CS 349 NETWORKS LAB
Assignment 1
PRIYANSHU SINGH 170101049

Q.1

1. Use **–c** flag with ping command. Ex: 'ping iitg.ac.in -c 4' will send 4 ECHO_REQUEST packets and waits for 4 ECHO_REPLY packets until the timeout expires.
2. Use **-i** flag with ping command. Ex: 'ping iitg.ac.in -i 4' will send request packets after every 4 seconds. Setting time interval between sending packets less than 0.2 seconds require super-user privileges.
3. Use **–l** flag with ping command (can be used by super-user only). Ex: 'sudo ping –l 10 yahoo.com ' will flood the network with 10 request packets. Normal users cannot send more than 3 packets at once.
4. ping –s <packetsize> is used to specify packet size. If the packet size is **32 bytes**, the total packet size would be **60 bytes** due to 8 bytes of ICMP header and 20 bytes of IP header.

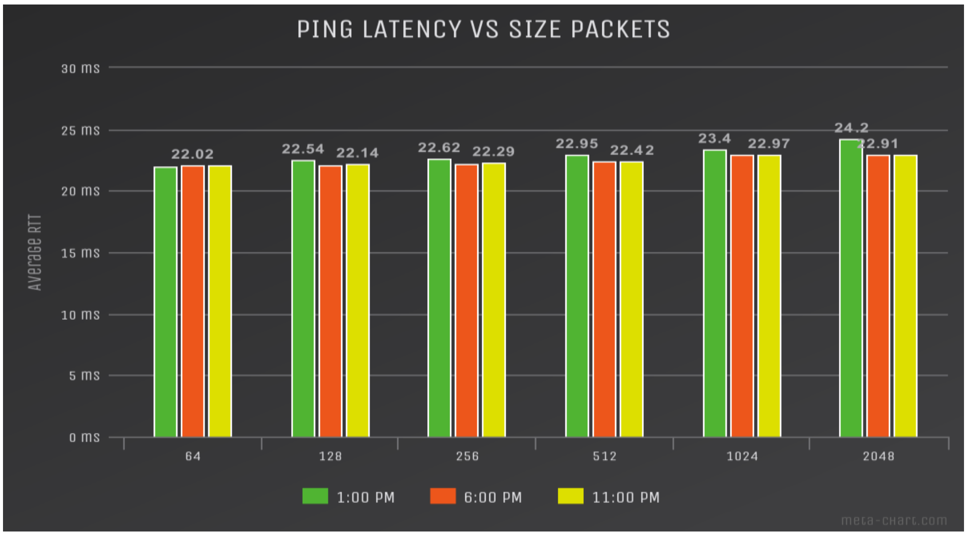
Q.2

The table below shows the various hosts chosen and their Average RTT delay (in milliseconds) at various times in a day. A ping utility located in New Jersey(*www.spfld.com*) was used for collecting this data.

Host	Location	1:00 PM	6:00 PM	11:00 PM
Facebook.com	Ireland	21.991	22.020	22.060
Facebook.in	Ireland	21.236	21.417	21.502
Bookmyshow.com	Kansas, USA	6.471	6.021	6.020
Libgen.is	Seychelles	105.157	109.043	138.948
Youtube.com	California, USA	40.087	39.941	47.737
Google.com	California, USA	39.823	40.276	40.465

- Packet loss greater than 0% was not observed as no packet was lost. However, packets maybe lost in case some firewall is deployed on server or when the network is down.
- There seems to be no obvious correlation between geographical distance and ping latency. Common logic suggests that the lesser the distance of server from our network the less is the ping latency. Such a connection can be seen with the fact that *Libgen.is* has more ping latency that *Google.com* but a reversal of this happens when we compare *Facebook.com* with *Google.com*. Hence there is only a **weak correlation** between them.

