CS 349: Networks Lab
Assignment 1

Akul Agrawal
160101085

Q1.

a) "-c" command is required to specify the number of ECHO_REQUESTs to send with ping command. For eg. for 5 requests of www.google.com, the command is "ping -c 5 www.google.com".
b) "-i" command is required to set time interval (in seconds) between two successive ping ECHO_REQUESTs. For eg. to set time interval to 3 sec, the command is "ping -i 3 www.google.com".
c) "-f" command is required to send ECHO_REQUEST packets to the destination one after another without waiting for a reply. For eg. ping -f www.google.com. The minimum time interval for sending such ECHO_REQUEST packets by normal users is 200ms. Thus, maximum rate = 5 requests per second.
d) "-s" command is required to set the ECHO_REQUEST packet size (in bytes). If the PacketSize is set to 64 bytes, the total packet size will be 72 bytes.

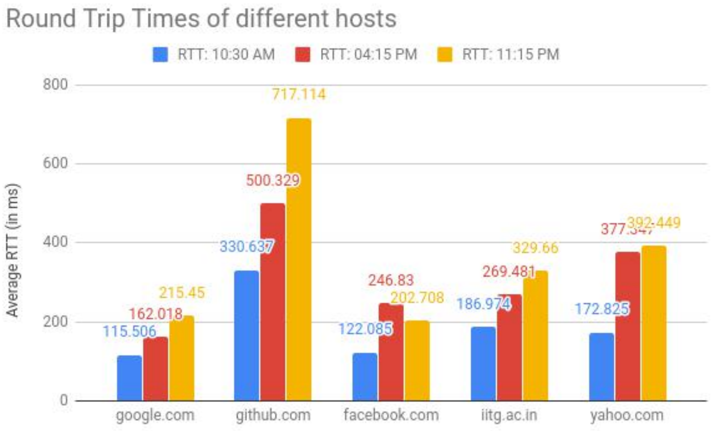
Q2.

Host	Avg RTT: 10:30 AM (ms)	Avg RTT: 04:15 PM (ms)	Avg RTT: 11:15 PM (ms)	Packet loss: 10:30 AM(%)	Packet loss: 04:15 PM(%)	Packet loss: 11:15 PM(%)
google.com	115.506	162.018	215.45	0	10	10
facebook.com	122.085	246.83	202.708	0	10	5
iitg.ac.in	186.974	269.481	329.66	5	5	15
yahoo.com	172.825	377.347	392.449	0	15	15
github.com	330.637	500.329	717.114	0	5	15

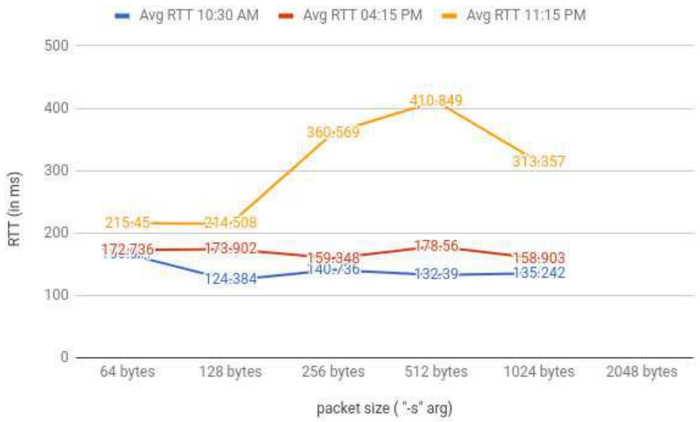
As it can be observed, there are cases with packet loss greater than 0%. This case arises due to one of the following reasons: Many routers are typically programmed to give lower priority to echoing of ICMP packets and drop them preferentially in favor of spending resources on genuine data. Just because there is a hop with high loss doesn't mean it's slowing down "real" traffic. It may only be throwing away ICMP. That's not necessarily good because it might mean the router is too busy, but it's not guaranteed. The router may also be programmed to limit the number of responses it sends to ICMP packets in an effort to mitigate DoS attacks.

Host	IP Address	Geographical Location	Avg RTT (in ms)
google.com	172.217.166.164	Mountain View, California, United States	164.325
facebook.com	31.13.79.35	London, England, United Kingdom	190.541
iitg.ac.in	14.139.196.22	Guwahati, Assam, India	262.038
yahoo.com	106.10.250.11	Omaha, Nebraska, United States	314.207
github.com	192.30.253.112	San Francisco, California, United States	516.027

From the above table, one can easily conclude that average measured RTTs are weakly correlated with the geographical distance of the hosts. Although both the hosts “google.com” and “github.com” are located in California, United States, and thus, have very similar geographical distance from IIT Guwahati, there is a huge difference in their RTTs. There exists a correlation between geographical distance and average RTT, since the propagation delay time = geographical distance / speed of signal (mostly almost equals speed of light). Since propagation delay time is added two times in RTT, more the geographical distance, more should be the RTT, if all other factors(routing efficiency, processing overhead, congestion, etc.) are kept constant. Also, with increase in distance, number of intermediate nodes like routers may increase, thus, increasing RTT due to the processing delay at each additional node. Although, as we saw, this correlation is quite week since there are a number of other factors affecting the RTT. One of the most important factors is the type of server. Server can be of many types, optimized to a particular functionality and having specific storage and processing power. Also, traffic on the servers might affect the RTT, more the traffic, more would be the RTT.



It can be observed from the tables that RTT hugely varied with time. On the same day, the RTT is, in general, least in the morning (10:30 AM in this case) and highest in the night (11:15 PM in this case). Thus, one may conclude that there is less network traffic in the morning, and more at night.



