Computer Networks
Lab Assignment 3 Report

Karthik Kancharla
190020020
February 15th, 2021

Questions

Question 1)

a) The option required to specify the number of echo requests to send with the ping command is **-c**. Eg: ping -c 4 www.amazon.in

b) The option required to set a time interval between two successive ping ECHO_REQUEST's is **-i**. Eg: ping -i 3 www.amazon.in (ping -i time).

c) An example command to send ECHO REQUEST packets to the destination one after another without waiting for a reply is ping -l 3 www.amazon.in. (ping -l time) Option used is **-l**.

   The limit for sending such ECHO_REQUEST for normal users is 3.

d) An example command to set the ECHO_REQUEST packet size is
   ping -s 64 www.amazon.in. (ping -s packetsize)
   The option used is **-s**.
   If the packet size is set to 64 bytes, the total packet size would be 92 bytes considering 8 bytes for ICMP header and 20 bytes for the IP address header.

Question 2)
5 hosts I am choosing are:

1) amazon.in
2) google.com
3) youtube.com
4) flipkart.com
5) myntra.com

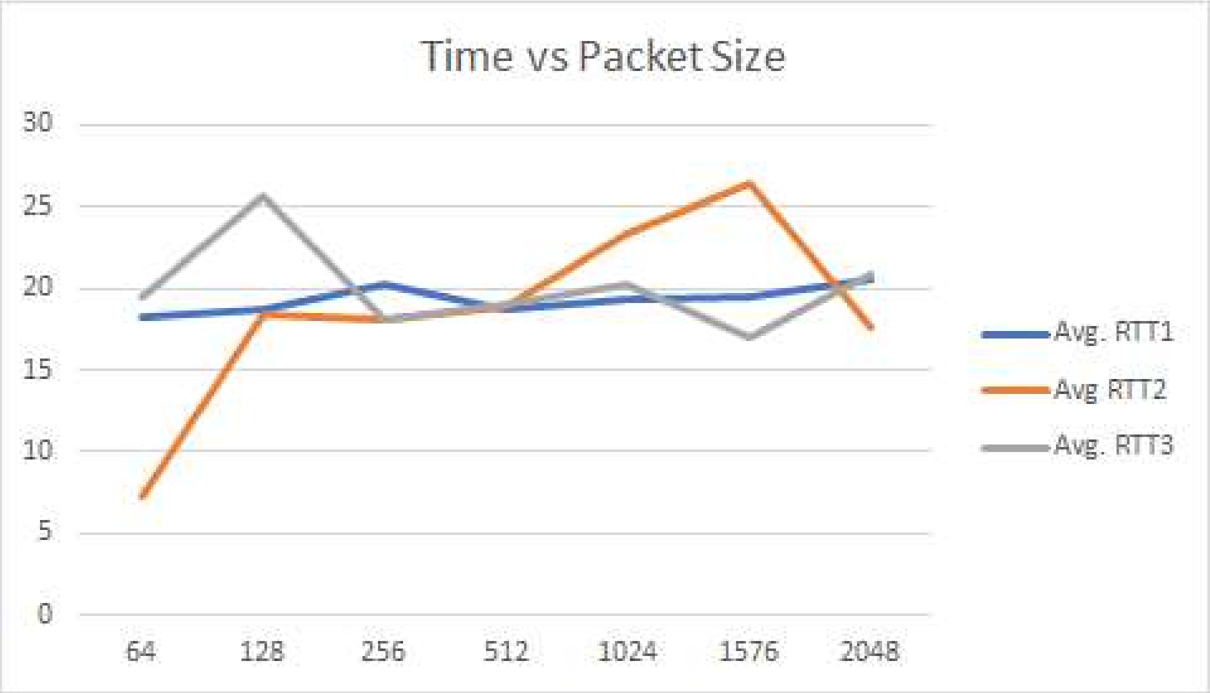
Domain Name	IP Address	Geolocation	Avg. RTT1 (6PM)	Avg.RTT2 (11PM)	Avg. RTT3 (10AM)	Total Avg. RTT
amazon.in	23.62.25.28	USA	18.075ms	7.241 ms	18.401ms	14.572ms
google.com	74.125.138.147	USA	34.846ms	34.973 ms	33.361 ms	34.393ms
youtube.com	74.125.196.93	USA	33.934 ms	33.680 ms	34.312 ms	33.975ms
flipkart.com	163.53.78.110	India	214.974 ms	217.300 ms	215.781 ms	216.018ms
myntra.com	23.62.24.177	USA	18.680 ms	17.828 ms	19.891 ms	18.799ms

I didn't see any case with packet loss. Everywhere 100% packets were transmitted. This might be because all the connections were fine and everything was well connected and the amount of traffic was also normal every time I pinged.

From the above table **we can see that RTT's are weakly correlated with geographical distance of the hosts**. It just depends on the number of routers that come in the way and the traffic at that time, also the proxy servers setup makes an effect to send pages quickly by that website. Transmission is not considered on a large scale as the transmission happens at speed of light.

I choose www.amazon.in to perform experiments with different packet sizes. And the results are mentioned in the below table.

	64	128	256	512	1024	1576	2048
Avg. RTT1	18.242ms	18.771ms	20.212ms	18.638ms	19.267ms	19.451ms	20.633 ms
Avg. RTT2	7.262 ms	18.460 ms	18.112 ms	18.837 ms	23.376 ms	26.393 ms	17.615 ms
Avg. RTT3	19.415 ms	25.614 ms	18.053 ms	18.968 ms	20.284ms	16.997ms	20.890 ms



Generally, higher RTT’s are seen at day time considering an indian(local) website because most users are active and server will be busy with heavy traffic. Similarly, at night lesser RTT’s would be seen.