Host chosen for pinging with different data size packets: - *facebook.com*



We can see from the plot that the ping latency is proportional to the packet size. The more is the packet size; the more is the transmission time and hence more latency. Also, more users are active in the afternoon than during evening or late night at *facebook.com*

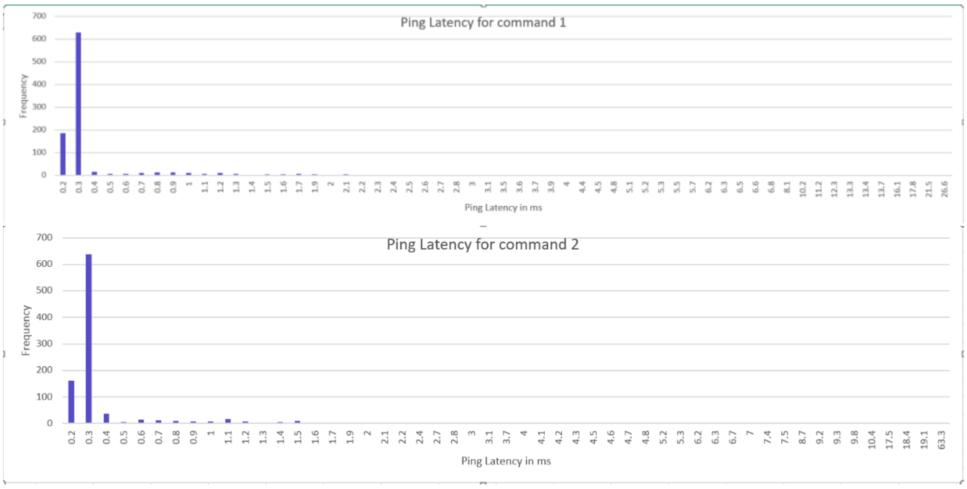
Q.3

IP address selected: - *172.16.116.136*

1. Packet loss in both commands in 0% (i.e. no packet is lost).
2. Table shown below captures the details:

S.No	Command	Minimum Lat	Maximum Lat	Average Lat	Median Lat
1	ping –c 1000 -n 172.16.116.136	0.171 ms	26.673 ms	0.688 ms	0.303 ms
2	ping -c 1000 -p ff00 172.16.116.136	0.158 ms	63.398 ms	0.698 ms	0.305 ms

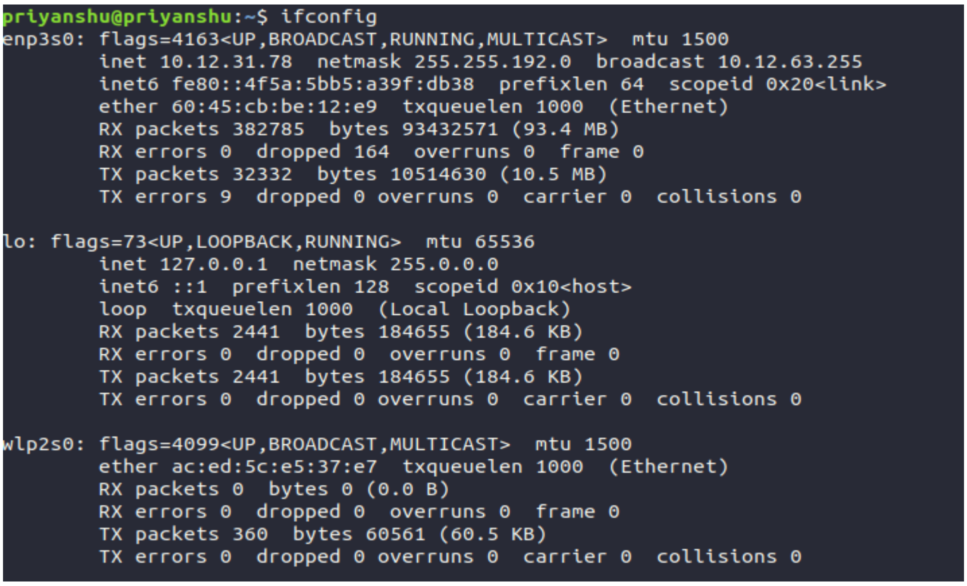
3. The distribution graphs are shown below (Ping Latency vs frequency) where latency is rounded to 1 decimal place:



4. With **–n** flag with ping, there would be no need for **DNS lookup** for domain name resolution leading to less time spent at each request. With **–p** flag we are giving a pattern that is to be sent in the packets. In this case the patter in ff00(1111111100000000). This pattern does not have enough transitions (for clock synchronization) and is likely to face problem in transmission. Hence it has on average more ping latency.