From the graph we can clearly observe that the Round Trip Time(RTT) is almost the same for all packets till size 1024 bytes. For packet size 2048 bytes, I didn't recieve any response. This can be explained by the fact that Maximum Transmission Unit(MTU) is 1500 Bytes by default. If the packet size is less than 1500 Bytes, then the data is padded to make the size 1500 Bytes. Hence, for packets with size less than 1500 Bytes, the RTT is same.

Q3.

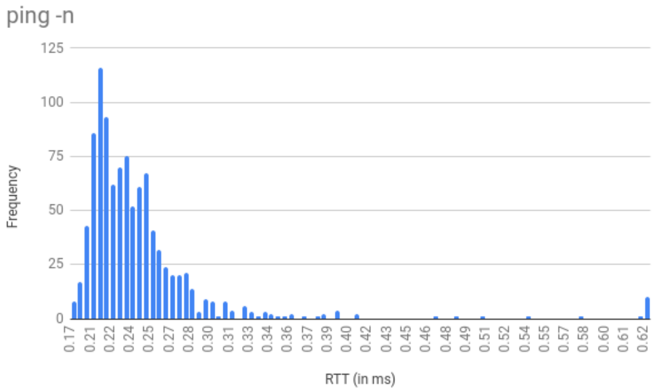
address pinged: 202.141.80.14

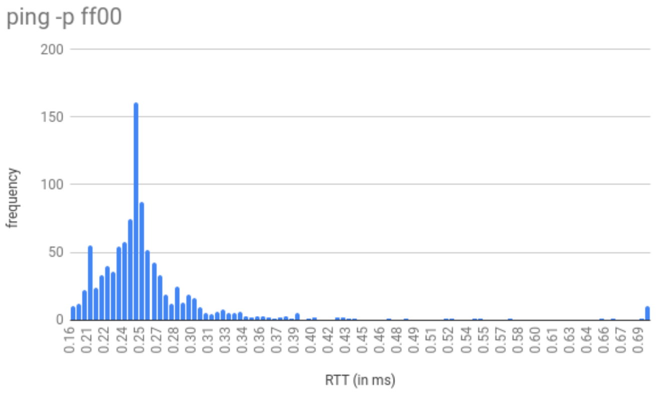
a) "ping -n": 0%
"ping -p ff00": 0%

b)

	minimum latency	maximum latency	mean latency	median latency
"ping -n"	0.171 ms	7.055 ms	0.261 ms	0.231 ms
"ping -p ff00"	0.156 ms	4.300 ms	0.274 ms	0.251 ms

c)





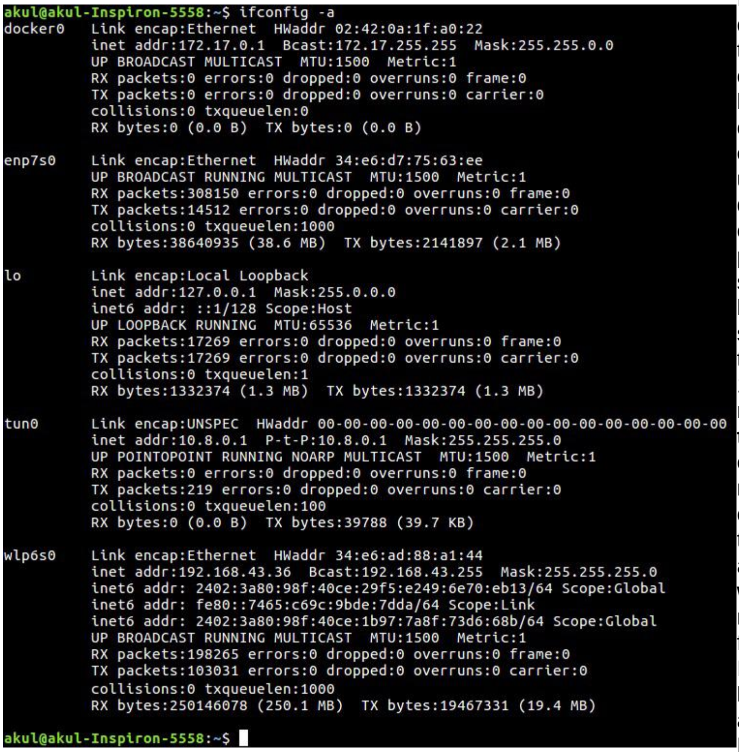
d) Both the graphs appear to have a normal distribution. The two different aspects in both the experiments are:

1. In the first case (ping -n), no attempt will be made to lookup symbolic names for host addresses. Thus, it only produces a numeric output and the lookup time for finding symbolic host name is saved, unlike in the 2nd case. Thus, the mean latency in 1st case is slightly lesser than the mean latency of 2nd case.

2. In the 2nd case (ping -p ff00), the command will cause the 16 pad bits of the sent packets to be filled with "ff00" in hex, i.e. 1111111100000000. However, in the 1st case, by default, the padding starts from "0001" in hex, and continues in ascending order. Thus, in 2nd case bit pattern has a single transition which can lead to loss of synchronisation and subsequently packet loss.

