



# CS 558: Computer Systems Lab

## Report on Network Diagnostic Commands & Socket Programming (Assignment 1)

---

### Group 5

Aditya Deshmukh	234101004	<a href="mailto:aditya.deshmukh@iitg.ac.in">aditya.deshmukh@iitg.ac.in</a>
Akshay Bhosale	234101006	<a href="mailto:a.bhosale@iitg.ac.in">a.bhosale@iitg.ac.in</a>
Ritik Kumar Koshta	234101044	<a href="mailto:r.koshta@iitg.ac.in">r.koshta@iitg.ac.in</a>
Rumit Gore	234101045	<a href="mailto:g.rumit@iitg.ac.in">g.rumit@iitg.ac.in</a>

---

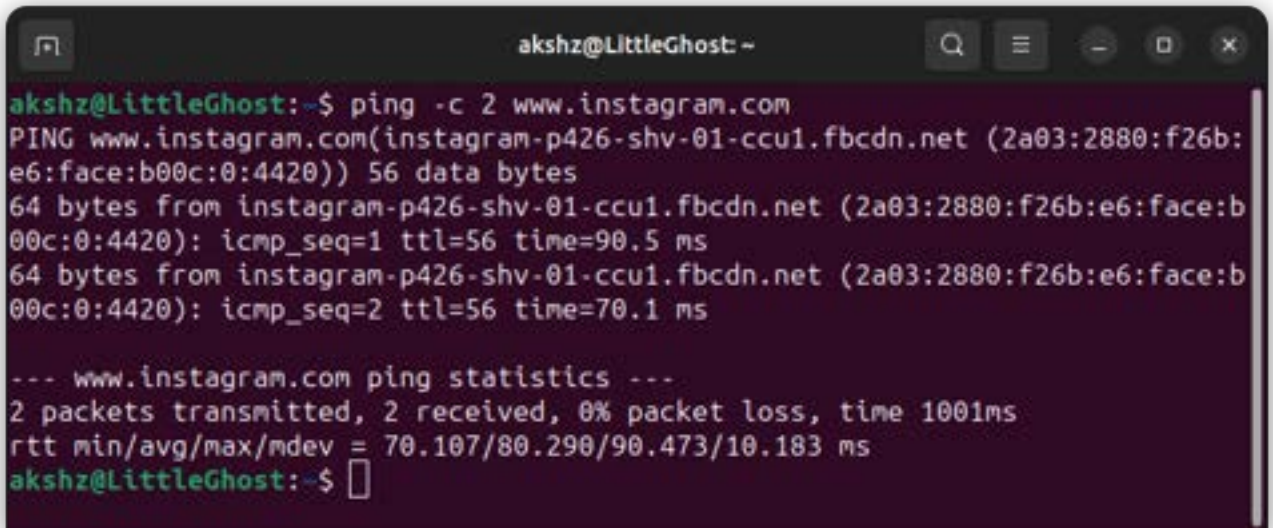
## Assignment – 1: Network Diagnostic Commands & Socket Programming

### Q1. Ping Command

The ping command is a network utility that sends small packets of data, known as ICMP ECHO\_REQUEST, to a specified destination and measures the round-trip time for the corresponding ECHO\_REPLY.

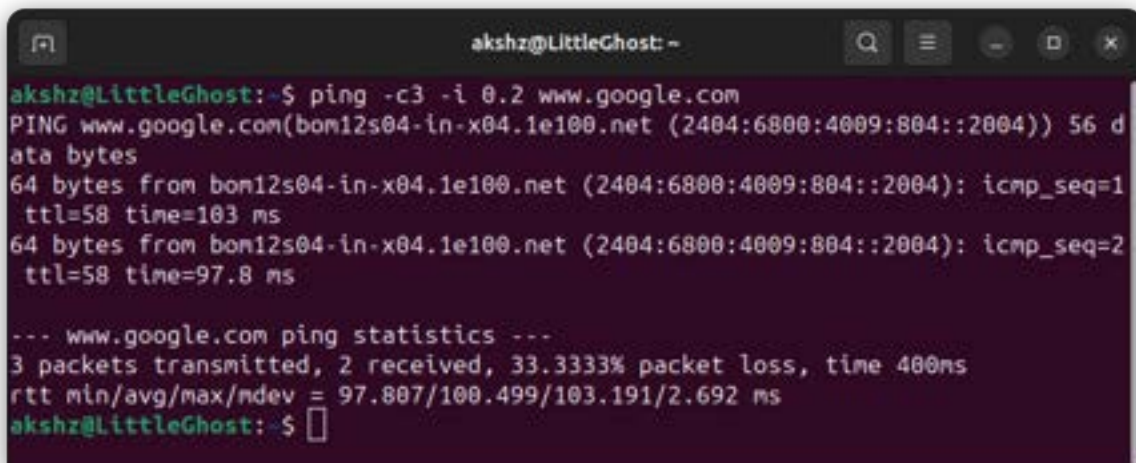
a) In Unix or GNU/Linux versions, the option to specify the number of echo requests with the ping command is -c. For example:

