# CS-349 Networks Lab
# Assignment – 1
# By- ABHINAV GUPTA (150123001)

**Ans-1:**

(a) ‘**-c**’ option is required to specify the number of echo requests to send with ping.

(b) ‘**-i**’ option is required to set time interval (in seconds), rather than the default one second interval, between two successive ping ECHO_REQUESTs.

(c) ‘**ping -f destination**’ command is used to send ECHO_REQUEST packets to the destination one after another without waiting for a reply. Such ECHO_REQUEST packets can be sent by normal users (not super user) only when ‘**minimal interval between each request is 200ms**’ .

(d) ‘**ping -s packet_size destination**’ is the command to set the ECHO_REQUEST packet size (in bytes). If the Packet Size is set to 64 bytes, the total packet size will be **92.**

**Ans-2:** The following reading were taken on 22 January,2018. Website **spfld.com/ping.html (**USA server**)**

| **Hosts** | **09:00 hrs** | **19:00 hrs** | **02:00 hrs** |
|---|---|---|---|
| www.google.com | 3.927/4.110/4.872 | 4.451/4.632/5.352 | 5.070/5.133/5.221 |
| www.flipkart.com | 239.588/239.723/240.282 | 238.982/239.077/239.935 | 238.912/239.054/239.810 |
| www.london.gov.uk | 12.451/13.042/16.250 | 12.783/12.935/13.272 | 11.625/12.372/13.027 |
| www.jyjjapan.jp | 192.153/196.805/202.101 | 199.757/215.400/237.200 | 191.576/198.104/201.170 |
| www.remgro.com | 101.778/102.511/102.708 | 103.038/103.179/104.345 | 101.846/102.461/102.693 |

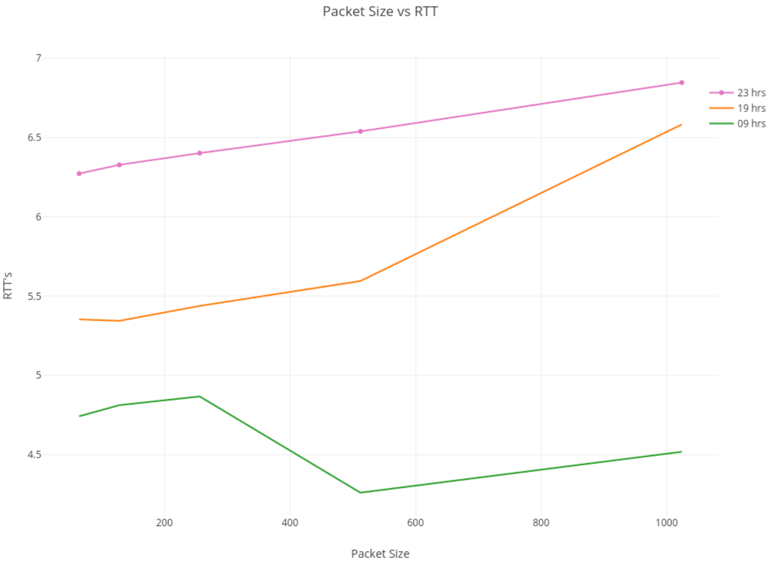
Each of the above reading represents the Minimum RTT/Average RTT/ Maximum RTT (all in ms).

There was no case of packet loss while doing the above experiment with packet size of 64B. But there can be packet losses and the reason for the same could be:

- There are many hosts which don’t respond to ping requests (Ex:- www.southafrica.net)
- Generally hosts reply for packet size of less than 1440B, above that they don’t respond
- There may be packet loss due to high traffic or congestion in network

I specifically choose the hosts located at different parts of world like USA, India, London, Japan and South Africa. I got the measured RTTs from **spfld.com** whose server is situated in **USA** so, it gives minimum RTT for USA and maximum for India/Japan. From the above data, we can see that there is a weakly positive correlation between measured RTTs and host’s geographical distance from the server because by increase in distance time of travel of the packets increase and hence the latency time. The reason for the weak correlation is that latency time also depends on various other factors like ‘**bandwidth of network**’, ‘**contention ratio**’, ‘**load at server**’ and ‘**traffic in the network**’.