Q4.

ifconfig



```
akul@akul-Inspiron-5558:~$ ifconfig -a
docker0   Link encap:Ethernet  HWaddr 02:42:0a:1f:a0:22
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

enp7s0    Link encap:Ethernet  HWaddr 34:e6:d7:75:63:ee
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:308150 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14512 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:38640935 (38.6 MB)  TX bytes:2141897 (2.1 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:17269 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17269 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:1332374 (1.3 MB)  TX bytes:1332374 (1.3 MB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.1  P-t-P:10.8.0.1  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:219 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:39788 (39.7 KB)

wlp6s0    Link encap:Ethernet  HWaddr 34:e6:ad:88:a1:44
          inet addr:192.168.43.36  Bcast:192.168.43.255  Mask:255.255.255.0
          inet6 addr: 2402:3a80:98f:40ce:29f5:e249:6e70:eb13/64 Scope:Global
          inet6 addr: fe80::7465:c69c:9bde:7dda/64 Scope:Link
          inet6 addr: 2402:3a80:98f:40ce:1b97:7a8f:73d6:68b/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:198265 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103031 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:250146078 (250.1 MB)  TX bytes:19467331 (19.4 MB)

akul@akul-Inspiron-5558:~$ 
```

Interfaces

docker0: bridge device for Docker. All traffic from the Docker containers flows over it to the Docker daemon, which handles routing on behalf of the container.

enp7s0: That's a change in how now udevd assigns names to ethernet devices.

en:ethernet

p7:bus number (7)

s0:slot number (0)

lo: a loopback device that is on all systems, even if they aren't connected to any network. It has an IP address of 127.0.0.1 and can be used to access network services locally.

tun0: Hardware network links can be either point to point or point to multipoint. tun can act as either, in my case it is acting as a point to point link. tun interfaces don't have default mac addresses.

wlp6s0: Just like enp7s0 is another name for eth0, wlp6s0 is a new name for wlan0. wl stands for wlan.

Interface details

Hwaddr: hardware address or MAC address which is unique to each Ethernet card

inet addr: indicates the machine IP address

Bcast: denotes the broadcast address

Mask: is the network mask which we passed using the netmask option.

UP: This flag indicates that the kernel modules related to the Ethernet interface has been loaded.

BROADCAST: shows that Ethernet device supports broadcasting - a feature to obtain IP address via DHCP.

RUNNING: The interface is ready to accept data.

ARP: option specific to broadcast networks. It enables the use of Address Resolution Protocol (ARP) to detect the physical addresses of hosts attached to the network. If disabled, ifconfig shows the NOARP flag.

MULTICAST: indicates that the Ethernet interface supports multicasting. Multiple devices can capture the same signal from the radio station but if and only if they tune to a particular frequency. Multicast allows a source to send a packet(s) to multiple machines as long as the machines are watching out for that packet.

MTU: short form for Maximum Transmission Unit is the size of each packet received by the Ethernet card. The value of MTU for all Ethernet devices by default is set to 1500, but can be changed by “sudo ifconfig eth0 mtu <size>”. Setting this to a higher value could hazard packet fragmentation or buffer overflows.

Metric: This option can take a value of 0,1,2,3... with the lower the value the more leverage it has. The value of this property decides the priority of the device. This parameter has significance only while routing packets.

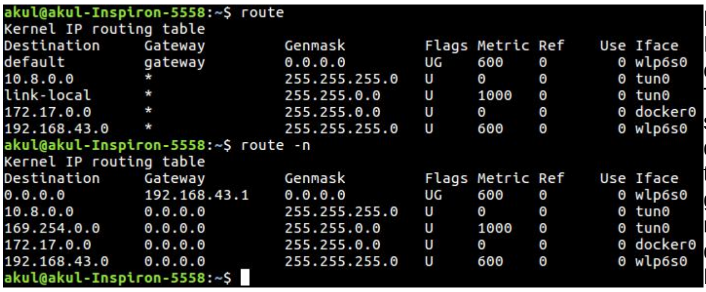
RX Packets, TX Packets: The next two lines show the total number of packets received and transmitted respectively. As you can see in the output, the total errors are 0, no packets are dropped and there are no overruns. If you find the errors or dropped value greater than zero, then it could mean that the Ethernet device is failing or there is some congestion in your network.

collisions: The value of this field should ideally be 0. If it has a value greater than 0, it could mean that the packets are colliding while traversing your network - a sure sign of network congestion.

txqueuelen: This denotes the length of the transmit queue of the device. Usually set it to smaller values for slower devices with a high latency such as modem links and ISDN.

RX Bytes, TX Bytes: These indicate the total amount of data that has passed through the Ethernet interface either way.

route



```
akul@akul-Inspiron-5558:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         gateway         0.0.0.0         UG    600    0        0 wlp6s0
10.8.0.0        *               255.255.255.0   U     0      0        0 tun0
link-local      *               255.255.0.0     U     1000   0        0 tun0
172.17.0.0      *               255.255.0.0     U     0      0        0 docker0
192.168.43.0    *               255.255.255.0   U     600    0        0 wlp6s0
akul@akul-Inspiron-5558:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.43.1    0.0.0.0         UG    600    0        0 wlp6s0
10.8.0.0        0.0.0.0         255.255.255.0   U     0      0        0 tun0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 tun0
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
192.168.43.0    0.0.0.0         255.255.255.0   U     600    0        0 wlp6s0
akul@akul-Inspiron-5558:~$ 
```

First entry tells the following:

If any traffic does not fit the traffic defined on any other rules then use this route. The address will be translated (which, say, wouldn’t be among the remaining destinations). Since it won't fit on rest of the routes, it will be routed using the gateway 192.168.43.1. The metric will not matter and it will be using the interface defined on the default route.

Rest of the entries tell the following:

Any traffic with given destination will not be using a gateway (that’s the * on the line), will be using a 255.255.255.0/255.255.0.0 net mask, route is UP (that's the meaning of the U) and Iface shows which interface the route uses.

