







## Question 1: Ping Command and its options

- a) To specify the number of echo requests to send: `ping -c <count> <IP_address>`  
 b) To set time interval between two successive ping requests: `ping -i <time> <IP_address>`  
 c) (I) `ping -l <preload> <address>` (II) Limit for sending such requests is 3 for normal users.  
 d) (I) `ping -s <packet_size><addr>` (II) 40 Bytes (32 bytes(payload size) + 8 bytes(ICMP header size))  
 If we consider the IP Address header size also i.e. 20 bytes, then the answer will be 60 Bytes .

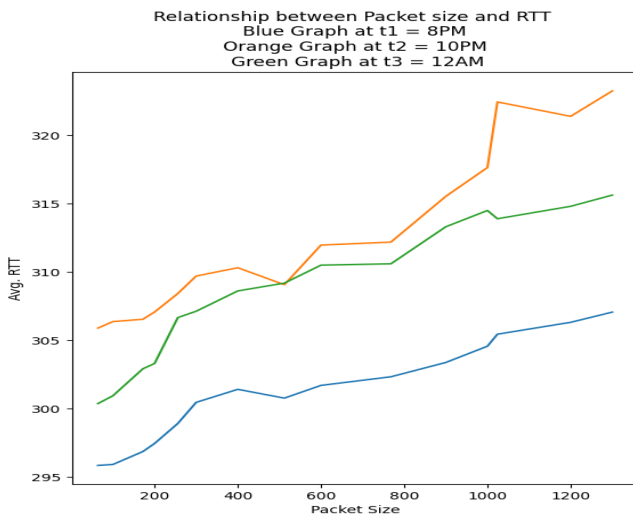
## Question 2: Dependence of ping RTT on different parameters

Hosts Chosen: 1. <https://www.google.com/> 2. <https://www.facebook.com/> 3. <https://in.yahoo.com/>  
 4. <https://codeforces.com/> 5. <https://louisvuitton.com/> 6. <https://www.spotify.com/in/>

Host IP Address	Location of the host	11 PM IST (RTT & %loss)	8 PM IST (RTT & %loss)	10 PM IST (RTT & %loss)	Avg. RTT (in ms)	Distance of host (in kms)	%age Loss of packets
173.194.216.104	Google, US 	299.515 ms, 0%	348.68 ms, 0%	299.49 ms, 0%	315.895	12619	0%
31.13.67.35	Facebook, Ireland, 	292.031 ms, 0%	339.88 ms, 0%	292.86 ms, 0%	308.256	7826	0%
74.6.143.26	Yahoo, US 	256.342 ms, 0%	279.08 ms, 0%	258.69 ms, 0%	264.704	12401	0%
81.27.240.126	Tver, Russia 	216.646 ms, 0%	327.40 ms, 0%	233.46 ms, 0%	259.168	4444	0%
23.218.122.23	France 	130.963 ms, 10%	167.31 ms, 15%	180.604 ms, 10%	159.62	6182	11.67%
35.186.224.25	London, UK 	249.10 ms, 0%	204.69 ms, 0%	193.40 ms, 0%	215.73	6834	0%

- a) RTTs for above hosts are weakly dependent on the geographical distance of hosts. The reason behind this observation is that the packet travels with speed of light between two hops. RTT strongly depends on the number of hops to the host because main latency comes by waiting in a queue of hops.
- b) The percentage packet loss is non-zero in my fifth host i.e. 23.218.122.23(louisvuitton.com), The reason behind this is that the data must travel through multiple links and routers during its trip across the network. When your data arrives at a router with full capacity, then it must wait for its turn before being sent across the wire. If a network device is falling very far behind, it won't have room for the new data to wait, so it discards that particular information which leads to packet loss.
- c) Host used to analyse the relationship between packet size and RTT :- Facebook( second host in the table ) i.e. 31.13.67.35. The graph is shown on the next page:-
- d) Effect of Packet Size on RTT for a single host:- It is visible from the above plot that as packet size increases, the RTT also increases with some exceptions in between. Every router and switch along the path has to receive the entire packet before it can forward it. The latency introduced at each point thus equals the speed of the inbound link in bits per second divided by the frame size in bits. Larger packet size leads to increased latency.

