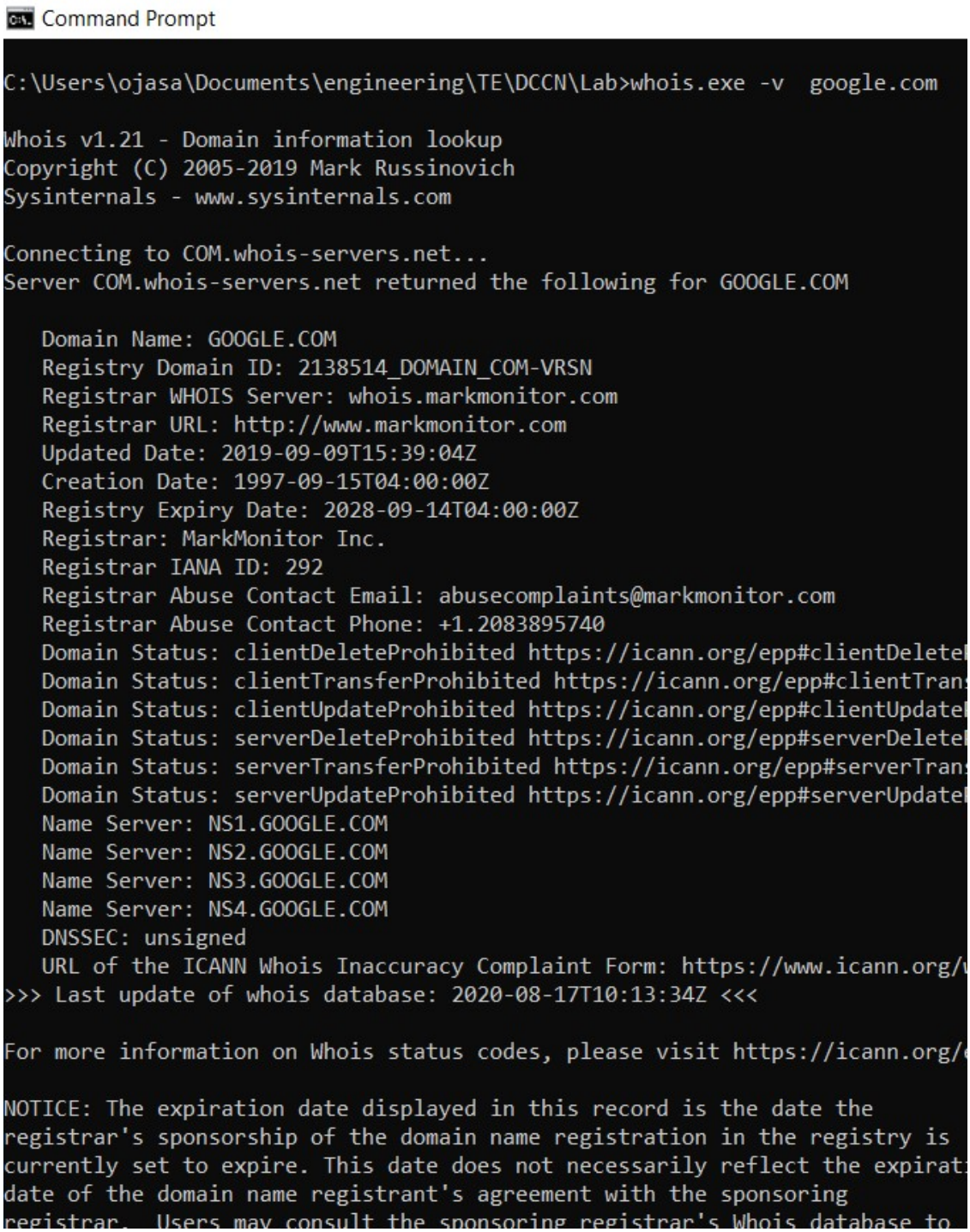
A Whois record contains all of the contact information associated with the person, group, or company that registers a particular domain name. Typically, each Whois record will contain information such as the name and contact information of the Registrant (who owns the domain), the name and contact information of the registrar Registrar (the organization or commercial entity that registered the domain name), the registration dates, the name servers, the most recent update, and the expiration date. Whois records may also provide the administrative and technical contact information (which is often, but not always, the registrant).

Whois Thick and Thin Models
There are two different data models for storing Whois resource information:

Thin Model. Thin Whois lookup only gives the registrar, name servers and registration dates. To acquire additional information, a secondary lookup at the registrar on file is necessary to attain full information on domain name ownership.

Thick Model. A thick Whois provides useful additional details beyond what is contained in a thin Whois record. Typically, the additional details contain contact (registrant, administrative, and technical) information. A lookup, then, will supply all the necessary information on who owns the domain, where it is registered, what name servers it uses, when it was registered and when it may expire.

Incase of google we can see that the Domain Name is Google.com
We can also see its domain ID which 2138514_DOMAIN_COM-VRSN
Then the whois server of the registrar and its url



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.