Q.4

1. **IFCONFIG** stands for Interface configuration and it is used to view and change the configuration of kernel-network interfaces on your system. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.



```
priyanshu@priyanshu:~$ ifconfig
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.12.31.78  netmask 255.255.192.0  broadcast 10.12.63.255
        inet6 fe80::4f5a:5bb5:a39f:db38  prefixlen 64  scopeid 0x20<link>
        ether 60:45:cb:be:12:e9  txqueuelen 1000  (Ethernet)
        RX packets 382785  bytes 93432571 (93.4 MB)
        RX errors 0  dropped 164  overruns 0  frame 0
        TX packets 32332  bytes 10514630 (10.5 MB)
        TX errors 9  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2441  bytes 184655 (184.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2441  bytes 184655 (184.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether ac:ed:5c:e5:37:e7  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 360  bytes 60561 (60.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

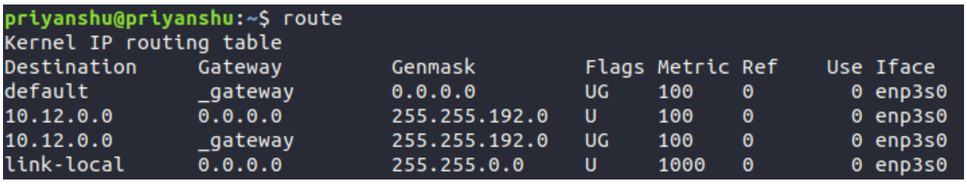
EXPLANATION

- Here, **eth0**, **lo** and **wlan0** are the names of the active network interfaces on the system.
- **enp3s0** is the first Ethernet interface. Additional Ethernet interfaces would be named **enp3s1**, **enp3s2**, etc.
- **lo** is the loopback interface. This is a special network interface that the system uses to communicate with itself.
- **wlp2s0** is the name of the first wireless network interface on the system. Additional wireless interfaces would be named **wlp2s1**, **wlp2s2**, etc.
- **UP** means that network interface is activated (with address and routing tables) and is accessible to the IP layer.
- **BROADCAST** means that interface supports broadcasting (and can hence obtain an IP address using DHCP).
- **RUNNING** signifies that the network driver has been loaded and has initialized the interface.

- **MULTICAST** tells us that multicasting support is enabled on this interface.
- **mtu** sets the maximum transfer unit of an interface. This setting is used to limit the maximum packet size transferred by the interface.
- **inet** is the IPv4 address assigned to the interface whereas **inet6** is the IPv6 address assigned to the interface.
- A **netmask** is a 32-bit binary mask used to divide an IP address into subnets and specify the network's available hosts.
- A **broadcast address** is a network address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.
- **txqueuelen length** sets the length of the transmit queue of the device. A transmission queue is a local queue that is used when a queue manager forwards messages to a remote queue manager through a message channel.
- **RX packets** is the number of packets received via an interface while **RX bytes** is the number of bytes received via an interface.
- **frame** counts misaligned frames; it means frames with a length not divisible by 8 and are hence discarded.
- **RX errors** are an aggregation of the total number of packets received with errors. This includes too-long-frames errors, ring-buffer overflow errors, crc errors, frame alignment errors, FIFO overruns, and missed packets.
- **RX overruns** count the number of times when there is FIFO overruns, caused by the rate at which the buffer gets full and the kernel isn't able to empty it.
- **dropped** counts things like unintended VLAN tags or receiving IPv6 frames when the interface is not configured for IPv6.
- **TX packets** is the number of packets transmitted via an interface while **TX bytes** is the number of bytes transmitted via an interface.
- **collisions** are the number of transmissions terminated due to CSMA/CD (Carrier Sense Multiple Access with Collision Detection).
- **Carrier** errors occur when there is a problem with the modulation of your signal.

2. Options available with the ifconfig command are: -
    - ifconfig –a: - displays all interfaces which are currently available, even if down.
    - ifconfig enp3s0 up: - Activate the network interface **enp3s0**.
    - ifconfig enp3s0 down: - Deactivate the network interface **enp3s0**.
    - ifconfig enp3s0 mtu 3000: - To change mtu of the interface to the desired value.
    - ifconfig –s: - Displays a short list of interfaces.

3. **ROUTE** command is used to show/manipulate the IP routing table. It is primarily used to setup static routes to specific host or networks via an interface.

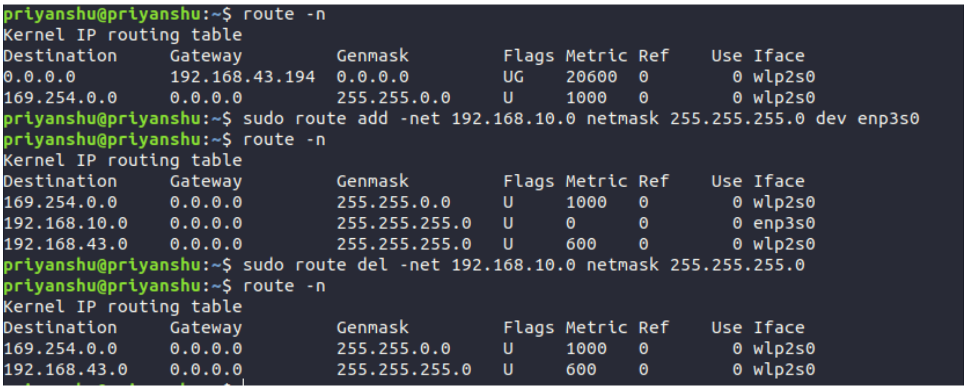