I chose **www.google.com** to repeat the experiment with different packet sizes from 64 bytes – 2048 bytes. I observed a change of nearly 10% in change of measured RTTs in a particular time period(diff. sizes) and we also observed that measured RTTs were different at different time of day. The probable reason for the first one could be due to larger packet size, it takes more time for transmission but the major part of RTTs is composed of connection establishment time and transmission time, it depends very less on the packet size though we can see a slight positive correlation. Observe that RTTs change for different time of day it may be possible due to the network’s high usage and congestion, as it totally explains the observed data. I got a 100% packet loss for packet size > 1440 bytes. For large packet size, the buffer at the immediate router gets filled up quickly hence losing packets.

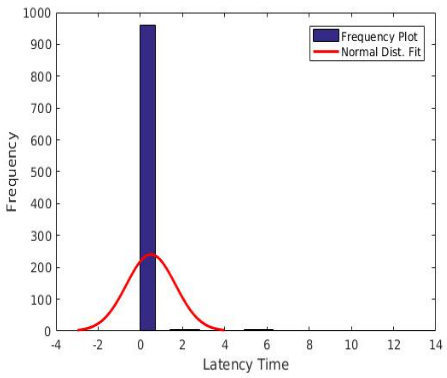


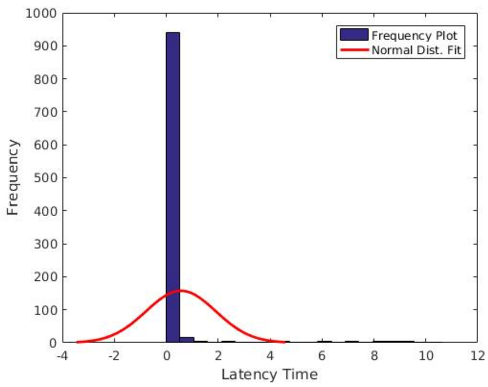
**Ans-3:** Ping host: **202.141.80.14**

**(a)** Packet loss % for both commands is 0.

**(b)** Ping Latency 1st (in ms): Minimum= 0.1770 , Maximum= 13.9000, Mean= 0.5008, Median= 0.3050
Ping Latency 2nd(in ms): Minimum= 0.1690, Maximum= 10.5000, Mean= 0.5653, Median= 0.3080

**(c)**





**(d)** The two experiments are different in two aspects that are ‘packets sent in exp. 2 are padded by data **ff00** in addition to header whereas in exp. 1 are empty with only header’ and ‘ping output in exp. 1 doesn’t show full hostname while the exp. 2 do the necessary lookup’. Due the data put up in packets in exp.2, avg. latency time increases as latency time is weakly positive correlated to packet size.

**Ans-4:** Output of ‘**ifconfig’ -** displays the status of the currently active interfaces :

```
abhinav@abhinav:~$ ifconfig
enp1s0      Link encap:Ethernet  HWaddr 50:7b:9d:79:14:4b
             inet addr:10.10.3.22  Bcast:10.10.63.255  Mask:255.255.192.0
             inet6 addr: fe80::a329:d3a0:8564:8c72/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:157997 errors:0 dropped:121 overruns:0 frame:0
             TX packets:42124 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:20087860 (20.0 MB)  TX bytes:3859140 (3.8 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:3272 errors:0 dropped:0 overruns:0 frame:0
             TX packets:3272 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1
             RX bytes:238756 (238.7 KB)  TX bytes:238756 (238.7 KB)
```

```
wlp2s0      Link encap:Ethernet  HWaddr 48:e2:44:57:cc:d5
            inet addr:10.42.0.1  Bcast:10.42.0.255  Mask:255.255.255.0
            inet6 addr: fe80::4ae2:44ff:fe57:ccd5/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:527 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:95138 (95.1 KB)
```