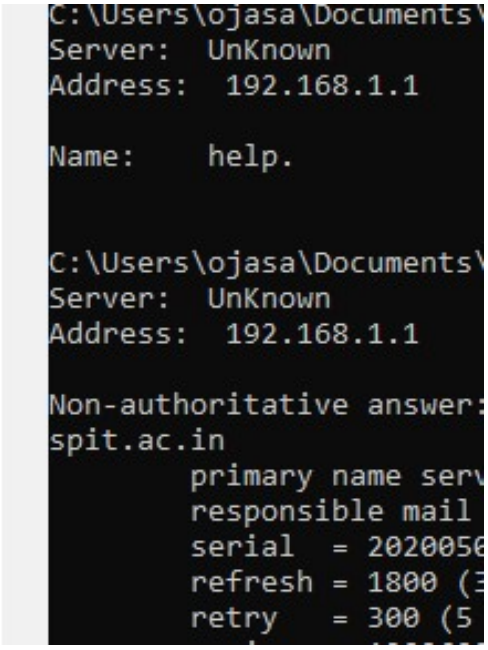
This geolocation program is not installed on our computers, but you can access one on the command line using the *curl* command, which can send HTTP requests and display the response. The following command uses *curl* to contact a public web service that will look up an IP address for you: curl ipinfo.io/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

```
Access to .IN WHOIS information is provided to assist persons
r informational purposes only ,and .IN does not guarantee its
rcumstances will you use this data to (a) allow, enable, or o
an the data recipient's own existing customers; or (b) enable
nably necessary to register domain names or modify existing r
cy.
```

```
Name:    spit.ac.in
Address:  43.252.193.19


C:\Users\ojasa\Documents\engineering\TE\DCCN\Lab>nslookup
Default Server:  UnKnown
Address:  192.168.1.1

>
```



Execution of whois on spit.ac.in :

C:\Users\ojasa\Documents\engineering\TE\DCCN\Lab>whois.exe -v spit.ac.in

Whois v1.21 - Domain information lookup
Copyright (C) 2005-2019 Mark Russinovich
Sysinternals - www.sysinternals.com

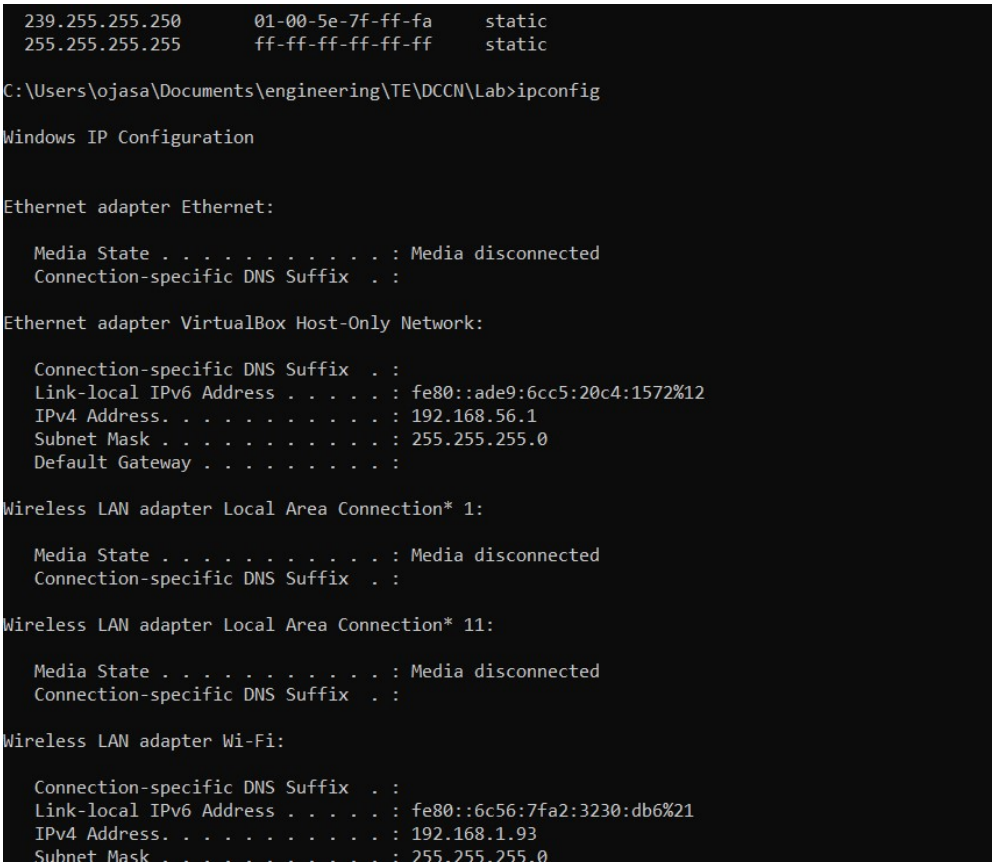
Connecting to IN.whois-servers.net...
Server IN.whois-servers.net returned the following for SPIT.AC.IN

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: 2020-08-24T09:52:29Z <<<

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.

```
Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4009:800::2004
          216.58.200.196


C:\Users\ojasa\Documents\engineering\TE\DCCN\Lab>arp -a

Interface: 192.168.56.1 --- 0xc
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.1.93 --- 0x15
  Internet Address      Physical Address      Type
  192.168.1.1           18-56-44-26-66-a3     dynamic
```



Conclusion :

I was able to execute and understand various network commands.

References :

1. https://en.wikipedia.org/wiki/Ping_(networking_utility)
2. https://en.wikipedia.org/wiki/Traceroute