ASSIGNMENT –1

CS 342: NETWORKS LAB (SEP – NOV 2020)

Name: Udandarao Sai Sandeep
Roll No.: 180123063

Q1.

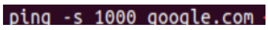
(a) The Option required to specify the number of echo requests is **-c**. For example, the following command would send 5 Echo Requests to google.com.

`ping -c 5 google.com`

(b) The Option required to specify the time interval (in seconds) is **-i**. For example, the following command would send Echo Requests with a time interval of 3 seconds.

`ping -i 3 google.com`

(c) The option required to keep sending Echo Requests without waiting for a reply is **-l preload** option. Only the superuser can select preload more than **3**. For normal users, the limit is 3 packets.

(d) The Option required to set the payload/data size is **s**. The command used is as follows: .

`ping -s 1000 google.com`

The above command would set the payload size to 1000 bytes. If the payload size is set to 32 bytes, then the total packet size (ping bytes) = 32+28 = 60 Bytes (extra 28 bytes for packet header = (ICMP Header (8 Bytes) + IP Header (20 Bytes))).

Q2. The six hosts chosen are: 1) codeforces.com 2) globat.com 3) iitd.ac.in 4) flikart.com 5) yahoo.co.jp 6) trademe.co.nz The experiment was carried at 3 different times (IST) ,12 am,10 am and 5 pm. The packet size is 64 bytes for all the hosts.

Destination Address	Server Location	Avg RTT (25) 12 am	Avg RTT (25) 10 am	Avg RTT (25) 5 pm	Final Avg RTT
codeforces.com	St. Petersburg, Russia	120.605ms	118.140ms	116.441ms	118.395ms
globat.com	Boston, US	25.920ms	24.357ms	24.361ms	24.879ms
iitd.ac.in	Delhi, India	267.321ms	277.616ms	308.122ms	284.353ms
flipkart.com	Bengaluru, India	227.125ms	256.163ms	287.152ms	256.816ms
yahoo.co.jp	Tokyo, Japan	204.127ms	210.152ms	194.684ms	202.987ms
trademe.co.nz	Wellington, New Zealand	209.703ms	272.564ms	230.093ms	237.456ms

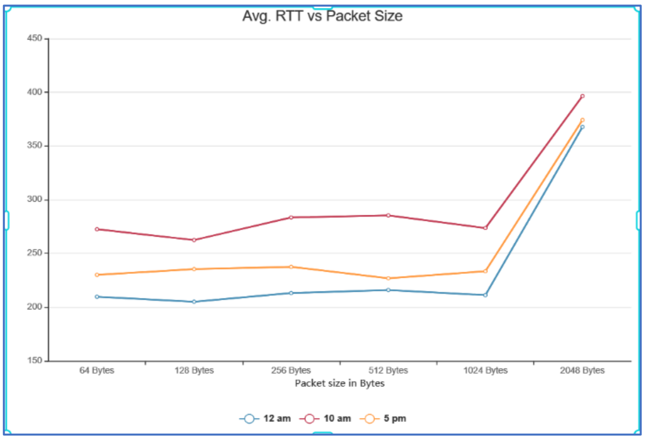
The site chosen for the Packet Size experiment is: **trademe.co.nz**

Size (Bytes) ->	64	128	256	512	1024	2048
Avg. RTT (12 am)	209.703ms	205.124ms	213.154ms	215.985ms	211.246ms	367.778ms
Avg. RTT (10 am)	272.564ms	262.506ms	283.512ms	285.417ms	273.687ms	396.542ms
Avg. RTT (5 pm)	230.093ms	235.461ms	237.561ms	226.915ms	233.481ms	374.265ms

(a)Average RTT Table is given above. There is a slight direct correlation between Avg. RTT and geographical distance between devices. More distance between source and destination indicates greater intermediate routers (more hops) and greater propagation delay. Each router may involve its own queueing delays. But geographical distance is not a huge factor, because it gets dominated by network congestion and server response speed.

(b)In the case of tradme.co.nz, there was a positive packet loss rate. The reason behind this might be increased collision probability (large distance), low server response capabilities and active firewalls that block the incoming ICMP packets. Sometimes, high network congestion may also cause packet loss (TTL exceeded).

(c)The flowing graph has been created using the table given above (Packet Size experiment).



