CS-349 Networks Lab

Assignment 1

By Mayank Wadhwani, Roll No: 170101038

Question 1: Ping

a) The option required to specify the number of echo requests to send with ping command:- **-c**.
b) The option required to set the time interval (in seconds) between two ping ECHO REQUESTS:- **-i**
c) We can send packets ECHO_REQUEST packets one after the another without waiting for another reply using the **-l** option with specifying the number of packets to be sent. However, we can also flood the server with a lot of requests (the number can shoot up to 15,000 in 5 seconds) using the **-f** option. If, however we are sending requests from a normal user, a maximum of 3 packets can only be sent using the **-l** option.
d) The command to set the ECHO_REQUEST packet size is: -
ping iitg.ernet.in -s value
where we can send requests to iitg.ernet.in with the size of the packet being equal to value.
The actual packet size if we specify the size to be 32 bytes is 60 bytes considering the 8 bytes for the ICMP header and the 20 bytes for the IP header.

Question 2: Ping carried on different hosts

The readings were taken at 11:00am, 4:00pm at 10:00 pm IST. The six hosts chosen are codeforces.com, amazon.in, ethz.ch, stanford.edu, gov.za, gouvernement.fr. Following is the data collected:

Destination Host Address	IP Address	Geographic Location	Avg RTT 1 (ms)	Avg RTT 2 (ms)	Avg RTT 3 (ms)	Total Avg RTT (ms)
codeforces.com	81.27.240.126	Moscow, Russia	122.384	122.165	121.903	207.2166
amazon.in	54.239.33.92	Bangalore, India	84.442	85.950	84.806	85.066
ethz.ch	129.132.19.216	Zurich, Germany	105.289	104.579	105	104.956
stanford.edu	171.67.215.200	Stanford, United States	67.506	67.021	67.244	67.256
gov.za	163.195.1.225	South Africa, Africa	275.108	275.131	274.812	275.015
gouvernement.fr	185.11.125.117	Paris, France	74.628	74.681	74.718	74.675667

Table 1: RTT of all 6 hosts

Data loss wasn't seen in any of the case during any time. However, packet loss may occur maybe because the network is down or there is firewall at the server side. It can also be because of too much congestion.

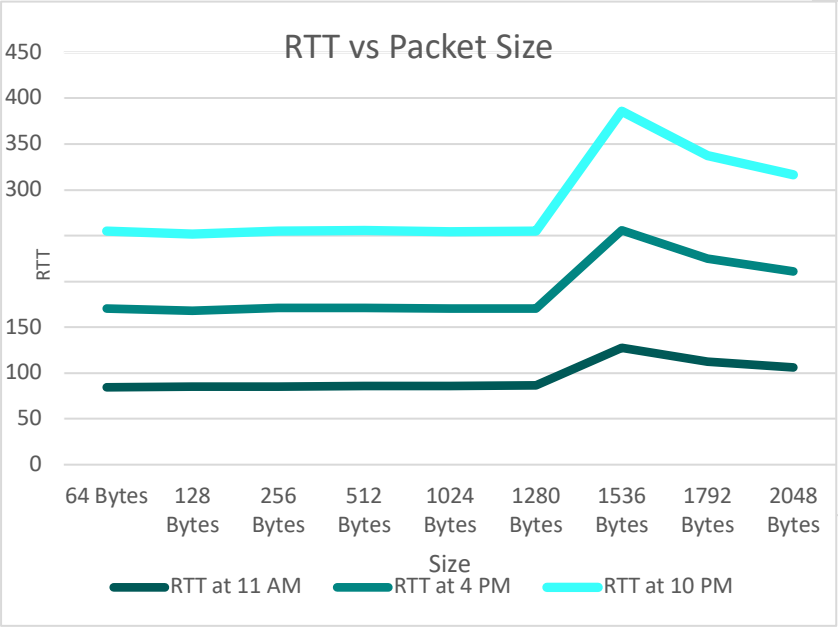
The host chosen for the next part is **amazon.**