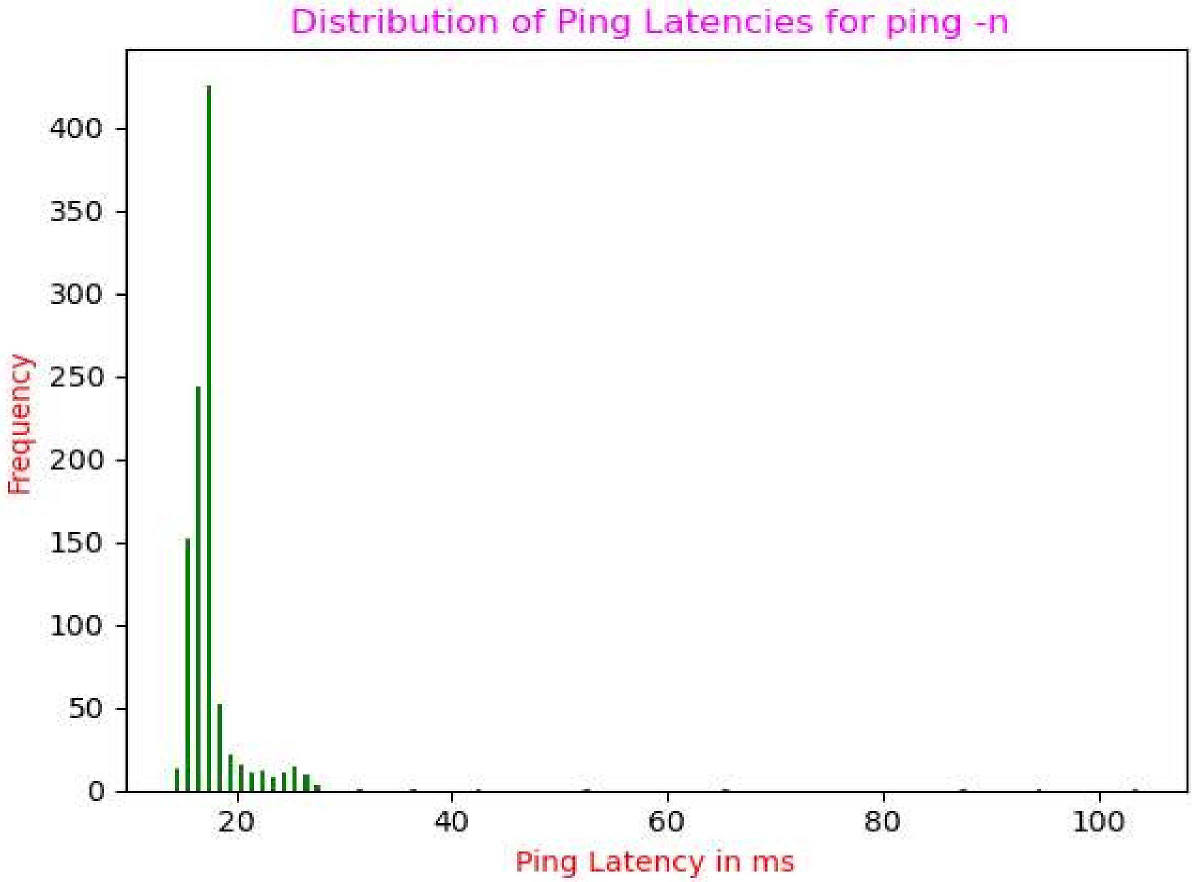
Question 3)

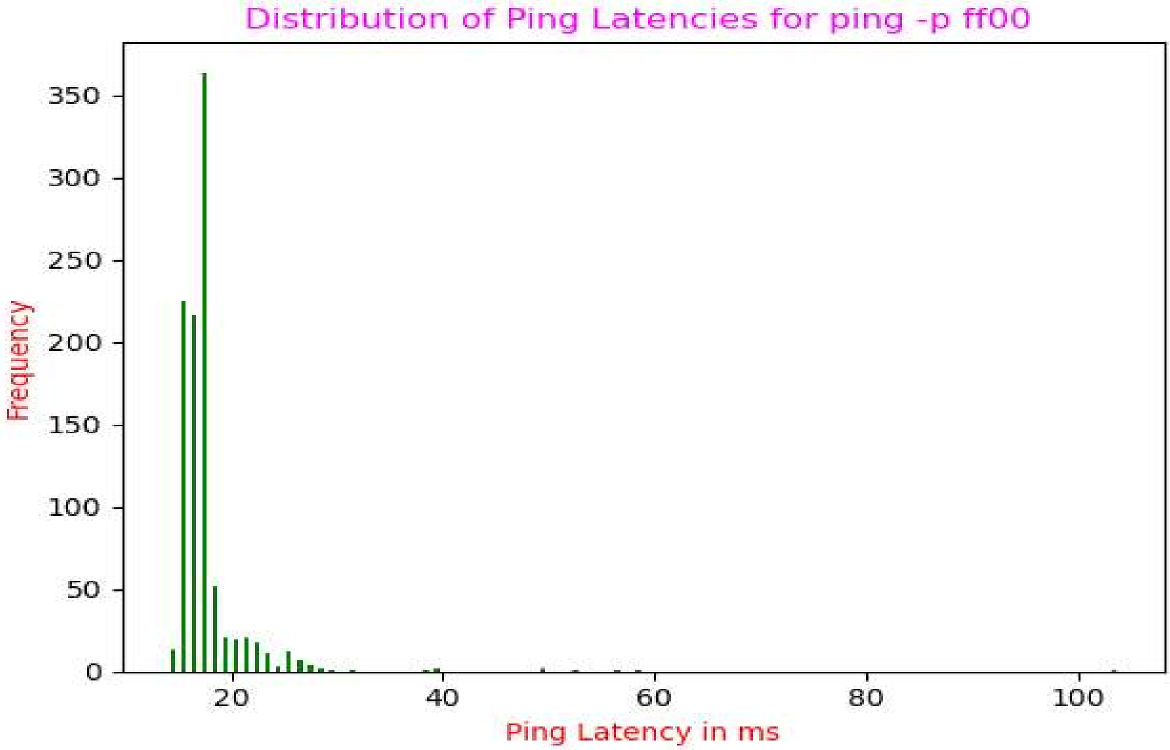
The IP address chosen by me is 142.250.67.74. This is the IP address of google.

a) Packet loss for ping -c 1000 -n 142.250.67.74 is 0.2%.
   Packet loss for ping -c 1000 -p ff00 142.250.67.74 is 0.2%.

b) Data in the table.

c) Graphs: These graphs and median latency were found using python code.