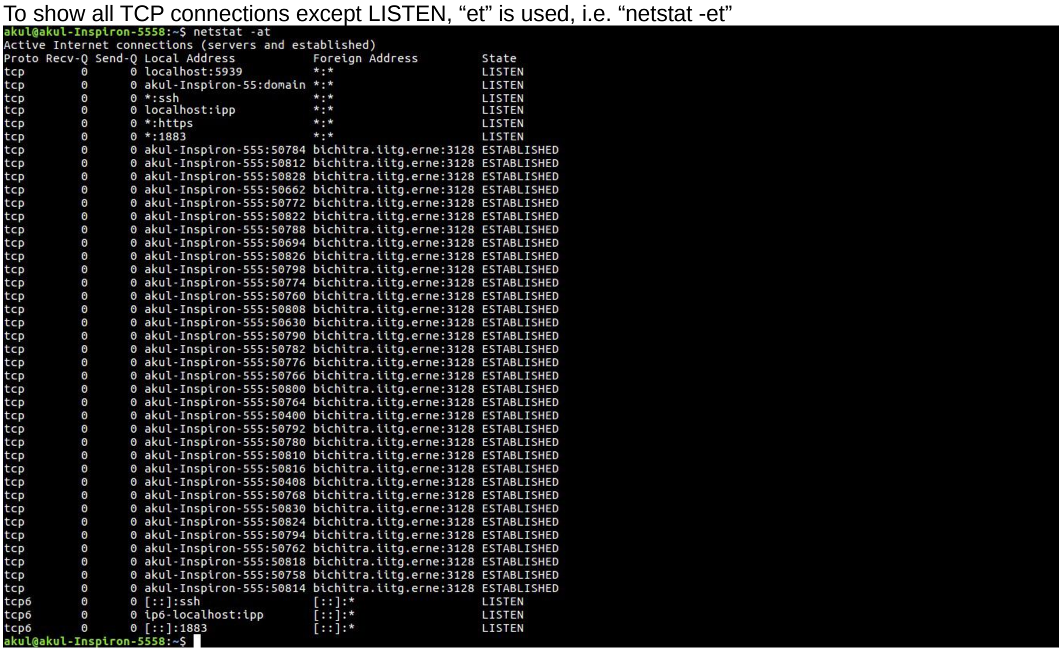
“-n” option of route is used to show numerical addresses instead of trying to determine symbolic host names.

Q5.

Netstat (network statistics) is basically a network utility tool that displays network connections (TCP, UDP), routing tables, number of network interface and network protocol statistics. It is often used to find problems in the network, or determine the amount of traffic over the network as a performance measurement. It is a command line tool used for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc. It is one of the most basic network service debugging tools, telling you what ports are open and whether any programs are listening on ports.

To show all TCP connections, parameters used is: “at”, and thus, the command is: “netstat -at”

To show all TCP connections except LISTEN, “et” is used, i.e. “netstat -et”



```
akul@akul-Inspiron-5558:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:5939          *:*                     LISTEN
tcp        0      0 akul-Inspiron-55:domain *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 *:https                 *:*                     LISTEN
tcp        0      0 *:1883                  *:*                     LISTEN
tcp        0      0 akul-Inspiron-555:50784 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50812 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50828 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50662 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50772 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50822 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50788 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50694 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50826 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50798 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50774 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50760 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50808 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50630 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50790 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50782 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50776 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50766 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50800 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50764 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50400 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50792 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50780 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50810 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50816 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50408 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50768 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50830 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50824 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50794 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50762 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50818 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50758 bichitra.iitg.erne:3128 ESTABLISHED
tcp        0      0 akul-Inspiron-555:50814 bichitra.iitg.erne:3128 ESTABLISHED
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
tcp6       0      0 [::]:1883               [::]:*                  LISTEN
akul@akul-Inspiron-5558:~$
```

Proto: protocol

Recv-Q: The count of bytes not copied by the user program connected to this socket.

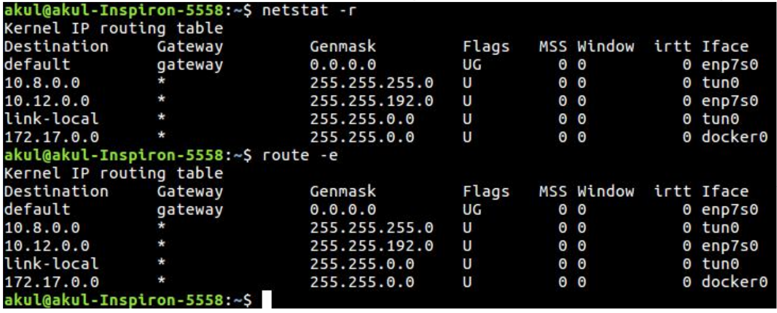
Send-Q: The count of bytes not acknowledged by the remote host.

State: LISTEN: Indicates that the server is ready to accept a connection

ESTABLISHED: Indicates that the server received the SYN signal (synchronize, this signal is only sent in the first packet) from the client and the session is established.

Local Address: It is the IP address of the local computer (your device) and the port number being used. This address is assigned by the router DHCP servers. DHCP (Dynamic Host Configuration Protocol) is a network protocol that automatically assigns IP address to your computer from a predefined range of number configured for a given network.

