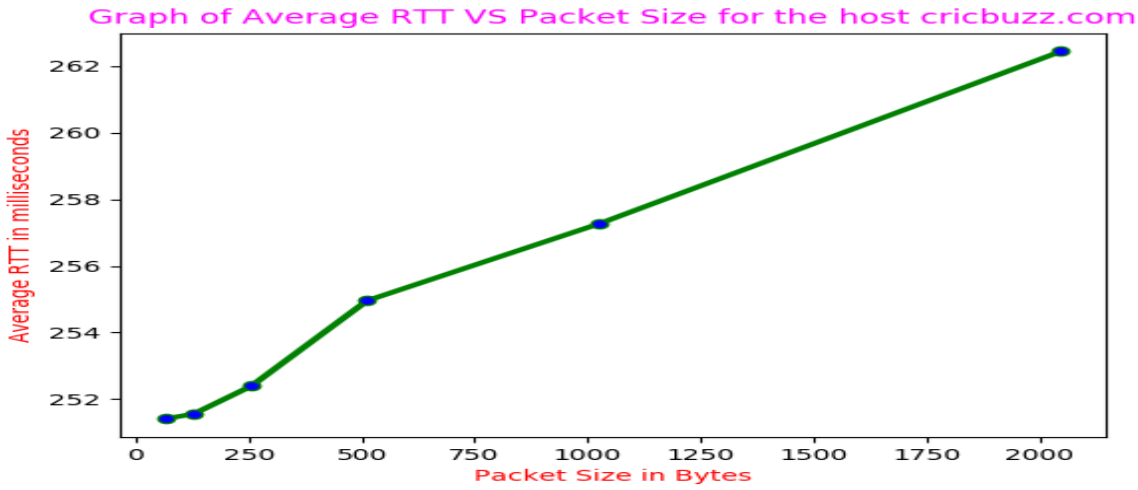
Submitted by **Hareesh Reddi** *Roll Number:* **150101051**

Question 1: a) ping -c count to specify the number of echo requests to send with ping.
b) ping -i interval to set time interval (in seconds), rather than the default one second interval, between two successive ping ECHO_REQUESTs. However, normal user cannot set interval to values less than 0.2 seconds.
c) ping -l preload to send ECHO_REQUEST packets to the destination one after another without waiting for a reply. **3** is the limit for sending such ECHO_REQUEST packets by normal users.
d) ping -s packetsize to set the ECHO_REQUEST packet size (in bytes). If the packetsize is set to 64 bytes, the total packetsize will be **72 bytes** as 8 bytes of ICMP header data is added to the 64 bytes.
Question2: **a)** I experimented at three different hours by pinging each host 20 times at three different hours of the day. Following table shows average RTT for all hosts at three different hours.

Host Name	**Distance(in miles)**	**Average RTT(28 Jan,10 AM) in ms**	**Average RTT(28 Jan,5 PM) in ms**	**Average RTT(28 Jan,9 PM) in ms**	**Overall Average RTT in milli seconds(ms)**	**Percentage Loss**
cricbuzz.com	7985	250.511	251.206	251.079	250.932	0
naukri.com	1901	248.569	249.586	250.145	249.433	0
amazon.com	7887	11.556	12.575	12.420	12.184	0
hackerearth.com	1901	No RTT	No RTT	No RTT	No RTT	100
yahoo.com	7534	35.997	38.907	37.293	37.339	0

b) Yes, there exists a case where packet loss is more than 0% i.e. hackerearth.com shows 100% packet loss. There could be restrictions on the source ip address that can access. **Packet loss** occurs when one or more packets of data travelling across a computer network fail to reach their destination, which can be because the firewall is blocking. Packet loss is mainly caused by **network congestion** or target IP address might not have any network device connected with it.
c) There is a **weak** positive correlation between distance and RTT. Mainly it will depend on how many routers/switches come into way, as transmission will not take much time. At each router there may be a delay, so if the packets have to go through **more distance**, thus more routers, therefore **longer** the RTT. However, Geographical distance b/n two hosts does not directly give the path through which the packet travels. **d)** I picked **cricbuzz.com** and experimented with different packet sizes ranging from 64-bytes to 2048-bytes.