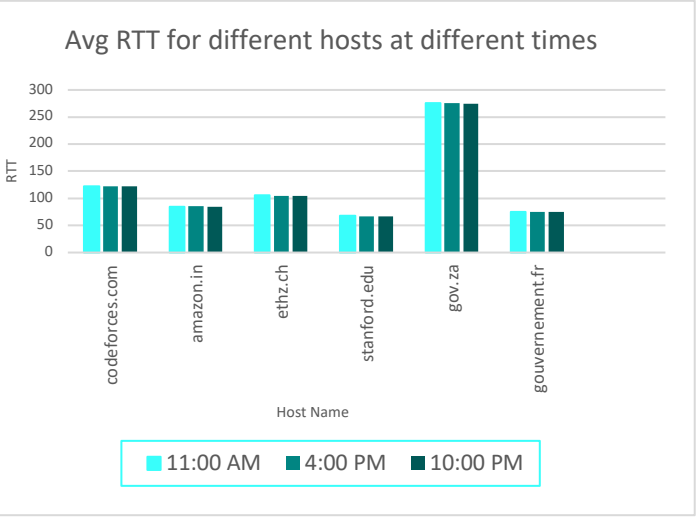
Size(Bytes)	64	128	256	512	1024	1280	1536	1792	2048
Avg RTT 1 (ms)	84.442	85.156	85.262	85.634	86.210	86.409	127.541	112.341	105.791
Avg RTT 2 (ms)	85.950	82.860	85.982	85.691	83.985	84.015	128.314	112.898	105.293
Avg RTT 3 (ms)	84.806	83.706	83.821	84.129	84.401	84.389	129.987	112.431	105.131

Table 2: RTT of one host with different size

RTT vs Distance

It is seen that RTT is **weakly positively related** to distance. This is positive as if the geographical distance is greater than it is possible that a greater number of routers are in the path which will result in a greater delay. Also, more distance means more propagation delay. But still this is a weak relation as there are a lot of other factors like network traffic which influences the RTT. As we see that the average RTT of stanford.edu is less than amazon.in, this means it takes more time for the packet to reach Bangalore than it takes to reach US. This is because RTT is mainly dependent on routers coming in





between the path. Routers can add the delay. So, it is possible the routers in the path for amazon.in take more time resulting in a greater RTT.

RTT vs Time

RTT depends on time. This is because at different times of the day, the congestion in the network is different. So, it is possible that at night time at 10 PM, a greater number of users were active resulting in a greater RTT than the other times of the day. So, from my data, it can be concluded that the network traffic is highest at 10 PM. Also different paths can have different transmission delays leading to different RTT.

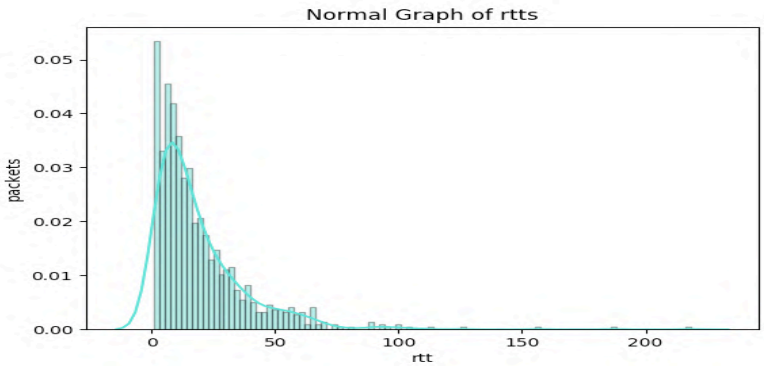
RTT vs Packet Size

It can be observed that very RTT varies slightly up to 1280 bytes. But when we increase the size further, there is a peak after 1500 bytes. This can be explained by the fact that the MTU (maximum transmission unit) is 1500 bytes by default. So, if the size of the packet exceeds this limit, it is broken in frames and sent and so the RTT increases. So RTT is more for cases where size is greater than 1500 as now we have to cover multiple trips.