Foreign address: It is the IP address and port number of the remote computer to which the socket is connected.



```
akul@akul-Inspiron-5558:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         gateway         0.0.0.0         UG        0 0          0 enp7s0
10.8.0.0        *               255.255.255.0   U         0 0          0 tun0
10.12.0.0       *               255.255.192.0   U         0 0          0 enp7s0
link-local      *               255.255.0.0     U         0 0          0 tun0
172.17.0.0      *               255.255.0.0     U         0 0          0 docker0
akul@akul-Inspiron-5558:~$ route -e
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         gateway         0.0.0.0         UG        0 0          0 enp7s0
10.8.0.0        *               255.255.255.0   U         0 0          0 tun0
10.12.0.0       *               255.255.192.0   U         0 0          0 enp7s0
link-local      *               255.255.0.0     U         0 0          0 tun0
172.17.0.0      *               255.255.0.0     U         0 0          0 docker0
akul@akul-Inspiron-5558:~$
```

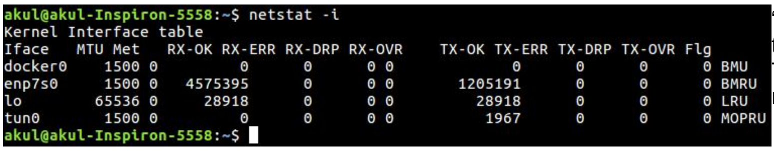
“netstat -r” displays kernel routing table. It has the same output as “route -e”.

MSS: It is the Maximum Segment Size and is the size of the largest datagram the kernel will construct for transmission via this route.

Window: It is the maximum amount of data the system will accept in a single burst from a remote host.

Irtt: It stands for “initial round trip time.”. The initial round-trip time is the value of RTT that the TCP protocol will use when a connection is first established. For most network types, the default value is okay, but for some slow networks, the time is too short and causes unnecessary retransmission. The irtt value can be set using the route command. Values of zero in these fields mean that the default is being used.

Iface: It is the network interface that the given route uses.



```
akul@akul-Inspiron-5558:~$ netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
docker0    1500 0         0      0      0 0             0      0      0      0 BMU
enp7s0     1500 0   4575395      0      0 0       1205191      0      0      0 BMRU
lo        65536 0     28918      0      0 0         28918      0      0      0 LRU
tun0       1500 0         0      0      0 0          1967      0      0      0 MOPRU
akul@akul-Inspiron-5558:~$
```

“netstat -i” can be used to display the network interface status There are 4 interfaces in my machine.

The loopback device is a special, virtual network interface that the 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 Purpose of Loopback

When a network interface is disconnected, for example, when an ethernet port is unplugged or wifi is turned off or not associated with an access point, no communication on that interface is possible, not even communication between your computer and itself. 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.

This is important for troubleshooting (it can be compared to looking in a mirror). The loopback device is sometimes explained as purely a diagnostic tool. But it is also helpful when a server offering a resource you need is running on your own machine.
For example, if I run a web server, I have all my web documents and could examine them file by file. I may be able to load the files in my browser too, though with server-side active content, it won't work the way it does when someone accesses it normally. So if I want to experience the same site others do, the best course is usually to connect to my own server. The loopback interface facilitates that.

Q6.

a)

time	google.com	facebook.com	iitg.ac.in	yahoo.com	github.com
10:30 am	10	8	6	12	13
04:15 pm	10	9	6	12	11
11:15 pm	7	9	7	11	10

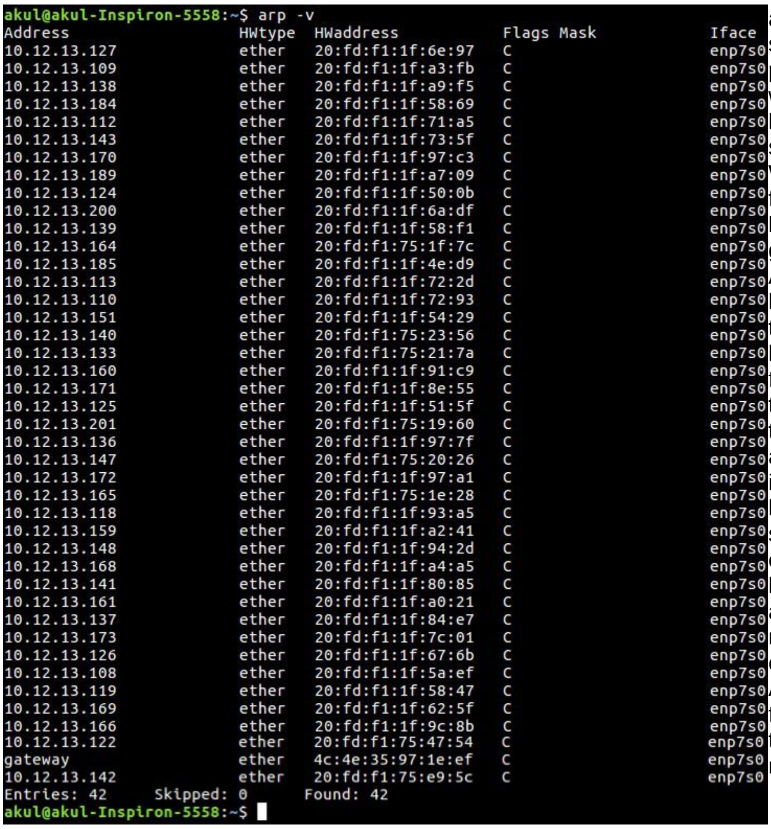
The following are the common hops between every host above: 192.168.43.1, 192.168.212.1, 172.21.63.1 and 118.185.210.162. Also, google.com and iitg.ac.in have an extra common hop: 182.19.106.198.