Packet Size(in Bytes)	64	128	256	512	1024	2048
Average RTT(in ms)	251.407	251.556	252.396	254.957	257.258	262.441



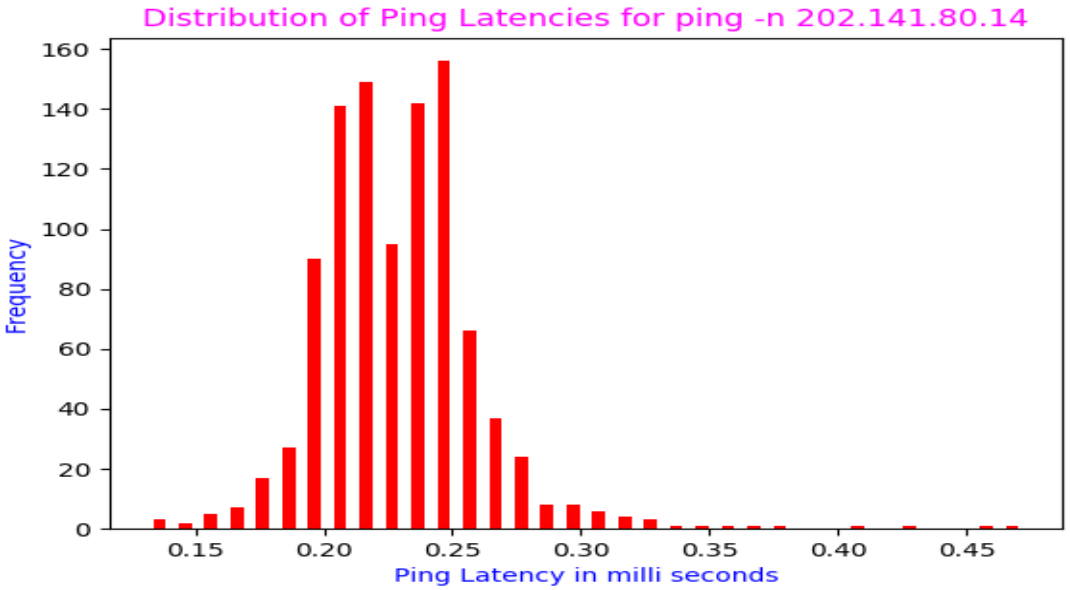
e) With **increase in packet size, RTT increases** as shown in the above figure. **Time influence on RTT** measurements can be understood as different hours correspond to different busy working hours of different continents. Higher RTTs are measured when it is daytime in Asia and Europe. It may be because of network traffic is busier, as more users are online.