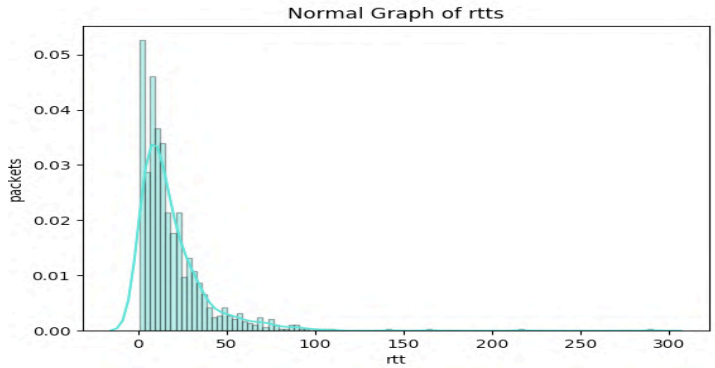
Question 3: 2 Separate commands of Ping

Following is the data collected and calculated when the different ping commands, one with -n option and the other with -p ff00 option is run.

Command	Packets Sent	Packets Received	Packet Loss Rate	Minimum Latency (ms)	Max Latency (ms)	Mean Latency (ms)	Median Latency (ms)
ping -n -c 1000 172.17.0.23	1000	999	0.1%	0.892	218.391	19.3329	13.253
ping -p ff00 -c 1000 172.17.0.23	1000	997	0.3%	1.082	290.88	19.41105	13.127



Graph Plotted running the ping command with -n option

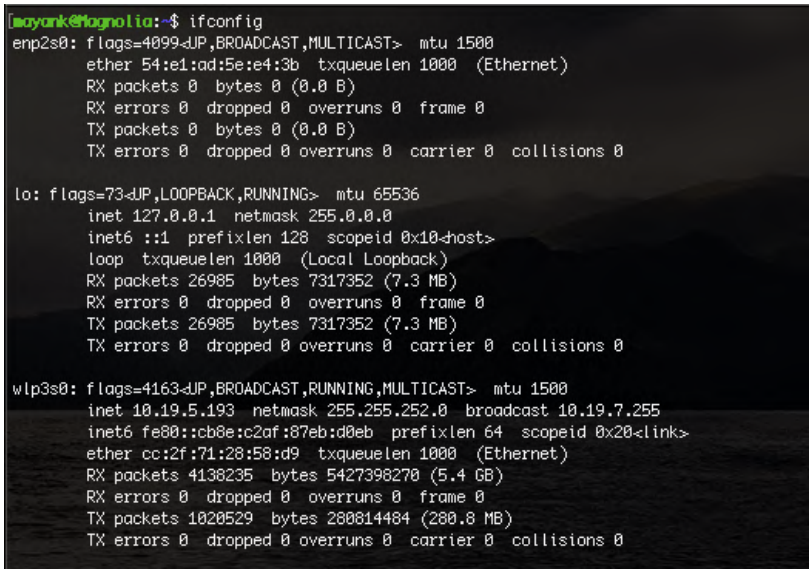


Graph Plotted running the ping command with -p ff00 option

After observing both the plots(histograms) of the ping command, we can make the following observations:

a) First of all, we notice that the mean latency is higher in the second case as compared to the first one. This is because when we use the '-n' option, **no attempt will be made to look up the symbols for the host addresses** and so we will save the extra latency.
b) And, we also notice the difference in the packet loss rate of the 2 commands. Packet loss rate is higher in the second case. This is because using -p ff00 will result in the pattern 1111 1111 0000 0000 to get appended to the packet being sent (this pattern can be used for diagnosing data-dependent problems in the network). So, this appended pattern will result in some problems including synchronization problems with the clock since now only one transition is present in the padding, that is from 1 to 0. As a result of which the clocks are now more likely to go out of synchronization in second case which will result in the observed higher packet loss rate.