(d) As seen the graph above, the Round-Trip Time remains approximately same till the Packet Size reaches 1024 Bytes. But when the packet size becomes 2048 bytes, there is a sudden increase in the RTT in all the three cases. The reason behind such an observation is that the default MTU (Maximum Transmission Unit) is 1500 Bytes. When the packet size exceeds the 1500 threshold limit, the packet is segregated to two frames of size 1500 bytes. Hence there is a sudden increase in RTT as there are two packets to be transmitted to the destination.

RTT (as shown in the first table) reaches its peak at 6 pm and reaches its minimum at 12 am as far as the Indian sites are concerned (due to network traffic {number of users}). All the sites have different target audience, and receive varied traffic from different countries. So, the network congestion is generally higher during the day time (with respect to the local time of the country from which the site receives majority traffic) than compared to the night times.

Q3. IP Address Used: 172.271.163.46 (google.com)

Command	Packets Sent	Packets Received	Packet Loss Rate	Min. Latency (ms)	Max. Latency (ms)	Mean Latency (ms)	Median Latency (ms)
`$ ping -n -c 1000 172.217.163.46`	1000	1000	0%	45.90	177.0	53.2627	51.0
`$ ping -p ff00 -c 1000 172.217.163.46`	1000	996	0.4%	45.60	187.0	53.738	51.2

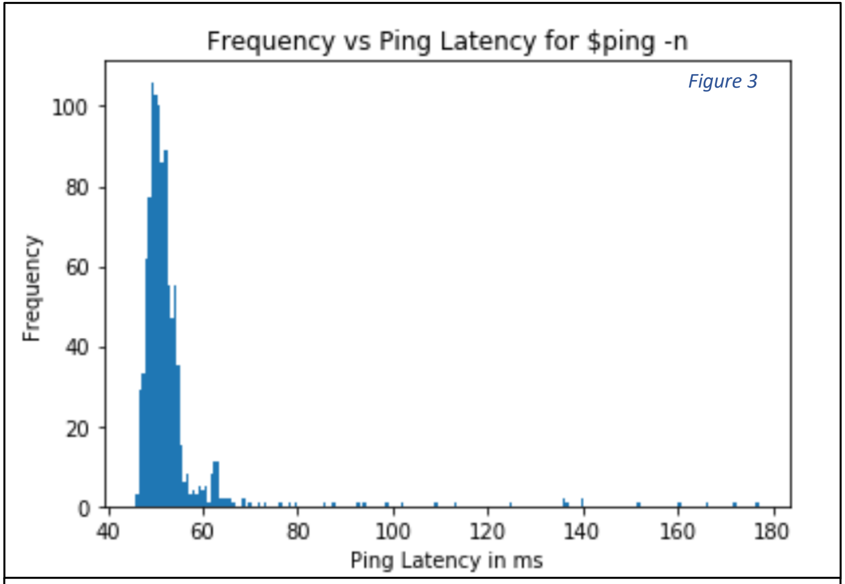


Figure 3

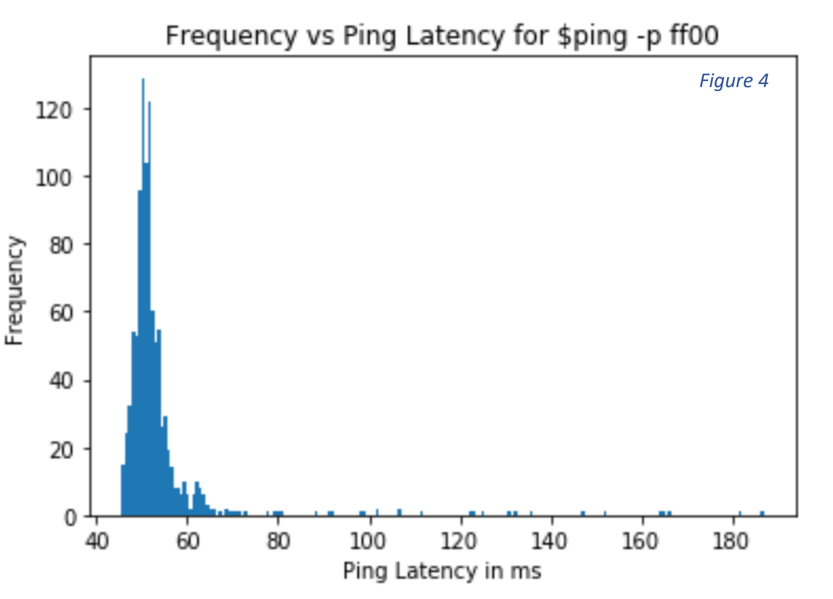


Figure 4

(a) As given in the table, Packet Loss Rate for the first command is 0% and for the second command is 0.4%.