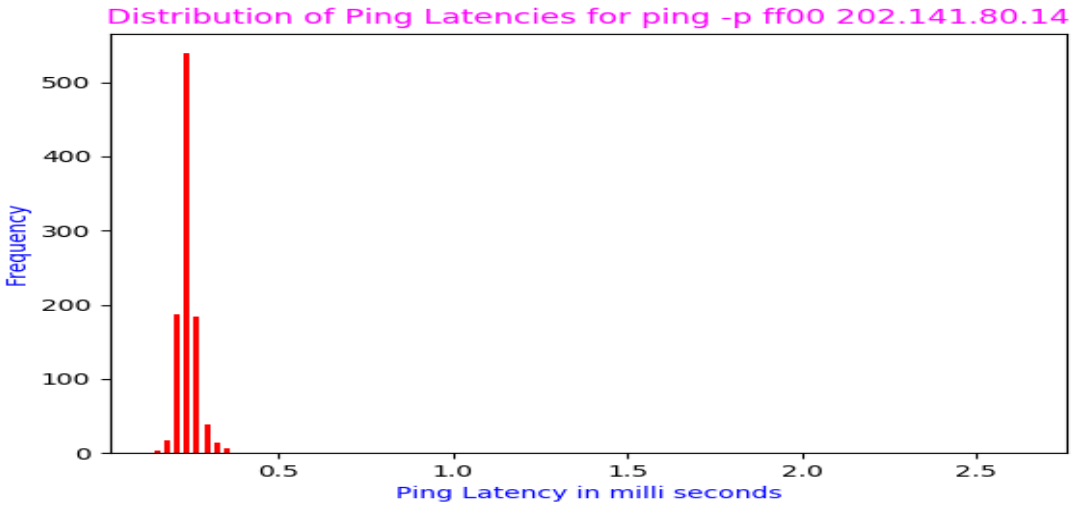
Question3: I chose IP Address **202.141.80.14** itself.

a) Packet loss rate for command ping -n 202.141.80.14 is **0.1%** and packet loss rate for command ping -p ff00 202.141.80.14 is **0.3%.**

b) Minimum, maximum, mean, and median latency of the pings that succeeded for ping –n 202.141.80.14 are **0.131, 0.473, 0.229, 0.228** milliseconds respectively and for ping -p ff00 202.141.80.14 are **0.136, 2.647, 0.246, 0.241** milliseconds respectively.

c & d) Graphs to visualize the **distribution** of **ping latencies**: **Distribution** for **ping –n 202.141.80.14 :** Out of 1000 packets transmitted, **999 were received** and RTT values lied between 0.131 and 0.473





Distribution for ping -p ff00 202.141.80.14 : Out of 1000 packets transmitted, **997 were received** and RTT values lied between 0.136 and 2.647 however 992 out of 997 have RTT values less than 0.4 i.e. only 5 values were more than 0.4 hence this variation is shown in the histogram with more than 99% values lying between 0.136 and 0.4 itself. I obtained mdev(**standard deviation**) values as **0.034 and 0.100 ms**

for ping -n 202.141.80.14 and ping -p ff00 202.141.80.14 respectively. ping -n <IP Address> goes **without dns resolution** but ping -p ff00 <IP Address> will go through it. ping -n <IP Address> carries **default Pattern** whereas ping -p ff00 <IP Address> carries the given(input) pattern 0xff00.

Question4: **a)** Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. If no arguments are given, ifconfig displays the status of the currently active interfaces.

```
hareesh@hareesh-Inspiron-5558:~$ ifconfig
enp7s0    Link encap:Ethernet  HWaddr 20:47:47:bb:f5:17
          inet addr:10.10.2.9  Bcast:10.10.63.255  Mask:255.255.192.0
          inet6 addr: fe80::5763:d3:21e7:bde4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4716156 errors:0 dropped:2172 overruns:0 frame:0
          TX packets:1514898 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3520973242 (3.5 GB)  TX bytes:159298313 (159.2 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:5783 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5783 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:356009 (356.0 KB)  TX bytes:356009 (356.0 KB)
```

Analysis & Explanation of Terminologies in the output of ifconfig:

Here enp7s0, lo are the names of the **active network interfaces** on the system.

1) enp7s0 is the first Ethernet interface. Next Ethernet interfaces would be named enp7s1, enp7s2, etc.

2) lo is the loopback interface. This is a special network interface, which the system uses to communicate with itself.