## Graph of RTT(in ms) vs Packet size(in bytes)



Effect of time of the day on RTT for a single host:- ISP gateway can handle a constant number of requests per second. The ping time is increased sometimes because during that time there may be more traffic for this ISP, therefore this high number of requests exceed the number of requests that ISP gateway can handle. So some of the requests including our request remain in the gateway queue and this may cause some delay in responding to these requests and finally increase RTT.

## Question 3: Analysis of two different ping commands

I have used my Facebook host i.e. 31.13.67.35 for these experiments.

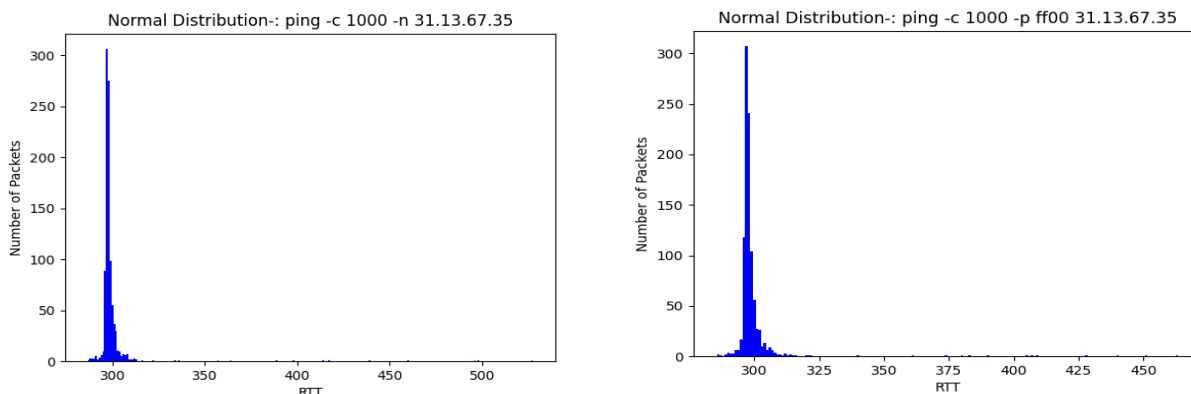
Commands used :-

- 1) `ping -c 1000 -n 31.13.67.35 >> ping1.txt`      2) `ping -c 1000 -p ff00 31.13.67.35 >> ping2.txt`

- a) 0% packet loss for command-1 and 0% packet loss for command-2.  
 b) I ran a python script and scrapped the time from the output files(`ping1.txt` & `ping2.txt`) created using the above commands. I put them into the list and calculated the required latencies using python.

Sr. No.	Min Latency	Max Latency	Avg Latency	Median Latency
Command-1	287.824 ms	427.600 ms	302.014 ms	302.125 ms
Command-2	288.073 ms	463.000 ms	304.174 ms	302.800 ms

- c) The graph shows normal distribution of ping time in milliseconds. The graph on the left shows the distribution of command -1( `ping -c 1000 -n 31.13.67.35 >> ping1.txt` ) while on the right shows the distribution of command -2( `ping -c 1000 -p ff00 31.13.67.35 >> ping2.txt` ).



- d) **Observation:** Mean ping latency for command-2 is **more than** mean latency in command-1. This is because in command 1 due to the option `-n` there is no reverse DNS lookup which makes it faster to execute. In the second command, we are sending pattern 1111111000000000 (`ff00`). Sending eight continuous 1s followed by eight consecutive 0s(padding with the packet) is always an error prone task (due to less randomization in this process) which raises synchronization problems. This requires resending the same packets to the host which results in higher latency.