(b) 1000 values of ping latencies were collected for each command. Using statistics module of python, mean, median, min and max value of the latencies was calculated (ignoring the ones that failed). The values are given in the above table.

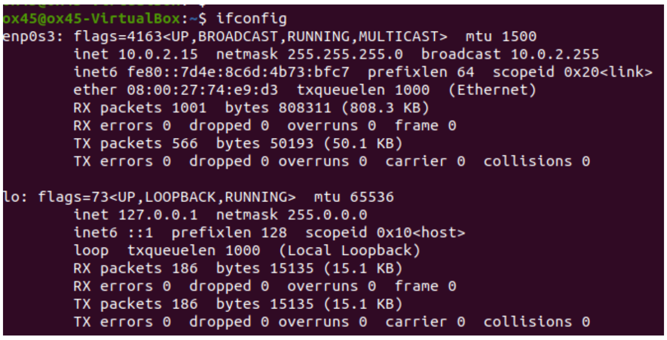
(c) Plot for both the cases is shown by figure 3 and figure 4. They were plotted using matplotlib library of python. These frequency histograms of the ping latency approximately match the normal distribution.

(d) The two experiments are almost similar. It differs slightly in two aspects:

(i) Mean Latency and the Median Latency of the first command is slightly lesser than that of the second command. The reason behind this is that the '-n' option in the first command produces numeric output only (Symbolic names for destination addresses will not be searched), hence making it slightly faster.

(ii) '-p ff00' option specifies the ping pattern. The second command sends two bytes, one filled with all ones and the other filled with all zeroes (1111111100000000). This option is used to figure out the data dependent network problems. Sometimes, not having enough transitions in the packet causes packet loss (in our case, we have only one transition from 1 to 0), which explains the higher packet loss rate (0.4%) in the second case.

Q4.(a) **'ifconfig'** stands for Interface Configuration. The command **ifconfig** (in Linux Terminal) displays information about all the network interfaces currently up and running on the system. The output can be seen in the screenshot (right). The first interface stands for **enp0s3** (Ethernet Network Peripheral # serial #). This naming system (based on geographic location of the connector) was introduced to avoid problems arising due to the classic interface naming convention (eth 0, eth1, etc.). The second interface **lo** stands for the loopback interface. It is used by system for self-communication and troubleshooting. **Flags 4163** denotes that the interface is up and ready to accept data, and the network supports multicasting and broadcasting. **MTU** (Max. Transmission Unit) denotes the size of the largest size of the packet that can be sent over a network layer in a single transaction**.** **Inet, netmask and broadcast addr** denote the machine IP address, available IP addresses and the broadcast addresses respectively. **Txquelen** denotes the length of the transmit queue of the device. **RX** and **TX Bytes** indicates total amount of the data received and transmitted over the interface. As seen in the image above, no **errors** and no **collisions** (indicator of network congestion, ideally should be 0) have been reported.



