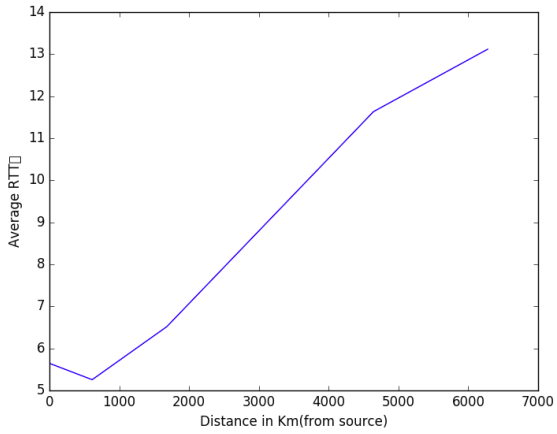
Akash Mahalik 150123004

Q1.
(a).“ping-**c count**” where count is the number of echo requests to send with the ping.
(b). “ping **-i interval**” where interval is Wait interval seconds between two successive ping ECHO_REQUESTs
(c). “ping **-l ECHO_REQUEST**” where ECHO_REQUEST is the number of packets to send to the destination one after another without waiting for a reply.Limit for sending such ECHO_REQUEST packets by normal users is 3.
(d). “ping **-s ECHO_REQUEST**” where ECHO_REQUEST is the packet size(in bytes) we want to send.If the PacketSize is set to 64 bytes, the total packet size will be 64 bytes + 8 bytes(ICMP header data) = 72 bytes.

Q2.

Address	Location	Time(11 AM)	Time(2PM)	Time(10PM)
64.233.184.138	Virginia(612)	5.149	5.256	4.481
108.174.10.10	California(4642.4)	10.993	11.633	12.276
192.229.182.210	New Jersey(0)	4.526	5.645	5.322
157.240.20.35	Hesse(6278)	13.066	13.118	12.712
54.192.35.118	Missouri(1682.5)	7.678	6.519	4.820

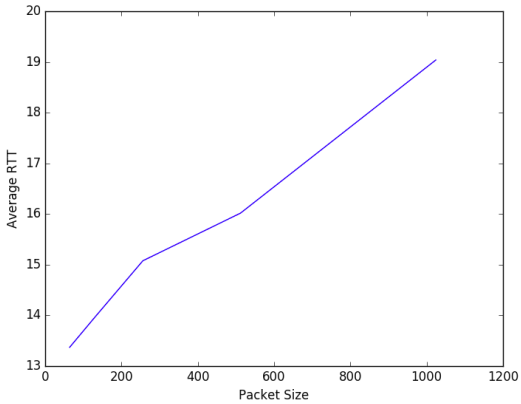
All the above hosts had 0% packet loss.
Possible reasons for packet loss:
(i).High Latency : If high latency is being experienced, the slower network could cause difficulty in delivering data packets in a consistent manner.
(ii).Inconsistent Jitter: Jitter is the spacing of packets being delivered.If the Jitter is inconsistent, there will be a timing issue on the receiving end.The result of this could be the loss of packets.
(iii)Software error: The software that is being used could either be faulty or have excessive bugs.This can result in more packets than normal being lost during transmission.