- **Link encap:Ethernet** - This denotes that the interface is an Ethernet related device.
- **Link encap:Local Loopback** - The loopback is a special, network interface that the computer uses to communicate with itself.
- **inet addr** - indicates the machine IP address.
- **inet6 addr** - first part is the address and **'/64'** indicates subnet size.
- **HWaddr 50:7b:9d:79:14:4b** - This is the hardware address or MAC address which is unique to each Ethernet card which is manufactured.
- **Scope** - Indicates type of addressing in inet5.
- **Bcast** - denotes the broadcast address. A broadcast message is sent to all the devices connected to a particular network.
- **Mask** – It is network mask.
- **UP** - This flag indicates that the kernel modules related to the Ethernet interface has been loaded.
- **BROADCAST** - Denotes that the device supports broadcasting - a necessary characteristic to obtain IP address via DHCP.
- **RUNNING** - The interface is ready to accept data.
- **MULTICAST** - This indicates that the Ethernet interface supports multicasting. 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 transferred by the Ethernet card. The value of MTU for all Ethernet devices by default is set to 1500.
- **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.
- **RX Packets, TX Packets** - The total number of packets received and transmitted respectively.
    - **packets:** The number of packets received/transferred via the interface.
    - **errors:** The number of damaged packets received/transferred.
    - **dropped:** The number of dropped packets due to reception/transfer errors.
    - **overruns:** The number of received packets that experienced data overruns.
    - **frame:** The number received packets that experienced frame errors.
    - **carrier:** The number received packets that experienced loss of carriers.
    - **collisions:** 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.

**Describing ifconfig command with various options :**

- Running ifconfig with no options will display the configuration of all active interfaces.
- **-a :** display all interfaces available (both active and inactive)
- **-s :** display a short list
- **-v :** more verbose for some error conditions
- **interface :** The name of the interface. This is usually a driver name followed by a unit number, for example eth0.
- To change the settings of the existing interfaces, following after type options can be used. Only the most used types are specified-
    - **up :** This flag causes the interface to be activated.
    - **down :** This flag causes the driver for this interface to be shut down.
    - **metric N :** This parameter sets the interface metric.
    - **mtu N :** This parameter sets the Maximum Transfer Unit (MTU) of an interface.
    - **dstaddr addr :** Set the remote IP address for a point-to-point link
    - **netmask addr :** Set the IP network mask for this interface.
    - **address :** The IP address to be assigned to this interface.

**Route command** is used to view and also modify the IP routing table. Output of route ·

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.10.0.254     0.0.0.0         UG    100    0        0 enp1s0
10.10.0.0       0.0.0.0         255.255.192.0   U     100    0        0 enp1s0
10.42.0.0       0.0.0.0         255.255.255.0   U     600    0        0 wlp2s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp1s0
```

- **Destination** : The destination network or destination host.
- **Gateway** : The gateway address or '*' if none set.
- **Genmask** : The netmask for the destination net.
- **Flag**s : U (route is up) G (use gateway)
- **Metric** : The distance to the target (usually counted in hops). It is not used by recent kernels.
- **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.

**Some of the relevant options of route** used are:

- **-n :** show numerical addresses instead of trying to determine the symbolic host names.
- **-v :** verbose operation
- **del :** delete a route
- **add :** add new route

**Ans-5: Netstat** is a command line utility that can be used to list out all the network (socket) connections on a system.
**'netstat -t'** command should be used to show all the TCP connections established.

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address         Foreign Address       State
tcp       0      0  10.10.3.22:53432      202.141.80.24:3128    ESTABLISHED
tcp       1      0  10.10.3.22:53562      202.141.80.24:3128    CLOSE_WAIT
tcp       0      0  10.10.3.22:53460      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:53566      202.141.80.24:3128    TIME_WAIT
tcp       0      0  10.10.3.22:52950      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:52952      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:53148      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:53370      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:52848      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:49158      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:53564      202.141.80.24:3128    TIME_WAIT
tcp       0      0  10.10.3.22:53446      202.141.80.24:3128    ESTABLISHED
tcp       0      0  10.10.3.22:53502      202.141.80.24:3128    ESTABLISHED
tcp      32      0  10.10.3.22:53530      202.141.80.24:3128    CLOSE_WAIT
tcp       1      0  10.10.3.22:53542      202.141.80.24:3128    CLOSE_WAIT
tcp       0      0  10.10.3.22:53448      202.141.80.24:3128    ESTABLISHED
```

**Explanations:**

- **Proto :** The protocol used for the given connection
- The "**Recv-Q**" and "**Send-Q**" columns tell us how much data is in the queue for that socket, waiting to be read (Recv-Q) or sent (Send-Q).
- The "**Local Address**" and "**Foreign Address**" columns tell to which hosts and ports the listed sockets are connected. The local address is the address of the machine on which netstat is running, and the foreign end is the other computer.
- The "**State**" column tells in which state the listed sockets are.