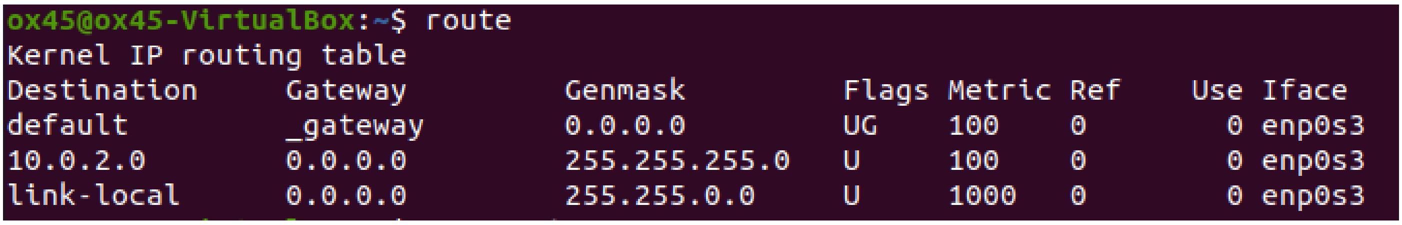
```
ox45@ox45-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::7d4e:8c6d:4b73:bfc7  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:74:e9:d3  txqueuelen 1000  (Ethernet)
        RX packets 1001  bytes 808311 (808.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 566  bytes 50193 (50.1 KB)
        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 186  bytes 15135 (15.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 186  bytes 15135 (15.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

(b) Options in **ifconfig** command:

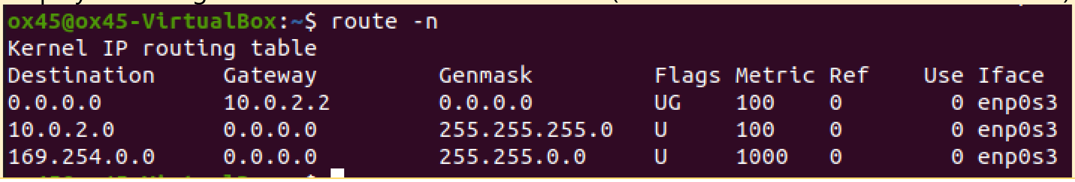
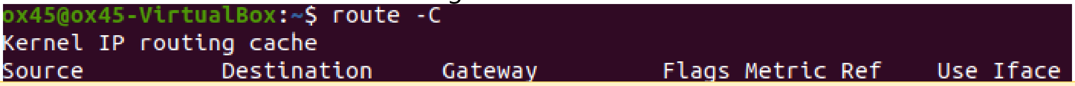
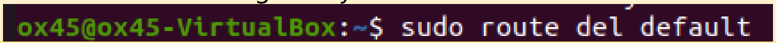
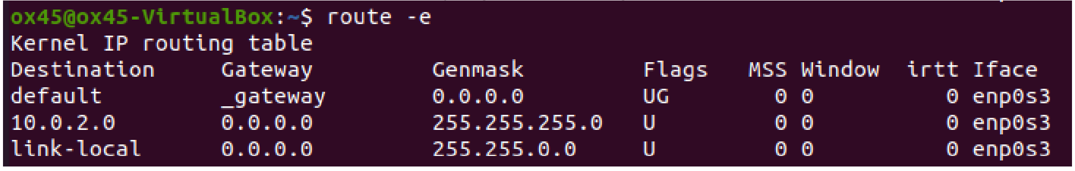
Option	Command	Explanation
-a	ifconfig -a	Displays information of all network interfaces (even if they are not currently running)
-v	ifconfig -v	Verbose mode, prints more information regarding certain errors
-s	ifconfig -s	Provides summarized and concise data of the interfaces (presented in a tabular format)
up, down	sudo ifconfig enp0s3 up	up: This causes the interface specified to get activated, makes it ready to accept data. down: This causes the driver of the specified network interface to shut down.

(c) The **route** command displays the kernel IP routing table of the device. The routing table is a data table that lists the paths to certain network destinations and the metrics (distances) associated with those paths. **Destination** column shows the Destination host name/address. **Gateway** column identifies the specific gateway address for the route. **Genmask** column identifies the netmask for the destination net. **Flags** (to describe the route) Column: **U** indicates that the route is Up, **G** indicates that the route is to a gateway. **Metric** column refers to the number of hops (distance) to reach the target. **Ref** column refers to the number of references to the route. **Use** column denotes the count of lookups for the specific root. **Iface** column indicates the interface to which the packets will be sent.



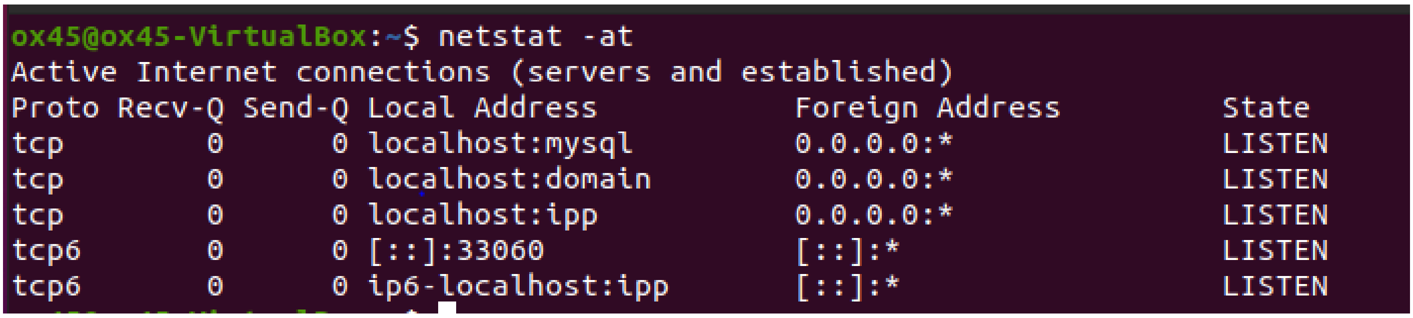
```
ox45@ox45-VirtualBox:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
```

(d) Options in **route** command:

Option	Command	Explanation
-n	route -n	Displays routing table in full numeric form (does not search for host names)  ox45@ox45-VirtualBox:~$ 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
-C	route -C	This command lists the kernel's routing cache information.  ox45@ox45-VirtualBox:~$ route -C Kernel IP routing cache Source Destination Gateway Flags Metric Ref Use Iface
del	sudo route del default	To delete the default gateway.  ox45@ox45-VirtualBox:~$ sudo route del default
-e	route -e	Uses the netstat (8)-format for displaying the routing table.  ox45@ox45-VirtualBox:~$ route -e Kernel IP routing table Destination Gateway Genmask Flags MSS Window irtt Iface default _gateway 0.0.0.0 UG 0 0 0 enp0s3 10.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 enp0s3 link-local 0.0.0.0 255.255.0.0 U 0 0 0 enp0s3

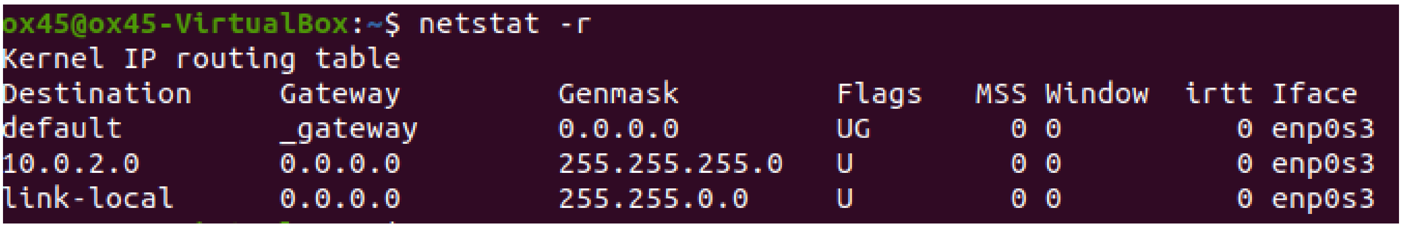
Q5. (a) **netstat** stands for network statistics. It is a command line tool used for viewing IP routing tables, viewing interface statistics, multicast memberships, performance measurement, troubleshooting, debugging, etc. Its primary function is to monitor network connections (both incoming and outcoming) associated with the device. It also provides information regarding the open ports and the programs running on them (if any).

(b) The command used for listing all the TCP ports is **netstat -at**. The parameter used is '**-at**'.