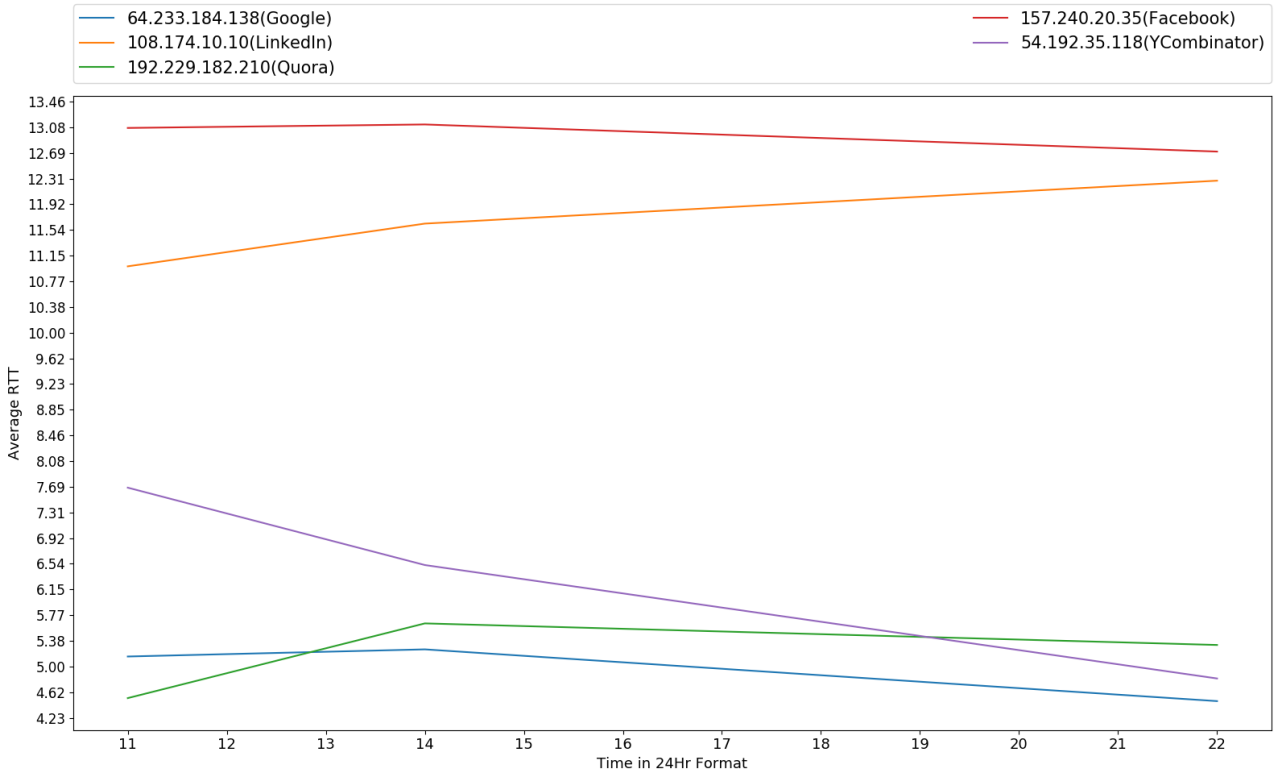
We observe that geographical distance and average RTT are somewhat correlated may be because as the distance increases the hop count increases and thus the average RTT also increases.

Host Used : **157.240.20.35**

Size of Packet	Avg RTT	Packet Loss
64 Bytes	13.370	0%
128 Bytes	13.945	0%
256 Bytes	15.078	0%
512 Bytes	16.018	0%
1024 Bytes	19.039	0%
2048 Bytes	N/A	100%



Increase in packet size increases the average RTT occurs because every network device on the path has to receive complete packet before it can transmit that packet forward.Larger packet size results in increase in receive time.This latency occurs at all points thus increasing the average RTT.



There's not much correlation with the time of the day and average RTT .Ycombinator has a drop in Average RTT during the night and almost every other host has almost the same RTT during the time the ping has been measured.

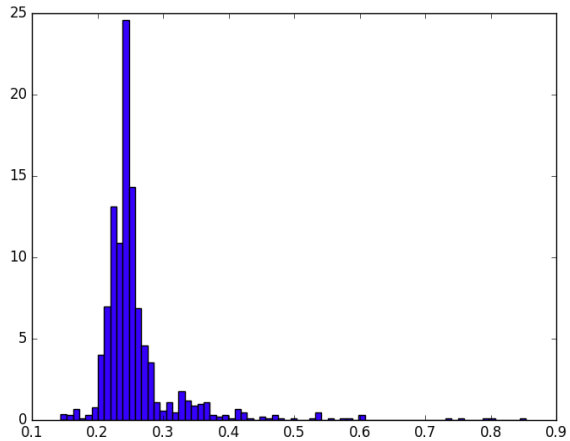
Q3.