```
priyanshu@priyanshu:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp3s0
10.12.0.0       0.0.0.0         255.255.192.0   U     100    0        0 enp3s0
10.12.0.0       _gateway        255.255.192.0   UG    100    0        0 enp3s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp3s0
```

EXPLANATION

- The **Destination** column identifies the destination network.
- The **Gateway** column identifies the defined gateway for the specified network. The 0.0.0.0 means that the network is locally connected on that interface and no more hops are needed to get to it.
- The **Genmask** column shows the netmask for the network.
- Under the Flags section, the **U** flag means the route is up, **G** means Gateway is used, and **H** means target is a host.
- **Metric** is the distance to the target (usually counted in hops). It is not used by recent kernels but may be needed by routing daemons.
- **Ref** is the number of references to this route. (Not used in the Linux kernel.)
- **Use** if the count of lookups for the route.
- **Iface** is the Interface to which packets for this route will be sent.

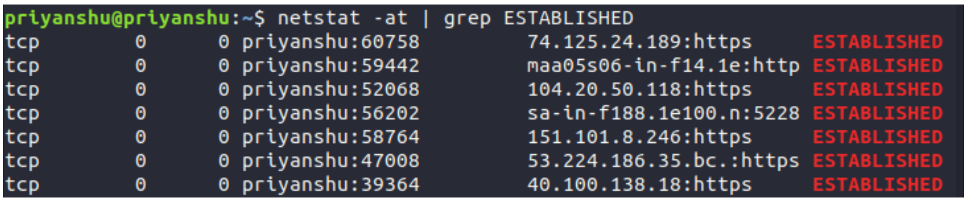
4. Options available with the route command are: -
   - route –n: - shows numerical addresses instead of trying to determine symbolic host names.
   - sudo route del default: - deletes the current default route, which is labelled “default” or 0.0.0.0 in destination field of the routing table.
   - sudo route add -net <IP> netmask <subnet> dev <Iface>: - This will add the network IP with the given subnet to the interface Iface.
   - sudo route del -net <IP> netmask <subnet>: - deletes the routing table entry for the given IP and subnet.



```
priyanshu@priyanshu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.43.194  0.0.0.0         UG    20600  0        0 wlp2s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlp2s0
priyanshu@priyanshu:~$ sudo route add -net 192.168.10.0 netmask 255.255.255.0 dev enp3s0
priyanshu@priyanshu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlp2s0
192.168.10.0    0.0.0.0         255.255.255.0   U     0      0        0 enp3s0
192.168.43.0    0.0.0.0         255.255.255.0   U     600    0        0 wlp2s0
priyanshu@priyanshu:~$ sudo route del -net 192.168.10.0 netmask 255.255.255.0
priyanshu@priyanshu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlp2s0
192.168.43.0    0.0.0.0         255.255.255.0   U     600    0        0 wlp2s0
```

Q.5

1. Netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc. It is used in network troubleshooting and performance measurement. Some of the uses of netstat command are:
   - Listing all the LISTENING ports of TCP and UDP connections. (*netstat –a*)
   - Showing Statistics by cprotocol. (*netstat –s*)
   - Displaying service name by PID (*netstat –tp*)
2. *netstat –at | grep ESTABLISHED* is used to list all established TCP connections.

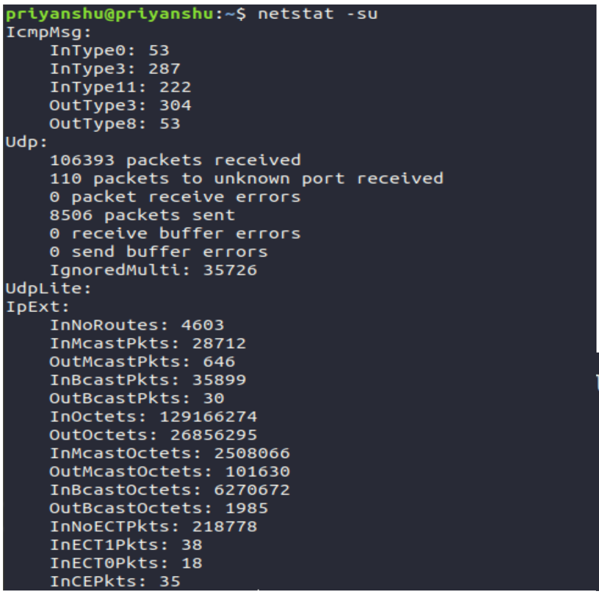