ping -c 2 www.instagram.com



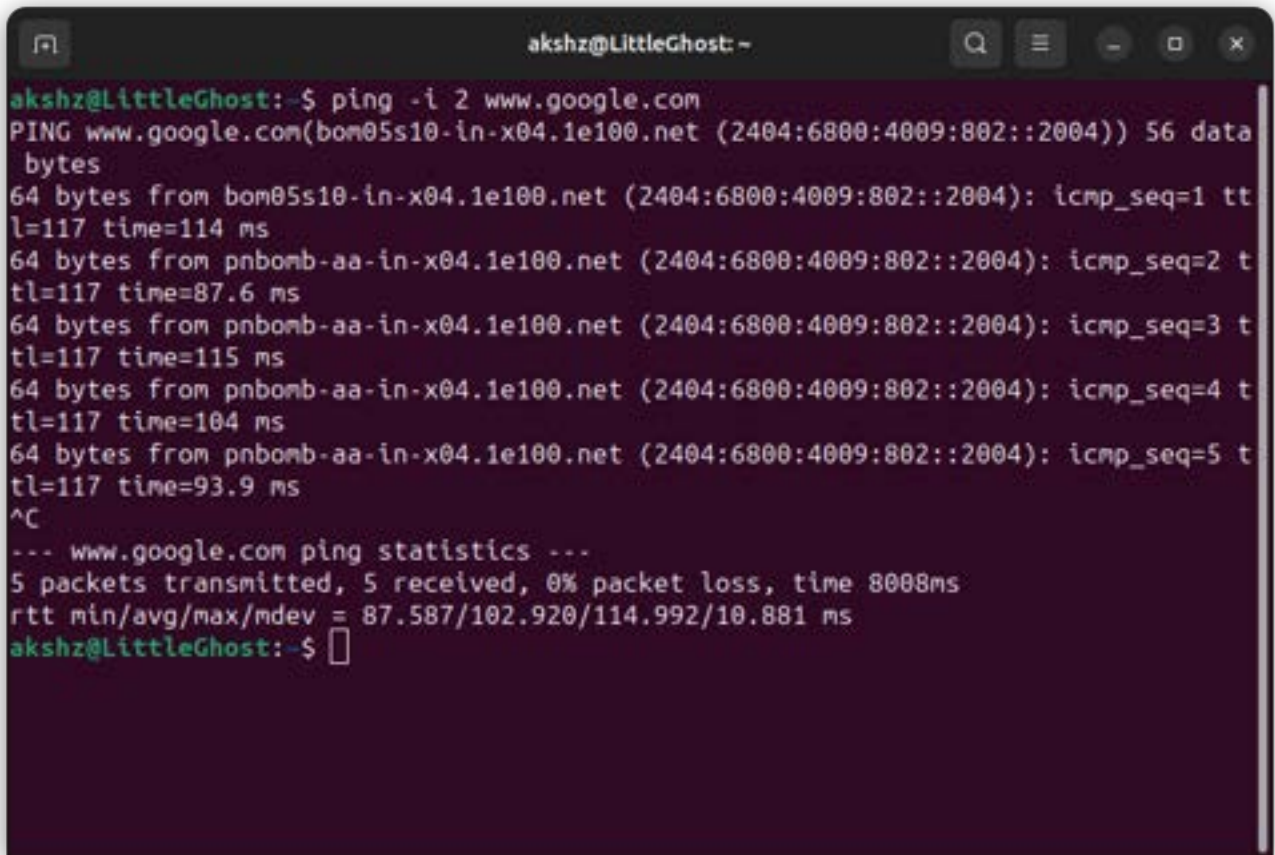
```
akshz@LittleGhost: ~  
akshz@LittleGhost:~$ ping -c 2 www.instagram.com  
PING www.instagram.com(instagram-p426-shv-01-ccu1.fbcdn.net (2a03:2880:f26b:e6:face:b00c:0:4420)) 56 data bytes  
64 bytes from instagram-p426-shv-01-ccu1.fbcdn.net (2a03:2880:f26b:e6:face:b00c:0:4420): icmp_seq=1 ttl=56 time=90.5 ms  
64 bytes from instagram-p426-shv-01-ccu1.fbcdn.net (2a03:2880:f26b:e6:face:b00c:0:4420): icmp_seq=2 ttl=56 time=70.1 ms  
  
--- www.instagram.com ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1001ms  
rtt min/avg/max/mdev = 70.107/80.290/90.473/10.183 ms  
akshz@LittleGhost:~$
```

c)The command to send ECHO\_REQUEST packets to the destination one after another without waiting for a reply is achieved using the -c and -i option. However, normal users typically have restrictions on using flood ping. To execute flood ping, the user usually needs superuser (root) privileges. The minimum interval for sending ICMP ECHO\_REQUEST packets for users other than the super-user is 200 milliseconds (0.2 Second). For ECHO\_REPLY in general it takes 1 second to come.



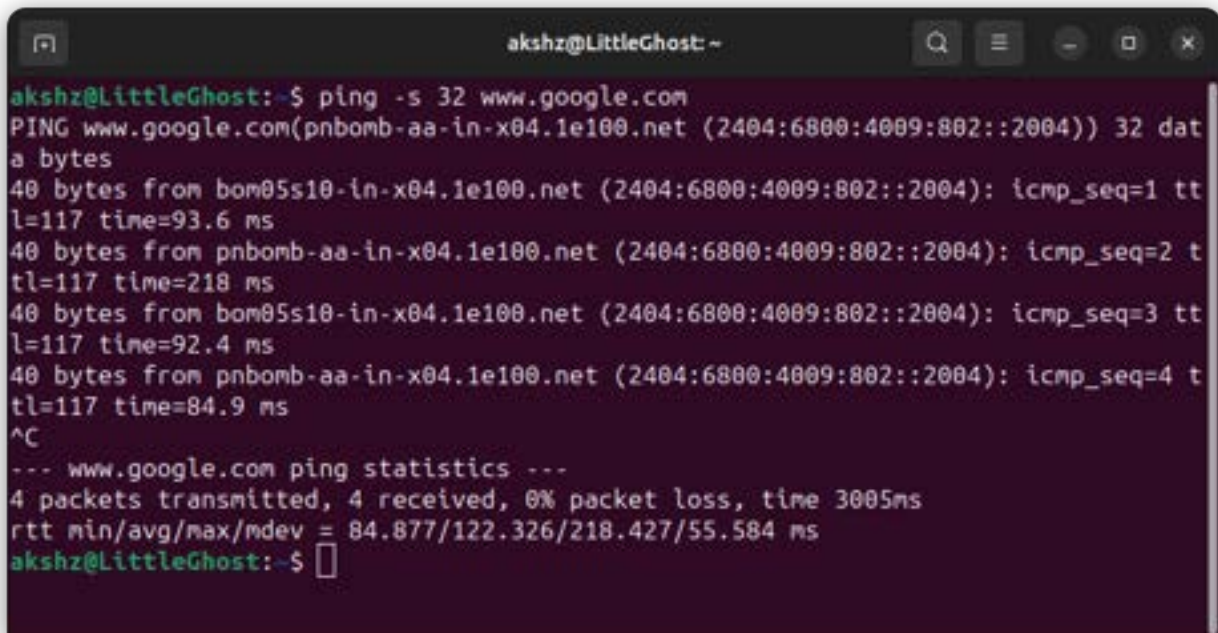
```
akshz@LittleGhost: ~  
akshz@LittleGhost:~$ ping -c3 -i 0.2 www.google.com  
PING www.google.com(bom12s04-in-x04.1e100.net (2404:6800:4009:804::2004)) 56 data bytes  
64 bytes from bom12s04-in-x04.1e100.net (2404:6800:4009:804::2004): icmp_seq=1 ttl=58 time=103 ms  
64 bytes from bom12s04-in-x04.1e100.net (2404:6800:4009:804::2004): icmp_seq=2 ttl=58 time=97.8 ms  
  
--- www.google.com ping statistics ---  
3 packets transmitted, 2 received, 33.3333% packet loss, time 400ms  
rtt min/avg/max/mdev = 97.807/100.499/103.191/2.692 ms  
akshz@LittleGhost:~$
```

- b) The option to set the time interval between two successive ping ECHO\_REQUESTs is -i. For example:  
ping -i 2 www.google.com



```
akshz@LittleGhost: ~  
akshz@LittleGhost:~$ ping -i 2 www.google.com  
PING www.google.com(bom05s10-in-x04.1e100.net (2404:6800:4009:802::2004)) 56 data  
bytes  
64 bytes from bom05s10-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=1 tt  
l=117 time=114 ms  
64 bytes from pnbomb-aa-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=2 t  
tl=117 time=87.6 ms  
64 bytes from pnbomb-aa-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=3 t  
tl=117 time=115 ms  
64 bytes from pnbomb-aa-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=4 t  
tl=117 time=104 ms  
64 bytes from pnbomb-aa-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=5 t  
tl=117 time=93.9 ms  
^C  
--- www.google.com ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 8008ms  
rtt min/avg/max/mdev = 87.587/102.920/114.992/10.881 ms  
akshz@LittleGhost:~$
```

- d) The command to set the ECHO\_REQUEST packet size in bytes is -s. For example:  
ping -s 32 www.google.com



```
akshz@LittleGhost: ~  
akshz@LittleGhost:~$ ping -s 32 www.google.com  
PING www.google.com(pnbomb-aa-in-x04.1e100.net (2404:6800:4009:802::2004)) 32 dat  
a bytes  
40 bytes from bom05s10-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=1 tt  
l=117 time=93.6 ms  
40 bytes from pnbomb-aa-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=2 t  
tl=117 time=218 ms  
40 bytes from bom05s10-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=3 tt  
l=117 time=92.4 ms  
40 bytes from pnbomb-aa-in-x04.1e100.net (2404:6800:4009:802::2004): icmp_seq=4 t  
tl=117 time=84.9 ms  
^C  
--- www.google.com ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3005ms  
rtt min/avg/max/mdev = 84.877/122.326/218.427/55.584 ms  
akshz@LittleGhost:~$
```

If the packet size is set to 32 bytes, the total packet size will be larger than 32 bytes due to the additional ICMP header and any other potential overhead. The ICMP header is 8 bytes, so the total packet size would be 40 bytes (32 bytes data + 8 bytes ICMP header).