(a). Packet loss for :
"**ping -n 202.141.80.14 -c 1000**" is 0%
"**ping -p ff00 202.141.80.14 -c 1000**" is 0%

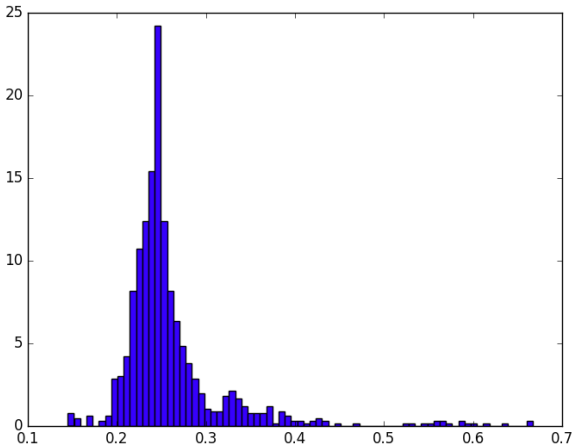
(b)	1st Command	2nd Command
Minimum -	0.144	0.145
Maximum -	207.994	22.681
Mean-	0.776	0.540
Median Latency -	6.915	1.943

(c). Normal distribution of ping latencies for

"**ping -n 202.141.80.14 -c 1000**"



"**ping -p ff00 202.141.80.14 -c 1000**"



(d). The first command has **very less outliers** with respect to the second command visible from the normalized distributions.From the statistical data we can conclude that filling out the packets with **"pad" bytes** (here "ff00") has a slower ping speed than the general ping command.

Q4.

```
akash@akash-Lenovo-Z51-70:~$ ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:32:32:af:3f
          inet addr:172.17.0.1  Bcast:0.0.0.0  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)

enp2s0    Link encap:Ethernet  HWaddr f0:76:1c:bb:6c:79
          inet addr:10.4.3.13  Bcast:10.4.63.255  Mask:255.255.192.0
          inet6 addr: fe80::aa7:ddd8:493:830d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:128350 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65626 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:159146795 (159.1 MB)  TX bytes:7033451 (7.0 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:3641 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3641 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:238390 (238.3 KB)  TX bytes:238390 (238.3 KB)
```

Description of Various Terms:

1.**docker0:** Here ‘docker’ has intercontainer communication enabled which means containers on a host can communicate with each other without any restrictions.’docker0’, a virtual Ethernet bridge that automatically forwards packets between any other network interfaces attached to it.
2.**enp2s0:** Here this basically tells us about the ethernet interface ‘en’ -> ethernet,’p2’ -> bus number (2), ’s0’ -> slot number (0)
3. **lo:** This represents the local server running on our computer denoted by ip address : 127.0.0.1
4.**Link encap:Ethernet** : This denotes that the interface is an ethernet related device
5.**Hwaddr:** This is the MAC address(hardware address) of our Ethernet card which is unique for each ethernet card manufactured .The first half part of this address contains the manufacturer code which is common for all ethernet cards manufactured by the same manufacturer and the rest denotes the device Id which should be unique for each device manufactured at same place.
6.**inet addr**: This indicates the IPv4 address of our machine.
7.**Bcast:** This indicates the broadcast address associated with our machine.A message sent to a broadcast address is typically received by all network-attached hosts, rather than by a specific host.
8.**Mask:** This indicates the network mask/subnet address.Subnet is a portion of a netwrok that shares a common address component.On TCP/IP networks, subnets are defined as all devices whose IP addresses have the same prefix. Dividing a network into subnets is useful for both security and performance reasons. IP networks are divided using a subnet mask.
9.**inet6 addr:** This indicates Ipv6 address of our machine.
10.**UP:** This flag indicates that the kernel modules related to the Ethernet interface has been loaded.
11.**BROADCAST:** This denotes that the Ethernet device supports broadcasting, which is essential to to obtain IP address via DHCP.
12.**RUNNING:** This indicates that the Ethernet interface is ready to accept the data.
13.**MULTICAST:** This indicates that the Ethernet interface supports multicasting.It allows a source to send packets to multiple machines as long as machines are watching out for that packet.
14.**MTU:** Maximum Transmission Unit is the size of each packet received by the Ethernet card.Setting this to a higher value could hazard packet fragmentation or buffer overflows.
15.**Metric:** The value of this property decides the priority of the device.This paramter has significance only while routing packets.The option can take value of 1,2,3,4,5....
16.**RX packets,TX Packets:** This represents the total number of packets received and transmitted respectively.If we find errors or dropped value greater than zero, then it could mean that the Ethernet device is failing or there is some congestion in the network.
17.**collisions:** Ideally this value should be 0.If it has a value greater than 0, it could mean that the packets ae colliding while traversing the network – a sign of congestion.
18.**txqueuelen:** This denotes the length of the transmit queue of the device.Usually set to smaller values for slower devices with a high latency such as modem links and ISDN.
19.**RX Bytes,TX Bytes:** Indicates the total amount of data that has passed either way through the Ethernet interface.As lon as there is some network traffic being generated via the Ethernet device,both RX and TX bytes will go increasing.

