Question 1

a) Option required to specify number of echo requests to send with ping :
-c *count*

b) Option required to set time interval (in seconds), between two successive ping ECHO_REQUESTs :
-i *interval*

c) 1. Command to send ECHO_REQUEST packets to the destination one after another without waiting for a reply:
ping –f *destination*
2. The limit for sending such ECHO_REQUEST packets by normal users (not super user):
Interval b/w successive requests can't be <0.2s

d) 1. Command to set the ECHO_REQUEST packet size (in bytes) :
ping –s *size*
2. If packet size is set to 64, total packet size total packet size will be :
64 + 8 = **72**

Question 2

- **Hosts chosen:**
  1. amazon.in
  2. amazon.com
  3. google.co.in
  4. sonyliv.com
  5. baidu.com
- **%Loss, Average RTT**

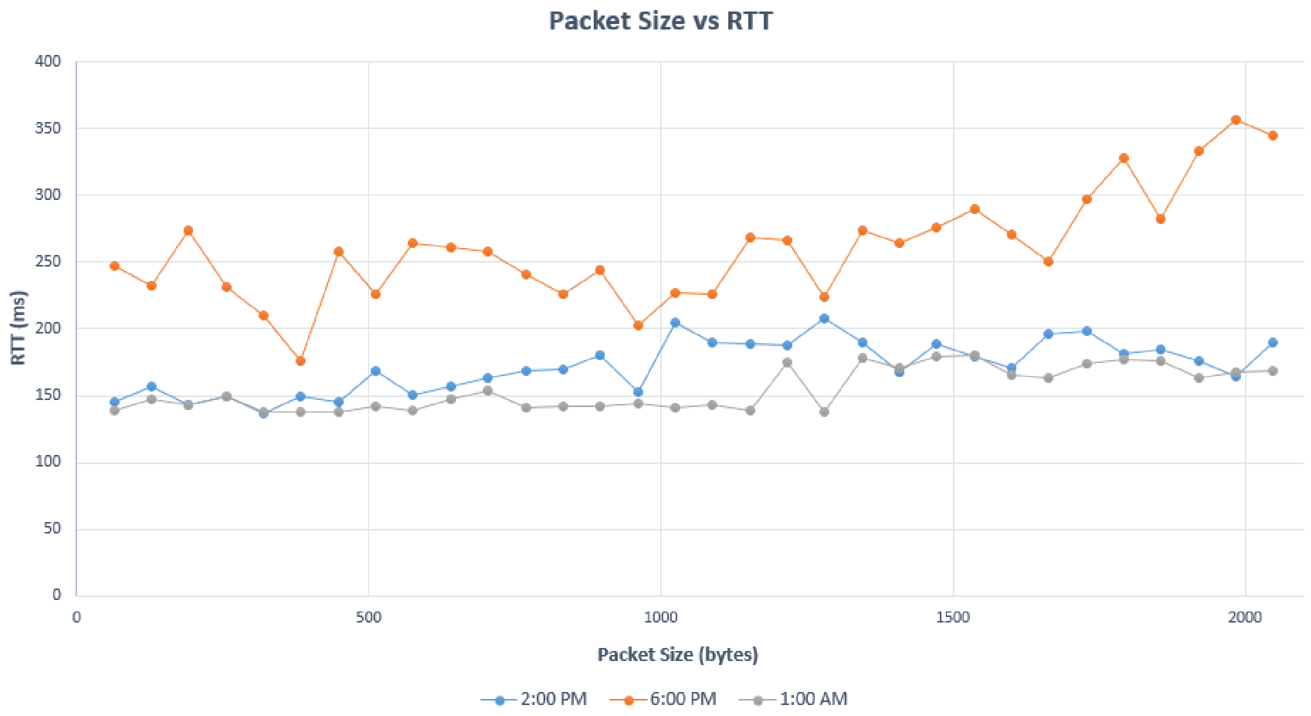
	2 PM IST		**Host Address**	**Host Location**	**Geographical Distance**
	%Loss	Avg. RTT			
amazon.in	0%	382.907	52.95.120.67	Dublin, Leinster, Ireland (IE) 🇮🇪	8400 km
amazon.com	0%	544.080	205.251.242.103	Ashburn, Virginia, United States (US) 🇺🇸	12676 km
google.co.in	0%	132.752	216.58.196.195	Mountain View, California, United States (US) 🇺🇸	12604 km
sonyliv.com	0%	155.858	52.77.194.240	Singapore, Central Singapore Community Development Counc, Singapore (SG) 🇸🇬	3053 km
baidu.com	10%	695.384	111.13.101.208	China (CN) 🇨🇳	2427 km