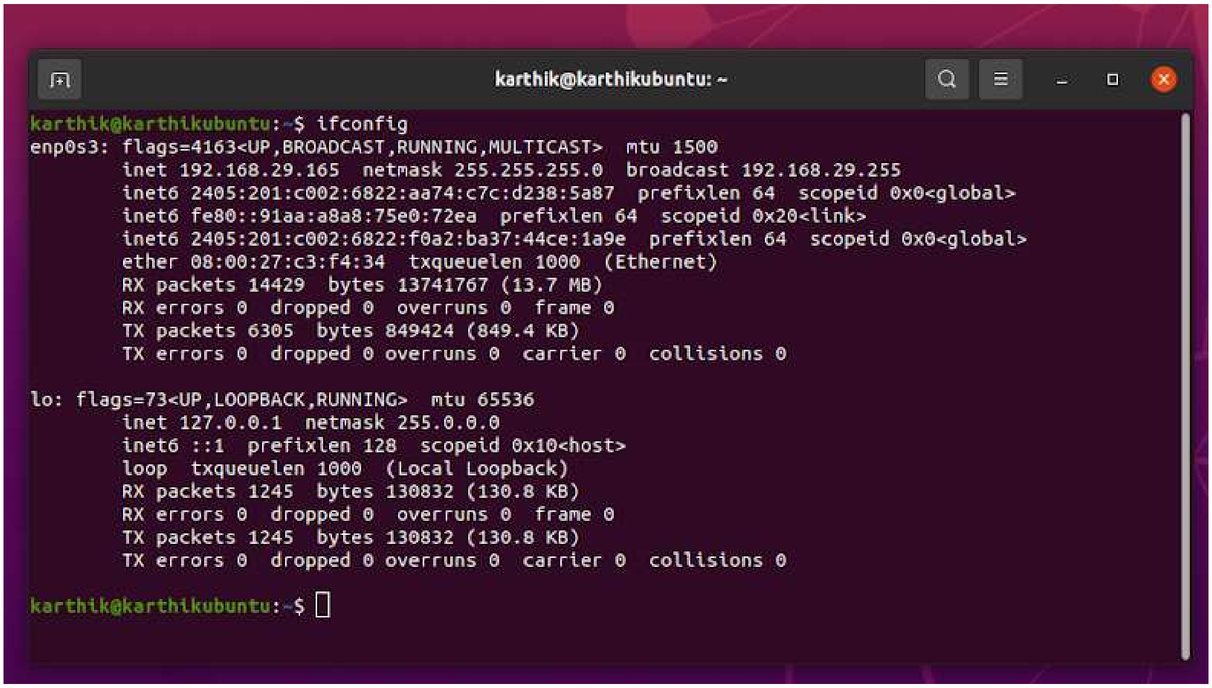
d) From the graphs, we can see that the maximum frequency of packets in a specific range is different using both the commands (~350 and ~400). There wasn't much difference in the graph pattern and packet loss rate, and the values of latency observed as the only difference was the pattern used. The maximum frequency was the only big difference which occurred due to the pattern used.

Output Table:

Command	Packets Sent	Packets Received	Packet Loss Rate	Min. Latency	Max. Latency	Mean Latency	Median Latency
ping -n 142.250.67.74	1000	998	0.2%	14.561ms	1040.939ms	18.769ms	17.0ms
ping -p ff00 142.250.67.74	1000	998	0.2%	14.450ms	1041.089ms	18.640ms	17.0ms

Question 4)

ifconfig:



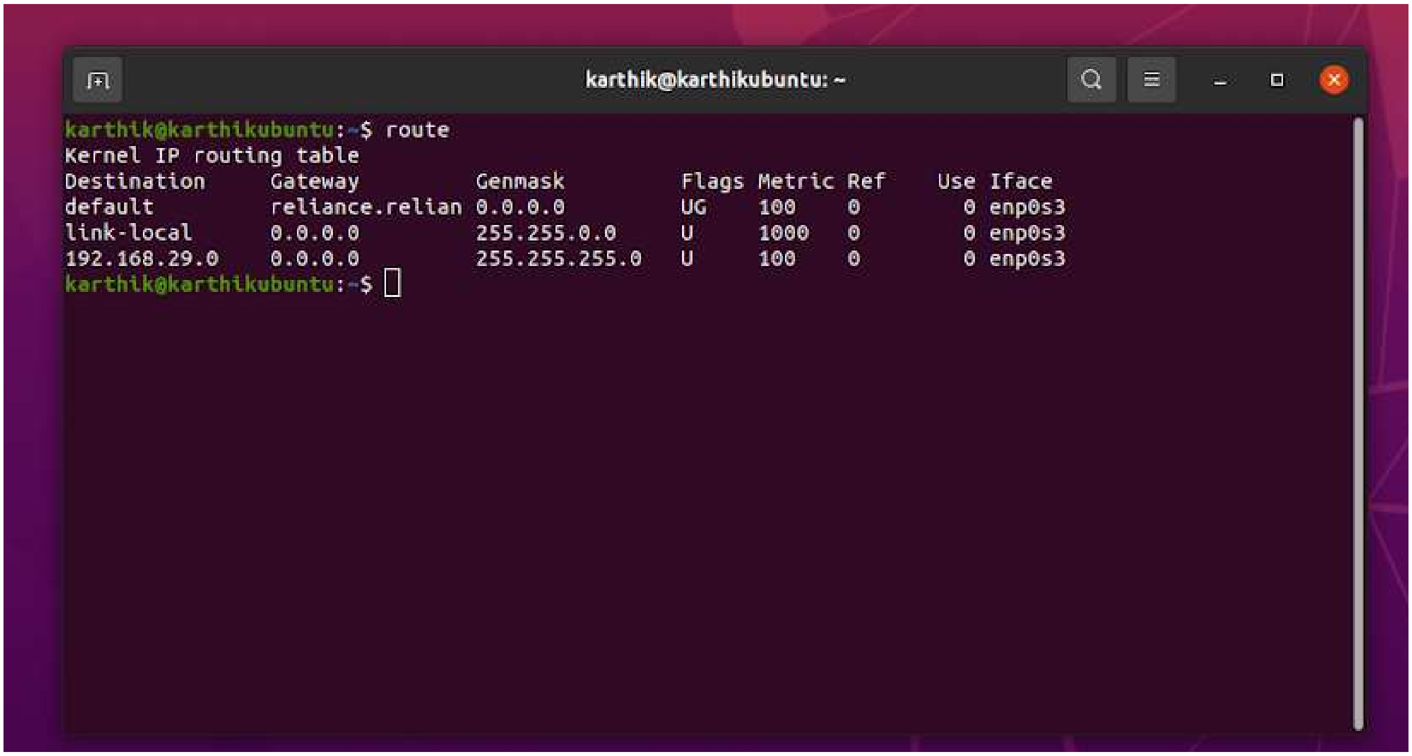
```
karthik@karthikubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.29.165  netmask 255.255.255.0  broadcast 192.168.29.255
        inet6 2405:201:c002:6822:aa74:c7c:d238:5a87  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::91aa:a8a8:75e0:72ea  prefixlen 64  scopeid 0x20<link>
        inet6 2405:201:c002:6822:f0a2:ba37:44ce:1a9e  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:c3:f4:34  txqueuelen 1000  (Ethernet)
        RX packets 14429  bytes 13741767 (13.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6305  bytes 849424 (849.4 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 1245  bytes 130832 (130.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1245  bytes 130832 (130.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

karthik@karthikubuntu:~$ 
```

- enp0s3 is the ethernet interface.
- Flags shows the sum of all numbers based on each of them like UP, broadcast, running, multicast (shows the sum of code for each of those in the brackets).
- mtu is the maximum transmission unit size and 1500 is the size and it is used because when data loss occurs, whole webpage is not reloaded instead that particular packet is loaded from server again and helps to save data.
- inet is the IP address.
- Netmask shows the netmask used.
- Broadcast shows the ethernet device address that supports broadcasting.
- inet6 shows the IP address in version 6.
- For RX, TX packets sizes and errors, drops, overruns, frame are shown. Transaction queue length, ether MAC address is shown.
- Similarly above things for loop back interface also which never shuts down.
- In loopback ethernet is not present because it doesn’t come because we didn’t give any specific internet connection like ethernet or wifi.

route:



```
karthik@karthikubuntu:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         reliance.relian 0.0.0.0         UG    100    0        0 enp0s3
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
192.168.29.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
karthik@karthikubuntu:~$ 
```

route command is used to show/manipulate the IP routing table. In the output:

- Destination column shows the destination network or destination host.
- Gateway column shows the the gateway address or '*' if none set.
- Genmask column shows the netmask for the destination net; '255.255.255.255' for a host destination and '0.0.0.0' for the **default** route.
- Flags column shows the possible flags such as U(up),H(host),G(gateway) etc.,
- Metric column shows the 'distance' to the target (usually counted in hops). It is not used by recent kernels, but may be needed by routing daemons.
- Ref column shows the number of references to this route.
- Use columns shows the count of lookups for the route.
- Iface column shows the interface to which packets for this route will be sent.

Some options in route:

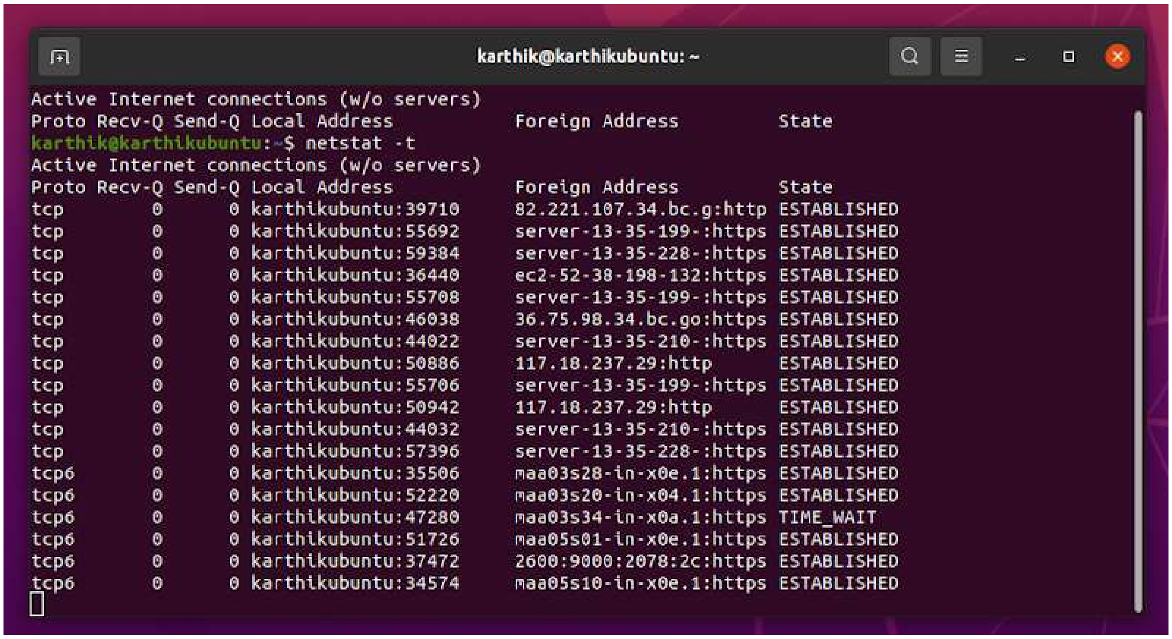
- -F to operate on the kernel's FIB (Forwarding Information Base) routing table. This is the default.
- -C to operate on the kernel's routing cache.
- -v to select verbose operation.
- -n to show numerical addresses instead of trying to determine symbolic host names.
- -ee to generate a very long line with all parameters from the routing table.
- del to delete a route.