Question 4: ifconfig and route commands



Ifconfig refers to **interface configuration** and the command is used to configure the kernel resident network interfaces. it is used to assign the IP address and netmask to an interface or to enable or disable a given interface. It is also used when the computer boots to set up the interfaces as necessary. Along with that it is also needed during debugging or when we need it for system tuning. **HWaddr** is the hardware address of the ethernet interface also known as mac address. It is of 48 bits. **iNET ADDR** is the IPV4 address assigned to the interface. **Bcast** denotes the broadcast address for the current network. **Mask** is the network mask which decides the potential size of the network. **UP** means the network interface is configured to be enabled. **Broadcast** means the ethernet device supports broadcasting which is a necessary characteristic to obtain IP address via DHCP. **MTU** is the maximum transmission unit which is a link layer characteristic which provides limit on the size of the Ethernet frame. **RX/TX** packets are the total number of packets received and transmitted respectively. It also shows the number of packets dropped and the overruns. **Collision** shows the number of packets that are colliding while traversing the network due to network congestion **RX/TX** bytes is the total amount of data in bytes that has passed through using the Ethernet interface. **Interrupt** is used by the NIC, network interface card. **Txqueuelen** is the field that provides the information about the configured length of transmission queue.

The options that can be specified with the ifconfig command are:

a) **'-a':** This option displays all the interfaces which are currently available even if they are down.
b) **'-s':** This option is used to display a short list.
c) **'-v':** This option is used to display the output in the verbose mode.
d) **'-up':** This option when specified with an interface, for example if we write the following command:
'ifconfig enp2s0 up' will activate the interface enp2so.
e) **'-mtu'** : This option is used to set the mtu, the maximum transmission unit foe the specified interface. For example, the following command:
'ifconfig enp2s0 mtu 9000' will set the mtu of the interface enp2s0 to be equal to 9000.

The **route** command shows the routing table of the device. As can be observed the screenshot, the **Destination** column refers to the destination network or the destination host. The **Gateway** column refers to the defined gateway for the specified network. An asterisk (*) can be seen sometimes which simply means that no forwarding gateway is required for the network. The **Genmask** column shows the netmask for the network. The **Iface** column shows the network interface.

```
mayank@Magnolia:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlp3s0
10.19.4.0       0.0.0.0         255.255.252.0   U     600    0        0 wlp3s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 wlp3s0
```

Destination address of 0.0.0.0 is used when no other rule is matched(subnet 0.0.0.0 means any ip will match this rule). The gateway of router which is 192.168.0.1 will be used since packets will be going out of router.

```
mayank@Magnolia:~$ route -A inet6
Kernel IPv6 routing table
Destination                    Next Hop                   Flag Met Ref Use If
ip6-localhost/128              [::]                       U    256 1     0 lo
fe80::/64                      [::]                       U    600 1     0 wlp3s0
[::]/0                         fe80::ed2:b5ff:fe1d:9658   UG   20600 5     0 wlp3s0
[::]/0                         fe80::1                    UG   20600 5     0 wlp3s0
ip6-localhost/128              [::]                       Un   0   6     0 lo
Magnolia/128                   [::]                       Un   0   3     0 wlp3s0
ip6-mcastprefix/8              [::]                       U    256 6     0 wlp3s0
[::]/0                         [::]                       !n   -1  1     0 lo
mayank@Magnolia:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.19.4.1       0.0.0.0         UG    600    0        0 wlp3s0
10.19.4.0       0.0.0.0         255.255.252.0   U     600    0        0 wlp3s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlp3s0
mayank@Magnolia:~$ route -v
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlp3s0
10.19.4.0       0.0.0.0         255.255.252.0   U     600    0        0 wlp3s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 wlp3s0
mayank@Magnolia:~$ route -C
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
mayank@Magnolia:~$ route --version
net-tools 2.10-alpha
+NEW_ADDRT +RTF_IRTT +RTF_REJECT +I18N +SELINUX
AF: (inet) +UNIX +INET +INET6 +IPX +AX25 +NETROM +X25 +ATALK +ECONET +ROSE -BLUETOOTH
HW: +ETHER +ARC +SLIP +PPP +TUNNEL -TR +AX25 +NETROM +X25 +FR +ROSE +ASH +SIT +FDDI +HIPPI +HDLC/LAPB +EUI64
```

The required flags are as follows:

a) **'-n':** The option shows numerical addresses instead of trying to determine symbolic host names. This is very useful if one is trying to determine why the route to one's nameserver has vanished.

b) **'-A family':** This option is used to display the specified family eq inet6.

c) **'v':** The option is used to display the output in verbose mode.

d) **'-C':** This option is used to operate on the kernel's routing cache.

e) **'—version':** This option is used to display the version of the route command.