- **Link encap: Ethernet**: This denotes that the interface is an Ethernet related device.
- **HWaddr 20:47:47:bb:f5:17** : This is the hardware address or MAC Address which is unique to each Ethernet card which is manufactured. First half denotes the manufacturer code that is same for all Ethernet cards manufactured by same manufacturer and the second half denotes the device id, which is unique.
- **inet addr: 10.10.2.9** : Indicates the machine IP address
- **Bcast: 10.10.63.255**: Denotes the broadcast address associated with that particular interface
- **Mask: 255.255.192.0**: This is the network mask that shows how much of the address is routable, which determines whether the computer can connect directly to a device on the LAN or whether it needs to send the packet to a router.
- **inet6 addr**: **fe80::5763:d3:21e7:bde4/64** :IPV6 address of the interface.
- **Scope** :**Link**: This means they are to be used in a directly attached network (link).
- **UP:** This flag indicates that the kernel modules related to the Ethernet interface has been loaded.
- **BROADCAST:** Denotes that the Ethernet 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.
- **MTU: 1500**:(Maximum Transmission Unit)This is the size of each packet received by the Ethernet card.
- **Metric:1**:This option can take a value of whole numbers with the lower the value the more gain through the use of a tool it has. The value of this property decides the priority of the device.
- **RX packets(4716156 errors:0 dropped:2172 overruns:0 frame:0):** Total number of received packets **1)** Errors: Number of damaged packets received.**2)** Dropped: Number of dropped packets due to reception errors.**3)** Overruns: Number of received packets that experienced data overruns.**4)** Frame: Number of received packets that experienced frame errors. Parameter has significance only while routing packets.
- **TX packets(1514898 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000)** :Total number of transmitted packets.

1) Errors: Number of packets that experienced transmission error.**2)** Dropped: Number of dropped transmitted packets due to transmission errors.**3)** Overruns: Number of transmitted packets that experienced data overruns.**4)** Carrier: Number received packets that experienced loss of carriers.
5) collisions: 0**:** The value of this field should ideally be 0. If it has a value greater than zero, it could mean that the packets are colliding while traversing your network - sign of network congestion.
6) txqueuelen: 1000: This denotes the length of the transmit queue of the device. In the above RX and TX, if we 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.
● **RX Bytes, TX Bytes** - These indicate the total amount of data that has passed through the interface.
Some options for ifconfig includes **1) -a** to Display information for all network interfaces, even if they are down. **2) -v** for verbose mode, to display additional information for certain error conditions. **3) Interface** This is usually a driver name followed by a unit no., e.g. enp7s0 for first Ethernet interface. **4) Up**-This flag causes the interface to be activated. **5) Down**-This flag causes the driver for this interface to be shut down. **6) metric N** Sets the interface metric, which is used by the interface to make routing decision. **7) add address** to add an IPv6 address to an interface. **8) del address** to remove an IPv6 address from an interface.

b) Explain the output of route command and its options:

```
hareesh@hareesh-Inspiron-5558:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.10.0.254     0.0.0.0         UG    100    0        0 enp7s0
10.10.0.0       *               255.255.192.0   U     100    0        0 enp7s0
link-local      *               255.255.0.0     U     1000   0        0 enp7s0
```

Route command is used to **view and manipulate the IP routing tables** in both UNIX and Windows based systems. This shows us how the system is currently configured and existing routes table. If a packet comes into the system and has a destination in the range 10.10.0.0 through 255.255.192.0, then it is forwarded to the gateway *, which is **0.0.0.0** — A special address which represents an invalid or non-existing destination. If the destination is not in this IP address range, it is forwarded to the **default gateway** (in this case 10.10.0.254) and that system will determine how to forward the traffic on to the next step towards its destination. Some of Flags are **U**(route is up)/**H**(target is host)/**G**(use gateway).

Some Options for route includes **-F** to operate on the kernel's **FIB** (Forwarding Information Base) routing table. This is the default.**-C** to operate on the kernel's routing cache.**-v** to select verbose operation.**-n** to show numerical addresses instead of trying to determine symbolic host names.**-e** to use **netstat(8)**-format for displaying the routing table.**-ee** will generate a very long line with all parameters from the routing table. **del** to delete a route. **add** to add a new route. **target** the destination network or host. You can provide IP addresses in dotted decimal or host/network names**.-net** if the target is a network or **–host** if the target is a host.