Question 5)

The netstat is a command-line network utility that displays network connections for TCP, routing tables and number of network interface and network protocol statistics.

netstat -t shows all the TCP connections established.

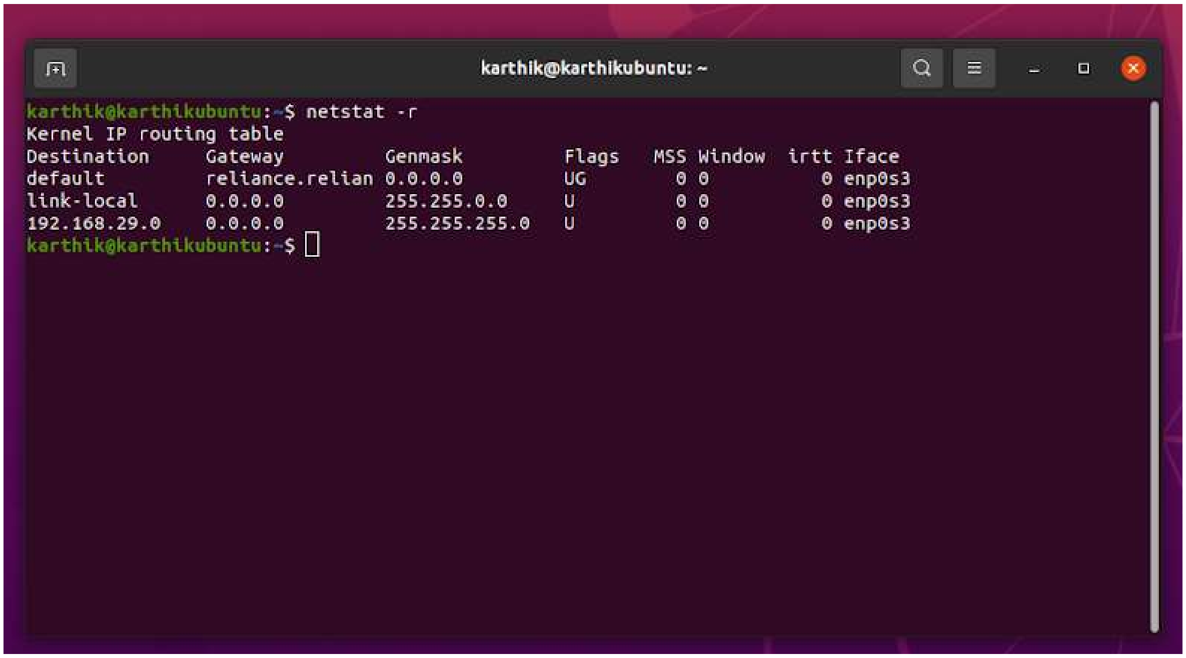
- Proto shows the name of protocol used by the socket such as TCP as shown in the screenshot.
- Recv -Q shows the count of bytes not copied by the user program connected to this socket.
- Send-Q shows the count of bytes not acknowledged by the remote host.
- Local Address column shows the address and port number of the local end of the socket.
- Foreign address column shows the address and port number of the remote end of the socket.
- State column shows the state of the socket such as ESTABLISHED, CLOSED etc.,



```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
karthik@karthikubuntu:~$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 karthikubuntu:39710     82.221.107.34.bc.g:http ESTABLISHED
tcp        0      0 karthikubuntu:55692     server-13-35-199-:https ESTABLISHED
tcp        0      0 karthikubuntu:59384     server-13-35-228-:https ESTABLISHED
tcp        0      0 karthikubuntu:36440     ec2-52-38-198-132:https ESTABLISHED
tcp        0      0 karthikubuntu:55708     server-13-35-199-:https ESTABLISHED
tcp        0      0 karthikubuntu:46038     36.75.98.34.bc.go:https ESTABLISHED
tcp        0      0 karthikubuntu:44022     server-13-35-210-:https ESTABLISHED
tcp        0      0 karthikubuntu:50886     117.18.237.29:http      ESTABLISHED
tcp        0      0 karthikubuntu:55706     server-13-35-199-:https ESTABLISHED
tcp        0      0 karthikubuntu:50942     117.18.237.29:http      ESTABLISHED
tcp        0      0 karthikubuntu:44032     server-13-35-210-:https ESTABLISHED
tcp        0      0 karthikubuntu:57396     server-13-35-228-:https ESTABLISHED
tcp6       0      0 karthikubuntu:35506     maa03s28-in-x0e.1:https ESTABLISHED
tcp6       0      0 karthikubuntu:52220     maa03s20-in-x04.1:https ESTABLISHED
tcp6       0      0 karthikubuntu:47280     maa03s34-in-x0a.1:https TIME_WAIT
tcp6       0      0 karthikubuntu:51726     maa05s01-in-x0e.1:https ESTABLISHED
tcp6       0      0 karthikubuntu:37472     2600:9000:2078:2c:https ESTABLISHED
tcp6       0      0 karthikubuntu:34574     maa05s10-in-x0e.1:https ESTABLISHED
```

netstat -r shows the kernel routing tables.