b) It is observed that the route to the hosts (not only the route, but also the number of hops) changes at different times of the day in the experiments. This is due to the fact that traffic is different at different times of the day. Thus, at different points of time, even for a same host, the packets may be directed (and as experimented, they are indeed directed) to different routes according to the one having the least traffic, according to the load balancing algorithms used.

c) A traceroute will show the number of layer3 hops from A to B. However, I could be going through hundreds of switches inbetween. I could also be going through a number of ISP routers running a layer 2 vpn which appears as a single hop. An MPLS(Multiprotocol Label Switching) network could hide its internals, or show its internals. I could have transparent firewalls in the path as well.Some servers/hosts along the path may have not been configured to respond to the ICMP Traffic or may have set up firewalls which block the ICMP Traffic. However, they still send the data to the next hop as there are results that follow. Many network providers disable ICMP traffic if their network is under heavy load. Either way, I cannot guarantee that every single device in the path will count as a hop. Because of the above points I mentioned, I could be going through 100 devices but it could look like 5 to me. It doesn't happen all the time though. If I see 15 hops it very well could be 15 hops.

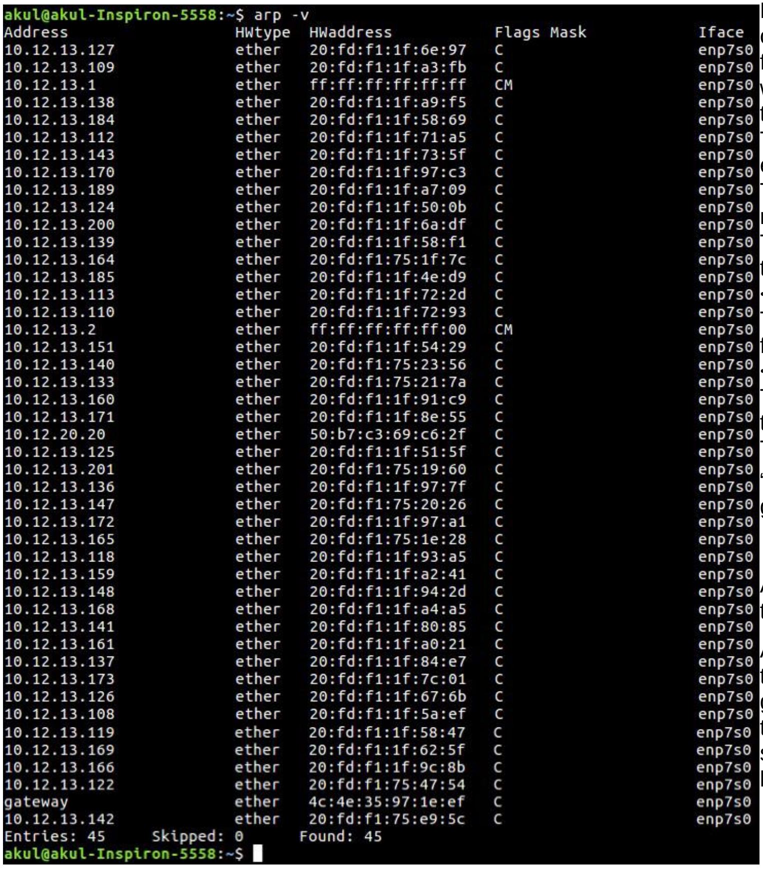
d) Yes, it all has to do with how traceroute works. On Unix operating systems, traceroute sends, by default, a sequence of User Datagram Protocol (UDP) packets, with destination port numbers ranging from 33434 to 33534 (and not ICMP echo requests as in case of ping). Most probably the *ping* is blocked/gets discarded, while *traceroute* uses an error message form a node/hop to determine the route. *Traceroute* is not a standard tool, in that it uses a trick to get the information. The trick is to manipulate the TTL, so the hop responds with an *ICMP* error (ICMP TTL exceeded), and that is why this is possible. Ping is straight ICMP from point A to point B, that traverses networks via routing rules. Traceroute 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 - so even if it uses ICMP, it is using it in a very different way.