```
ox45@ox45-VirtualBox:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp6       0      0 [::]:33060              [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
```

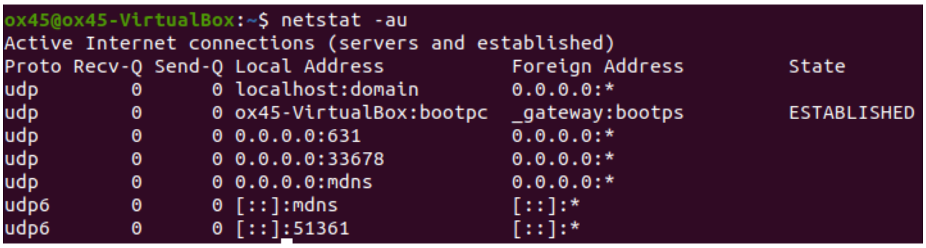
(c) **netstat -r** command is used to display the kernel IP routing table of the device. The routing table is a data table that lists the routes to certain network destinations. **Destination** column shows the Destination host name/address. **Gateway** column identifies the specific gateway address for the route. **Genmask** column identifies the netmask for the destination net. **Flags** (to describe the route) Column: **U** indicates that the route is Up, **G** indicates that this specific route is connected to a gateway. The next three columns are applied to the TCP connections (associated with that particular route). **MSS** (Maximum Segment Size) specifies the maximum amount of data(bytes) that can be received in a single TCP Segment. **Window** is the maximum burst of data that can be accepted for TCP connections over this route. **irtt** (Initial Round Trip Time) is the round-trip time that is determined by the TCP three-way handshake, used to estimate the optimum window size of a connection. **Iface** column indicates the interface to which the packets will be sent.