Question 5: a) netstat (***net****work* ***stat****istics*) is a command-line network utility tool that displays network connections for the **T**ransmission **C**ontrol **P**rotocol (both incoming and outgoing), routing tables, and a number of network interface (network interface controller/software-defined network interface) and network protocol statistics. It is used for finding problems in the network, to determine the amount of traffic on the network as a performance measurement and for checking your network configuration and activity.

b)To list out TCP connections which are ESTABLISHED, I used **netstat –t | grep -e ESTABLISHED**

```
hareesh@hareesh-Inspiron-5558:~$ netstat -t | grep -e ESTABLISHED -e State
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 10.10.2.9:54398         202.141.80.24:3128      ESTABLISHED
tcp        0      0 10.10.2.9:52474         202.141.80.24:3128      ESTABLISHED
tcp        0      0 10.10.2.9:54366         202.141.80.24:3128      ESTABLISHED
tcp        0      0 10.10.2.9:54410         202.141.80.24:3128      ESTABLISHED
tcp        0      0 10.10.2.9:54414         202.141.80.24:3128      ESTABLISHED
tcp        0      0 10.10.2.9:54418         202.141.80.24:3128      ESTABLISHED
tcp        0      0 10.10.2.9:54432         202.141.80.24:3128      ESTABLISHED
```

I used **netstat -t | grep -e ESTABLISHED -e State** just to show the first row containing above 6 fields including State. **c) The fields are:**

1) Proto: The name of the protocol (tcp, udp, raw) used by the socket which is tcp in this case.

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

3) Send-Q: The count of bytes yet to be acknowledge by the remote host.

4) Local address: Address and port number of the local end of socket.

5) Foreign Address: Address and port number of the remote end of the socket.

6) State: The state of the socket connected in between the Local Address and Foreign Address. These states represent the three-way handshake communication system that TCP uses.

```
hareesh@hareesh-Inspiron-5558:~$ netstat -r
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 enp7s0
10.10.0.0       *               255.255.192.0   U         0 0          0 enp7s0
link-local      *               255.255.0.0     U         0 0          0 enp7s0
```

d) netstat -r shows the kernel routing information. The **fields** are

1) Destination: The destination network or destination host**.** Default IPV4 address for my system.

2) Gateway: The gateway to which the routing entry points**. 3) Genmask:** The net mask for the destination net; 255.255.192.0 for a host destination and 0.0.0.0 for the default route. 4) **Flags:** This signifies route is up or to gateway or host. **5) MSS:** Default maximum segment size for TCP connection over route. **6) Windows:** Default windows size over this route. **7) irtt:** Initial RTT (Round Trip Time). The kernel uses this to guess about the best TCP protocol parameter without waiting on slow answer. **8)Iface:** Interface to which packets for this route will be sent.

e) netstat –i | wc -l gives <number of interfaces>+2 as the output and **netstat -i** can be used to display network interface status. My machine has **two** interfaces, which I found from the above command.

```
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:511 errors:0 dropped:0 overruns:0 frame:0
          TX packets:511 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:35976 (35.9 KB)  TX bytes:35976 (35.9 KB)
```

f) Loopback Interface: Whenever a network interface is disconnected, for example, when an Ethernet port is unplugged or Wi-Fi is not associated with an access point or turned off i.e. 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 that applications running on our computer can always connect to servers on the same machine. It is used mainly for **diagnostics and troubleshooting**, to connect to servers running on the local machine and it is a **logical, virtual interface**, which a machine uses to communicate to itself.