**'netstat -r'** command is used to list the kernel routing table.

```
Kernel IP routing table
Destination   Gateway       Genmask        Flags   MSS   Window  irtt  Iface
default       10.10.0.254   0.0.0.0        UG      0         0      0  enp1s0
10.10.0.0     *             255.255.192.0  U       0         0      0  enp1s0
10.42.0.0     *             255.255.255.0  U       0         0      0  wlp2s0
link-local    *             255.255.0.0    U       0         0      0  enp1s0
```

- The "**Destination**" column indicates the pattern that the destination of a packet is compared to.
- The "**Gateway**" column tells the computer where to send a packet that matches the destination of the same line. An asterisk ( * ) here means "send locally", because the destination is supposed to be on the same network.
- The "**Genmask**" column is the subnet mask that is used for the connection
- The "**Flags**" column shows which flags apply to the current line. "**U**" means Up (active line),"**G**" means line uses a Gateway.
- The "**MSS**" column lists the value of the Maximum Segment Size for this line. Nowadays, most
- computers have no problems with the most commonly used maximum packet sizes, so this column usually has the value of 0, meaning "no changes".

- The “**Window**” column is like the MSS column in that it gives the option of altering a TCP parameter. In this case that parameter is the default window size, which indicates how many TCP packets can be sent before at least one of them has to be ACKnowledged. Like the MSS, this field is usually 0, meaning “no changes”.
- The “**irtt**” column stands for Initial Round Trip Time and may be used by the kernel to guess about the best TCP parameters without waiting for slow replies. In practice, it’s not used much, so 0 here.
- The “**Iface**” column tells which network interface should be used for sending packets that match the destination. If your computer is connected to multiple subnets on multiple network cards.

**‘netstat -i’** command is used to display network interface status.

```
Kernel Interface table
Iface      MTU  Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp1s0     1500   0     9516513     0     2913 0     4532947     0      0      0 BMRU
lo         65536  0       23185     0        0  0      23185     0      0      0 LRU
wlp2s0     1500   0       55167     0       16  0      70699     0      0      0 BMRU
```

**Loopback Interface:** The loopback device is a virtual network interface that computer uses to communicate with itself. It is used for diagnostics and troubleshooting, and to connect to servers running on local machine. If we ping the virtual address of the machine then it loopbacks till keyboard interrupt is given.

```
abhinav@abhinav:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.113 ms
^C
--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.113/0.113/0.113/0.000 ms
```

**Ans-6:** **i)** Online Tool used: **http://network-tools.com**

| Hosts | Hop Counts | | | No. of Common Hops | Common Hops |
|---|---|---|---|---|---|
| | 12:00 | 17:00 | 22:00 | | |
| **www.google.com** | **5** | **5** | **5** | **2** | 207.86.208.17, 207.86.208.62 |
| **www.flipkart.com** | **15** | **15** | **15** | **12** | 207.86.208.17, 207.88.13.122, 207.88.14.189, 206.111.5.34, 66.110.56.6, 66.110.57.21, 66.110.59.2, 180.87.15.25, 180.87.15.6, 172.31.219.42, 115.110.250.34, 163.53.78.128 |
| **www.london.gov.uk** | **4** | **4** | **4** | **3** | 67.219.148.9, 184.105.25.73, 206.53.174.11 |
| **www.jyjjapan.jp** | **17** | **17** | **17** | **10** | 67.219.148.9, 184.105.25.73, 184.105.81.173, 184.105.81.177, 72.52.92.121, 206.223.123.112, 1.208.14.26, 1.208.13.170, 2`11.60.0.194, 115.68.77.132 |
| **www.remgro.com** | **11** | **11** | **11** | **11** | 67.219.148.9, 184.105.25.73, 184.105.81.169, 184.105.213.70, 184.105.223.166, 72.52.92.165, 72.52.92.214, 193.239.117.110, 213.239.229.74, 213.239.229.10, 148.251.118.180 |
| | | | | | |