## Question 4: ifconfig and route commands

- a) The output of `ifconfig`(interface configuration) command:-

```

kartikay@krtky:~/Desktop/codes$ ifconfig
enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether 54:bf:64:51:c0:0f txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 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 1941 bytes 177251 (177.2 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1941 bytes 177251 (177.2 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.29.47 netmask 255.255.255.0 broadcast 192.168.29.255
        inet6 fe80::3609:9ad5:54e1:d529 prefixlen 64 scopeid 0x20<link>
        inet6 2405:201:5801:e03c:3cf2:8231:bd5c:f18b prefixlen 64 scopeid 0x0<global>
        inet6 2405:201:5801:e03c:2023:1f8c:62e:dfba prefixlen 64 scopeid 0x0<global>
        ether 00:bb:60:7c:6a:e6 txqueuelen 1000 (Ethernet)
        RX packets 847702 bytes 379567779 (379.5 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 683188 bytes 158166323 (158.1 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

This command is used to view and change the configuration of the network interfaces on your system. If no arguments are given, ifconfig displays the status of the currently active interfaces. The output of running ifconfig is described below:-

- The **enp2s0** is the wired ethernet interface, **wlp2s0** is the wireless ethernet interface and **lo** is the loopback interface which is a virtual network interface that is used by the computer to communicate to itself.
- **MTU** is the 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.
- **RX and TX Packets** are the number of packets received and transmitted through the interface respectively. The number of bytes corresponding to each are also specified. The number of packets dropped, overrun, collided, or had error transmitting while receiving or transmitting is also mentioned for both RX and TX packets.
- **Txqueuelen** denotes the transmit queue length of the device.
- **Inet addr** and **inet6 addr** are the IPV4 and IPV6 address assigned when the machine is connected to the network.
- **Bcast** denotes the Broadcast Address (address at which all devices connected to the network are enabled to receive datagrams).
- **Mask** is the network mask which we passed using the netmask option. This is required to extract the network address and host address from the IP address.
- **Flags** denote the status of the interface and its facilities. Example, the UP flag indicates an active interface. Running flag indicates that the interface is ready to accept data. Broadcast flag indicates that a broadcast address has been set. Multicast flag indicates that the interface supports multicasting, i.e., it allows a source to send a packet to multiple machines if the machines are watching out for that packet.

b) Various options about network interfaces and its flags can be specified along with ifconfig:

- a : Display information for all network interfaces, even if they are down.
- s : Display a short list in a format identical to the command **netstat -i**.
- v : Verbose mode; display additional information for certain error conditions.
- up : This flag causes the interface to be activated.
- down : This flag causes the driver for this interface to be shut down.
- mtu N : sets the maximum transfer unit of an interface (limit the maximum packet size).

c) Route command manipulates and displays the system's IP routing tables. Output Explanation:-

```

kartikay@krtky:~/Desktop/codes$ route
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
default        reliance.relian 0.0.0.0         UG    600    0      0 wlo1
link-local     0.0.0.0         255.255.0.0     U      1000   0      0 wlo1
192.168.29.0   0.0.0.0         255.255.255.0   U      600    0      0 wlo1

```

- **Destination** : The destination network or destination host.
- **Gateway** : It points to the gateway through which network can be reached (\* if none set)
- **Genmask** : It is the netmask for the destination net; The value is 255.255.255.255 for a host destination and 0.0.0.0 for the default route.
- **Flags** : These are status indicators. Flag U denote that the route is up, Flag G signifies that the route is to a gateway. Flag H signifies that route is to a host i.e. the dest. is a complete host address.
- **Metric** : The Metric indicates the associated cost of using the indicated route. This is useful for determining the efficiency of a certain route from two points in a network.
- **Ref** : Indicates the number of references to this route.

- Use : Indicates the count of lookups for the route.
- Iface: Interface to which packets for this route will be sent.

d)