```
priyanshu@priyanshu:~$ netstat -at | grep ESTABLISHED
tcp        0      0 priyanshu:60758         74.125.24.189:https     ESTABLISHED
tcp        0      0 priyanshu:59442         maa05s06-in-f14.1e:http ESTABLISHED
tcp        0      0 priyanshu:52068         104.20.50.118:https     ESTABLISHED
tcp        0      0 priyanshu:56202         sa-in-f188.1e100.n:5228 ESTABLISHED
tcp        0      0 priyanshu:58764         151.101.8.246:https     ESTABLISHED
tcp        0      0 priyanshu:47008         53.224.186.35.bc.:https ESTABLISHED
tcp        0      0 priyanshu:39364         40.100.138.18:https     ESTABLISHED
```

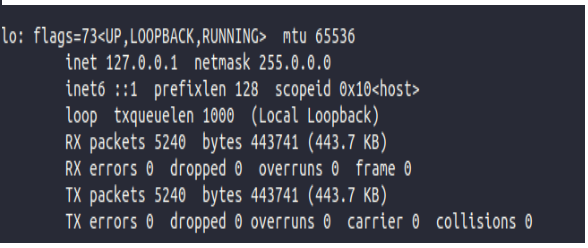
EXPLANATION

- **The “Proto” column (1st column)** tells us if the socket listed is TCP or UDP.
- **The “Recv-Q” and “Send-Q” columns (2nd and 3rd column)** tell us how much data is in the queue for that socket, waiting to be read (Recv-Q) or sent (Send-Q). In short: if this is 0, everything’s ok, if there are non-zero values anywhere, there may be trouble.
- **The “Local Address” and “Foreign Address” columns (4th and 5th column)** tell which hosts and ports the listed sockets are connected. The local end is always on the computer on which you’re running netstat and the foreign end is about the other computer.
- **The “State” column** tells in which state the listed sockets are. The TCP protocol defines states, including “LISTEN” (wait for some external computer to contact us) and “ESTABLISHED” (ready for communication).“CLOSE WAIT” means that the foreign or remote machine has already closed the connection, but that the local program somehow hasn’t followed suit. “TIME_WAIT” tells that the local machine has closed the connection.

3. *netstat –r* displays the kernel routing table. (shows the same result as the route command). Here **MSS** displays the maximum segment size of datagram. The **Window** is the maximum amount of data the system will accept in a single burst from a remote host. The acronym **irtt** stands for “initial round trip time.”
4. *netstat -i* is used to display the status of all network interfaces. My PC currently has 3 network interfaces which are **lo** (loopback interface), **enp3s0**(ethernet interface) and **wlp2s0**(WIFI interface).
5. *netstat –su* command is used for showing statistics of all UDP connections.