**ii)** Route to same host changes at different time of the day due to differing traffic pattern. Routing may change due to considerations of different servers along the way, such as server load and availability. If the routing is not dynamic and let’s say a server is down, it will lead to undelivered requests. Some servers are extremely busy during day time due to heavy computations or too many request while during night, they may be ideal so in such case we may route the requests to these servers at night and change during day time.

**iii)** There are cases when traceroute doesn’t provide the complete path, at times the request is acknowledged but the host name is not provided and at times timeout is returned. The primary reason could be existence of firewall which is configured to block these packets or a secondary (very unlikely though) reason could be that router is dropping packets going through it. This is usually caused by three reasons either the router is overloaded, the router having a software or physical failure or the router is configured to do so (null route/black holes).

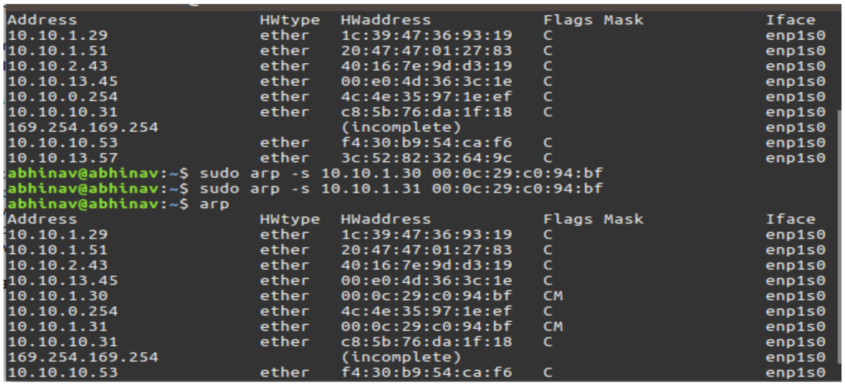
**iv)** Yes, it is possible to find the route to certain hosts which fail to respond with ping experiment (e.g. www.southafrica.net). Ping works on straight ICMP (Internet control Message Protocol) Traceroute works very different from ping even though it uses ICMP. Traceroute works by targeting the final hop, but limiting the TTL (Time To Live) 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 the host. There are some hops, which don't give an ICMP echo reply so, we don't get a reply for ping request but we are able to trace the route

**Ans-7:** ARP table can be seen using the '**arp**' or '**arp -v**' command. The output of arp command:

```
Address          HWtype   HWaddress          Flags Mask     Iface
192.168.43.1      ether   38:a4:ed:c2:d1:a1    C            enxd46e0e7f8a65
202.141.81.2              (incomplete)                      enp1s0
```

- ➢ Address column of the table shows the IP addr of the machine connected to a network
- ➢ Hwtype specifies the type of hardware.
- ➢ Hwaddress column shows the mac address corresponding the particular entry in the table.
- ➢ ARP cache entries may be marked with the following flags: C(complete), M(permanent).
- ➢ Interface shows the network interface type for the corresponding entry.

By the command '**sudo arp -s <ip> <MAC Address>**', we added the following entry.