	6 PM IST		**Host Address**	**Host Location**	**Geographical Distance**
	%Loss	Avg. RTT			
amazon.in	0%	371.658	52.95.116.115	Dublin, Leinster, Ireland (IE) 🇮🇪	8400 km
amazon.com	0%	587.257	205.251.242.103	Ashburn, Virginia, United States (US) 🇺🇸	12676 km
google.co.in	0%	109.761	216.58.196.195	Mountain View, California, United States (US) 🇺🇸	12604 km
sonyliv.com	0%	242.189	52.77.194.240	Singapore, Central Singapore Community Development Counc, Singapore (SG) 🇸🇬	3053 km
baidu.com	50%	777.711	111.13.101.208	China (CN) 🇨🇳	2427 km

	1 AM IST		**Host Address**	**Host Location**	**Geographical Distance**
	%Loss	Avg. RTT			
amazon.in	0%	296.971	52.95.116.115	Dublin, Leinster, Ireland (IE) 🇮🇪	8400 km
amazon.com	0%	469.602	205.251.242.103	Ashburn, Virginia, United States (US) 🇺🇸	12676 km
google.co.in	0%	88.291	216.58.196.195	Mountain View, California, United States (US) 🇺🇸	12604 km
sonyliv.com	0%	151.929	52.77.194.240	Singapore, Central Singapore Community Development Counc, Singapore (SG) 🇸🇬	3053 km
baidu.com	0%	577.436	220.181.57.216	Beijing, Beijing, China (CN) 🇨🇳	2427 km

- **Ping to baidu.com has non zero packet loss at first two observations by following reason :**
  - Data must travel through multiple devices and links during its trip across your network. If one of these links is at full capacity when your data arrives, then it must wait its turn before being sent across the wire (this is known as queuing).If a network device is falling very far behind, it won't have room for the new data to wait (queue), so it does the only thing it can, which is to discard the information.

- Note that **RTTs for above hosts are weakly correlated with geographic distance of host** as packet travels with speed of light between two hops. It Is strongly correlated with number of hops to the host because main latency comes by waiting in queue of hop.
- **I have used 'sonyliv.com' host to analyse relationship between packet size and RTT.**
- From graph, we can observe that **packet size is weakly correlated with RTT.**
- **Effect of packet size on RTT:**
  - Every router and switch along the path has to receive the entire packet/frame before it can forward it. The latency introduced at each point thus equals the speed of the inbound link in bps divided by the frame size in bits. Larger frames = increased latency.
- **Effect of time of day on RTT:**
  - Internet service provider gateway can handle constant number of request per second. So in some hours of day it is seen that ping time increased, this is because of in that time there are more internet users from this ISP that sending request to gateway, therefore this high number of requests exceed the number of requests that ISP gateway can handle. so some of the request included your ping request should be remains in gateway queue and this may cause some delay in answering these request and finally it can increase the ping time.