```
kartikay@krtky:~$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 192.168.29.1 0.0.0.0 UG 600 0 0 wlo1
169.254.0.0 0.0.0.0 255.255.0.0 U 1000 0 0 wlo1
192.168.29.0 0.0.0.0 255.255.255.0 U 600 0 0 wlo1
kartikay@krtky:~$ route -e
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default reliance.relian 0.0.0.0 UG 0 0 0 wlo1
link-local 0.0.0.0 255.255.0.0 U 0 0 0 wlo1
192.168.29.0 0.0.0.0 255.255.255.0 U 0 0 0 wlo1
kartikay@krtky:~$ sudo route add -net 175.56.76.0 netmask 255.255.255.0 wlo1
[sudo] password for kartikay:
kartikay@krtky:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default reliance.relian 0.0.0.0 UG 600 0 0 wlo1
link-local 0.0.0.0 255.255.0.0 U 1000 0 0 wlo1
175.56.76.0 0.0.0.0 255.255.255.0 U 0 0 0 wlo1
192.168.29.0 0.0.0.0 255.255.255.0 U 600 0 0 wlo1
kartikay@krtky:~$ sudo route del -net 175.56.76.0 netmask 255.255.255.0 wlo1
kartikay@krtky:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default reliance.relian 0.0.0.0 UG 600 0 0 wlo1
link-local 0.0.0.0 255.255.0.0 U 1000 0 0 wlo1
192.168.29.0 0.0.0.0 255.255.255.0 U 600 0 0 wlo1
```

Various options along with route command are (screenshot is attached above):

1. -n: show numerical addresses instead of trying to determine symbolic hostnames.
2. -e: use netstat-format for displaying the routing table.
3. add: add a new route. While adding a new route, -net specifies the destination network and netmask specifies the Genmask.
4. del: While deleting a route, -net specifies destination network to be deleted and netmask specifies the Genmask.

## Question 5: netstat command( network statistics )

a) Netstat is a command-line network utility tool that displays network connections for the TCP, routing tables, and several network interfaces. It is one of the most basic network debugging tools and is used to find problems in the network and to determine the amount of traffic on the network as a performance measurement by telling what ports are open and whether any programs are listening on ports.

b) netstat -at is used to show all TCP connections. We can use netstat -at | grep "ESTABLISHED" command to show only the ESTABLISHED TCP connections.

```
kartikay@krtky:~$ netstat -at | grep "ESTABLISHED"
tcp6      0      0 krtky:33728      whatsapp-cdn6-shv:https ESTABLISHED
tcp6      0      0 krtky:57924      g2600-140f-dc00-0:https ESTABLISHED
tcp6      0      0 krtky:46986      2404:6800:4003:c03:5228 ESTABLISHED
```

c) The output of the netstat -r command is as follows:

```
kartikay@krtky:~$ netstat -r
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default reliance.relian 0.0.0.0 UG 0 0 0 wlo1
link-local 0.0.0.0 255.255.0.0 U 0 0 0 wlo1
175.56.76.0 0.0.0.0 255.255.255.0 U 0 0 0 wlo1
192.168.29.0 0.0.0.0 255.255.255.0 U 0 0 0 wlo1
```

-> netstat -r shows the kernel routing table of the machine. The output of the command is explained below:

1. **Destination:** It indicates the pattern that the destination of a packet is compared to. While sending a packet over the network, this table is examined in top-down fashion, and the first line that matches is the destination for the packet.
2. **Gateway:** It indicates where to send a packet that matches the destination of the same line. An asterisk means send the packet locally as the destination is on the same network.
3. **Genmask:** It is the netmask for the destination network. It tells the number of bits from the start of the IP address used to identify the subnet.
4. **Flags:** This column indicates which flags apply to the current table line. Flag U indicates that the route is up, Flag G signifies that the route is to a gateway. Moreover, Flag H signifies that the route is to a host which means that the destination is a complete host address.
5. **MSS:** Maximum Segment Size is the size of the 4 largest datagram that the kernel constructs for transmission via this route. It is a TCP parameter which is used to split packets when the destination cannot handle large packets.
6. **Window:** This column indicates the window size which denotes how many TCP packets can be sent before at least one of them has to be acknowledged.
7. **irtt:** Initial Round Trip Time is used by the kernel to guess about the best TCP parameters without waiting for slow replies.
8. **Iface:** it indicates which network interface to be used for sending packets that match the destination.

d) netstat -i is used to display the status of all network interfaces. As it can be seen in the screenshot, three network interfaces are present on my system (calculated using netstat -i | wc -l)