Q7. Command to show the full ARP table for my machine is: “arp -v”.



```
akul@akul-Inspiron-5558:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
10.12.13.127             ether   20:fd:f1:1f:6e:97   C                     enp7s0
10.12.13.109             ether   20:fd:f1:1f:a3:fb   C                     enp7s0
10.12.13.138             ether   20:fd:f1:1f:a9:f5   C                     enp7s0
10.12.13.184             ether   20:fd:f1:1f:58:69   C                     enp7s0
10.12.13.112             ether   20:fd:f1:1f:71:a5   C                     enp7s0
10.12.13.143             ether   20:fd:f1:1f:73:5f   C                     enp7s0
10.12.13.170             ether   20:fd:f1:1f:97:c3   C                     enp7s0
10.12.13.189             ether   20:fd:f1:1f:a7:09   C                     enp7s0
10.12.13.124             ether   20:fd:f1:1f:50:0b   C                     enp7s0
10.12.13.200             ether   20:fd:f1:1f:6a:df   C                     enp7s0
10.12.13.139             ether   20:fd:f1:1f:58:f1   C                     enp7s0
10.12.13.164             ether   20:fd:f1:75:1f:7c   C                     enp7s0
10.12.13.185             ether   20:fd:f1:1f:4e:d9   C                     enp7s0
10.12.13.113             ether   20:fd:f1:1f:72:2d   C                     enp7s0
10.12.13.110             ether   20:fd:f1:1f:72:93   C                     enp7s0
10.12.13.151             ether   20:fd:f1:1f:54:29   C                     enp7s0
10.12.13.140             ether   20:fd:f1:75:23:56   C                     enp7s0
10.12.13.133             ether   20:fd:f1:75:21:7a   C                     enp7s0
10.12.13.160             ether   20:fd:f1:1f:91:c9   C                     enp7s0
10.12.13.171             ether   20:fd:f1:1f:8e:55   C                     enp7s0
10.12.13.125             ether   20:fd:f1:1f:51:5f   C                     enp7s0
10.12.13.201             ether   20:fd:f1:75:19:60   C                     enp7s0
10.12.13.136             ether   20:fd:f1:1f:97:7f   C                     enp7s0
10.12.13.147             ether   20:fd:f1:75:20:26   C                     enp7s0
10.12.13.172             ether   20:fd:f1:1f:97:a1   C                     enp7s0
10.12.13.165             ether   20:fd:f1:75:1e:28   C                     enp7s0
10.12.13.118             ether   20:fd:f1:1f:93:a5   C                     enp7s0
10.12.13.159             ether   20:fd:f1:1f:a2:41   C                     enp7s0
10.12.13.148             ether   20:fd:f1:1f:94:2d   C                     enp7s0
10.12.13.168             ether   20:fd:f1:1f:a4:a5   C                     enp7s0
10.12.13.141             ether   20:fd:f1:1f:80:85   C                     enp7s0
10.12.13.161             ether   20:fd:f1:1f:a0:21   C                     enp7s0
10.12.13.137             ether   20:fd:f1:1f:84:e7   C                     enp7s0
10.12.13.173             ether   20:fd:f1:1f:7c:01   C                     enp7s0
10.12.13.126             ether   20:fd:f1:1f:67:6b   C                     enp7s0
10.12.13.108             ether   20:fd:f1:1f:5a:ef   C                     enp7s0
10.12.13.119             ether   20:fd:f1:1f:58:47   C                     enp7s0
10.12.13.169             ether   20:fd:f1:1f:62:5f   C                     enp7s0
10.12.13.166             ether   20:fd:f1:1f:9c:8b   C                     enp7s0
10.12.13.122             ether   20:fd:f1:75:47:54   C                     enp7s0
gateway                  ether   4c:4e:35:97:1e:ef   C                     enp7s0
10.12.13.142             ether   20:fd:f1:75:e9:5c   C                     enp7s0
Entries: 42     Skipped: 0      Found: 42
akul@akul-Inspiron-5558:~$
```

arp -a: Shows the entries of the specified hosts. If the **hostname** parameter is not used, **all** entries will be displayed. The entries will be displayed in alternate (BSD) style.
When an incoming packet destined for a host machine on a particular local area network arrives at a gateway, the gateway asks the ARP program to find a physical host or MAC address that matches the IP address. The ARP program looks in the ARP cache and, if it finds the address, provides it so that the packet can be converted to the right packet length and format and sent to the machine. If no entry is found for the IP address, ARP broadcasts a request packet in a special format to all the machines on the LAN to see if one machine knows that it has that IP address associated with it. A machine that recognizes the IP address as its own returns a reply so indicating. ARP updates the ARP cache for future reference and then sends the packet to the MAC address that replied.



```
akul@akul-Inspiron-5558:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
10.12.13.127             ether   20:fd:f1:1f:6e:97   C                     enp7s0
10.12.13.109             ether   20:fd:f1:1f:a3:fb   C                     enp7s0
10.12.13.1               ether   ff:ff:ff:ff:ff:ff   CM                    enp7s0
10.12.13.138             ether   20:fd:f1:1f:a9:f5   C                     enp7s0
10.12.13.184             ether   20:fd:f1:1f:58:69   C                     enp7s0
10.12.13.112             ether   20:fd:f1:1f:71:a5   C                     enp7s0
10.12.13.143             ether   20:fd:f1:1f:73:5f   C                     enp7s0
10.12.13.170             ether   20:fd:f1:1f:97:c3   C                     enp7s0
10.12.13.189             ether   20:fd:f1:1f:a7:09   C                     enp7s0
10.12.13.124             ether   20:fd:f1:1f:50:0b   C                     enp7s0
10.12.13.200             ether   20:fd:f1:1f:6a:df   C                     enp7s0
10.12.13.139             ether   20:fd:f1:1f:58:f1   C                     enp7s0
10.12.13.164             ether   20:fd:f1:75:1f:7c   C                     enp7s0
10.12.13.185             ether   20:fd:f1:1f:4e:d9   C                     enp7s0
10.12.13.113             ether   20:fd:f1:1f:72:2d   C                     enp7s0
10.12.13.110             ether   20:fd:f1:1f:72:93   C                     enp7s0
10.12.13.2               ether   ff:ff:ff:ff:ff:00   CM                    enp7s0
10.12.13.151             ether   20:fd:f1:1f:54:29   C                     enp7s0
10.12.13.140             ether   20:fd:f1:75:23:56   C                     enp7s0
10.12.13.133             ether   20:fd:f1:75:21:7a   C                     enp7s0
10.12.13.160             ether   20:fd:f1:1f:91:c9   C                     enp7s0
10.12.13.171             ether   20:fd:f1:1f:8e:55   C                     enp7s0
10.12.20.20              ether   50:b7:c3:69:c6:2f   C                     enp7s0
10.12.13.125             ether   20:fd:f1:1f:51:5f   C                     enp7s0
10.12.13.201             ether   20:fd:f1:75:19:60   C                     enp7s0
10.12.13.136             ether   20:fd:f1:1f:97:7f   C                     enp7s0
10.12.13.147             ether   20:fd:f1:75:20:26   C                     enp7s0
10.12.13.172             ether   20:fd:f1:1f:97:a1   C                     enp7s0
10.12.13.165             ether   20:fd:f1:75:1e:28   C                     enp7s0
10.12.13.118             ether   20:fd:f1:1f:93:a5   C                     enp7s0
10.12.13.159             ether   20:fd:f1:1f:a2:41   C                     enp7s0
10.12.13.148             ether   20:fd:f1:1f:94:2d   C                     enp7s0
10.12.13.168             ether   20:fd:f1:1f:a4:a5   C                     enp7s0
10.12.13.141             ether   20:fd:f1:1f:80:85   C                     enp7s0
10.12.13.161             ether   20:fd:f1:1f:a0:21   C                     enp7s0
10.12.13.137             ether   20:fd:f1:1f:84:e7   C                     enp7s0
10.12.13.173             ether   20:fd:f1:1f:7c:01   C                     enp7s0
10.12.13.126             ether   20:fd:f1:1f:67:6b   C                     enp7s0
10.12.13.108             ether   20:fd:f1:1f:5a:ef   C                     enp7s0
10.12.13.119             ether   20:fd:f1:1f:58:47   C                     enp7s0
10.12.13.169             ether   20:fd:f1:1f:62:5f   C                     enp7s0
10.12.13.166             ether   20:fd:f1:1f:9c:8b   C                     enp7s0
10.12.13.122             ether   20:fd:f1:75:47:54   C                     enp7s0
gateway                  ether   4c:4e:35:97:1e:ef   C                     enp7s0
10.12.13.142             ether   20:fd:f1:75:e9:5c   C                     enp7s0
Entries: 45     Skipped: 0      Found: 45
akul@akul-Inspiron-5558:~$ 
```