Question 3

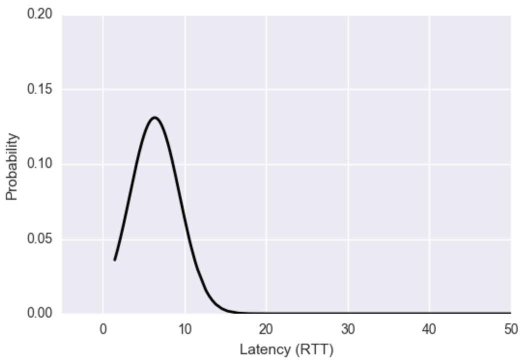
- **IP Address used**: 172.16.114.140

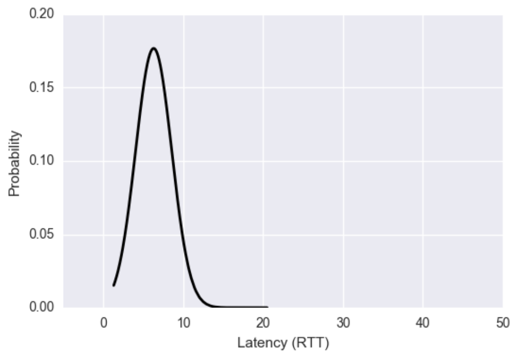
- **Output of commands are stored in files as following**:
  1. ping –c 1000 -n 172.16.114.140 >> ping_n.txt
  2. ping –c 1000 –p ff00 172.16.114.140 >> ping_p.txt

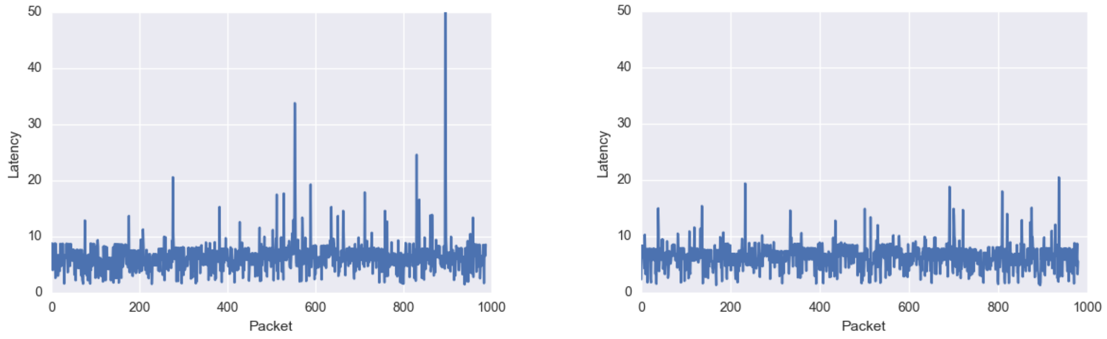
- **Method used to analyse:**
  By running python script I have time from output file and put them into list and plotted it.

a) b)	%Loss	Minimum	Maximum	Mean	Median	SD
Command 1	1.2.% (12/1000)	1.47	52.2	6.36	6.05	3.04
Command 2	2% (20/1000)	1.30	20.5	7.02	6.78	2.25

c) **Plots :**





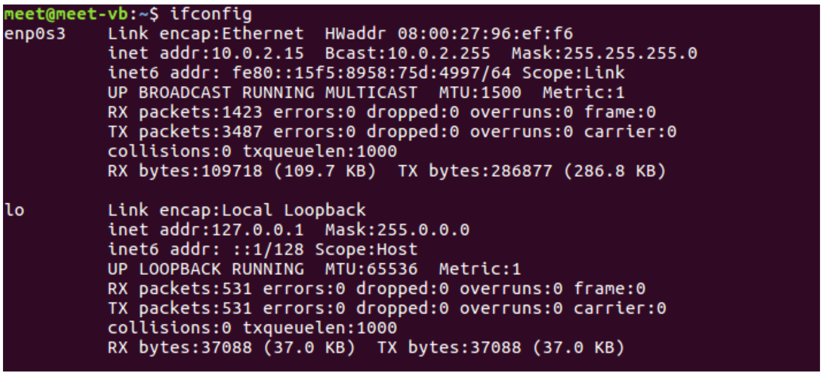


(a) Command 1 **(b) Command 2**

d) Behavior observed :

- In second case, we are sending pattern 1111111100000000. Sending continuous 1s and 0s is always a error prone task and synchronization issue come into picture. This requires resending of same packets. This results in high latency.
- Also in first case, no attempt will be made to look up symbolic names for host address. So it is likely that case 2 is more likely to have more RTT.
- Both of above supports that second case have more RTT than first case.