```
ox45@ox45-VirtualBox:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
```

(d) The option used to display status of all network interfaces on the device is **'-i'** (Command: **netstat -i**). By using the mentioned command, I figured out that my current device uses **two** network interfaces (**enp0s3** {Ethernet Network Peripheral # serial #} and **lo** {loopback interface}).

(e) The option used to display all the UDP connections prevalent on the device is **'-au'** (Command: **netstat -au**). **UDP** (User Datagram Protocol) facilitates the exchange of messages between computing devices in a network.

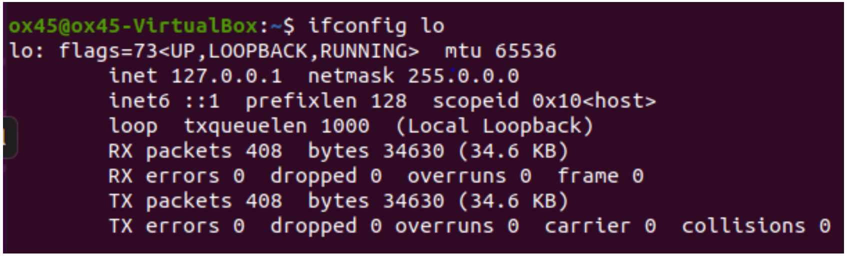


```
ox45@ox45-VirtualBox:~$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 ox45-VirtualBox:bootpc  _gateway:bootps         ESTABLISHED
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:33678           0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp6       0      0 [::]:mdns               [::]:*
udp6       0      0 [::]:51361              [::]:*
```

(f) Loop-back interface is the network interface (virtual) used by the device to make connections with itself. Any traffic that a computer program sends on this loopback network is addresses to the same source device. The standard domain name for this address is **localhost.** Its Benefits are:

i) As the loopback address remains static, it is the most secure method to identify a device on the current network.

ii) This interface is up and running as long as the route to that IP Address is available. So, it can be used for troubleshooting and diagnostic purposes.



```
ox45@ox45-VirtualBox:~$ ifconfig lo
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 408  bytes 34630 (34.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 408  bytes 34630 (34.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

iii) Many network related command line functions use the loopback interface for various purposes.

Q6. (a) Traceroute is a network tool used to display the specific route taken by data packets across an IP network. This tool will show the numbers of hops, and also display information about each sequential hop. It tells the user about how the user's ISP connects to the internet.

(b) The flowing table displays the number of hops (metric) to reach the Destination Addresses:

Host Name->	codeforces.com	globat.com	iitd.ac.in	flipkart.com	yahoo.co.jp	trademe.co.nz
No. of Hops (12 am)	12	13	10	8	12	11
No. of Hops (10 am)	12	13	10	8	13	10
No. of Hops (6 pm)	13	13	10	8	12	10

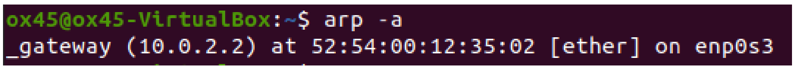
In all the cases, 45.79.12.202 and 45.79.12.201 are common, since these Addresses are associated with the Internet Service Provider. 45.79.12.2 is a common hop to all the sites, except codeforces.com. codeforces.com, flipkart.com and trademe.co.nz share a common hop (45.79.12.9). After the first few hops all the routes diverge from each other and share no common hops. These common hops arise due to same internet networking patterns and pre-defined protocols for diverting network traffic (at initial stages).