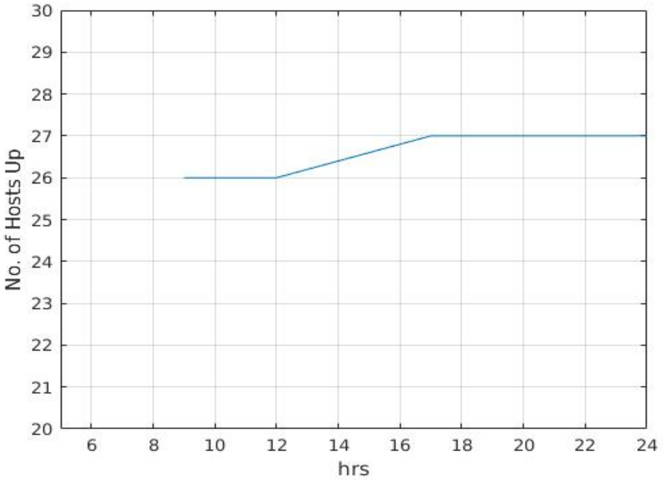
```
Address                  HWtype  HWaddress           Flags Mask            Iface
10.10.1.29               ether   1c:39:47:36:93:19   C                     enp1s0
10.10.1.51               ether   20:47:47:01:27:83   C                     enp1s0
10.10.2.43               ether   40:16:7e:9d:d3:19   C                     enp1s0
10.10.13.45              ether   00:e0:4d:36:3c:1e   C                     enp1s0
10.10.0.254              ether   4c:4e:35:97:1e:ef   C                     enp1s0
10.10.10.31              ether   c8:5b:76:da:1f:18   C                     enp1s0
169.254.169.254                  (incomplete)                              enp1s0
10.10.10.53              ether   f4:30:b9:54:ca:f6   C                     enp1s0
10.10.13.57              ether   3c:52:82:32:64:9c   C                     enp1s0
abhinav@abhinav:~$ sudo arp -s 10.10.1.30 00:0c:29:c0:94:bf
abhinav@abhinav:~$ sudo arp -s 10.10.1.31 00:0c:29:c0:94:bf
abhinav@abhinav:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.10.1.29               ether   1c:39:47:36:93:19   C                     enp1s0
10.10.1.51               ether   20:47:47:01:27:83   C                     enp1s0
10.10.2.43               ether   40:16:7e:9d:d3:19   C                     enp1s0
10.10.13.45              ether   00:e0:4d:36:3c:1e   C                     enp1s0
10.10.1.30               ether   00:0c:29:c0:94:bf   CM                    enp1s0
10.10.0.254              ether   4c:4e:35:97:1e:ef   C                     enp1s0
10.10.1.31               ether   00:0c:29:c0:94:bf   CM                    enp1s0
10.10.10.31              ether   c8:5b:76:da:1f:18   C                     enp1s0
169.254.169.254                  (incomplete)                              enp1s0
10.10.10.53              ether   f4:30:b9:54:ca:f6   C                     enp1s0
```

By the command '**sudo arp -d <ip> '** delete an entry. Dynamic entries stay cached for 60 seconds (can be checked in the file "/proc/sys/net/ipv4/neigh/default/gc_stale_time") while static entries stay for about 4 hours in the arp table.

**Trial And Error method to find the timeout:**
An approach **similar to the binary search** can be used to get the desired value. Connect the machine to a new network and then after every 5 mins, check of the entry in table is updated. Let the entry be updated in the i th check. This means that the cache was refreshed between the i-1 and the i check. Now disconnect from this network and wait for (i-1)*5 minutes + 2min + 30 sec. If the entry still exist at this time that means that the cache is cleared after this time and before 5*i minutes. Apply this approach iteratively to get the result.

**Yes, a single ethernet card can have multiple IP's assigned to it, this process is known as IP aliasing**. With this, one node on a network can have multiple connections to a network, each serving a different purpose. In a lot of scenarios, multiple IP addresses are used such as when a single server hosts multiple domain names, when we use two operating system simultaneously one background and another as virtual machine we use different two different ip addresses to communicate among them, even though the MAC address is same( MAC address of our machine). If IP's with same MAC address are on different subnet then there is no problem in packet routing as each router's table contains single ip with a specific MAC. But if IP's with same MAC are on same subnet (be it on same machine or different machines), there will be conflict as router will not know to which ip it has to route as ARP table contains many IP's with same MAC hence less correct transmission.

**Ans-8:** Command- '**nmap –n –sP <Subnet Range>**' Subnet chosen: **172.16.112.0/26**



| | | | | | | |
|---|---|---|---|---|---|---|
| **Hrs.** | : | 9 | 12 | 17 | 22 | 24 |
| **No. of Hosts Up:** | | 26 | 26 | 27 | 27 | 27 |

(out of 64 hosts scanned)