Question 5: Netstat

a) **Netstat** (network statistics) command is used to print the network connections, routing tables, interface statistics, masquerade connections, and multi-cast memberships. It also lists the network connections that currently exist between your machine and other machines, as well as sockets 'listening' for connections from other machines.

b) Required parameters to show all the required tcp connections is '**-at'.** So the required command is: **netstat -at.** The **Proto** column defines the name of the protocol ex. TCP or UDP. The column **Recv-Q** and **Send-Q** indicates the data in the queue to be received and sent respectively. The column **Local Addresses** refers to the IP address of the local computer along with the port number used. The next column, **Foreign Address** is the IP address and the port number of the remote computer to which the socket is connected. Finally, **State** indicated the state of a TCP connection. **Listen** means waiting for external host to contact while **Established** means it is ready for communication.

```
mayank@Magnolia:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:postgresql    0.0.0.0:*               LISTEN
tcp        0      0 localhost:5433          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:microsoft-ds    0.0.0.0:*               LISTEN
tcp        0      0 localhost:44515         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.1:gpsd            0.0.0.0:*               LISTEN
tcp        0      0 localhost:43461         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:daap            0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:netbios-ssn     0.0.0.0:*               LISTEN
tcp        0      0 localhost:40243         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 Magnolia:47076          maa05s10-in-f14.1:https ESTABLISHED
tcp      680      0 localhost:47898         Magnolia:netbios-ssn    ESTABLISHED
tcp        0    172 Magnolia:ssh            10.19.6.71:56059        ESTABLISHED
tcp        0      0 Magnolia:49374          maa03s31-in-f4.1e:https ESTABLISHED
tcp        0      0 Magnolia:53716          sc-in-f188.1e100.n:5228 ESTABLISHED
tcp        0      0 Magnolia:52446          maa03s28-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:55298          edge-star-shv-02-:https ESTABLISHED
tcp        0      0 Magnolia:51404          ingress-westus-10:https ESTABLISHED
tcp        0      0 Magnolia:56574          maa05s02-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:49430          maa05s09-in-f3.1e:https ESTABLISHED
tcp        0      0 Magnolia:55304          edge-star-shv-02-:https ESTABLISHED
tcp        0      0 Magnolia:44206          maa05s06-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:49018          maa05s06-in-f3.1e:https ESTABLISHED
tcp        0      0 Magnolia:38086          sc-in-f189.1e100.:https ESTABLISHED
tcp        0      0 Magnolia:59124          maa05s09-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:46324          13.107.6.171:https      ESTABLISHED
tcp        0      0 Magnolia:36384          maa05s02-in-f3.1e:https TIME_WAIT
tcp        0      0 Magnolia:36382          maa05s02-in-f3.1e:https ESTABLISHED
tcp        0      0 Magnolia:48388          maa05s04-in-f3.1e:https ESTABLISHED
tcp        0      0 Magnolia:59168          maa05s09-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:53096          maa03s31-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:53052          maa03s31-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:52466          maa03s28-in-f14.1:https ESTABLISHED
tcp        0      0 Magnolia:netbios-ssn    localhost:47898         ESTABLISHED
tcp        0      0 Magnolia:58468          server-13-33-142-:https ESTABLISHED
tcp6       0      0 [::]:microsoft-ds       [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:gpsd      [::]:*                  LISTEN
tcp6       0      0 [::]:daap               [::]:*                  LISTEN
tcp6       0      0 [::]:netbios-ssn        [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
```

c) **netstat -r** command shows the kernel routing table. Following is the required explanation of all fields:

Destination shows the pattern that is checked with the destination of the packet by the kernel. So, whenever we have to send some packet, we compare the destination of the packet with all destinations listed in the routing table and in this way we determine where to send the packet. Once there's a match, the **Gateway** column is then used to determine where to send the packet from the computer. Next, the **Genmask** column specifies how many bits from the start of the IP address are used to identify the subnet. 255 refers to the non-zero part whereas the 0 part refers to the other parts to the destination. The flags describing the route can be seen from the **Flags** column where **U** means interface is up, **G** means that the route uses a gateway. There are more options however my computer's routing table doesn't have those yet namely, **H** only a single host be reached through the route and so on. The **MSS** refers to

```
mayank@Magnolia:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 wlp3s0
10.19.4.0       0.0.0.0         255.255.252.0   U         0 0          0 wlp3s0
link-local      0.0.0.0         255.255.0.0     U         0 0          0 wlp3s0
```

Maximum segment size which means the maximum size of the datagram the kernel will construct for transmission via the route. The **Window** the maximum count of data he system will accept in a single burst. The acronym **IRTT** refers to initial round trip time and is used by the kernel to guess about the best TCP parameters without waiting for slow replies.

d) **Netstat -i** can be used to display the status of all network interfaces. As can be seen from the screenshot, I have 3 interfaces in my computer namely enp2s0, lo and wlp3s0.

```
mayank@Magnolia:~$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp2s0    1500        0      0      0 0             0      0      0      0 BMU
lo       65536    36843      0      0 0         36843      0      0      0 LRU
wlp3s0    1500  5618807      0      0 0       1388198      0      0      0 BMRU
```

e) The required option to check all the UDP connections is **'-aus'.** Thus the comand is: **netstat -aus.**

```
mayank@Magnolia:~$ netstat -aus
IcmpMsg:
    InType0: 1423
    InType3: 5590
    InType11: 1125
    OutType3: 4536
    OutType8: 3202
Udp:
    368206 packets received
    632 packets to unknown port received
    0 packet receive errors
    47844 packets sent
    0 receive buffer errors
    9 send buffer errors
    IgnoredMulti: 65088
UdpLite:
IpExt:
    InMcastPkts: 66549
    OutMcastPkts: 4610
    InBcastPkts: 123324
    OutBcastPkts: 1503
    InOctets: 6939615377
    OutOctets: 328344131
    InMcastOctets: 5234180
    OutMcastOctets: 618234
    InBcastOctets: 18567597
    OutBcastOctets: 237114
    InNoECTPkts: 5358067
```

f) When network interface is disconnected, for example when the ethernet cable is loose and gets detached from the computer, no communication is possible, not even between the computer and oneself. **The Loopback Interface** is a special kind of virtual network (it does not physically exist) that one's computer uses to communicate with itself. It serves many purposes including troubleshooting and diagnostics and also helps in connecting to servers running on the local machine. For example, if we run a web server, we have all our files and media and could examine them file by file. We may even load the files in the browser as well though with server-side active content. So, if we want to experience the same site others do, the best course is usually to connect to one's own server.

Question 6: Traceroute

The hop counts were measured at different times of the day, 11 AM, 4 PM and at 10 PM.

	codeforces.com	amazon.in	ethz.ch	stanford.edu	gov.za	gouvernement.fr
Hop Count #1	9	8	12	12	17	10
Hop Count #2	9	8	12	12	17	10
Hop Count #3	9	8	12	12	18	10

a) Going through all the data from all websites, it was noted that all websites passed through the router with IP 213.239.245.237. Also, it was also noted that in the case of amazon.in, codeforces.com, stanford.edu and gouvernement.fr one more router with IP 213.239.245.241 was also same. Also in the case of ethz.ch and amazon.in, a router with IP 213.239.245.218 was same.