QUESTION 4

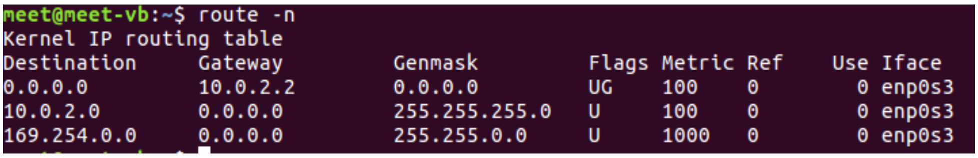


```
meet@meet-vb:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:96:ef:f6
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::15f5:8958:75d:4997/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1423 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3487 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:109718 (109.7 KB)  TX bytes:286877 (286.8 KB)

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:531 errors:0 dropped:0 overruns:0 frame:0
          TX packets:531 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:37088 (37.0 KB)  TX bytes:37088 (37.0 KB)
```

Output	
Link encap:Ethernet	**This denotes that the interface is an Ethernet related device.**
inet addr	It indicates the machine IP address
Bcast	It denotes the broadcast address
Mask	It 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	It denotes that the Ethernet device supports broadcasting - a necessary characteristic to obtain IP address via DHCP.
NOTRAILERS	It indicate that trailer encapsulation is disabled. Linux usually ignore trailer encapsulation so this value has no effect at all.
RUNNING	The interface is ready to accept data.
MULTICAST	This indicates that the Ethernet interface supports multicasting.
MTU	It is the size of each packet received by the Ethernet card.
Metric	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
txqueuelen	This denotes the length of the transmit queue of the device.
RX Bytes, TX Bytes	These indicate the total amount of data that has passed through the Ethernet interface either way.
Options	
up	This flag causes the interface to be activated. It is implicitly specified if an address is assigned to the interface.
down	This flag causes the driver for this interface to be shut down.
-a	Display all interfaces which are currently available, even if down
interface	The name of the interface. This is usually a driver name followed by a unit number, for example eth0 for the first Ethernet interface.

- **Explaination of route command :**



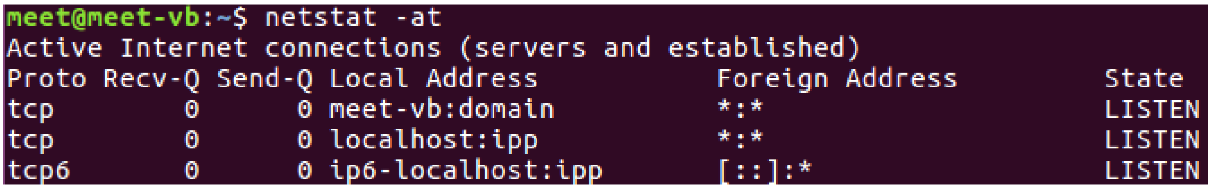
```
meet@meet-vb:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
```

Output	
Destination	The destination network or destination host.
Gateway	The gateway address or '*' if none set.
Genmask	The netmask for the destination net; '255.255.255.255' for a host destination and '0.0.0.0' for the default route.
Flags	U : route is uo and G : use gateway
Metric	The 'distance' to the target (usually counted in hops). It is not used by recent kernels, but may be needed by routing daemons.
Ref	Number of references to this route.
Use	Count of lookups for the route.
Iface	Interface to which packets for this route will be sent.
Options	
del	Delete a route
add	Add a new route
target	The destination network or host
-net	Target is network
-host	Target is a host
-v	Verbose
-n	Show numerical addresses instead of trying to determine symbolic host names.

Question 5

- **Netstat and its use:**
    - netstat ("network statistics") is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface (network interface controller or software-defined network interface) and network protocol statistics.
    - It is useful to show which programs on network are active right now, helps network admin to keep eye on invalid/suspicious network connection, helps in finding problems in network and determine traffic on network.

- **To show all the TCP connections established :** ***netstat -at***

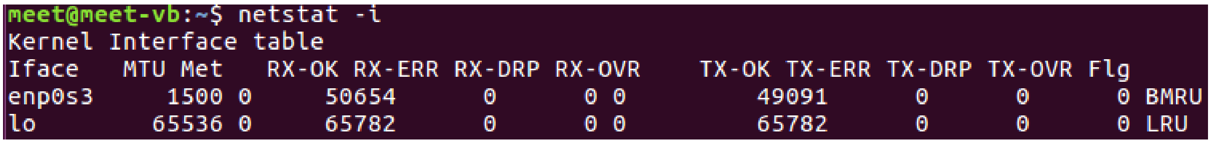


```
meet@meet-vb:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 meet-vb:domain          *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
```

