Name – Prerak M. Parekh
Third Year Computers
Roll No – 40
Data Communication and Computer Networks
Date - 21/08/2020

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.

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

Some Basic command line Networking utilities

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

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

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

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

The syntax in Windows is:

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

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

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

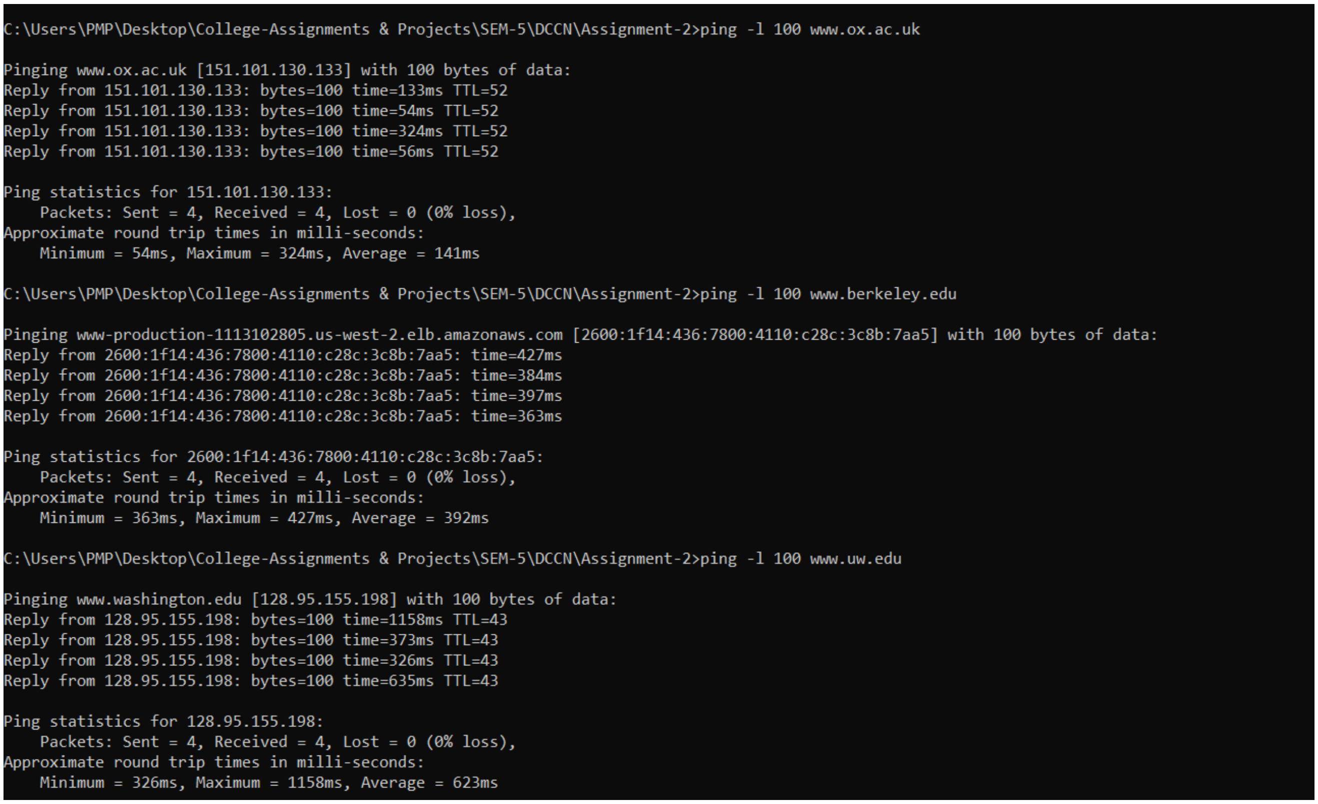
```
ping -c 10 google.com > ping_c10_s64_google.log
```

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).

Answer 1: I have tries three places in the following order

1. www.uw.edu
2. www.berkeley.edu
3. www.ox.ac.uk

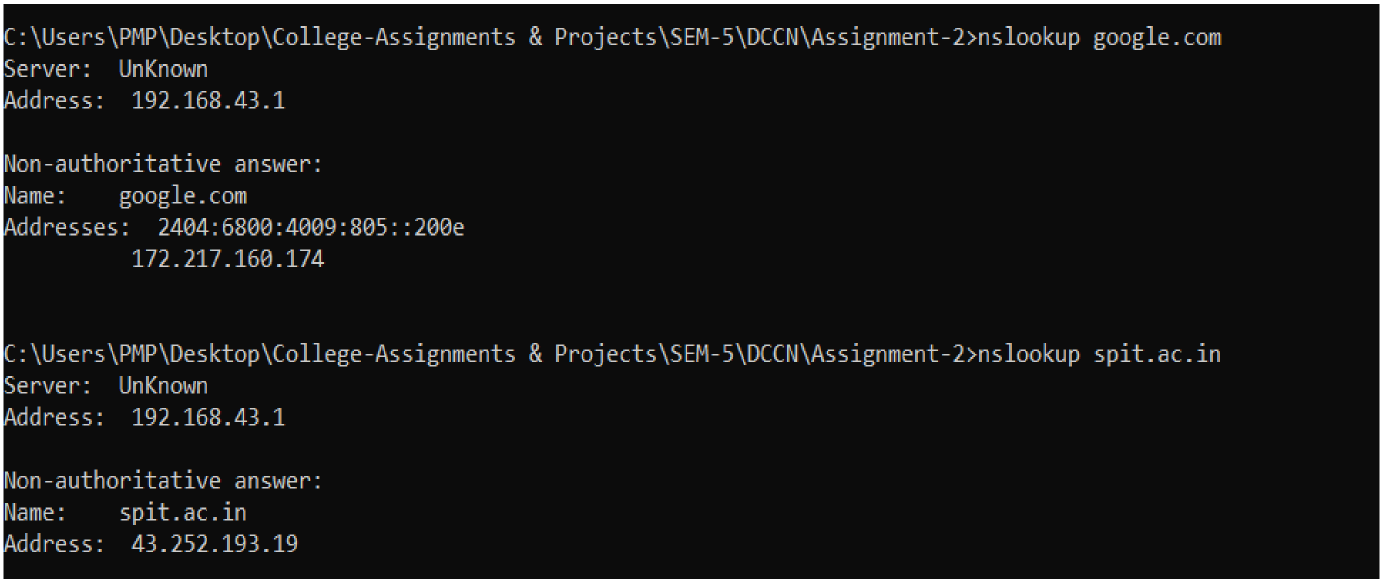
and have got the below observation.



```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>ping -l 100 www.ox.ac.uk

Pinging www.ox.ac.uk [151.101.130.133] with 100 bytes of data:
Reply from 151.101.130.133: bytes=100 time=133ms TTL=52
Reply from 151.101.130.133: bytes=100 time=54ms TTL=52
Reply from 151.101.130.133: bytes=100 time=324ms TTL=52
Reply from 151.101.130.133: bytes=100 time=56ms TTL=52

Ping statistics for 151.101.130.133:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 54ms, Maximum = 324ms, Average = 141ms

C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>ping -l 100 www.berkeley.edu

Pinging www-production-1113102805.us-west-2.elb.amazonaws.com [2600:1f14:436:7800:4110:c28c:3c8b:7aa5] with 100 bytes of data:
Reply from 2600:1f14:436:7800:4110:c28c:3c8b:7aa5: time=427ms
Reply from 2600:1f14:436:7800:4110:c28c:3c8b:7aa5: time=384ms
Reply from 2600:1f14:436:7800:4110:c28c:3c8b:7aa5: time=397ms
Reply from 2600:1f14:436:7800:4110:c28c:3c8b:7aa5: time=363ms

Ping statistics for 2600:1f14:436:7800:4110:c28c:3c8b:7aa5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 363ms, Maximum = 427ms, Average = 392ms

C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>ping -l 100 www.uw.edu

Pinging www.washington.edu [128.95.155.198] with 100 bytes of data:
Reply from 128.95.155.198: bytes=100 time=1158ms TTL=43
Reply from 128.95.155.198: bytes=100 time=373ms TTL=43
Reply from 128.95.155.198: bytes=100 time=326ms TTL=43
Reply from 128.95.155.198: bytes=100 time=635ms TTL=43

Ping statistics for 128.95.155.198:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 326ms, Maximum = 1158ms, Average = 623ms
```