Question6: a) I used http://network-tools.com tool and five hosts of my choice for traceroute experiment

Host Name	Hop Count(8 AM)	Hop Count(2 PM)	Hop Count(6 PM)
cricbuzz.com	12	12	12
naukri.com	22	22	22
amazon.com	22	22	24
hackerearth.com	28(Trace aborted/Reached firewall)	28(Trace aborted/Reached firewall)	28(Trace aborted/Reached firewall)
yahoo.com	10	10	10

The **common hops** of the five hosts are as follows:

Host Name	Common Hops(IP Address)
cricbuzz.com	67.219.148.9, 184.105.25.73, 206.53.202.9

naukri.com	52.95.30.50, 52.93.128.16, 52.93.8.18, 52.93.8.17, 52.93.8.16, 52.93.8.27
amazon.com	207.86.208.17, 207.88.13.122, 207.88.14.189, 52.95.217.244, 176.32.125.229
hackerearth.com	184.105.25.73, 184.105.81.173, 184.105.81.177, 184.105.65.10, 184.105.64.66, 192.145.251.159
yahoo.com	67.219.148.9, 184.105.25.73, 206.53.202.12, 216.115.97.207, 216.115.104.116

b) Yes, route to same host **can change** at different times of the day. This may be because destination host utilizes **multiple Internet servers** to handle incoming requests, so it shows different IP addresses. There is **fast switching** with which after a packet was sent to the next hop, the routing information about how to get to the destination is stored in a **fast cache**. When the router receives another packet that is directed to the same destination it uses the cache. Therefore, some router's IP are selected from the routing table, which was used earlier and now is found to be inactive then another router IP will be selected.

c) Yes, traceroute for **hackerearth.com** did not find complete paths to the hosts as it shows trace aborted at the end. Traceroute is unable to find complete paths to some host because **Firewall** of that host might be blocking our IP, or we need to increase max hops, as packets **might not reach to destination within fixed max hops**. Other reason may be packet loss between various routers in between the path.

d) Yes, it is possible that tracerouting to certain hosts may be possible even though same host fail to respond to ping experiment. Failing ping is might be because of packet transmission is blocked or packet is discarded, while Traceroute uses an **error message** from a hop to find the route. Traceroute uses a trick to get the information, which is to manipulate the TTL (Time to Live), so the hop responds with an **ICMP error** (ICMP TTL exceeded).

Question7: a) I used **arp -e (Linux style)** command to show the full ARP table for my machine.

b) The Table has six columns:

1) **Address** - This column represents IP address of network connections.

2) **Hwtype** - This represent hardware type of this machine.

3) **Hwaddress** - This represents hardware address of the machine of respective rows network connection.

4) **Flag** - 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. 5) **Mask** - This represent Genmask.

6) **Iface** - This represent network interface of respective rows connection.