- **Explaination:**

Proto	Tells socket is TCP or UDP
Local Address	IP and port of clocal computer
Recv-Q Send-Q	Count of bytes not copied by user prog connected to socket, tells us how much data is in queue for that socket waiting to be read or sent
Foreign Address	IP and port of foreign device (other end socket)
State	Tells about state of listed sockets, LISTEN : wait for external computer to contact us, ESTABLISH : ready to communicate, TIME_WAIT : waiting to be closed

- **Output of 'netsh –r' :**
  It shows kernel routing table, i.e. it shows same output as route command does. Explained in above question.
- **To display network interface status:**
  'netstat –i' , as you can see it shoed enp0s3 and lo (loopback interface)



```
meet@meet-vb:~$ 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
enp0s3    1500 0     50654      0      0 0         49091      0      0      0 BMRU
lo       65536 0     65782      0      0 0         65782      0      0      0 LRU
```

- **Loopback interface:**
  - The loopback interface is a **virtual interface**. The only purpose of the loopback interface is to return the packets sent to it, i.e. whatever you send to it is received on the interface.
  - It makes little sense to put a default route on the loopback interface, because the only place it can send packets to is the imaginary piece of wire that is looped from the output of the interface to the input.
  - It is used mainly for diagnostics and troubleshooting and connect to servers running on local machine.

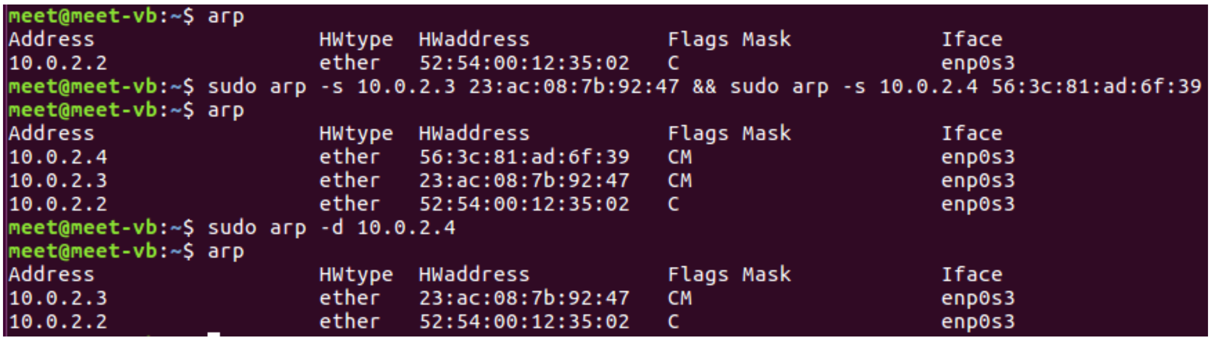
Question 6

Host Time Slot	amazon.in (7 hops common to all three routes)	amazon.com (16 hops common to all three routes)	google.co.in (8 hops common to all three routes)	sonyliv.com (9 hops common to all three routes)	baidu.com (8 hops common to all three routes)
2 PM	25	42	13	27	28
6 PM	32	41	13	27	28
1 AM	32	41	13	27	29

- **Reason for change in route to same host at different time of the day:**
  - The same company host doesn't means that they are on the same network architecture, so route and ping might be different if they are connected to different network elements (proxy, firewall, load balancers).
  - As they are on different subnetworks, they also might be in different datacentres which means different physical location, so route to the same host is different.
- **Cases when traceroute doesn't find complete path :**
  - It didn't happen in case of above 5 hosts but in general it can happen, because many people block ICMP/ping for security reasons, like preventing hackers from getting information about open ports and staving off denial of service attacks.
  - When ping is blocked, the server doesn't respond at all, resulting in "request timed out" messages that prevent traceroute from ever being able to map the path to the final destination.
- **Way find the route to certain hosts which fail to respond with ping experiment :**
  - Ping won't work if the ICMP port is closed somewhere along its way to your destination server. In this case, use TCP protocol to send TCP SYN packets instead of ICMP Echo packets.
  - These tools don't establish a complete TCP connection with the destination. Firewalls generally leave the TCP/IP port open, greatly reducing the risk to run into issues with "traditional" traceroute.
  - When using a TCP/IP-based tool, packets are more likely to make it to their final destination

Question 7

- Command to show full ARP table : **arp**
- ARP command to add entry : ***sudo arp –s ip_addrress HWadress***
- ARP command to delete entry: ***sudo arp –d ip_address***
- ARP command to change entry: just add entry, it will overwrite the existing ip address's corresponding details.