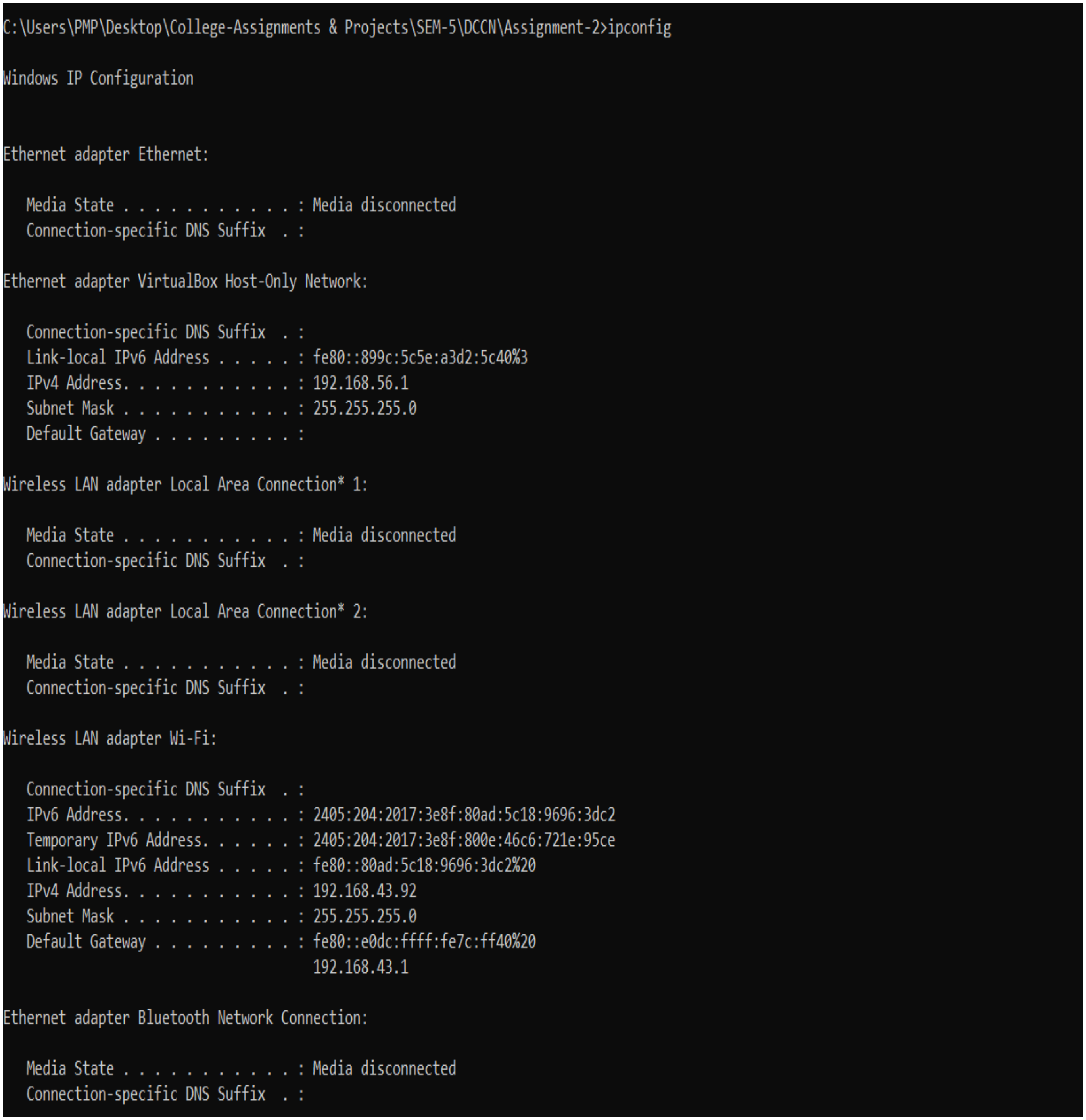
As seen from the above we see that as the distance from my location to the pinged place location increases the TTL time also increases. Like getting response from www.uk.edu (in United Kingdom) is faster then getting response from www.uw.edu (University of Washington). Hence it can be concluded that distance is a major factor in getting in the repsonse and has good impact on the response time.