## Q2.

I have pinged the following 6 servers and made this chart.

Server Name	Time	RTT (ms)	Packet loss %
<a href="http://www.apple.com">www.apple.com</a> (Germany)	11:00 AM	308.262	0
	04:30 PM	275.459	0
	11:00 PM	254.170	0
<a href="http://www.premierleague.com">www.premierleague.com</a> (US)	11:00 AM	176.475	0
	04:30 PM	478.376	0
	11:00 PM	243.807	0
<a href="http://www.indiansuperleague.com">www.indiansuperleague.com</a> (Europe france)	11:00 AM	307.076	0
	04:30 PM	262.682	0
	11:00 PM	347.117	0
<a href="http://www.flipkart.com">www.flipkart.com</a> (India)	11:00 AM	187.720	0
	04:30 PM	250.730	0
	11:00 PM	232.590	0
<a href="http://www.espn.com">www.espn.com</a> (Netherland)	11:00 AM	241.721	0
	04:30 PM	272.851	0
	11:00 PM	231.613	0
<a href="http://www.youtube.com">www.youtube.com</a> (US)	11:00 AM	312.617	0
	04:30 PM	322.667	8
	11:00 PM	181.085	0

**Packet Loss:** There was packet loss in some of the cases (www.youtube.com) , it may have caused due to:

- **Network Congestion:** Busy networks may drop packets during peak usage, prioritizing some over others.
- **Faulty Network Equipment:** Broken routers or switches can lead to packet loss due to hardware issues or misconfigurations.
- **Wireless Interference:** Interference from devices or obstacles in wireless networks can disrupt packet transmission.
- **Jitter and Latency:** Variations in network latency may cause packets to arrive at different times, resulting in losses.
- **Congestion at Intermediate Nodes:** Routers or switches along the data path may drop packets if overwhelmed by incoming traffic.
- **Buffer Overflows:** Full buffers in network devices can cause the dropping of subsequent packets during congestion.
- **Firewall or Security Measures:** Strict security rules may intentionally drop packets as part of network protection

### Relation of RTT and geographical distance

From my experiments, it became evident that there is a weak correlation between measured Round-Trip Times (RTTs) and geographical distance. From what I observed, the time it takes for data to go back and forth (RTT) doesn't strongly depend on how far the locations are. It's not just about physical distance; there are other things like the complicated setup of the internet, how it decides the route for data, and the devices it goes through, like routers. These factors make the connection between RTT and the distance between places not very strong.

### Relation of RTT and packet size:

Size of Packet	Avg RTT (ms)
64	173
128	255
256	259
384	273
512	291
1024	363
1600	rejected

The host used for testing is the www.youtube.com server with the IP address 209.85.202.138, located in California, US. Packet sizes were varied from 64 bytes to 1500 bytes, with requests beyond 1500 bytes being rejected by the ISP. Despite the variations in packet size, there is no clear trend observed in the ping latency. However, a slight increase in the time taken can be noted as the packet size increases. This suggests that packet size and ping latency have a weak relationship.

### Relation of RTT and time of the day

Through my experiments, I observed that the Round-Trip Time (RTT) is influenced by the time of day. During peak hours, there was an increase in RTTs, indicating higher network congestion. Conversely, during off-peak hours, RTTs were lower, suggesting reduced contention for network resources.