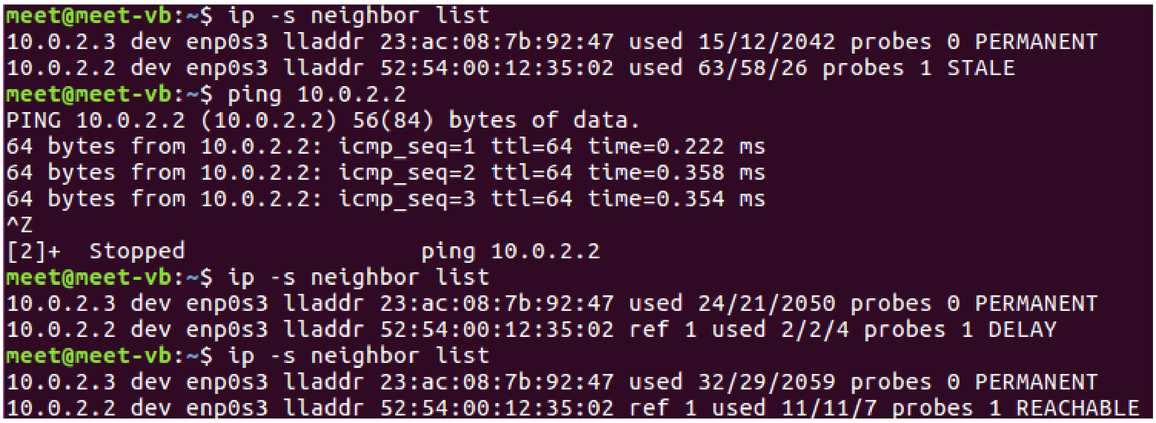


```
meet@meet-vb:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.2                 ether   52:54:00:12:35:02   C                     enp0s3
meet@meet-vb:~$ sudo arp -s 10.0.2.3 23:ac:08:7b:92:47 && sudo arp -s 10.0.2.4 56:3c:81:ad:6f:39
meet@meet-vb:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.4                 ether   56:3c:81:ad:6f:39   CM                    enp0s3
10.0.2.3                 ether   23:ac:08:7b:92:47   CM                    enp0s3
10.0.2.2                 ether   52:54:00:12:35:02   C                     enp0s3
meet@meet-vb:~$ sudo arp -d 10.0.2.4
meet@meet-vb:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.3                 ether   23:ac:08:7b:92:47   CM                    enp0s3
10.0.2.2                 ether   52:54:00:12:35:02   C                     enp0s3
```

- **Explanation of output of ARP table:**

Address	**IP address**
HWtype	Hardware type, here it is Etherrnet
HWaddress	MAC address
Flags	Each permanent entries are marked with M and published entries have the P flag. Complete entry in the ARP cache will be marked with the C flag.
Iface	Interface to which address mapping is assigned

- **How long do entries stay cached in the ARP table** :
    - It stays for 60 seconds in cache. There are subtle differences between an neighbor cache entry actually falling out of the cache entirely or just being marked as stale/invalid.
    - At some point between base_reachable_time/2 and 3*base_reachable_time/2, the entry will still be in the cache, but it will be marked with a state of STALE.
    - You should be able to view the state with "ip -s neighbor show",