nslookup — The command nslookup <host> will do a DNS query to find and report the IP address (or addresses) for a domain name or the domain name corresponding to an IP address. To do this, it contacts a "DNS server." Default DNS servers are part of a computer's network configuration. (For a static IP address in Linux, they are configured in the file /etc/network/interfaces that you encountered in the last lab.) You can specify a different DNS server to be used by nslookup by adding the server name or IP address to the command: nslookup <host> <server>



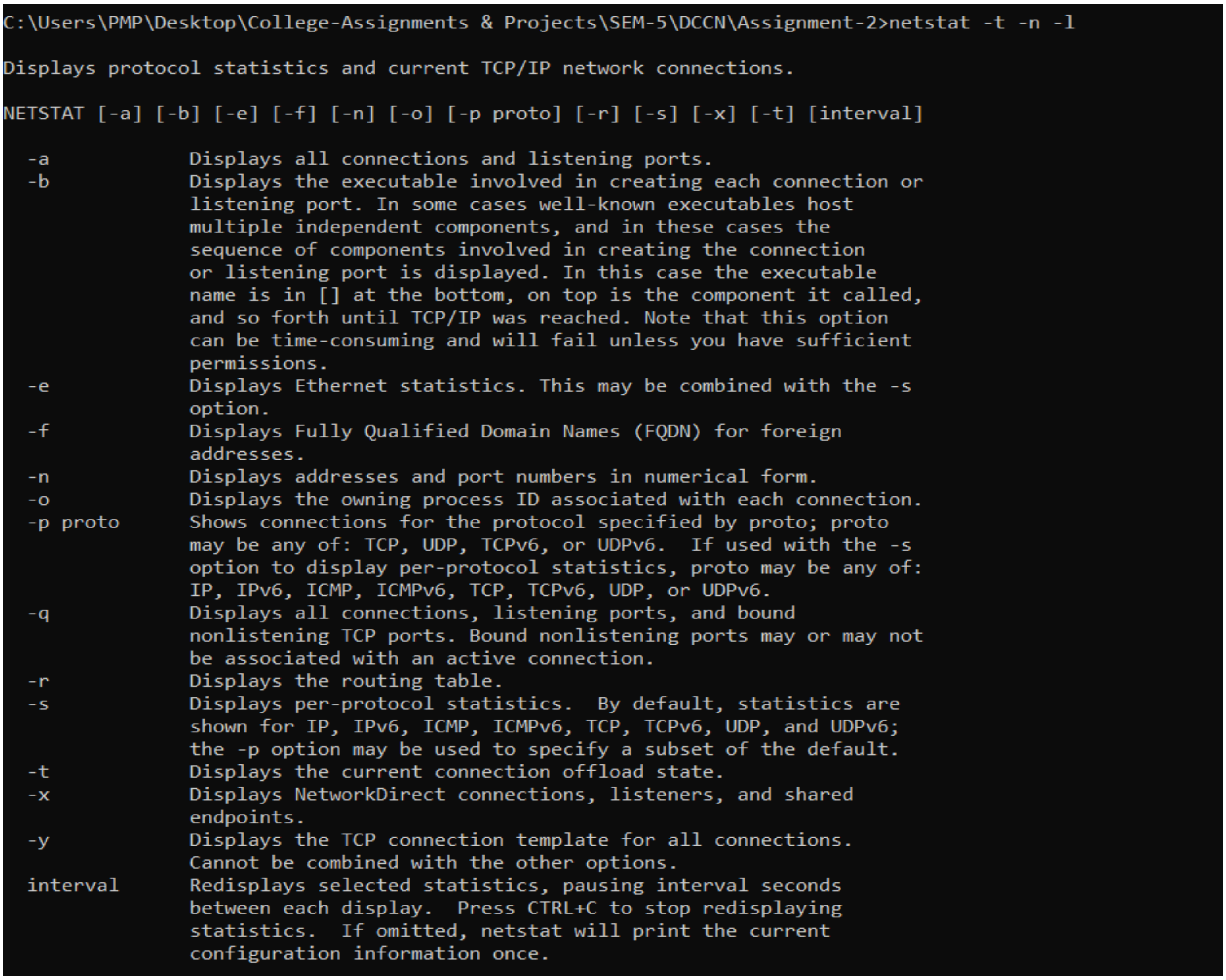
```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>nslookup google.com
Server:  UnKnown
Address:  192.168.43.1

Non-authoritative answer:
Name:    google.com
Addresses:  2404:6800:4009:805::200e
          172.217.160.174


C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>nslookup spit.ac.in
Server:  UnKnown
Address:  192.168.43.1

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

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!!!.)



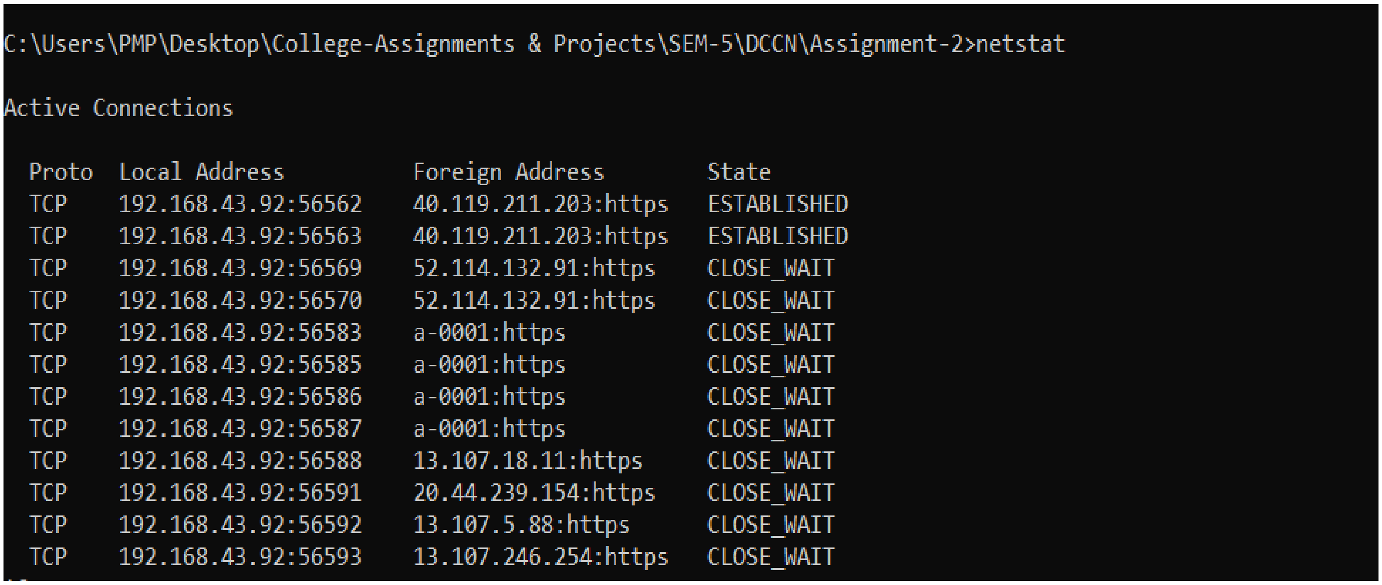
```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::899c:5c5e:a3d2:5c40%3
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2405:204:2017:3e8f:80ad:5c18:9696:3dc2
   Temporary IPv6 Address. . . . . . : 2405:204:2017:3e8f:800e:46c6:721e:95ce
   Link-local IPv6 Address . . . . . : fe80::80ad:5c18:9696:3dc2%20
   IPv4 Address. . . . . . . . . . . : 192.168.43.92
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::e0dc:ffff:fe7c:ff40%20
                                       192.168.43.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