- Destination column shows the destination IP address.
- Gateway column shows the gateway to which the routing entry points, If no gateway is used, asterisk is used.
- Genmask column shows the network mask for the route.
- Flags column shows the flags that describe the route. Eg: U,G,H,D,M etc.,
- MSS column shows the maximum segment size. (size of the largest data segment used for the transmission.
- Window column shows the maximum amount of data the system will accept in a single burst from a remote host. (default size of the window)
- irtt is the initial round trip time.
- Iface column shows the network interface used.

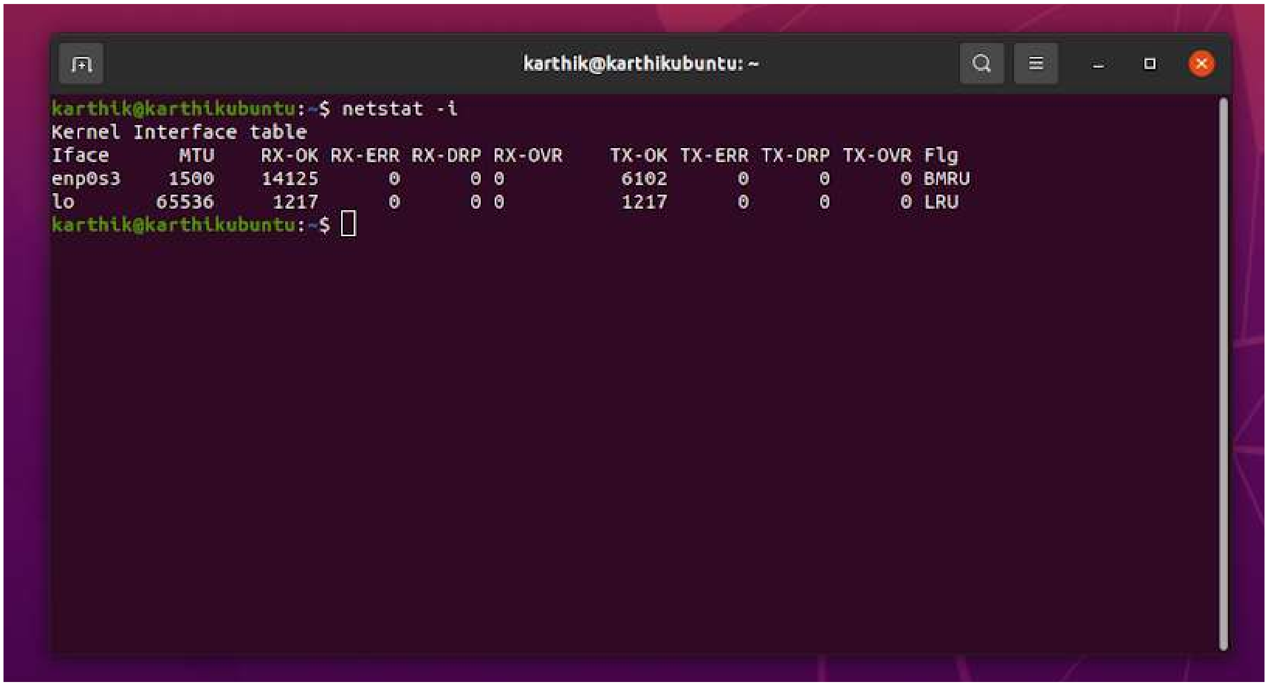


```
karthik@karthikubuntu:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         reliance.relian 0.0.0.0         UG        0 0          0 enp0s3
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
192.168.29.0    0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
karthik@karthikubuntu:~$
```

To display network interface status, the option -i is used.

There are 2 network interfaces as shown in the screenshot by using the above command.

The second one shown in the screenshot is the loopback interface. The **loopback** device is a special, virtual network **interface** that your computer uses to communicate with itself. It is used mainly for diagnostics and troubleshooting, and to connect to servers running on the local machine. It is useful because it is an interface with an IP address which will never go down.



```
karthik@karthikubuntu:~$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3    1500    14125      0      0 0          6102      0      0      0 BMRU
lo       65536     1217      0      0 0          1217      0      0      0 LRU
karthik@karthikubuntu:~$ 
```

Question 6)

	amazon.in	google.com	youtube.com	flipkart.com	myntra.com
Hop count1	22	16	13	9	10
Hop count2	22	13	16	9	11
Hop count3	22	14	16	9	11

a) Yes, there exists some common hops between different routes. One of them is reliance.reliance which is my internet provider. Also one of them is 10.1.48.1 which may be the adjacent IP address for my router.

b) The route is not the same at all times because the traffic and number of users will vary with time, so each time a route with least time is chosen to reduce delay. As google has many users, you can see that variation here easily.

c) Traceroute may not find complete paths in some cases, there may be various reasons such as: the firewall or other security device blocks the request. It may wait too much at the destination and the time to live for the packet may also expire, also there maybe sudden power shutdown at the server also. This didn’t happen with the hosts I have chosen.

d) Yes, it is possible to find the route to certain hosts which fail to respond with a ping experiment. This may happen because ICMP packets may be blocked, which are used by ping and UDP packets which are used by traceroute may not be blocked. In this case, traceroute works and ping doesn’t work. So, it is possible.

Question 7) Using the command arp, we can show the full ARP Table of my machine.

There are a total of 6 columns.

- Address column shows the IP address of the connections,
- HWtype shows the Hardware type of the connection used.
- HWaddress shows the hardware address.
- Flags column shows the flag accordingly. (C - Complete Entry (Dynamically Learned) , M - Manual Entry, P - Published entry.)
- Mask
- Iface column shows the network interface used.

sudo arp -d is used to delete an arp entry.
sudo arp -s *IPaddress* *MACaddress* is used to add entry to the arp table.