(c) The route to the hosts changes at different times of the day to avoid network congestion. The intermediate routers redirect and rebalance the network traffic to a certain route, in order to reduce network load. This decreases the rate of packet loss. This path selection technique is employed to reduce average RTT times of data packets.

(d) In certain cases, traceroute tool is not able to reach the final destination IP Address. The tool returns back the error message “Traceroute request timed out”. There are several reasons why a certain router blocked the incoming ICMP packet at a certain stage. It is habitual that for security reasons, ICMP/ping is blocked to prevent hackers from obtaining information about the open ports and to avoid service cyber-attacks. Firewalls might be present at certain stages which blocks the server from responding. Many network providers also disable ICMP response to decrease network congestion.

(e) Yes, it is possible to find routes to specific hosts (that have failed to return a response using ping) with the traceroute tool. This is because Ping and Traceroute follow different set of instructions for their implementation. Ping is a simple and a quick utility to tell if the specified server is accessible or not. Ping works by sending an ICMP packet to the specified IP Address and then waiting for a reply. If a ping to a specific destination fails, the most probable reason is that server is down or not sending a response. Contrastingly, the traceroute tool employs the TTL (Time to Live) feature of the data packet. When a packet cannot reach the final destination, the final node in the route returns the packet and identifies itself. The traceroute tool sends packets with increasing TTL values, until the destination Address is reached. So, gradually, the traceroute tool identifies all the intermediate hosts. So, even if the end server is blocking a response, the traceroute tool will the obtain the data about the end server and find a viable route as well.

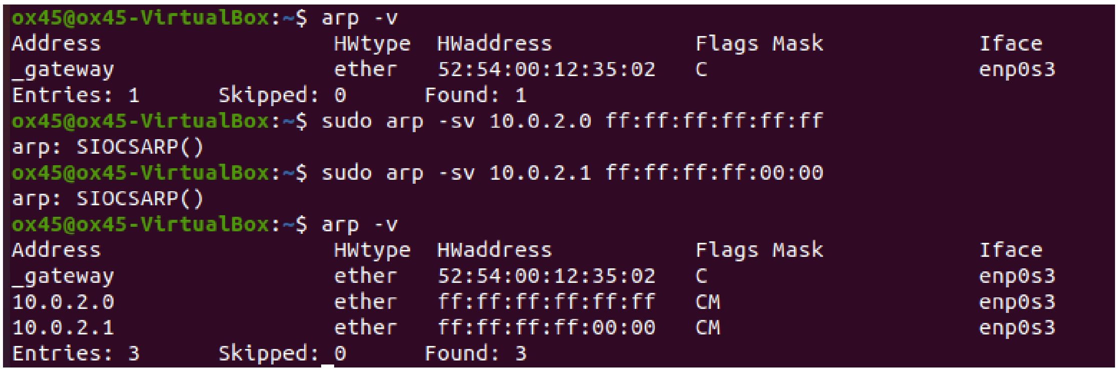
Q7. (a) **ARP** stands for Address Resolution Protocol. Majority of the devices use IP addresses to send/receive data, but the actual communication (data transfer) involves the usage of physical address (**MAC** {Media Access Control} address). So, here the ARP Table provides the needed information to translate the given IP address to appropriate physical (MAC) address. The command required to see the all the entries of the ARP table is **arp -a.** The columns of the **arp -a** result are as follows: **| Hostname | IP Address | MAC Address | Interface | Hardware Type |**



```
ox45@ox45-VirtualBox:~$ arp -a
_gateway (10.0.2.2) at 52:54:00:12:35:02 [ether] on enp0s3
```

Hostname and IP Address: Label and Address assigned to the host. MAC Address: MAC {Media Access Control} Address is the hardware identification number assigned to each device on a network. **Interface:** It specifies the network interface over which communication is happening. HWType: It specifies the type of hardware used for the network transmitting the ARP message.