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.).



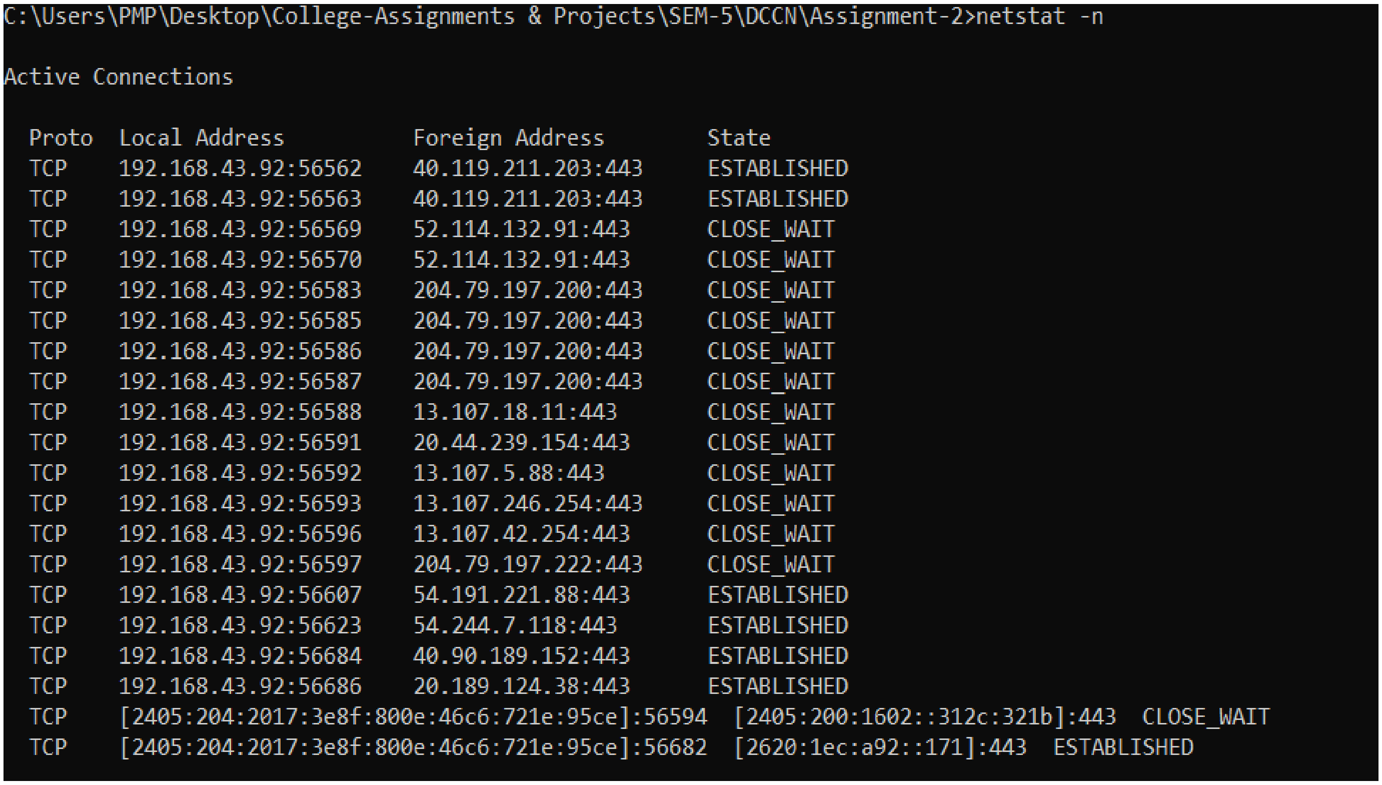
```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>netstat -t -n -l

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]

  -a            Displays all connections and listening ports.
  -b            Displays the executable involved in creating each connection or
                listening port. In some cases well-known executables host
                multiple independent components, and in these cases the
                sequence of components involved in creating the connection
                or listening port is displayed. In this case the executable
                name is in [] at the bottom, on top is the component it called,
                and so forth until TCP/IP was reached. Note that this option
                can be time-consuming and will fail unless you have sufficient
                permissions.
  -e            Displays Ethernet statistics. This may be combined with the -s
                option.
  -f            Displays Fully Qualified Domain Names (FQDN) for foreign
                addresses.
  -n            Displays addresses and port numbers in numerical form.
  -o            Displays the owning process ID associated with each connection.
  -p proto      Shows connections for the protocol specified by proto; proto
                may be any of: TCP, UDP, TCPv6, or UDPv6.  If used with the -s
                option to display per-protocol statistics, proto may be any of:
                IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
  -q            Displays all connections, listening ports, and bound
                nonlistening TCP ports. Bound nonlistening ports may or may not
                be associated with an active connection.
  -r            Displays the routing table.
  -s            Displays per-protocol statistics.  By default, statistics are
                shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;
                the -p option may be used to specify a subset of the default.
  -t            Displays the current connection offload state.
  -x            Displays NetworkDirect connections, listeners, and shared
                endpoints.
  -y            Displays the TCP connection template for all connections.
                Cannot be combined with the other options.
  interval      Redisplays selected statistics, pausing interval seconds
                between each display.  Press CTRL+C to stop redisplaying
                statistics.  If omitted, netstat will print the current
                configuration information once.
```