Each complete entry in the ARP cache will be marked with the C flag. Permanent entries are marked with M and published entries have the P flag.

The commands to delete and add entries in ARP table are as follows.

To run these commands, one has to run them as a root user (use sudo).

To delete an entry from the ARP table, the command is "arp –d <address>".

To add/modify MAC address used for an IP, the command is "arp -s <ip_addr> <MAC_addr>".

The entries stay cached in the ARP table for 60 seconds (1 minute).

This can be found from command:

"cat /proc/sys/net/ipv4/neigh/default/gc_stale_time".

A trial & error method to discover the timeout value is:

Add an entry in the arp table, check the arp table continuously (load at gaps of, say 5 ms) and note the time at which the entry becomes stale (in the output of ip -s neigh list).

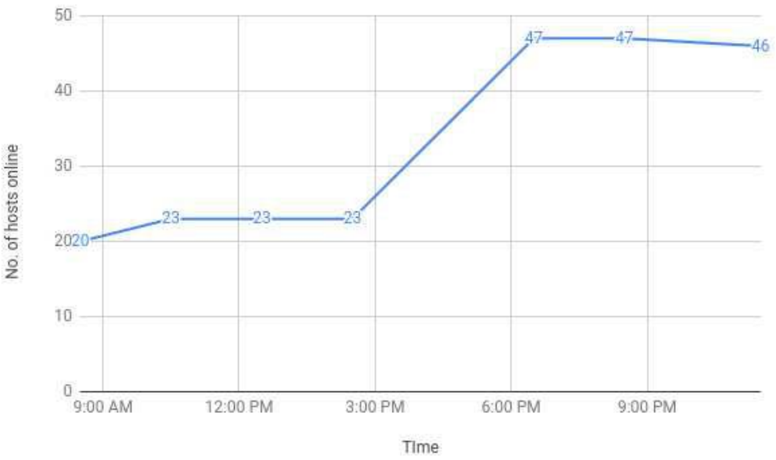
If two IP addresses were mapped to the same MAC Address in an ARP cache of a PC, the packets sent to either IPs will actually be sent to the machine having actual MAC Address same as that in the PC's ARP.

Although, if two IP addresses have the same MAC address, the problem will be even bigger. If the two Pcs are in different LANs, then there wouldn't be any issue. But if the two IP addresses with same MAC addresses are in the same LAN, then all the traffic directed to that MAC address maybe distributed among the two. And if, say one of them was a server, then the server load will be distributed among the two, thus, forcing unnecessary load on other PC.

Functionality:

When a source device want to communicate with another device, source device checks its Address Resolution Protocol (ARP) cache to find it already has a resolved MAC Address of the destination device. If it is there, it will use that MAC Address for communication. If ARP resolution is not there in local cache, the source machine will generate an Address Resolution Protocol (ARP) request message. The source broadcast the Address Resolution Protocol (ARP) request message to the local network. The message is received by each device on the LAN since it is a broadcast. Each device compares the Target Protocol Address (IPv4 Address of the machine to which the source is trying to communicate) with its own Protocol Address (IPv4 Address). Those who do not match will drop the packet without any action. When the targeted device checks the Target Protocol Address, it will find a match and will generate an Address Resolution Protocol (ARP) reply message. The destination device will update its Address Resolution Protocol (ARP) cache and send the Address Resolution Protocol (ARP) reply message as a unicast. The source machine will process the Address Resolution Protocol (ARP) reply from destination and update its Address Resolution Protocol (ARP) cache with the Sender Hardware Address and Sender Protocol Address it received from the Address Resolution Protocol (ARP) reply message.

Q8.



Command used: "nmap –n –sP 172.16.112.0/24"

From the graph, it is observed that the number of hosts are quite low in the morniing. This number takes a big jump (more than double), when observed in the evening. Thus, more number of people are using the lan in the evening as compared to morning. This also seems logical because the morning faces a lan ban, unlike the evening. The number of hosts online starts to decrease after that.