```
priyanshu@priyanshu:~$ netstat -su
IcmpMsg:
    InType0: 53
    InType3: 287
    InType11: 222
    OutType3: 304
    OutType8: 53
Udp:
    106393 packets received
    110 packets to unknown port received
    0 packet receive errors
    8506 packets sent
    0 receive buffer errors
    0 send buffer errors
    IgnoredMulti: 35726
UdpLite:
IpExt:
    InNoRoutes: 4603
    InMcastPkts: 28712
    OutMcastPkts: 646
    InBcastPkts: 35899
    OutBcastPkts: 30
    InOctets: 129166274
    OutOctets: 26856295
    InMcastOctets: 2508066
    OutMcastOctets: 101630
    InBcastOctets: 6270672
    OutBcastOctets: 1985
    InNoECTPkts: 218778
    InECT1Pkts: 38
    InECT0Pkts: 18
    InCEPkts: 35
```



```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 5240  bytes 443741 (443.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5240  bytes 443741 (443.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

6. The loopback device is a special, virtual network interface that your computer uses to communicate with itself. It is used mainly for diagnostics and troubleshooting, and to connect to servers running on the local machine. The loopback interface does not represent any actual hardware but exists so applications running on your computer can always connect to servers on the same machine.

Q.6

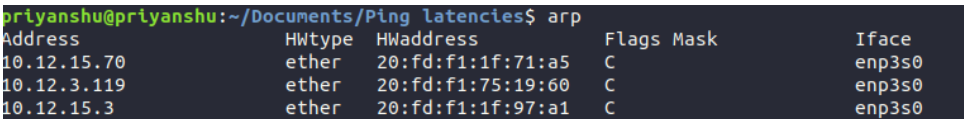
1.

Host	11 PM	1 PM	6PM
Libgen.is	13	13 (1 different from 11 PM)	13 (same as 11 PM)
Facebook.in	14	14 (2 different from 11 PM)	14 (2 different from 11 PM)
Facebook.com	14	14 (all same from 11 PM)	14 (2 different from 11 PM)
Google.com	13	13 (2 different from 11 PM)	13 (1 different from 11 PM)
Youtube.com	13	13 (1 different from 11 PM)	13 (1 different from 11 PM)
Bookmyshow.com	11	11 (6 hops missing)	11 (6 hops missing)

2. Route to same host can change during different times of day due to different congestion in network at different times. Network tries to do **load balancing** and hence the different paths at distinct times.
3. Yes, it is possible that traceroute is not able to find complete paths for some hosts, which was the case with *hotstar.com.* This may be due to some firewall deployed by Hotstar to block out incoming ICMP packets or due to congestion in the network. In cases such as *bookmyshow.com* some hops went missing but we reached the destination anyway.
4. Yes, it is possible for traceroute to find the route to hosts which fail to respond with ping experiment. This is because although both use ICMP, but their mechanisms are different. Ping uses direct ICMP message from A to B and expects reply from B, which may be blocked due to firewall. Tracert works by targeting the final hop but limiting the TTL and waiting for a time exceeded message, and then increasing it by one for the next iteration. Therefore, the response it gets is not an ICMP echo reply to the ICMP echo request from the host along the way, but a time exceeded message from that host.

Q.7

1. Command ***arp*** is used to show the full ARP table. ARP stands for Address Resolution Protocol. This protocol is used by network nodes to match IP addresses to MAC addresses.



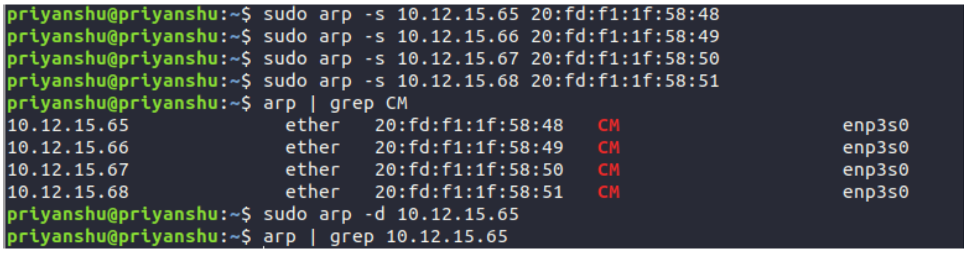
```
priyanshu@priyanshu:~/Documents/Ping latencies$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.12.15.70              ether   20:fd:f1:1f:71:a5   C                     enp3s0
10.12.3.119              ether   20:fd:f1:75:19:60   C                     enp3s0
10.12.15.3               ether   20:fd:f1:1f:97:a1   C                     enp3s0
```

Explanation:

- **HWtype** denotes the type of hardware link between the user’s IP and this IP.
- **HWaddress** denotes the MAC address corresponding to this IP.