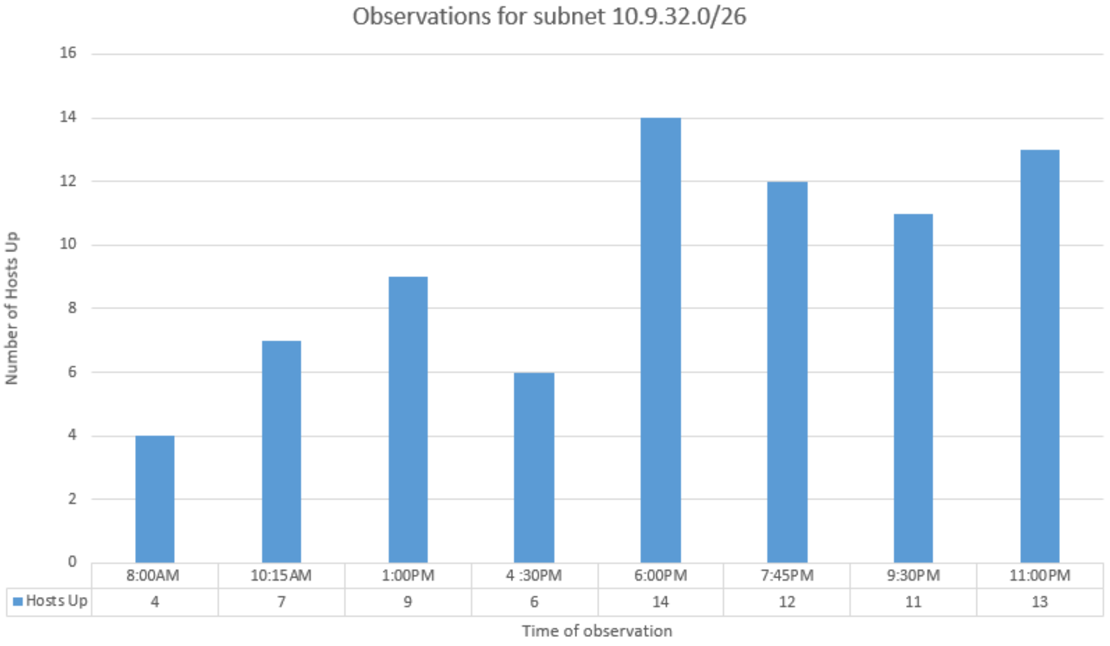
```
meet@meet-vb:~$ ip -s neighbor list
10.0.2.3 dev enp0s3 lladdr 23:ac:08:7b:92:47 used 15/12/2042 probes 0 PERMANENT
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 used 63/58/26 probes 1 STALE
meet@meet-vb:~$ ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56(84) bytes of data.
64 bytes from 10.0.2.2: icmp_seq=1 ttl=64 time=0.222 ms
64 bytes from 10.0.2.2: icmp_seq=2 ttl=64 time=0.358 ms
64 bytes from 10.0.2.2: icmp_seq=3 ttl=64 time=0.354 ms
^Z
[2]+  Stopped                 ping 10.0.2.2
meet@meet-vb:~$ ip -s neighbor list
10.0.2.3 dev enp0s3 lladdr 23:ac:08:7b:92:47 used 24/21/2050 probes 0 PERMANENT
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 ref 1 used 2/2/4 probes 1 DELAY
meet@meet-vb:~$ ip -s neighbor list
10.0.2.3 dev enp0s3 lladdr 23:ac:08:7b:92:47 used 32/29/2059 probes 0 PERMANENT
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 ref 1 used 11/11/7 probes 1 REACHABLE
```

    - When in the STALE state like show above, if I ping 10.0.2.2, it will send the packet to right away. A second or so later it will usually send an ARP request for who has 10.0.2.2 in order to update it's cache back to a REACHABLE state.
    - **So entries in arp table are never timed out, they just changes their state from one to another.**

- **What will happen if two IP addresses map to the same Ethernet address :**
  Case i)
  If there is an host with IP = IP_1 and MAC = MAC_1 and IP=IP_2 and MAC = MAC_1 (possible as some devices have option to change its MAC address) then only ip which is connected to MAC later in point of time is able to respond to ping command.
  Case ii)
  If we have entry manually (using arp –s command) to other IP by arp add command then both of them respond to ping command.

QUESTION 8

- Nmap ("Network Mapper") is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.
- **Command** : nmap –l 10.9.32.1/26
- **LAN subnet chosen for analysis :** 10.9.32.1/26 (Second floor of Kameng hostel, IITG)



- **Trend I observed is increase in number of hosts online during hostel hours.**