The **netstat command** displays that what is the network status and protocol statistics.



```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.43.92:56562    40.119.211.203:https   ESTABLISHED
  TCP    192.168.43.92:56563    40.119.211.203:https   ESTABLISHED
  TCP    192.168.43.92:56569    52.114.132.91:https    CLOSE_WAIT
  TCP    192.168.43.92:56570    52.114.132.91:https    CLOSE_WAIT
  TCP    192.168.43.92:56583    a-0001:https           CLOSE_WAIT
  TCP    192.168.43.92:56585    a-0001:https           CLOSE_WAIT
  TCP    192.168.43.92:56586    a-0001:https           CLOSE_WAIT
  TCP    192.168.43.92:56587    a-0001:https           CLOSE_WAIT
  TCP    192.168.43.92:56588    13.107.18.11:https     CLOSE_WAIT
  TCP    192.168.43.92:56591    20.44.239.154:https    CLOSE_WAIT
  TCP    192.168.43.92:56592    13.107.5.88:https      CLOSE_WAIT
  TCP    192.168.43.92:56593    13.107.246.254:https   CLOSE_WAIT
```

Displays active TCP connections, however, addresses and port numbers are expressed numerically and no attempt is made to determine names.



```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.43.92:56562    40.119.211.203:443     ESTABLISHED
  TCP    192.168.43.92:56563    40.119.211.203:443     ESTABLISHED
  TCP    192.168.43.92:56569    52.114.132.91:443      CLOSE_WAIT
  TCP    192.168.43.92:56570    52.114.132.91:443      CLOSE_WAIT
  TCP    192.168.43.92:56583    204.79.197.200:443     CLOSE_WAIT
  TCP    192.168.43.92:56585    204.79.197.200:443     CLOSE_WAIT
  TCP    192.168.43.92:56586    204.79.197.200:443     CLOSE_WAIT
  TCP    192.168.43.92:56587    204.79.197.200:443     CLOSE_WAIT
  TCP    192.168.43.92:56588    13.107.18.11:443       CLOSE_WAIT
  TCP    192.168.43.92:56591    20.44.239.154:443      CLOSE_WAIT
  TCP    192.168.43.92:56592    13.107.5.88:443        CLOSE_WAIT
  TCP    192.168.43.92:56593    13.107.246.254:443     CLOSE_WAIT
  TCP    192.168.43.92:56596    13.107.42.254:443      CLOSE_WAIT
  TCP    192.168.43.92:56597    204.79.197.222:443     CLOSE_WAIT
  TCP    192.168.43.92:56607    54.191.221.88:443      ESTABLISHED
  TCP    192.168.43.92:56623    54.244.7.118:443       ESTABLISHED
  TCP    192.168.43.92:56684    40.90.189.152:443      ESTABLISHED
  TCP    192.168.43.92:56686    20.189.124.38:443      ESTABLISHED
  TCP    [2405:204:2017:3e8f:800e:46c6:721e:95ce]:56594  [2405:200:1602::312c:321b]:443  CLOSE_WAIT
  TCP    [2405:204:2017:3e8f:800e:46c6:721e:95ce]:56682  [2620:1ec:a92::171]:443  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: telnet <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

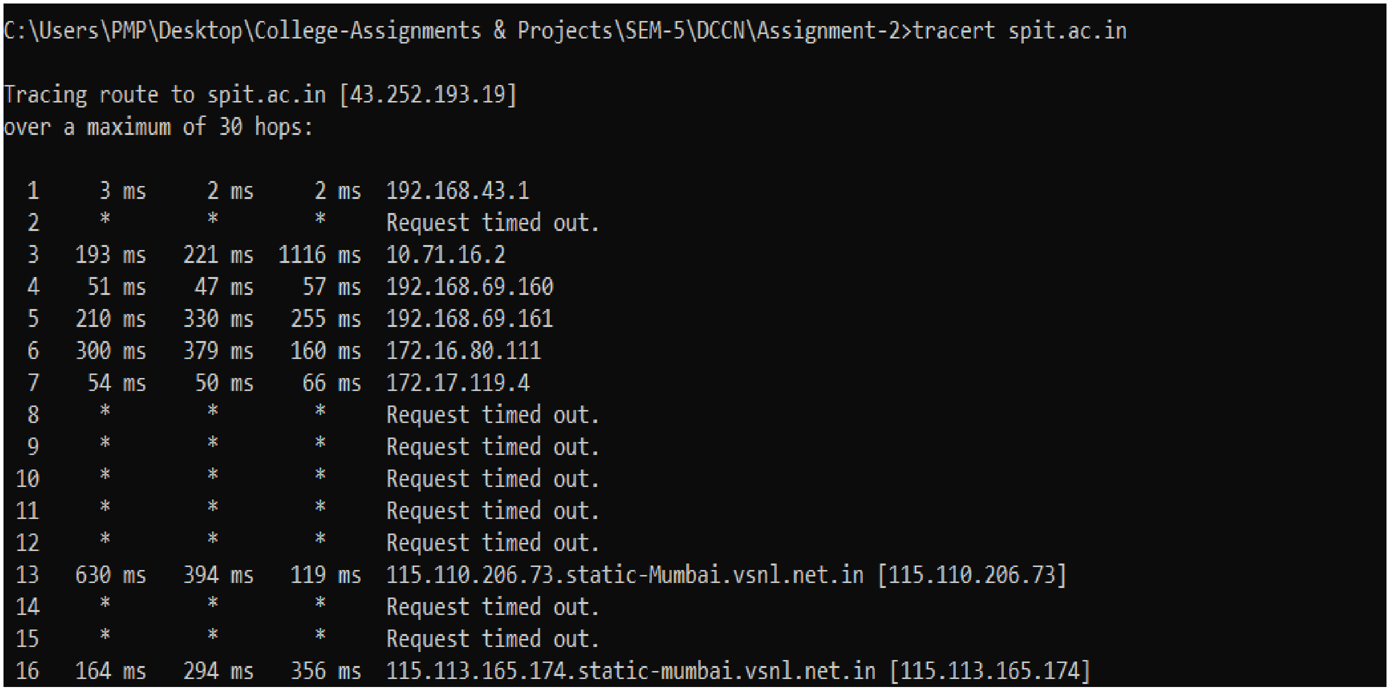
The path taken through a network, can be measured using traceroute. The syntax for the command in Linux is:

```
traceroute <hostname>
```

The syntax in Windows is:

```
tracert <hostname>
```

You can specify either a hostname (e.g., spit.ac.in) or an IP address (e.g., 43.252.193.19).

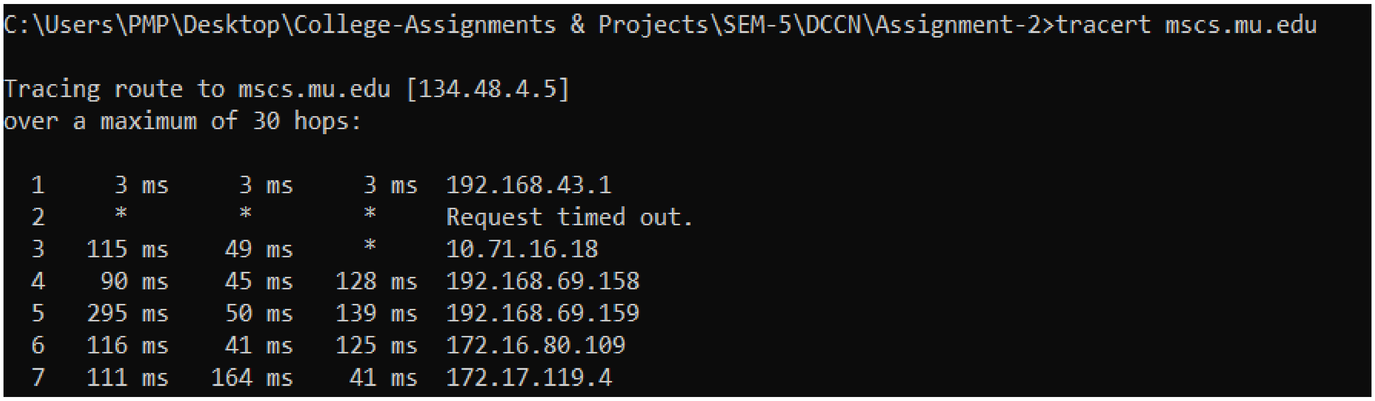


```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>tracert spit.ac.in

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

  1     3 ms     2 ms     2 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3   193 ms   221 ms  1116 ms  10.71.16.2
  4    51 ms    47 ms    57 ms  192.168.69.160
  5   210 ms   330 ms   255 ms  192.168.69.161
  6   300 ms   379 ms   160 ms  172.16.80.111
  7    54 ms    50 ms    66 ms  172.17.119.4
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13   630 ms   394 ms   119 ms  115.110.206.73.static-Mumbai.vsnl.net.in [115.110.206.73]
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16   164 ms   294 ms   356 ms  115.113.165.174.static-mumbai.vsnl.net.in [115.113.165.174]
```

1.2.1 Experiments with Traceroute

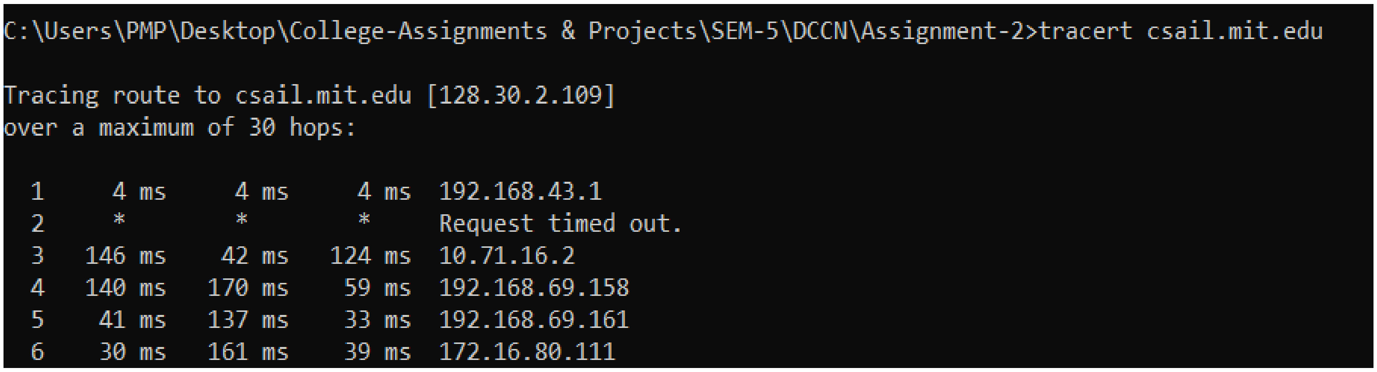
From **your machine** traceroute to the following hosts:

1. mscs.mu.edu



```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>tracert mscs.mu.edu

Tracing route to mscs.mu.edu [134.48.4.5]
over a maximum of 30 hops:

  1     3 ms     3 ms     3 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3   115 ms    49 ms     *     10.71.16.18
  4    90 ms    45 ms   128 ms  192.168.69.158
  5   295 ms    50 ms   139 ms  192.168.69.159
  6   116 ms    41 ms   125 ms  172.16.80.109
  7   111 ms   164 ms    41 ms  172.17.119.4
```

2. csail.mit.edu



```
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>tracert csail.mit.edu

Tracing route to csail.mit.edu [128.30.2.109]
over a maximum of 30 hops:

  1     4 ms     4 ms     4 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3   146 ms    42 ms   124 ms  10.71.16.2
  4   140 ms   170 ms    59 ms  192.168.69.158
  5    41 ms   137 ms    33 ms  192.168.69.161
  6    30 ms   161 ms    39 ms  172.16.80.111
```

<u>Exercise 2:</u> (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.

Output in the text file

1. math.hsw.edu

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

  1     *        6 ms     4 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3   120 ms    60 ms   124 ms  10.71.16.2
  4   769 ms    79 ms   138 ms  192.168.69.158
  5   116 ms    42 ms    51 ms  192.168.69.161
```

2. www.hws.edu

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

  1     4 ms     4 ms     3 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3   771 ms    87 ms   126 ms  10.71.16.2
  4   157 ms    55 ms   116 ms  192.168.69.162
  5   135 ms   154 ms    61 ms  192.168.69.165
  6    77 ms    53 ms   130 ms  172.16.80.113
```

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.

Original Path –

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

  1     *        6 ms     4 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3   120 ms    60 ms   124 ms  10.71.16.2
  4   769 ms    79 ms   138 ms  192.168.69.158
  5   116 ms    42 ms    51 ms  192.168.69.161
```

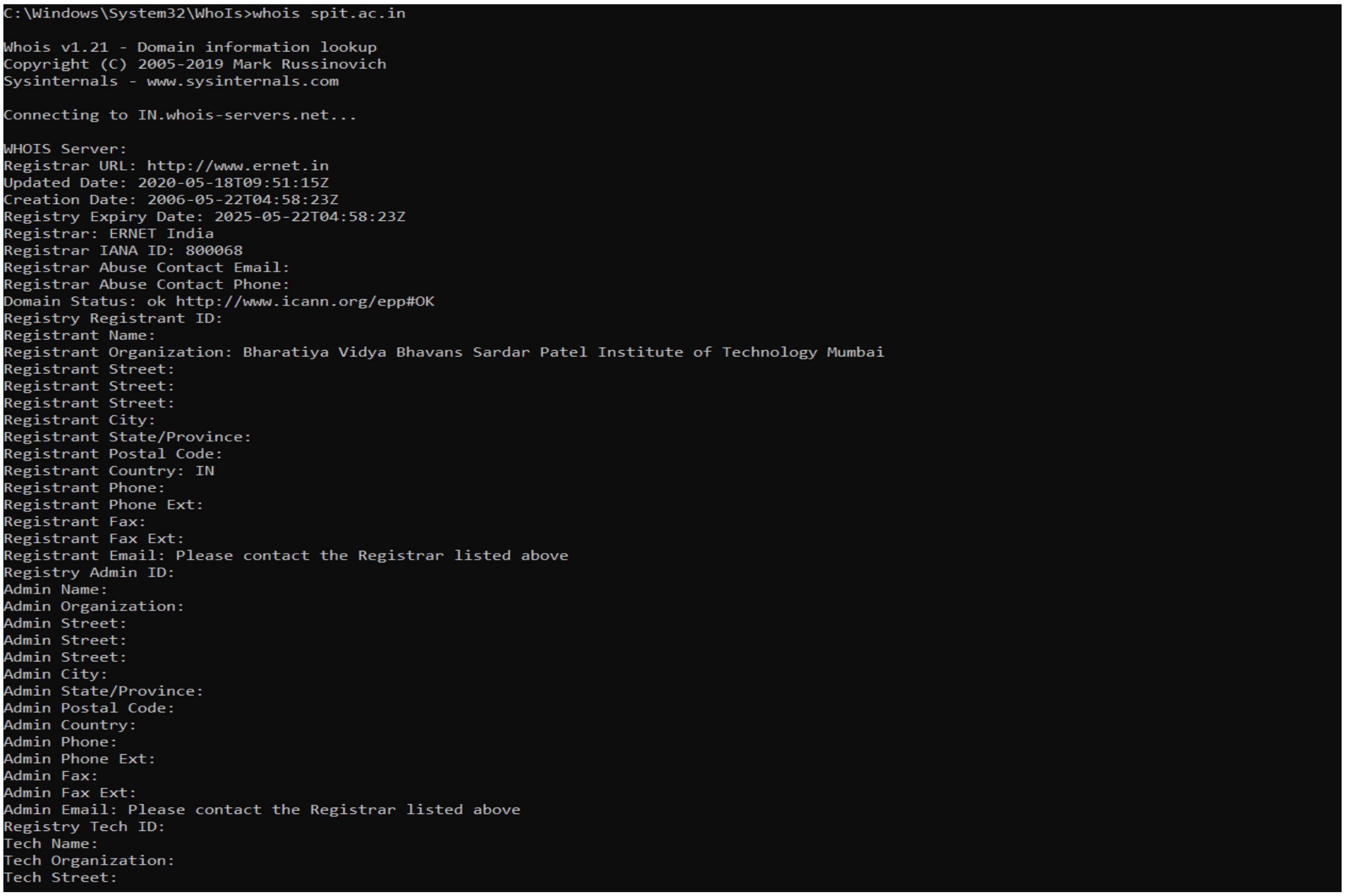
New Path –

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

  1     5 ms     3 ms     3 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3    64 ms    36 ms    56 ms  10.71.16.2
  4    62 ms    50 ms    56 ms  192.168.69.160
  5    51 ms    47 ms    59 ms  192.168.69.161
```

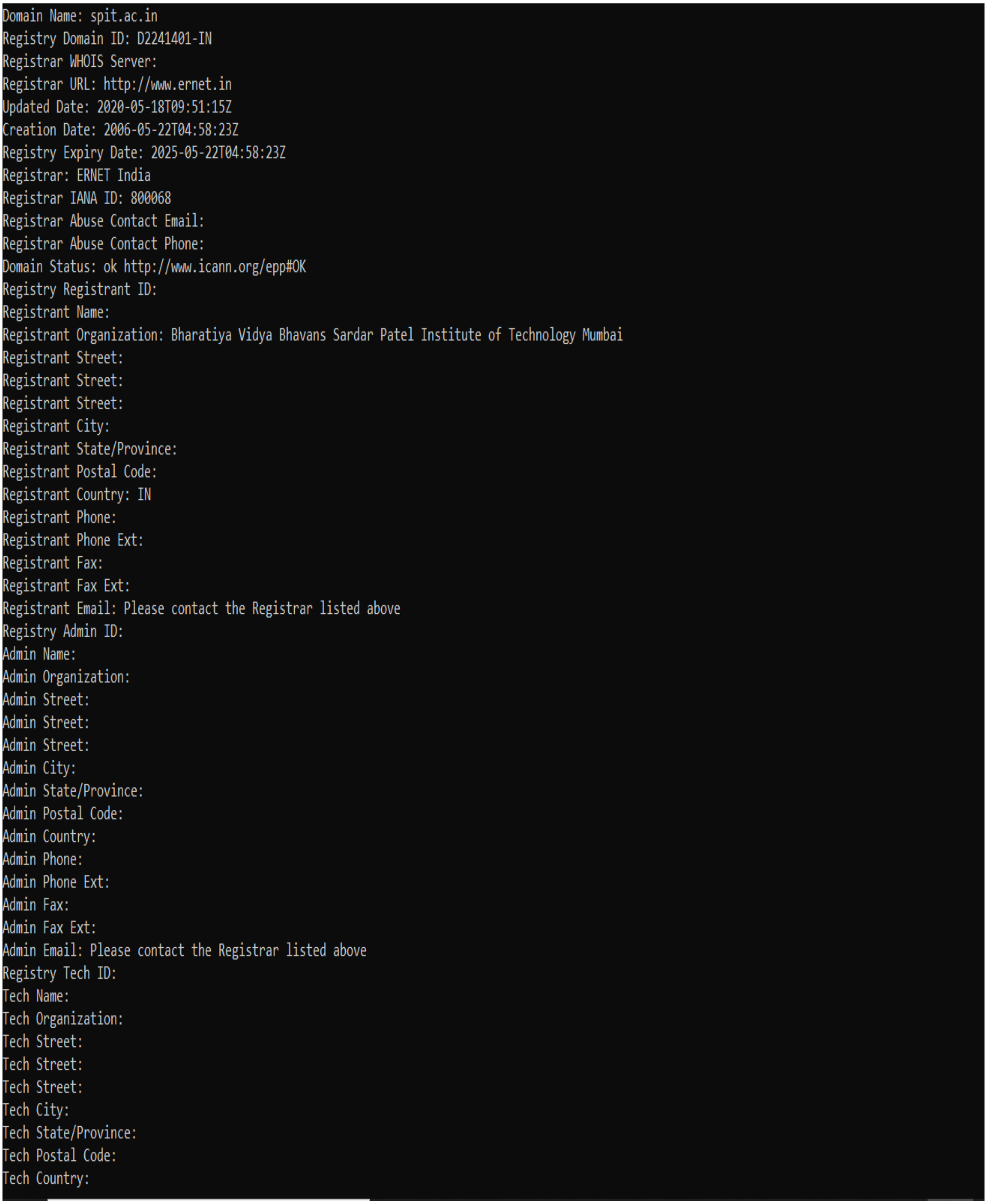