c) An entry is added manually in the arp table. **sudo arp -s IP_Address MAC_Address** is used to add a static ARP entry in ARP table with given ip address and with given mac address. **sudo arp -d IP_Address** is used to delete a static entry having given IP Address from ARP table. However, the entry **would not be** removed from the arp table after this command. This changes its hardware address to a sign of **<incomplete>** instead. Deleting things from caches is hard and expensive. Its way more efficient to invalidate an entry and wait if it is replaced before it is finally removed.

```
hareesh@hareesh-Inspiron-5558:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.10.12.5               ether   ec:8e:b5:fc:70:19   C                     enp7s0
10.10.10.53              ether   f4:30:b9:54:ca:f6   C                     enp7s0
10.10.2.43               ether   40:16:7e:9d:d3:19   C                     enp7s0
10.10.13.23              ether   fc:45:96:84:03:7d   C                     enp7s0
10.10.0.254              ether   4c:4e:35:97:1e:ef   C                     enp7s0
10.10.1.51               ether   20:47:47:01:27:83   C                     enp7s0
10.10.10.31              ether   c8:5b:76:da:1f:18   C                     enp7s0
10.10.0.58               ether   98:29:a6:0b:aa:e9   C                     enp7s0
hareesh@hareesh-Inspiron-5558:~$ ping 10.10.3.26 -c 1
PING 10.10.3.26 (10.10.3.26) 56(84) bytes of data.
64 bytes from 10.10.3.26: icmp_seq=1 ttl=64 time=0.383 ms

--- 10.10.3.26 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.383/0.383/0.383/0.000 ms
hareesh@hareesh-Inspiron-5558:~$ ping 10.10.3.23 -c 1
PING 10.10.3.23 (10.10.3.23) 56(84) bytes of data.
64 bytes from 10.10.3.23: icmp_seq=1 ttl=64 time=0.324 ms

--- 10.10.3.23 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.324/0.324/0.324/0.000 ms
hareesh@hareesh-Inspiron-5558:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.10.3.23               ether   f8:ca:b8:28:de:8b   C                     enp7s0
10.10.12.5               ether   ec:8e:b5:fc:70:19   C                     enp7s0
10.10.10.53              ether   f4:30:b9:54:ca:f6   C                     enp7s0
10.10.2.43               ether   40:16:7e:9d:d3:19   C                     enp7s0
10.10.13.23              ether   fc:45:96:84:03:7d   C                     enp7s0
10.10.0.254              ether   4c:4e:35:97:1e:ef   C                     enp7s0
10.10.1.51               ether   20:47:47:01:27:83   C                     enp7s0
10.10.10.31              ether   c8:5b:76:da:1f:18   C                     enp7s0
10.10.0.58               ether   98:29:a6:0b:aa:e9   C                     enp7s0
10.10.3.26               ether   d0:bf:9c:07:c8:76   C                     enp7s0
10.10.10.31              ether   c8:5b:76:da:1f:18   C                     enp7s0
```

d) **Adding** new hosts to the ARP table can be done by just **pinging another IP address** in the same intranet. Then a new pair for (new_host_ip_addr, new_host_hw_addr) will be inserted into APR table. Above picture indicates two new entries after ping to two address **10.10.3.26 and 10.10.3.23**. Therfore, these two new hosts are added into the ARP Table.
e) We can use **cat /proc/sys/net/ipv4/neigh/default/gc_stale_time** command to find how long entries stay cached in arp table. It is **60** seconds in my machine.
f) **Trial and error method:** It may be possible to use bisection method and try by error. For example, we guess the timeout value of 20 minutes and then make the system clock 20 minutes faster and see what happens. Try 10 minutes if the arp cache has been cleared or some value bigger if it has not.
g) Having two devices with the same MAC address on the same LAN is bad because they will confuse switches, they will both attempt to respond to the same traffic - it is a mess, devoutly to be avoided. However, so long as those devices with the same MAC addresses are not connected to the same Ethernet or Wi-Fi network, nothing bad happens because those MAC addresses never leave the network that the NIC is immediately connected to. If the two devices are in different LANs then nothing will happen.
h) **Be specific on how all hosts on the subnet operate:** On a subnet (machines plugged into the same set of hubs and switches), the machines talk to each other with their MAC addresses which uniquely identifies each Ethernet/Wi-Fi card. Machines, *a priori*, do not know MAC addresses; they just know IP addresses. Therefore, when machine A wants to send a packet to machine *B*, it sends a broadcast frame following the ARP protocol; the packet says: "hey, does anybody knows the MAC address of *B*"? If someone responds with the information ("B has MAC address xx:xx:xx:xx:xx:xx") then *A* will be able to send its data to B.

Question8: **a)** I chose **172.16.112.0/25** LAN subnet address and used the command nmap –n –sP 172.16.112.0/25. Following table shows **number of hosts online** at different hours of the day (I took 6 readings)

Time in Hours(**24 Hour Format**)	01	05	08	14	17	21
Number of Hosts Online	29	29	28	30	29	29

b) The graph against time is to see if there are any hourly trends to when computers are switched ON or OFF in my LAN i.e. out of total 128 hosts, below graph shows the number of hosts online.