Using ifconfig with different options:-

1. “**ifconfig -a**” : display all interfaces which are currently available
2. “**ifconfig -s**” : display a short list in tabular form
3. “**ifconfig enp2s0**” : Display configuration of the device enp2s0 as described above.
4. “**sudo ifconfig enp2s0 up**” : If this network device is inactive, it will activate.(doesn’t work without sudo)
5. “**sudo ifconfig enp2s0 down**”: If this network device is active, it will deactivate.(doesn’t work without sudo)

Output of route command :

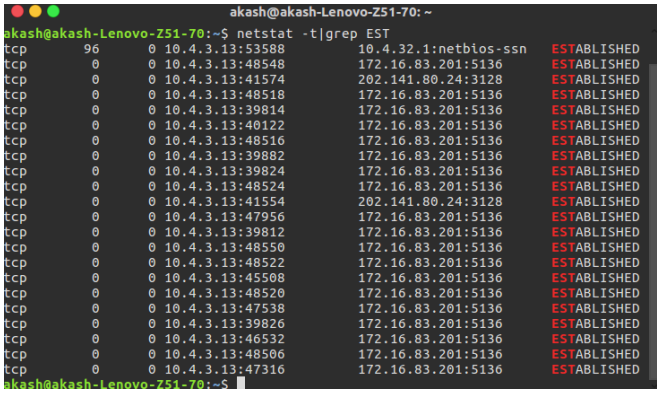
```
akash@akash-Lenovo-Z51-70:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.4.0.254      0.0.0.0         UG    100    0        0 enp2s0
10.4.0.0        *               255.255.192.0   U     100    0        0 enp2s0
link-local      *               255.255.0.0     U     1000   0        0 docker0
172.17.0.0      *               255.255.0.0     U     0      0        0 docker0
```

Route command is used to manipulate the kernel’s IP routing tables.Primary use is to set up static routes to specific routes or specific hosts or networks via an interface after it has been configured with the ifconfig program
Using Route with different options :-
1. “**route -C”:** Operate on kernel’s routing cache
2. “**route -n”:** Show numerical addresses instead of trting to determine symbolic host names.Useful to determine why the route of your nameserver has vanished.
3. “**route add default gw 192.168.1.254 eth0**”: This will set the default gateway to 192.168.1.254.

Q5.

Netstat(network statistics) is a command line utility that tells us about all the tcp/udp/unix socket connections on our system.Provides list of all connections that are currently established or are in waiting state.This tool is extremely usefule in identifying the port numbers on which an application is working and we can also make sure if an application is working or not on the port it is supposed to work.Also displays various other network related info such as routing tables,interface statistics, masquerade connections,multicast memberships etc.

netstat -t|grep EST” is used to show all TCP connections.



Explanations:
1. The “**Proto**” column tells us if the socket listed is tcp,udp or raw
2. The “**Recv-Q**” column tells us the count of bytes note copied by the user program connected to this socket.
3. The “**Send-Q**” column tells us the count of bytes not acknowledged by the remote host.
4. The “**Local Address**” column tells us the Address and port number of the local end of the socket.
5. The “**Foreign Address**” column tells us the Addres and port number of the remote end of the socket.
6. The “**State**” column tells us the state of the socket.Ex – ESTABLISHED,SYN_SENT,FIN_WAIT1 etc.