Whois — The *whois* command can give detailed information about domain names and IP addresses. If it is not installed on the computers then install it with command sudo apt-get install whois in. *Whois* can tell you what organization owns or is responsible for the name or address and where to contact 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.
I used whois on spit.ac.in and got the following output



```
C:\Windows\System32\WhoIs>whois 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...

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:
```



```
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:
```

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.

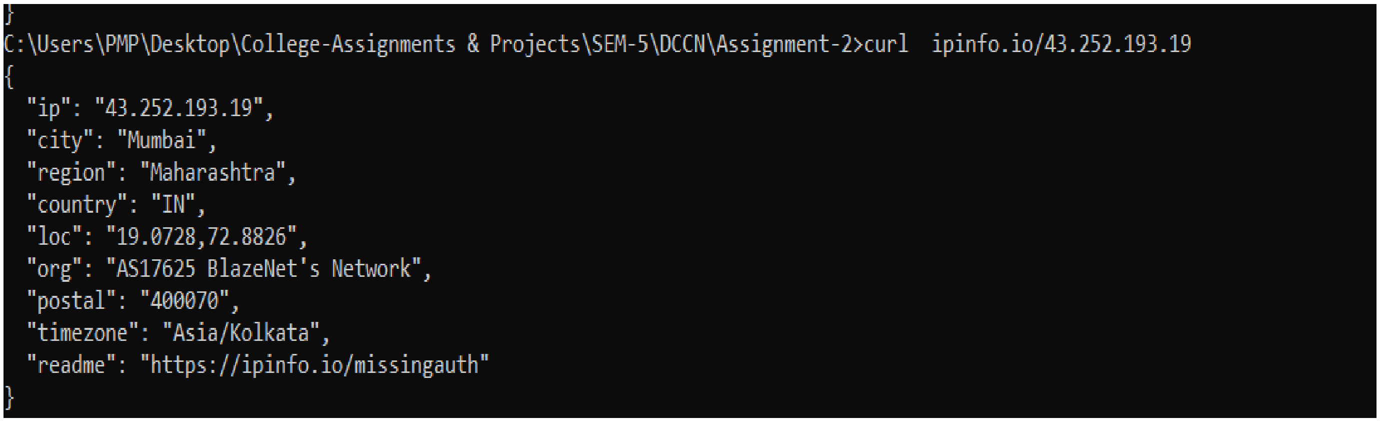
The outside IP address of spit.ac.in is **43.252.193.19**. I found it out by using "**tracert spit.ac.in**" command or we can also use ping command like **"ping spit.ac.in"**.

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

I have found the location of spit with the help of I.P. address that I got by "tracert spit.ac.in" command



```
}
C:\Users\PMP\Desktop\College-Assignments & Projects\SEM-5\DCCN\Assignment-2>curl  ipinfo.io/43.252.193.19
{
  "ip": "43.252.193.19",
  "city": "Mumbai",
  "region": "Maharashtra",
  "country": "IN",
  "loc": "19.0728,72.8826",
  "org": "AS17625 BlazeNet's Network",
  "postal": "400070",
  "timezone": "Asia/Kolkata",
  "readme": "https://ipinfo.io/missingauth"
}
```

Conclusion - After completing the above experiment, I have understood the basics of the networking utilities that we can implement. I came to know about various commands like ping, tracert, ipconfig etc by which we can easily monitor our networking.

References –

1. https://new.edmodo.com/view-office-online/view/1392187432/doc/CEL%2051%20lab%202