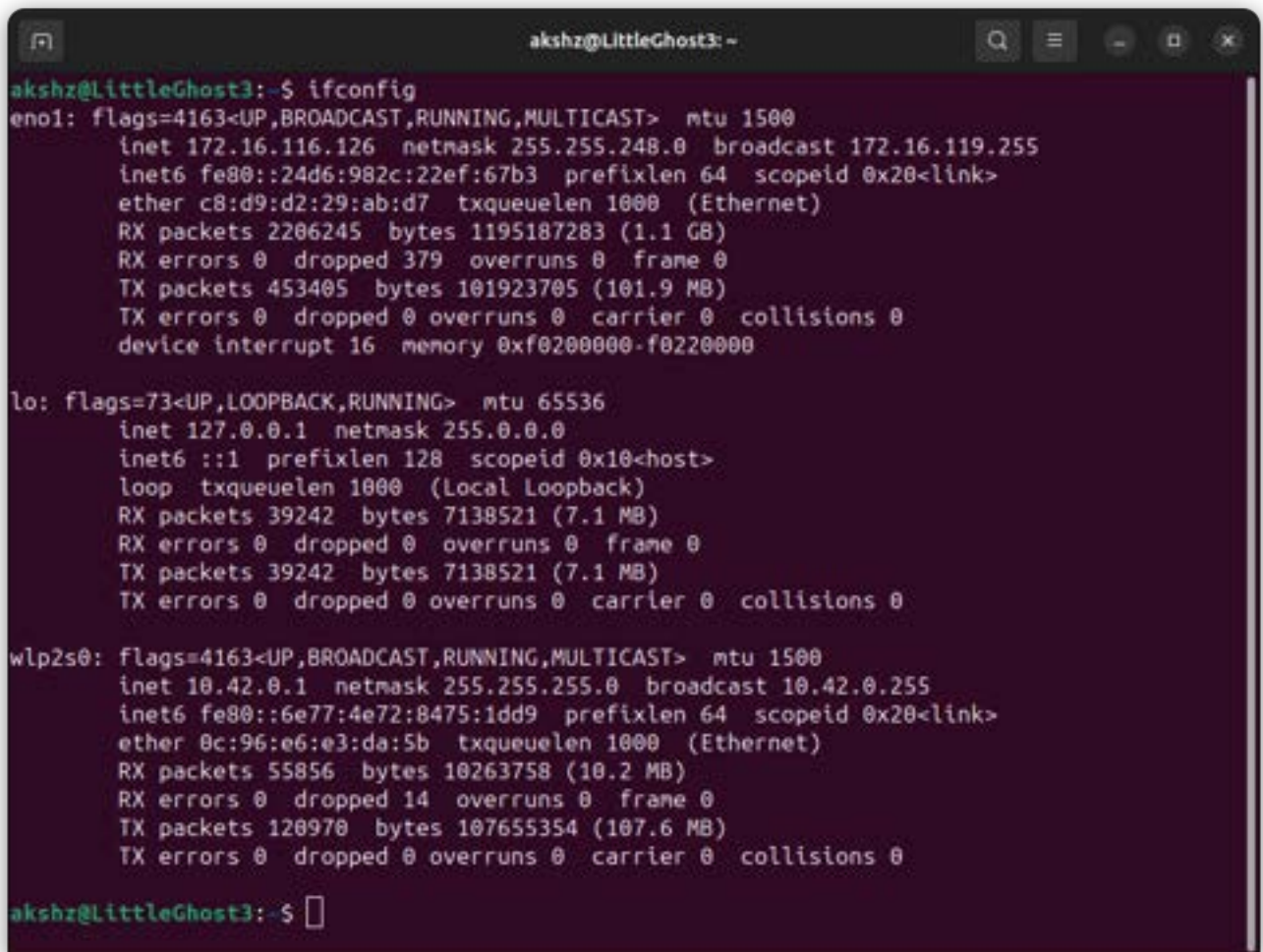
---



### Q3. ifconfig and route commands

The '**ifconfig**' command is a tool used in Unix-like operating systems to provide information about network interfaces on a device. It helps users inspect and configure network settings, offering details on interface names, statuses (UP or DOWN), hardware (MAC) addresses, and assigned IP addresses. By displaying key network metrics such as packet transmission and errors, users can assess the health and activity of their network interfaces. This command is valuable for network troubleshooting, configuration adjustments, and gaining insights into the overall network connectivity of a system.

#### a) ifconfig command and its output



```

akshz@LittleGhost3: ~
akshz@LittleGhost3:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.116.126 netmask 255.255.248.0 broadcast 172.16.119.255
    inet6 fe80::24d6:982c:22ef:67b3 prefixlen 64 scopeid 0x20<link>
    ether c8:d9:d2:29:ab:d7 txqueuelen 1000 (Ethernet)
    RX packets 2206245 bytes 1195187283 (1.1 GB)
    RX errors 0 dropped 379 overruns 0 frame 0
    TX packets 453405 bytes 101923705 (101.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xf0200000-f0220000

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 39242 bytes 7138521 (7.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 39242 bytes 7138521 (7.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.42.0.1 netmask 255.255.255.0 broadcast 10.42.0.255
    inet6 fe80::6e77:4e72:8475:1dd9 prefixlen 64 scopeid 0x20<link>
    ether 0c:96:e6:e3:da:5b txqueuelen 1000 (Ethernet)
    RX packets 55856 bytes 10263758 (10.2 MB)
    RX errors 0 dropped 14 overruns 0 frame 0
    TX packets 120970 bytes 107655354 (107.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

akshz@LittleGhost3:~$
  
```

In the report, the `ifconfig` command was used to gather information about the network interfaces. Here's a breakdown of the output:

#### Interface Information:

The output shows details about each network interface. For example, if there's an Ethernet connection, it might be labeled as "eth0."

It provides information like the interface's status (whether it's active or inactive), its hardware (MAC) address, and assigned IP addresses.

**Link Status:**

You can quickly see if an interface is UP (active) or DOWN (inactive). Additionally, it gives stats on the number of packets sent and received.

**IP Address Information:**

The assigned IP addresses (both IPv4 and IPv6) are displayed for each active interface. You'll also find details about the subnet mask and broadcast address, giving insights into the network configuration.

**MTU (Maximum Transmission Unit):**

The MTU value is included, which tells us the maximum size of a data packet that can be sent over the network.

**Network-related Statistics:**

Statistics related to network traffic, errors, and other metrics are provided. This includes the number of packets transmitted and received.

**Additional Information:**

Depending on the system, extra details may be given, such as media type, transmission speed, and network driver information.

**Loopback Interface (lo):**

The loopback interface (lo) is special and is used for internal communication within the system. It usually has the IP address 127.0.0.1.

This information helps in understanding the current state and configuration of the network interfaces on the system.

**b) some(4) options provided with the ifconfig command****1) ifconfig <interface>:**

This basic form of the command followed by the name of a specific network interface (e.g., eth0 or wlan0) displays detailed information about that particular interface.

Example: ifconfig eno1

```

akshz@LittleGhost3: ~
akshz@LittleGhost3:~$ ifconfig eno1
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.116.126 netmask 255.255.248.0 broadcast 172.16.119.255
    inet6 fe80::24d6:982c:22ef:67b3 prefixlen 64 scopeid 0x20<link>
    ether c8:d9:d2:29:ab:d7 txqueuelen 1000 (Ethernet)
    RX packets 2217767 bytes 1198355159 (1.1 GB)
    RX errors 0 dropped 379 overruns 0 frame 0
    TX packets 459469 bytes 105240583 (105.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xf0200000-f0220000

akshz@LittleGhost3:~$ 

```

2) ifconfig <interface> up and ifconfig <interface> down:

The up option activates a network interface, enabling it to send and receive data. Conversely, the down option deactivates the interface, preventing it from participating in network communication.

Examples: ifconfig eno1 up

ifconfig eno2 down

You can observe the flag changes after running these 2 commands.

```

akshz@LittleGhost3: ~
akshz@LittleGhost3:~$ sudo ifconfig eno1 down
akshz@LittleGhost3:~$ ifconfig eno1
eno1: flags=4098<BROADCAST,MULTICAST> mtu 1000
    ether c8:d9:d2:29:ab:d7 txqueuelen 1000 (Ethernet)
    RX packets 2229062 bytes 1203424862 (1.2 GB)
    RX errors 0 dropped 379 overruns 0 frame 0
    TX packets 466096 bytes 108635084 (108.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xf0200000-f0220000

akshz@LittleGhost3:~$ sudo ifconfig eno1 up
akshz@LittleGhost3:~$ ifconfig eno1
eno1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1000
    ether c8:d9:d2:29:ab:d7 txqueuelen 1000 (Ethernet)
    RX packets 2229062 bytes 1203424862 (1.2 GB)
    RX errors 0 dropped 379 overruns 0 frame 0
    TX packets 466096 bytes 108635084 (108.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xf0200000-f0220000

akshz@LittleGhost3:~$ 

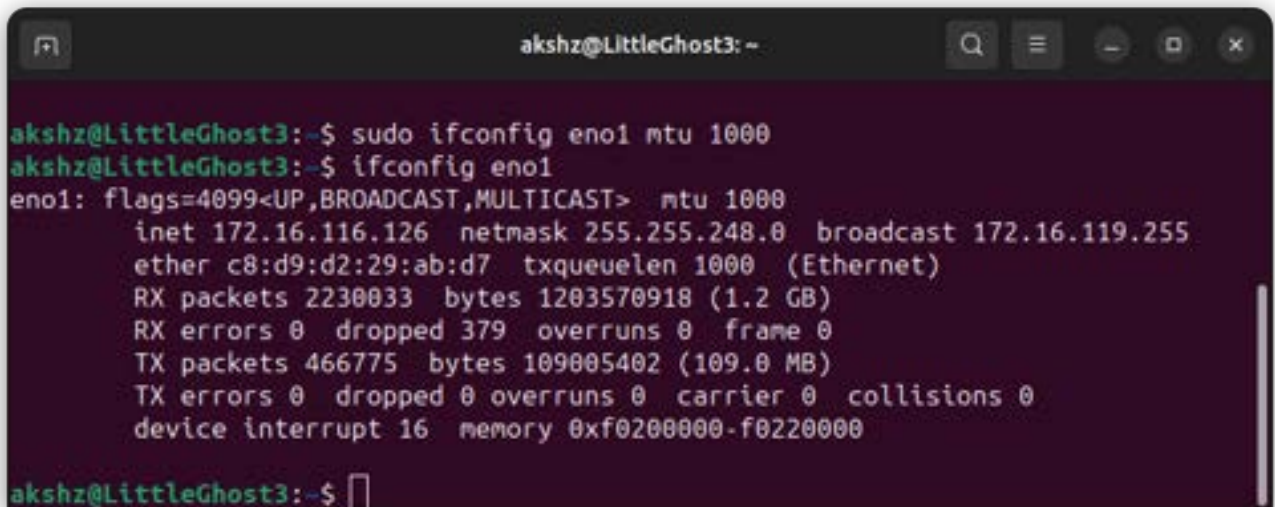
```