(b) To delete an entry from the ARP Table, we have to use the following command: **arp - d <Address>.** To add an entry into the ARP Table, we can use the following command: arp -sv <Address> <MAC Address>. It is important to note that both the commands require super user privileges (sudo).



```
ox45@ox45-VirtualBox:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   52:54:00:12:35:02   C                     enp0s3
Entries: 1      Skipped: 0      Found: 1
ox45@ox45-VirtualBox:~$ sudo arp -sv 10.0.2.0 ff:ff:ff:ff:ff:ff
arp: SIOCSARP()
ox45@ox45-VirtualBox:~$ sudo arp -sv 10.0.2.1 ff:ff:ff:ff:00:00
arp: SIOCSARP()
ox45@ox45-VirtualBox:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   52:54:00:12:35:02   C                     enp0s3
10.0.2.0                 ether   ff:ff:ff:ff:ff:ff   CM                    enp0s3
10.0.2.1                 ether   ff:ff:ff:ff:00:00   CM                    enp0s3
Entries: 3      Skipped: 0      Found: 3
```

As seen in the above screenshot, Address 10.0.2.0 and Address 10.0.2.1 have been added to the ARP table.

(c) ARP Table contains IP Addresses that have the same subnet. ARP Table helps in communication between devices that are present in the same IP subnet. When a message is to be sent from device A to device B, the computer first consults the routing table to determine if A and B belong to the same subnet or not. If the belong to the same subnet, then device A uses ARP Table to translate IP Address of B to MAC (Physical) Address, over which direct transmission of the data packets can take place. But, if devices A and B don’t belong to the same IP subnet, then the computer will then search in the routing table for a router/gateway to which the data packets have to be sent (since the IP Address of B has been deemed currently unreachable). ARP Table will then be used to determine the hardware address of the specific router, to which the data packets can be sent for further transmission.

(d) Upon experimenting with the task given in the question, the ping to the IP Address produced ambiguous results. In some cases, there was 100% packet loss, in other cases some cases, varied packet loss rates were reported. But theoretically the outcome is undefined. ARP Table contains two entries with same IP Address but different MAC Addresses (MAC1 and MAC2). Suppose initially, the request is sent to the first device (MAC1), it will be stored in the ARP Cache, and will be referenced for the subsequent requests until ARP Table refresh (timeout) occurs. After the refresh, either of the two devices can be chosen (depending on the algorithm employed). Then the process repeats. The system may alert the system admin of Duplicate address Detection. . So, at any given instance of time, only one device will be running. If ARP refresh rate is small enough, request received from device 1 might be responded to device 2. Hence, the output of such a ping request will be ambiguous.

Q8.

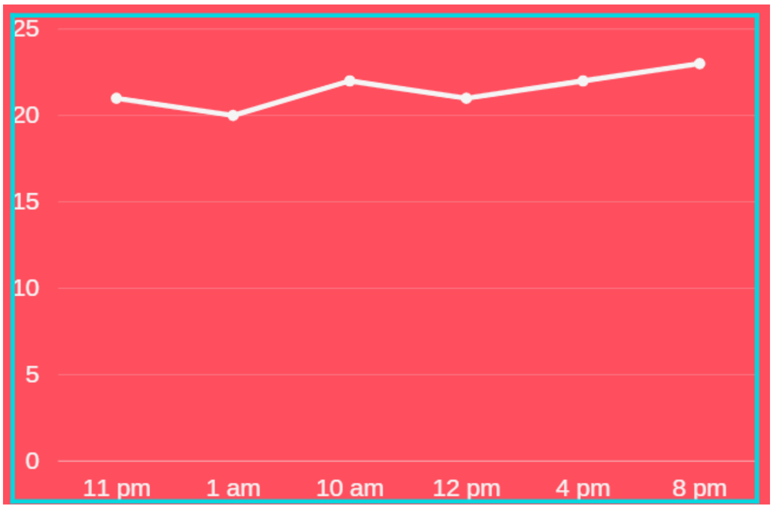
(a) The command used to check the active devices belonging to the sub net is: **nmap -sP <Address (in CIDR format)>**

(b) The command used to detect the firewall settings active on the host: **nmap -sA <IP Address>**

(c) The table shows the number of active users at different time of the day:

11 pm	1 am	10 am	12 pm	4 pm	8 pm
21 users	20 users	22 users	21 users	22 users	23 users

'



Trend: The number of active hosts remained almost same throughout the day. No specific pattern could be observed.

*Note: Because the lab PC required superuser privilege's for installing nmap, the command arp –a -n was used to check active hosts on the LAN network.