```

kartikay@krtky:~$ netstat -l
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 15624 0 0 0 15624 0 0 0 LRU
wlo1 1500 400305 0 0 0 295879 0 0 0 BMU
kartikay@krtky:~$ netstat -su
IcmpMsg:
  InType3: 751
  InType11: 48
  OutType3: 1231
Udp:
  22934 packets received
  499 packets to unknown port received
  0 packet receive errors
  19539 packets sent
  0 receive buffer errors
  0 send buffer errors
  IgnoredMulti: 27
UdpLite:
IpExt:
  InMcastPkts: 2883
  OutMcastPkts: 2741
  InBcastPkts: 27
  OutBcastPkts: 12
  InOctets: 37495586
  OutOctets: 8278866
  InMcastOctets: 232268
  OutMcastOctets: 325727
  InBcastOctets: 3166
  OutBcastOctets: 752
  InNoECTPkts: 67918

```

e) `netstat -su` is used to show the statistics of all UDP connections. The screenshot is attached.

f) The loopback device is a virtual network interface that the system uses to communicate with itself. It is not an actual hardware but helps the applications running on the machine to connect to servers on the same machine. The IPv4 address for accessing loopback interface is 127.0.0.1. It is used majorly for diagnostics and troubleshooting, and to connect to servers running on the local machine. Apart from using it as a diagnostic tool, it is used when a server offering a resource is running on the system itself. The loopback interface is shown in the screenshot of `ifconfig` command in the previous question under the head `lo`.

## Question 6: Traceroute

a) Traceroute is a network diagnostic tool used to track

the pathway taken by a packet on an IP network from source to destination in real-time. It lists the IP addresses of all the routers it pinged in between. Traceroute also records the time taken for each hop the packet makes during its route to the destination. Traceroute is a useful tool for determining the response delays and routing loops present in a network pathway across packet switched nodes. It also helps to locate any points of failure( denoted by an asterisk) encountered while enroute to a certain destination.

b) The hop counts for each host in each time slot are shown in the table below.

Hop Count Time	Google	Facebook	Yahoo	Codeforces	LouisVuitton	Spotify
4:30 PM	28	16	18	61 ( incomplete )	35	12
6:30 PM	28	16	18	62 ( incomplete )	36	12
9:00 PM	28	16	20	61 ( incomplete )	35	12

Yes, there existed some common hops as all packets initially pass through the default gateway within the home network. The common hops were 192.168.29.1 (my device) and 10.15.80.1, 172.16.26.5, 172.17.1.70.

c) While going to [codeforces.com](https://codeforces.com) server at 6:30pm and [louisvuitton.com](https://louisvuitton.com) at 6:30 PM, the packets went through an additional router which was the only change in their path. The reason for this would most probably be some hardware failure on the path or network congestion. Migration of destination VM servers across data centres may also have caused a change in hop count. Variations in the amount of traffic that the different hops are experiencing as the packets transverse the network also results in different output. Another reason might be the fast switching technique which changes the routing table.

d) Incomplete tracing of route occurred in the case of [codeforces.com](https://codeforces.com). Loss of ICMP/UDP reply packets from intermediate hosts or no reply from the host can be a reason for this. The reason can also be from the sender's side(sender timeout or ICMP/UDP packet not sent with incremental TTL value). Sometimes routers and servers have firewall enabled which either blocks the ICMP traffic or hides the IP Addresses of the hosts to be traced by traceroute. If none of the above problems is present, then there must be a fault in the router itself.

e) Yes, it is possible to find paths using traceroute in cases where ping fails. In ping, intermediate hosts forward the ICMP packets and the destination host replies, thus ping relies on the reply packet. Traceroute works by sending the packets of data with low survival time (Time to Live – TTL) which specifies how many hops the packet can survive before it is returned. Each intermediate host needs to respond with an ICMP/UDP packet. Hence, even if the destination host doesn't respond to ping, a partial path can always be found provided the given source is not blocked from receiving responses. Moreover, if you are not able to connect to a server, traceroute can be used to find the epicenter of the failure.

## Question 7: Arp Command(Address Resolution Protocol)