### 3) ifconfig <interface> mtu <new\_MTU\_value>:

This option allows you to set the Maximum Transmission Unit (MTU) for a network interface. The MTU represents the maximum size of a data packet that can be transmitted over the network.

Example: `ifconfig eno1 mtu 1500`



```

akshz@LittleGhost3:~$ sudo ifconfig eno1 mtu 1000
akshz@LittleGhost3:~$ ifconfig eno1
eno1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1000
    inet 172.16.116.126 netmask 255.255.248.0 broadcast 172.16.119.255
    ether c8:d9:d2:29:ab:d7 txqueuelen 1000 (Ethernet)
    RX packets 2230033 bytes 1203570918 (1.2 GB)
    RX errors 0 dropped 379 overruns 0 frame 0
    TX packets 466775 bytes 109005402 (109.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xf0200000-f0220000
akshz@LittleGhost3:~$

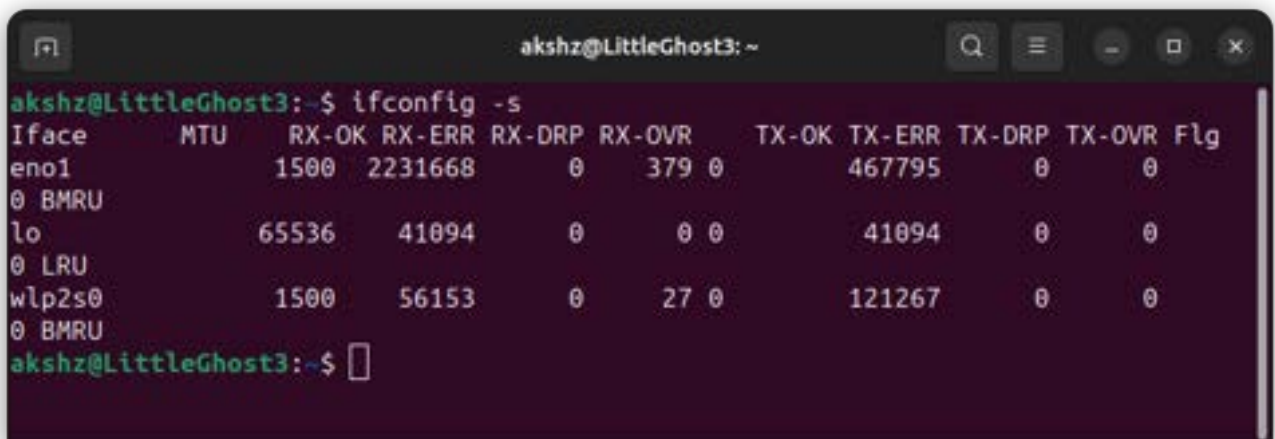
```

### 4) ifconfig -s:

This option is used to display a concise summary of network interfaces. It provides a simplified tabular overview that includes information such as interface names, MTU (Maximum Transmission Unit) values, and the count of transmitted and received packets.

Example: `ifconfig -s`

Using `ifconfig -s` is particularly useful when you want a quick overview of multiple interfaces and their basic statistics. It offers a summary that is easier to interpret at a glance.



```

akshz@LittleGhost3:~$ ifconfig -s
Iface      MTU      RX-OK RX-ERR RX-DRP RX-OVR      TX-OK TX-ERR TX-DRP TX-OVR Flg
eno1       1500     2231668 0      379 0      467795 0      0
0 BMRU
lo         65536     41094 0      0 0      41094 0      0
0 LRU
wlp2s0     1500     56153 0      27 0      121267 0      0
0 BMRU
akshz@LittleGhost3:~$

```

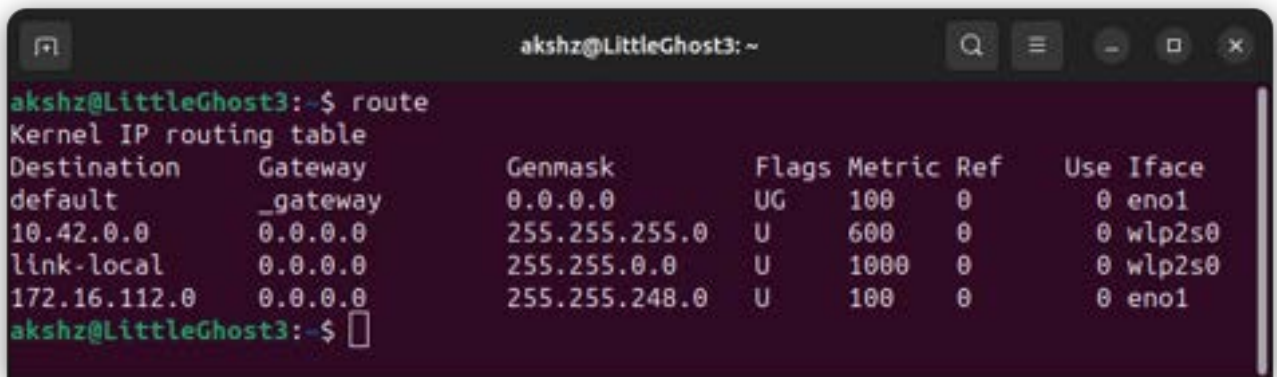
## c) Output of route command

The `route` command displays the Kernel IP routing table on a Unix-like operating system. Here's an explanation of the key components of the output:

Table Columns:

- Destination: Represents the destination network or IP address to which the route applies.
- Gateway: Specifies the next hop, or gateway, for reaching the destination. If the destination is on the local network, this may be set to 0.0.0.0.
- Genmask: Denotes the subnet mask associated with the destination network, indicating the range of IP addresses covered by the route.
- Flags: Indicates various flags that provide additional information about the route. Common flags include U (route is up), G (gateway), and H (target is a host).
- Metric: Represents a metric value that reflects the cost of the route. Lower metric values generally indicate more preferred routes.
- Ref: The reference count indicates the number of routes that share the same gateway.
- Use: Shows how many times the route has been used. This count can be useful in determining the activity or importance of a route.
- Iface: Specifies the network interface associated with the route. This is the interface through which the data is sent to reach the destination.