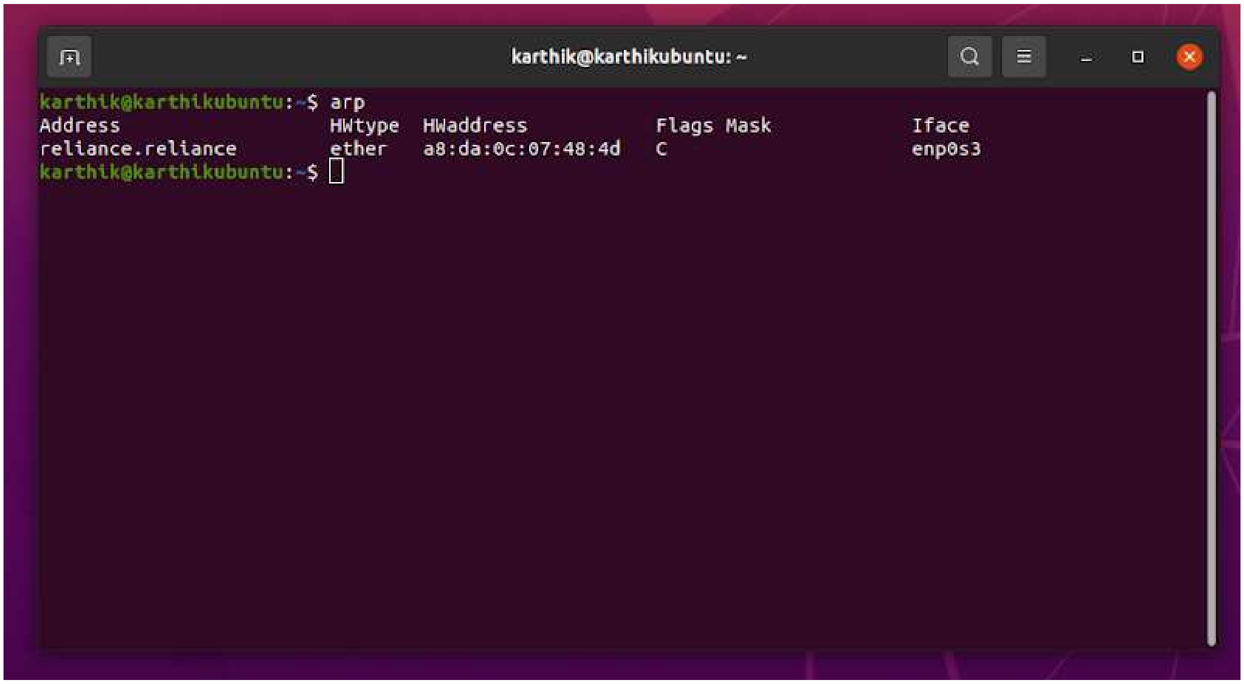
When you try to add an entry to the arp table, it is like telling the device that there exists another device with the specified ip address and mac address in their own network or nearby location. Deleting is like telling it to forget that device or not use it while doing traceroute or something else.

The entries stay cached in the arp table for about 60 seconds. This is also called arp cache timeout. The command used to find the time is given below:
cat /proc/sys/net/ipv4/neigh/default/gc_stale_time

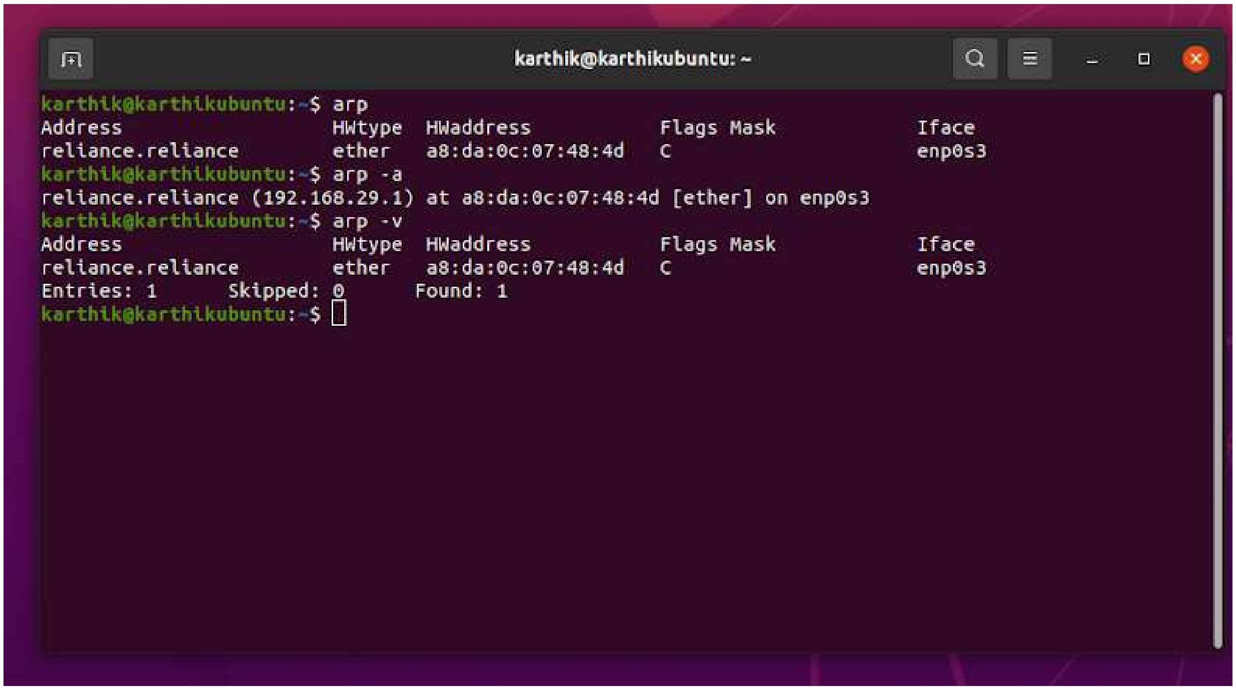
A possible trail and error method would be to add an entry, and then wait for some 2min and then see if the entry is still present or not. This way we can find the cache time. May be we could just change the system time and then check for faster checking.