b) In the case of gov.za we see that during the day and night time, there is a change in one of the hops, at day time the packet goes through the router with IP 5.11.10.177 but at night time it goes through 5.11.10.102. This difference can be explained by **load balancing,** i.e. the packets tend to take the path where there is less traffic and congestion. So this technique is used to reduce the load of a congested path

c) In our case, the paths to amazon.in, stanford.edu, codeforces.com and gouvernement.fr websites were left incomplete. This may occur because of several reasons. It is possible that the ICMP reply sent by the intermediate hosts may get lost . The sender may haven't sent the ICMP packet with incremented TTL value. The routers and servers in the path may have **firewall protection** which either blocks the ICMP traffic or hides the IP addresses of the hosts. Also many network providers disable ICMP traffic if the network is under heavy load.

d) Yes, it is possible to find a partial path even though ping fails. This is because their working is not exactly identical. Traceroute works by sending the packets of data with limited TTL (time to live) which specifies how many more steps the packet can take before it is returned back. When this TTL value becomes 0, the router identifies this and returns the packet back thus tracing the partial path. So all intermediate routers need to reply with an ICMP/UDP packet. If however due to some other reasons, the destination is unable to answer to the ping, a partial path can still be found. Though in recent years it has also become possible to sneak through firewalls of the routers which will again help us to find a partial path.

Question 7 ARP Address Resolution Protocol

a) We can see the full arp table by using the **arp** command. We can see the IP addresse, the MAC addresses and the the corresponding network interface of the computer. The explanation of each columns are as follows:

- **Address:** This column lists all the IP addresses of the machines used to uniquely identify the connection of the network.
- **HWaddress:** The corresponding MAC addresses of the computer with IP address specified in the first column.
- **Iface:** Denotes the interface connecting the local computer and the other computer on the network. For example the computer with MAC address e0:ac:cb:9c:15:78 is connected to my computer via the wlp3so inters=face.
- **Flags:** These represent if the MAC addresses have been learnt(C), manually set(M), published ie announced by another node than the requested) or is incomplete.
- **HWType:** Indicated that the host has ethernet interface.

```
mayank@Magnolia:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
156.99.224.35.bc.google  ether   (incomplete)                              enp2s0
5.85.222.35.bc.googleus  ether   (incomplete)                              enp2s0
_gateway                 ether   ec:44:76:74:60:43   C                     wlp3s0
10.19.6.71               ether   e0:ac:cb:9c:15:78   C                     wlp3s0
```

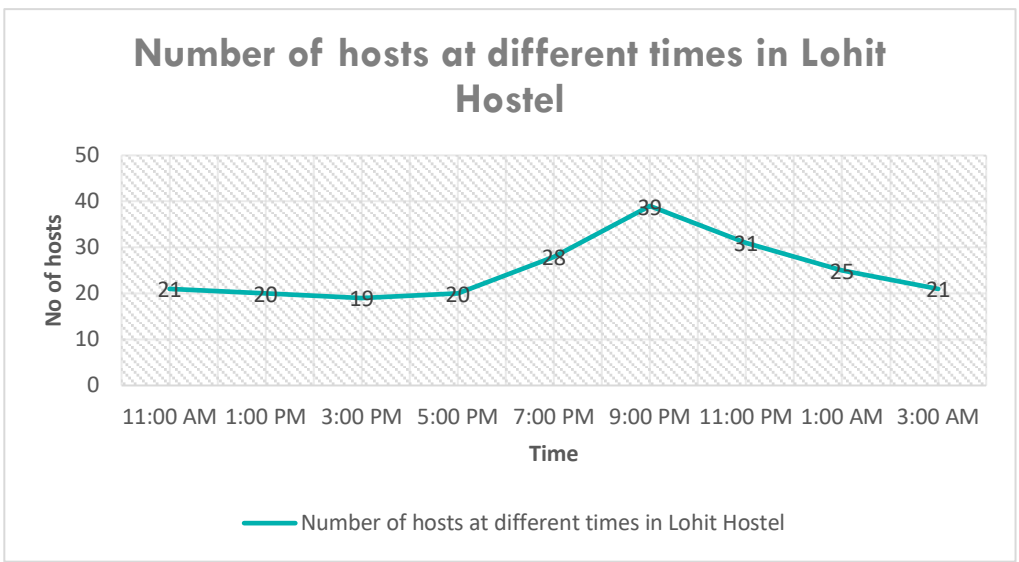
b) **'sudo arp-s <ip address> <mac address>'** is used to add and **'sudo arp -d <ip address>'** is used to delete an entry.