When I ran the route command, I observed the following routing table:



```

akshz@LittleGhost3: ~
akshz@LittleGhost3:~$ route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use    Iface
default          _gateway       0.0.0.0         UG    100    0      0      eno1
10.42.0.0        0.0.0.0        255.255.255.0   U     600    0      0      wlp2s0
link-local       0.0.0.0        255.255.0.0     U     1000   0      0      wlp2s0
172.16.112.0     0.0.0.0        255.255.248.0   U     100    0      0      eno1
akshz@LittleGhost3:~$

```

#### Default Route:

The default route (0.0.0.0) directs all traffic not matching any specific route to the gateway (\_gateway) through the interface eno1. The UG flag indicates it's a default route.

#### Local Network Routes:

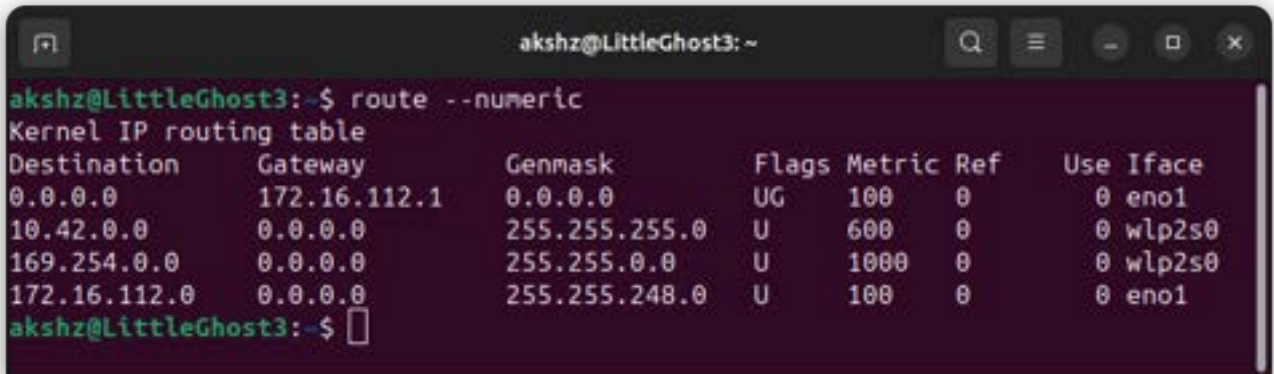
There are specific routes for local networks:

- The destination 10.42.0.0 with a subnet mask of 255.255.255.0 is reachable through the interface wlp2s0.
- The destination "link-local" with a subnet mask of 255.255.0.0 is also reachable through the interface wlp2s0.
- The destination 172.16.112.0 with a subnet mask of 255.255.248.0 is accessible through the interface eno1.

#### d) Four options of the route command.

##### 1) route -n or route --numeric:

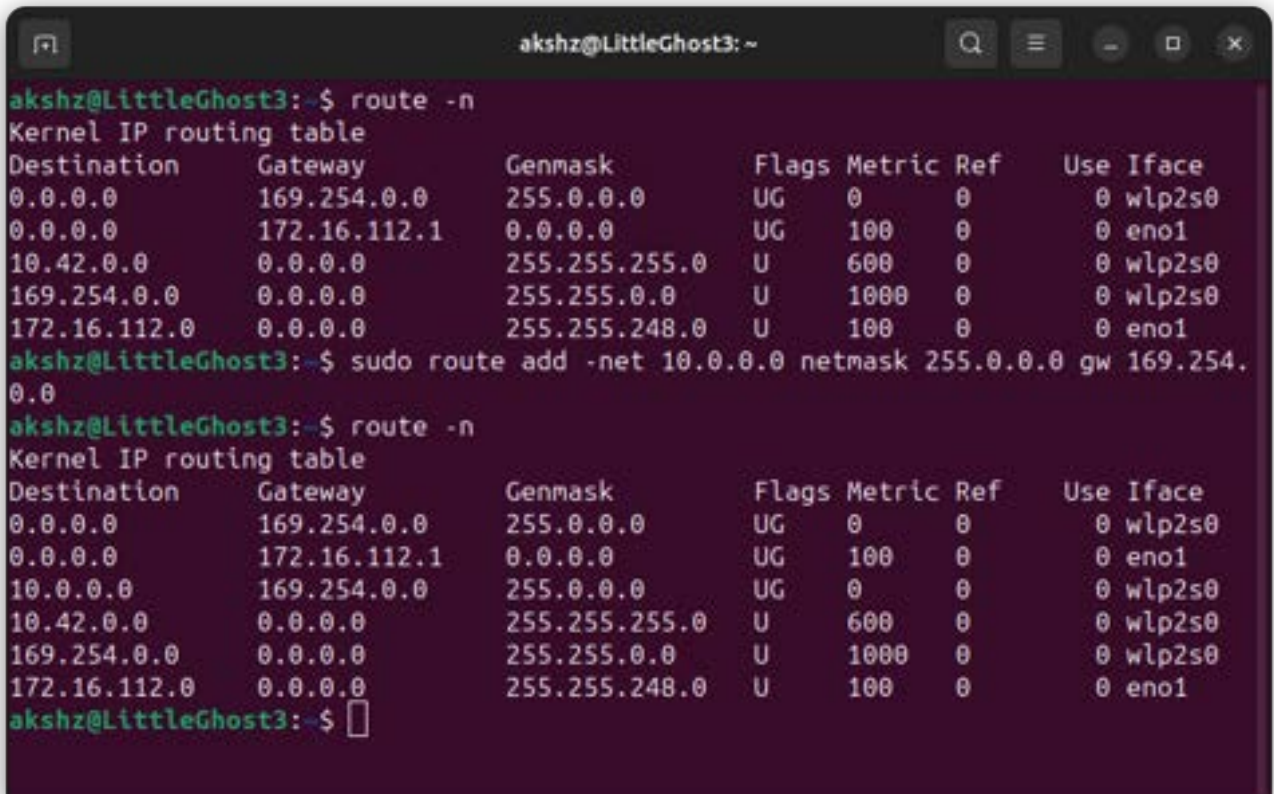
This option displays the numerical IP addresses in the output without attempting to resolve hostnames. It provides a faster display and is often used when you don't need to see hostnames.



```
akshz@LittleGhost3:~$ route --numeric
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          172.16.112.1    0.0.0.0          UG    100    0      0 eno1
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 wlp2s0
172.16.112.0     0.0.0.0         255.255.248.0    U     100    0      0 eno1
akshz@LittleGhost3:~$
```

##### 2) route add <destination> gw <gateway>:

This option is used to add a new route to the routing table. It specifies the destination network or host and the gateway through which the traffic should be routed.



```
akshz@LittleGhost3:~$ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          169.254.0.0     255.0.0.0        UG    0      0      0 wlp2s0
0.0.0.0          172.16.112.1    0.0.0.0          UG    100    0      0 eno1
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 wlp2s0
172.16.112.0     0.0.0.0         255.255.248.0    U     100    0      0 eno1
akshz@LittleGhost3:~$ sudo route add -net 10.0.0.0 netmask 255.0.0.0 gw 169.254.0.0
akshz@LittleGhost3:~$ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          169.254.0.0     255.0.0.0        UG    0      0      0 wlp2s0
0.0.0.0          172.16.112.1    0.0.0.0          UG    100    0      0 eno1
10.0.0.0         169.254.0.0     255.0.0.0        UG    0      0      0 wlp2s0
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 wlp2s0
172.16.112.0     0.0.0.0         255.255.248.0    U     100    0      0 eno1
akshz@LittleGhost3:~$
```