- **Flags** are used to describe the status of the table entries, e.g. Flag C is used to denote a complete entry in the ARP table, permanent entries are marked with M etc.
- **Mask** denotes the subnet mask.
- **Iface** denotes the interface corresponding to this table entry.

2. **Deletion** can be done using: '*sudo arp -d <IP Address>*' while addition can be done using the: '*sudo arp –s <IP Address> <MAC Address>*'. ARP entries that are manually added will have a '**M**' flag.

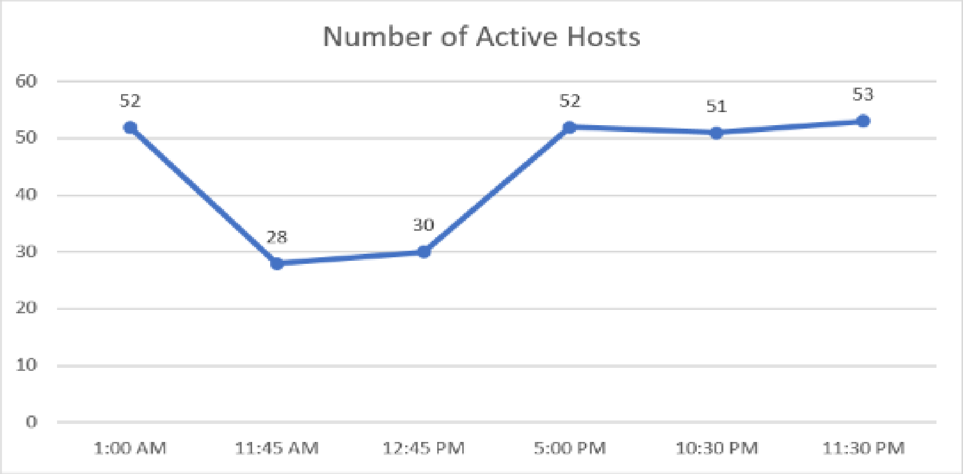


```
priyanshu@priyanshu:~$ sudo arp -s 10.12.15.65 20:fd:f1:1f:58:48
priyanshu@priyanshu:~$ sudo arp -s 10.12.15.66 20:fd:f1:1f:58:49
priyanshu@priyanshu:~$ sudo arp -s 10.12.15.67 20:fd:f1:1f:58:50
priyanshu@priyanshu:~$ sudo arp -s 10.12.15.68 20:fd:f1:1f:58:51
priyanshu@priyanshu:~$ arp | grep CM
10.12.15.65              ether   20:fd:f1:1f:58:48   CM                    enp3s0
10.12.15.66              ether   20:fd:f1:1f:58:49   CM                    enp3s0
10.12.15.67              ether   20:fd:f1:1f:58:50   CM                    enp3s0
10.12.15.68              ether   20:fd:f1:1f:58:51   CM                    enp3s0
priyanshu@priyanshu:~$ sudo arp -d 10.12.15.65
priyanshu@priyanshu:~$ arp | grep 10.12.15.65
```

3. If an ARP entry is not used a specific amount of time called the **ARP timeout** the entry is removed from the caching table. There is no standard value for this amount of time and is usually dependent on vendor. **Static entries** remain in the table **forever** and are never removed. Default ARP timeout for a machine if found by '*cat /proc/sys/net/ipv4/neigh/default/gc_stale_time*'. A trial-and-error method to find out the timeout value is to ping a random host, then check after regular intervals, say 30 sec to see if the entry corresponding to that IP is there in ARP table or not. Using this we can get the approximate value.
4. If two IP addresses map to the same MAC address, then if the two devices are on the same LAN, then there is a problem of ambiguity, but if they are separated by a router or switch, then it won't be an issue as they won't see each other directly. On a subnet, the machines talk to each other with their MAC addresses which uniquely identifies each NIC card. Machines do not know MAC addresses of other machines beforehand. When machine X wants to send a packet to machine Y it only has its IP address. It sends a broadcast following the ARP protocol, asking if any machine has the specified IP address, if so, it receives the corresponding MAC address in the reply, and can start communicating with the other machine.

Q.8

I used command '*nmap –n –sP 172.16.112.0/24*' to check how many users are active in the CSE dept. 256 IP addresses were checked. Following is the trend:



As we can see a smaller number of students are active during morning and afternoon hours. There is lot more students during the evening and the late-night hours suggesting IITG junta is not a morning one.