“**netstat -r**” command displays the routing information maintained by the kernel to compute the next hop for a packet.

```
akash@akash-Lenovo-Z51-70:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         172.16.68.254   0.0.0.0         UG        0 0          0 enp2s0
link-local      *               255.255.0.0     U         0 0          0 docker0
172.16.68.0     *               255.255.252.0   U         0 0          0 enp2s0
172.17.0.0      *               255.255.0.0     U         0 0          0 docker0
```

“**Destination**”: destination network or destination host.

"**Gateway**" : gateway address.
"**Genmask**": netmask for the destination net; 255.255.255.255 for a host destination and 0.0.0.0 for the default route.
"**Flag G**" indicates path to route packets via a gateway.
"**Flag U**" means that the route is up.
"**MSS**" is the default maximum segment size for TCP connections over this route.
"**Window**" referes to the default windows size for TCP connections over this route.
"**irtt**" Initial RTT.The kernel uses this to guess about the best TCP protocol parameters without waiting on answers.

"**netstat -i**" can be used to display network interface status.

```
akash@akash-Lenovo-Z51-70:~$ 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
enp2s0    1500 0    153803      0   1128 0         41051      0      0      0 BMU
lo       65536 0     16573      0      0 0         16573      0      0      0 LRU
```

This shows that the number of interfaces in my machine is **3** namely **docker0,enp2s0** and **lo**.
Loopback interface performs the following functions:
1.**Device Identification :** The loopback interface is used to identify the device.While any interface address can be used to determine if the device is online, the loopback address is the prefered method.
2. **Routing information:** The loopback address is used by protocols such as OSPF to determine protocol specific properties for the device or network.
3. **Packet filtering :** Stateless firewall filters can be applied to the loopback address to filter packets originating from, or destined for, the Routing Engine.

Q6.

Address	Hop Count(11AM)	Hop Count(2PM)	Hop Count(8PM)
64.233.184.138	21	21	21
108.174.10.10	5	5	5
192.229.182.210	9	9	9
157.240.20.35	12	12	12
54.192.35.118	19	19	19

(i).Host 2,3,4 have the same initial hop address – **67.219.148.9**
(ii).The traceroute path does not change sometimesw when the traceroute experiment is repeated at different times of the day.The reason for this could be congestion in the network during different times of the day.If a particular **router is busy** at some time then the routing would be done to some other router thus changing the traceroute.It can also happen if a route fails or there is overloading.
(iii). '**Timed out**' hops occur due to overloaded router or improper connection between 2 devices.
(iv). Ping is waiting for ICMP echo reply from the final hop and it is possible that the host is not configured to send reply and just keeps on waiting infinitely.Traceroute depends on a type of "time exceeded" message as the packet passes through a router the TTL is decremented until, when the TTL reaches zero, the packet is destroyed.So it is possible to find a route to a host which doesn't respond to ping.

Q7.

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.

```
akash@akash-Lenovo-Z51-70:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
kulsi.iitg.ernet.in              (incomplete)                              enp2s0
10.4.2.6                 ether   f8:ca:b8:1b:6c:76   C                     enp2s0
10.4.0.254               ether   4c:4e:35:97:1e:ef   C                     enp2s0
10.4.2.15                ether   b0:5a:da:59:b3:c1   C                     enp2s0
10.4.32.1                ether   2c:56:dc:1a:d1:bf   C                     enp2s0
```

(i). Address - IP address of the device is denoted by the first column.
(ii). Hwtype – Type of hardware is denoted by the second columnd like ether **.**
(iii). HWaddress – MAC address of that device is denoted in the third column **4c:4e:35:97:1e:ef**
(iv).Flags – indicate if the macadress has been learned,manually set,published or is incomplete.

(v).Iface - refers to the network interface here its enp2s0.