### 3) route del <destination>:

This option is used to delete a route from the routing table. It specifies the destination network or host that should be removed from the routing table.

```

akshz@LittleGhost3: ~
akshz@LittleGhost3:~$ route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          169.254.0.0    255.0.0.0      UG      0      0      0 wlp2s0
0.0.0.0          172.16.112.1   0.0.0.0        UG      100    0      0 eno1
10.0.0.0         169.254.0.0    255.0.0.0      UG      0      0      0 wlp2s0
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 wlp2s0
172.16.112.0     0.0.0.0        255.255.248.0  U       100    0      0 eno1
akshz@LittleGhost3:~$ sudo route del -net 10.0.0.0 netmask 255.0.0.0
akshz@LittleGhost3:~$ route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          169.254.0.0    255.0.0.0      UG      0      0      0 wlp2s0
0.0.0.0          172.16.112.1   0.0.0.0        UG      100    0      0 eno1
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 wlp2s0
172.16.112.0     0.0.0.0        255.255.248.0  U       100    0      0 eno1
akshz@LittleGhost3:~$ 

```

### 4) route -e

This option displays extended information, including additional details about the route, such as the reference count, use count, and flags.

```

akshz@LittleGhost3: ~
akshz@LittleGhost3:~$ route -e
Kernel IP routing table
Destination      Gateway         Genmask         Flags  MSS Window  irtt Iface
0.0.0.0          169.254.0.0    255.0.0.0      UG      0  0      0 wlp2s0
default          _gateway       0.0.0.0        UG      0  0      0 eno1
10.42.0.0        0.0.0.0        255.255.255.0  U       0  0      0 wlp2s0
link-local       0.0.0.0        255.255.0.0    U       0  0      0 wlp2s0
172.16.112.0     0.0.0.0        255.255.248.0  U       0  0      0 eno1
akshz@LittleGhost3:~$ 

```



**Q 4. Answer the following questions related to netstat command.**

**a) What is the command netstat used for?**

The netstat command is a command-line network tool that displays network status and provides detailed information about how a computer communicates with other computers or network devices.

**b) What parameters for netstat should you use to show all the established TCP connections? Include a screenshot of this list for your computer and explain all the fields of the table in the output.**

To show all established TCP connection, you can use the following command:

**Command:** `netstat -an | grep "Established"`

**Output:**

```
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ netstat -an | grep -l "tcp" | grep "ESTABLISHED"
tcp        0      0 172.16.118.135:50428 172.217.194.188:5228 ESTABLISHED
tcp        0      0 172.16.118.135:42990 142.250.76.174:443  ESTABLISHED
tcp        0      0 172.16.118.135:48138 35.190.80.1:443    ESTABLISHED
tcp        0      0 172.16.118.135:52058 34.120.214.181:443 ESTABLISHED
tcp        0      0 172.16.118.135:52074 34.120.214.181:443 ESTABLISHED
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

**c) What does "netstat -r" show? Explain all the fields of the output.**

The netstat -r command shows the kernel routing table.

**Command:** `netstat -r`

**Output:**

```
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ netstat -r
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default _gateway 0.0.0.0 UG 0 0 0 eno1
link-local 0.0.0.0 255.255.0.0 U 0 0 0 eno1
172.16.112.0 0.0.0.0 255.255.248.0 U 0 0 0 eno1
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

**Fields in the output include:**

- (i) **Destination:** Destination network or IP address.
- (ii) **Gateway:** Next-hop gateway for the destination.
- (iii) **Genmask:** Network mask.
- (iv) **Flags:** Routing flags (e.g., U for Up, G for Gateway).
- (v) **MSS:** It talks about the maximum segment size of TCP connection along this route.
- (vi) **Window:** The tcp window size along this route.
- (vii) **Irtt:** The initial round trip time used.
- (viii) **Iface:** Network interface associated with the route.

**d) What option of netstat can be used to display the status of all network interfaces?**

**By using netstat, figure out the number of interfaces on your computer.**

To display the status of all network interfaces, you can use the following command:

**Command:** netstat -i

**Output:**

```
172.16.112.0 0.0.0.0 255.255.248.0 0 0 0 eno1
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ netstat -i
Kernel Interface table
Iface MTU RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eno1 1500 2080070 0 30 0 436982 0 0 0 BMRU
lo 65536 44847 0 0 0 44847 0 0 0 LRU
wlp2s0 1500 0 0 0 0 0 0 0 0 BMU
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

The number of interfaces of our computer is 2,

- (i) **eno1**: onboard Ethernet (wired) adapter.
- (ii) **lo**: loopback device.

e) What option of netstat can be used to show the statistics of all UDP connections?

Run the command for this purpose on your computer and show the output.

This command provides statistics for various UDP-related parameters.

**Command:** netstat -su

**Output:**

```
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ netstat -su
IcmpMsg:
  InType3: 687
  OutType3: 785
Udp:
  86405 packets received
  748 packets to unknown port received
  0 packet receive errors
  97595 packets sent
  0 receive buffer errors
  81 send buffer errors
  IgnoredMulti: 9528
UdpLite:
IpExt:
  InMcastPkts: 1264
  OutMcastPkts: 9212
  InBcastPkts: 9529
  InOctets: 339501262
  OutOctets: 89746437
  InMcastOctets: 266087
  OutMcastOctets: 1818432
  InBcastOctets: 3620480
  InNoECTPkts: 526452
  InECT0Pkts: 5
MPTcpExt:
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

f) Show and explain the function of loopback interface.

It is a virtual interface, which is always active after it has been configured. For this, we assign a special IP Address to our system for testing purposes which is called loop back address

**Purpose:**

- It is used for testing the network stack of our device without relying on external interfaces
- It is used to test whether the server or service is working correctly.
- It is used in web development purpose for testing the websites on the server

```

172.16.112.0 0.0.0.0 255.255.248.0 0 0 0 eno1
aditya@aditya-HP-ProDesk-600-G4-PCI-MT: ~$ netstat -i
Kernel Interface table
Iface      MTU      RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eno1       1500    2080070      0      30 0      436982      0      0      0 BMRU
lo         65536    44847      0      0 0      44847      0      0      0 LRU
wlp2s0     1500      0      0      0 0          0      0      0      0 BMU
aditya@aditya-HP-ProDesk-600-G4-PCI-MT: ~$

```

## Q5. traceroute

The 'traceroute' tool is used to analyze and diagnose the route that data packets take from a source to a destination on a network. It helps identify network delays, diagnose packet loss, debug connectivity issues, check routing configurations, and reveal the network path and performance characteristics. It's a valuable tool for network troubleshooting and topology mapping.

a)

Server Name	Time	Hop Count
<a href="http://www.apple.com">www.apple.com</a> (Germany)	06:00 PM	No path
	02:00 PM	No path
	11:00 PM	No path
<a href="http://www.premierleague.com">www.premierleague.com</a> (US)	06:00 PM	16
	02:00 PM	17
	11:00 PM	17
<a href="http://www.indiansuperleague.com">www.indiansuperleague.com</a> (Europe france)	06:00 PM	8
	02:00 PM	9
	11:00 PM	9
<a href="http://www.flipkart.com">www.flipkart.com</a> (India)	06:00 PM	No path
	02:00 PM	No path
	11:00 PM	No path
<a href="http://www.espncricinfo.com">www.espncricinfo.com</a> (Netherland)	06:00 PM	8
	02:00 PM	9
	11:00 PM	9

<a href="http://www.youtube.com">www.youtube.com</a> (US)	06:00 PM	12
	02:00 PM	13
	11:00 PM	13

The initial hops (Hops 1 to 5) are common between both routes, representing local network and Airtel's network.