If two IP addresses map to the same ethernet address, then requests will be sent normally. But while sending, the packets coming from the server will not be able to follow a specific path, accordingly, they will follow the path with least traffic, and reach the wrong machine instead of the one from which the request is sent. Nothing works properly. Even in the subnet (group of devices), the same will happen, packets will be randomly received by each machine. There will be no use of the network as nothing works properly at that time.

Command Outputs:



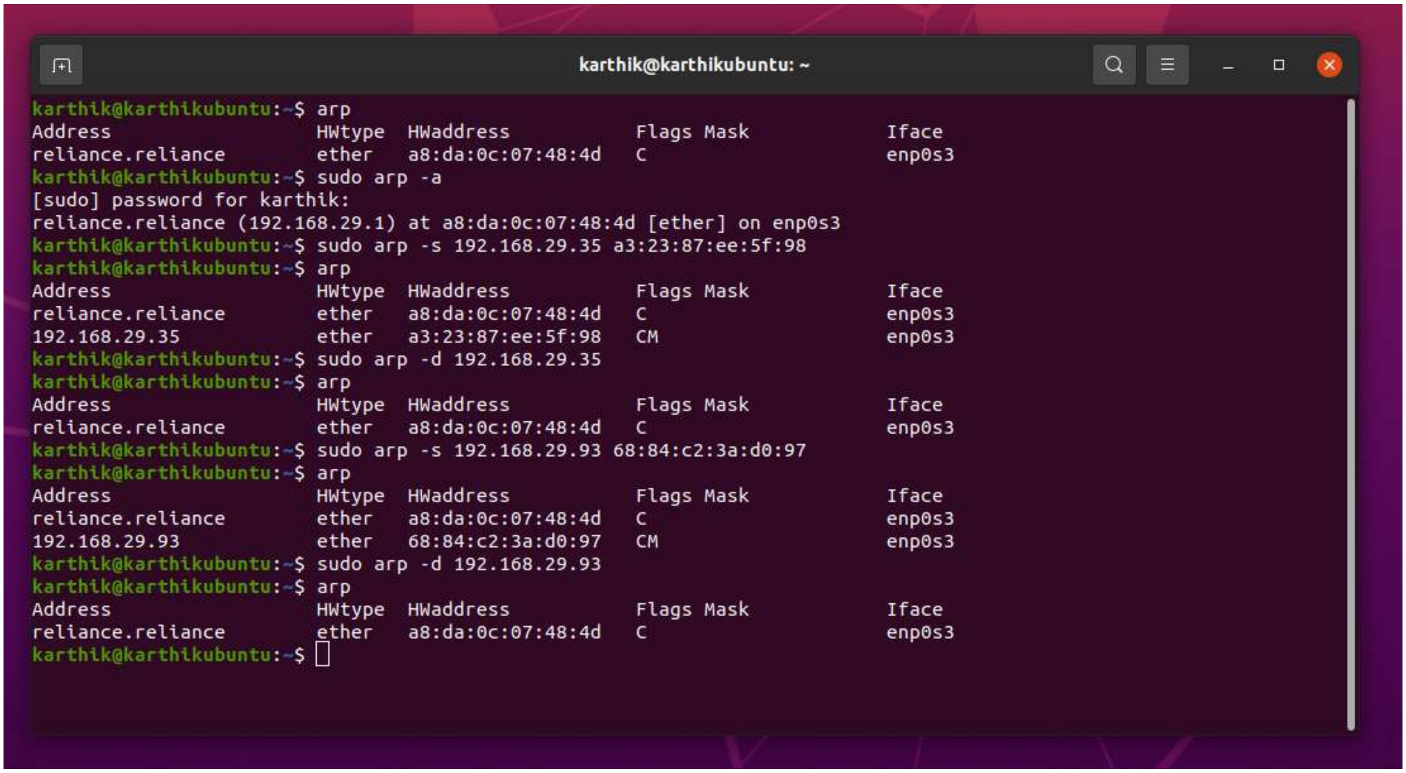
```
karthik@karthikubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
karthik@karthikubuntu:~$ 
```



```
karthik@karthikubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
karthik@karthikubuntu:~$ arp -a
reliance.reliance (192.168.29.1) at a8:da:0c:07:48:4d [ether] on enp0s3
karthik@karthikubuntu:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
Entries: 1      Skipped: 0      Found: 1
karthik@karthikubuntu:~$ 
```

Adding and deleting entries to the arp table:

I have added and deleted 2 entries from the arp table. You can see that in the below screenshot.



```
karthik@karthikubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
karthik@karthikubuntu:~$ sudo arp -a
[sudo] password for karthik:
reliance.reliance (192.168.29.1) at a8:da:0c:07:48:4d [ether] on enp0s3
karthik@karthikubuntu:~$ sudo arp -s 192.168.29.35 a3:23:87:ee:5f:98
karthik@karthikubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
192.168.29.35            ether   a3:23:87:ee:5f:98   CM                    enp0s3
karthik@karthikubuntu:~$ sudo arp -d 192.168.29.35
karthik@karthikubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
karthik@karthikubuntu:~$ sudo arp -s 192.168.29.93 68:84:c2:3a:d0:97
karthik@karthikubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
192.168.29.93            ether   68:84:c2:3a:d0:97   CM                    enp0s3
karthik@karthikubuntu:~$ sudo arp -d 192.168.29.93
karthik@karthikubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
reliance.reliance        ether   a8:da:0c:07:48:4d   C                     enp0s3
karthik@karthikubuntu:~$ 
```