To **add/delete** entry in arp table we need sudo access.
Add: sudo arp -i enp2s0 -s <IP address> <Hwaddress> (-i enp2s0 because my ethernet interface is enp2s0)

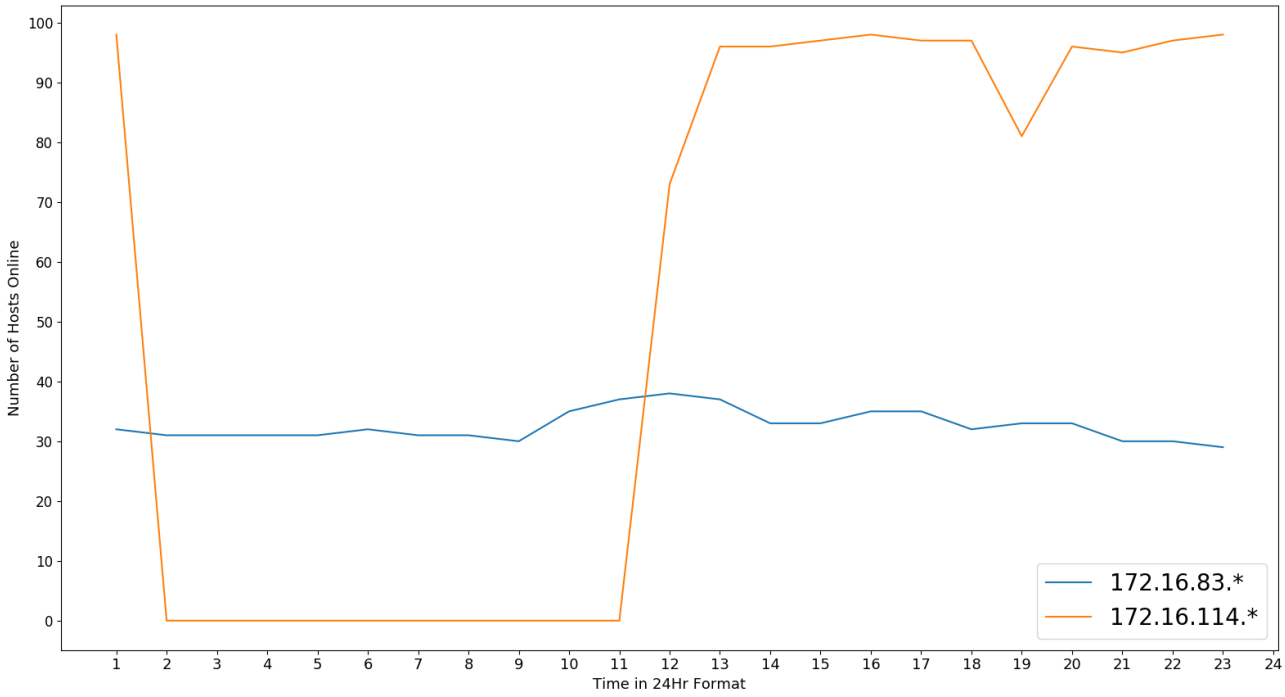
```
akash@akash-Lenovo-Z51-70:~$ sudo arp -s 10.1.43.138 00:00:22:33:33:33 -i enp2s0
akash@akash-Lenovo-Z51-70:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.4.3.16                ether   e0:d5:5e:21:52:7f   C                     enp2s0
10.4.3.17                ether   e0:d5:5e:21:52:7f   C                     enp2s0
10.4.3.20                ether   70:4f:57:67:f4:2c   C                     enp2s0
10.1.43.138              ether   00:00:22:33:33:33   CM                    enp2s0
10.4.3.21                ether   3c:a8:2a:a9:b6:55   C                     enp2s0
```

Delete: sudo arp -d <IP address>
We can get the **default arp cache timeout** by :
"**cat /proc/sys/net/ipv4/neigh/enp2s0/gc_stale_time**" (enp2s0 because my ethernet interface is enp2s0).
This gives timeout time as 60 secs.

We can estimate this time using a trial and error method. First we assume a timeout time. Then we see if the cache entry times out after that time. If no we decrease the time and check again. If yes then increase the assumed timeout time and wait for this increased time. This way we estimate the timeout time.

Yes, it is quite possible for a machine with **one MAC address to have two IP addresses on the same** (LAN) segment. The fundamental reason for this is that the MAC address is a hardware or burned-in address (data link layer) while IP addresses are from the network layer where more than one IP address can be assigned from your LAN segment to a single physical interface.

Q8.



The two LAN subnet address are specified in the plot and we can see that the one with subnet address 172.16.83.* has **almost equal number of hosts online most of the day** and the one with 172.16.114.* is switched off during LAN BAN and is mostly accessed after 12pm in the afternoon.