```

kartikay@krtky:~$ arp
Address HWtype HWaddress Flags Mask Iface
reliance.reliance ether 64:cc:22:10:32:7e CM wlo1
175.56.76.0 (incomplete) wlo1

```

a) To see the full arp table, `arp` command is used. ARP table stores IP addresses and the corresponding MAC addresses of the hosts on the network. It can be used to find out the destination MAC address while sending packets to other hosts.



### Explanation of the output:-

- **Address** : It is the IP address of the connected host on the network.
- **HWtype** : It indicates whether the host has an ethernet interface.
- **HWaddress** : It is the corresponding MAC address.
- **Flags Mask**: They indicate if the mac address has been learned by the system by connecting to the host(denoted by C flag) or/and manually set(denoted by M flag).
- **Iface** : It denotes the interface connecting them.

- b) `sudo arp -s <ip_address> <MAC_address>` command is used to add an entry.  
`sudo arp -d <ip_address>` command is used to delete an entry.

```
kartikay@krtky:~$ sudo arp -s 192.168.29.2 64:cc:22:10:32:7e
kartikay@krtky:~$ sudo arp -s 192.168.29.3 64:cc:22:10:32:7e
kartikay@krtky:~$ sudo arp -s 192.168.29.4 64:cc:22:10:32:7e
kartikay@krtky:~$ sudo arp -s 192.168.29.5 64:cc:22:10:32:7e
kartikay@krtky:~$ arp
Address                  HWtype  HWaddress           Flags Mask    Iface
192.168.29.4             ether    64:cc:22:10:32:7e    CM            wlo1
192.168.29.5             ether    64:cc:22:10:32:7e    CM            wlo1
reliance.reliance        ether    64:cc:22:10:32:7e    CM            wlo1
192.168.29.2             ether    64:cc:22:10:32:7e    CM            wlo1
175.56.76.0              (incomplete)
192.168.29.3             ether    64:cc:22:10:32:7e    CM            wlo1
kartikay@krtky:~$ sudo arp -d 192.168.29.2
kartikay@krtky:~$ sudo arp -d 192.168.29.3
kartikay@krtky:~$ sudo arp -d 192.168.29.4
kartikay@krtky:~$ sudo arp -d 192.168.29.5
kartikay@krtky:~$ arp
Address                  HWtype  HWaddress           Flags Mask    Iface
reliance.reliance        ether    64:cc:22:10:32:7e    CM            wlo1
175.56.76.0              (incomplete)
```

- c) ARP only works between devices in the same IP subnet. When a device with IP address X needs to send a packet to a device with IP address Y, the first thing it does is looking up its routing table to determine if IP address Y belongs to a subnet it can directly reach through its network interface and if it does, then devices X uses ARP to map IP address Y to a physical Ethernet address. But if the two IP Addresses are on different subnets it will look in its routing table for a route to the destination network, and then it will send its packet to the appropriate router. In some cases, devices on a subnet may respond on behalf of other devices outside the subnet, such as a gateway acting as an ARP proxy.

- d) Forcefully replacing the ethernet address of an existing entry and pinging resulted in a 100% packet loss. This is because the ping command operates on layer 3(the network layer) while ARP operates on layer 2(the link layer). So, when you ping an IP address, the layer 3 header is built first and passed to layer 2. At layer 2, the PC checks its ARP table for the corresponding MAC address. The device sends out an ARP request to the destination MAC address. If one of the devices reading that frame has that IP address, then it sends out an ARP reply with a destination MAC address of the device that sent the ARP request and its own MAC address as the source MAC address. If the device reading the ARP request doesn't have the particular IP address in the request, then the requesting device does not receive a reply to its request. Because no reply comes to the system, ping connection timeout occurs eventually.

## Question 8: Nmap

- a) IP Address of my LAB PC is 172.16.114.184/25.

I have used `nmap -n -sP 172.16.114.184/25` command to check which PCs of my subnet are up.

- b) I used `sudo nmap -sA -T4 172.16.114.184/25` command to detect firewall settings of my Lab PC.

- c) Plot of number of hosts online vs Time of the Day. We can see that LAN has the maximum number of active hosts during the day hours i.e. ( 11 AM-5 PM ) and number of active hosts start falling after 2 am and are minimum in morning hours.

Number of hosts online vs Time of the Day

Time of the day	No. of hosts
9 AM	48
11 AM	75
1 PM	64
4 PM	89
6 PM	67
2 AM	63