c) The parameter that determine how long the entries in the cache of the ARP module of the kernel remain valid and when they get deleted from the cache is gc_scale_time. We can use the command **'cat/proc/sys/ney/ipv4/neigh/default/gc_scale_time'** to find the value of timeout. Entries which are modified stay cached in the ARP table for **60 seconds.** The required trial and error method goes in this way: Add a temporary variable to the ARP table. Now in every small intervals of time say 3 seconds, keep checking if the changes are reflected in the arp table. The time when it we finally notice a change is the approximate timeout. Another approach that can be used is binary search. So we again add a temporary variable and keep checking after regular intervals of time say 3000 milliseconds. Once we get a timeout, we again add and just before the timeout reduce the regular interval to 1500 milliseconds and continue. This will be helpful in finding the timeout value with precision.

```
mayank@Magnolia:~$ sudo arp -s 10.19.5.1 ff:ff:ff:ff:ff:ff
mayank@Magnolia:~$ sudo arp -s 10.19.5.2 ff:ff:ff:ff:ff:ff
mayank@Magnolia:~$ sudo arp -s 10.19.5.3 ff:ff:ff:ff:ff:ff
mayank@Magnolia:~$ sudo arp -s 10.19.5.4 ff:ff:ff:ff:ff:ff
mayank@Magnolia:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   ec:44:76:74:60:43   C                     wlp3s0
5.85.222.35.bc.googleus          (incomplete)                              enp2s0
10.19.5.2                ether   ff:ff:ff:ff:ff:ff   CM                    wlp3s0
156.99.224.35.bc.google          (incomplete)                              enp2s0
10.19.6.71               ether   e0:ac:cb:9c:15:78   C                     wlp3s0
10.19.5.1                ether   ff:ff:ff:ff:ff:ff   CM                    wlp3s0
10.19.5.3                ether   ff:ff:ff:ff:ff:ff   CM                    wlp3s0
10.19.5.4                ether   ff:ff:ff:ff:ff:ff   CM                    wlp3s0
mayank@Magnolia:~$ sudo arp -d 10.19.5.1
mayank@Magnolia:~$ sudo arp -d 10.19.5.2
mayank@Magnolia:~$ sudo arp -d 10.19.5.3
mayank@Magnolia:~$ sudo arp -d 10.19.5.4
mayank@Magnolia:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   ec:44:76:74:60:43   C                     wlp3s0
5.85.222.35.bc.googleus          (incomplete)                              enp2s0
156.99.224.35.bc.google          (incomplete)                              enp2s0
10.19.6.71               ether   e0:ac:cb:9c:15:78   C                     wlp3s0
```

d) If we come across a situation where 2 IP's map to the same Ethernet address, then we will incur 100% data packet loss if either of the IP is pinged. MAC address is used for sending the packets when we have to communicate with machines on the same subnet range. So if we get the MAC address, we will have a corresponding 2 IP addresses and the router in this situation will not know where to send the packet. In such a situation, the router would not send the packet to either of the devices which would mean 100% packet loss rate. In the other case i.e. when the device have different subnet ranges, we refer to the ARP table and can learn from it that the IP addresses have ethernet address of the routers that connect those 2 subnets. ARP tables is referred in that situations and the packets are sent to the router which then uses its own routing table to send to the packet to the required destination device. So in this case there is not much problem as the MAC address on the other subnet would contain only one corresponding IP address

Question 8 NMAP



The following command is used on the terminal. The IP's taken are from Lohit hostel.

nmap -n -sP 10.19.4.1/22

From the above graph, it can be clearly seen that the number of hosts are almost constant in the afternoon time and shoots upto 39 in the 9:00 pm which is expected. It again starts to drop after 1:00 AM. So the time between 8-9 can be considered to be the peak time.