Hop 1: \_gateway (192.168.94.70)

Hop 2: 192.168.29.10

Hop 3: 192.168.28.65

Hop 4: 192.168.31.10

Hop 5: 192.168.31.33

The common hops between the two routes are Hops 1 to 5 and Hop 7. The routes start diverging beyond Hop 7. The differences in the subsequent hops indicate different paths taken to reach the respective destinations.

b) I checked and noticed that the route to the same place can switch during the day. This happens because of things like the way the network adjusts itself, spreads out internet traffic evenly, deals with busy times, and follows different rules at different hours. It's like the network figuring out the best path to take depending on what's happening at that moment.

c) During my testing for Flipkart and Apple server using traceroute, I observed instances where the traceroute did not complete successfully. The absence of responses marked as "\*" \* \*" at certain hops suggests that the routers at those points did not respond to the ICMP requests sent by the traceroute tool. This can occur due to various reasons, including firewall or router configurations, load balancers not responding, network filtering, routing policies prioritizing certain traffic, destination router configurations, or temporary network issues such as congestion or failures. It's worth noting that the lack of responses at specific hops may not necessarily indicate a problem, as some routers choose not to respond for security or operational reasons. The crucial aspect is that the traceroute eventually reaches the final destination.

d) Certainly! We can determine the route to a host even if ping is unresponsive. While ping is a straightforward ICMP communication from source to destination, traversing networks based on routing rules, `traceroute` employs a distinct methodology. Although both tools use ICMP, they differ significantly in their approach.

'Traceroute' functions by targeting the final destination but restricts the Time To Live (TTL) for each packet. It waits for a "time exceeded" message, then increments the TTL for the next attempt. Consequently, the received response is not an ICMP echo reply directly from the host but rather a "time exceeded" message from the intermediate host. Despite utilizing ICMP, `traceroute` operates in a unique manner, offering insights into the network path through which the packets travel.



**Q6. Answer the following questions regarding network addresses.**

- a) How do you show the full ARP table for your machine? Explain each column of the ARP table.

**Command:** `arp -a`

```
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ arp -a
? (172.16.112.30) at 00:03:0f:1d:ab:f0 [ether] on eno1
? (172.16.112.31) at 00:03:0f:1a:fc:ea [ether] on eno1
? (172.16.112.28) at 00:03:0f:1d:ab:ec [ether] on eno1
? (172.16.112.27) at 00:03:0f:1d:ab:c4 [ether] on eno1
gateway (172.16.112.1) at f8:0b:cb:cb:49:e4 [ether] on eno1
? (172.16.112.54) at 00:03:0f:1a:ff:70 [ether] on eno1
? (172.16.112.50) at 00:03:0f:1b:61:02 [ether] on eno1
? (172.16.112.49) at 00:03:0f:1b:61:22 [ether] on eno1
? (172.16.112.51) at 00:03:0f:1b:60:bc [ether] on eno1
? (172.16.112.58) at 00:03:0f:1a:fd:00 [ether] on eno1
? (172.16.112.34) at 00:03:0f:1d:ab:fe [ether] on eno1
? (172.16.112.37) at 00:03:0f:1d:ab:32 [ether] on eno1
? (172.16.112.39) at 00:03:0f:1a:ff:c4 [ether] on eno1
? (172.16.112.33) at 00:03:0f:1d:ab:58 [ether] on eno1
? (172.16.112.35) at 00:03:0f:1d:ab:1a [ether] on eno1
? (172.16.112.46) at 00:03:0f:1d:ac:10 [ether] on eno1
? (172.16.112.42) at 00:03:0f:1d:ab:f6 [ether] on eno1
? (172.16.112.44) at 00:03:0f:1d:ac:0c [ether] on eno1
? (172.16.112.45) at 00:03:0f:1d:ac:0e [ether] on eno1
? (172.16.112.47) at 00:03:0f:1d:ab:36 [ether] on eno1
? (172.16.112.43) at 00:03:0f:1a:fd:06 [ether] on eno1
? (172.16.112.66) at 00:03:0f:1a:fc:38 [ether] on eno1
? (172.16.118.133) at 84:a9:3e:8a:8b:b0 [ether] on eno1
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

The Arp cache entry shows the following information –

`<hostname>< (IP Address)> at<MAC address><protocol type>on<network interface>`

- (i) Hostname: It indicates the hostname, if not available print “?”.
- (ii) IP Address: It indicates the ip address of the host.
- (iii) Mac Address: It indicates the hardware address.
- (iv) Protocol type: It indicates the type of protocol, generally we use eth which denotes ethernet protocol.
- (v) Network Interface: It indicates the network interface on which the host is connected.

- b) Check and explain what happens if you try and use the ARP command to add or delete an entry to the ARP table. Find out how to add, delete or change entries in the ARP table. Use this mechanism to add at least four new hosts to the ARP table and include a printout.

**Adding an ARP Entry –**

**Command:** `sudo arp -i eno1 -s 192.168.1.1 00:11:22:33:44:55`

```
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ sudo arp -i eno1 -s 192.168.1.1 00:11:22:33:44:55
[sudo] password for aditya:
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ sudo arp -i eno1 -s 192.168.1.1 00:11:22:33:44:55
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ sudo arp -i eno1 -s 192.168.1.1 00:11:22:33:44:55
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ sudo arp -i eno1 -s 192.168.1.1 00:11:22:33:44:55
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

In above figure, we manually added 4 Arp entries.

```
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ arp -a
? (172.16.112.30) at 00:03:0f:1d:ab:f0 [ether] on eno1
? (172.16.112.31) at 00:03:0f:1a:fc:ea [ether] on eno1
? (172.16.112.28) at 00:03:0f:1d:ab:ec [ether] on eno1
? (172.16.112.27) at 00:03:0f:1d:ab:c4 [ether] on eno1
_gateway (172.16.112.1) at f8:0b:cb:cb:49:e4 [ether] on eno1
? (172.16.112.54) at 00:03:0f:1a:ff:70 [ether] on eno1
? (172.16.112.50) at 00:03:0f:1b:61:02 [ether] on eno1
? (172.16.112.49) at 00:03:0f:1b:61:22 [ether] on eno1
? (172.16.112.51) at 00:03:0f:1b:60:bc [ether] on eno1
? (192.168.1.1) at 00:11:20:38:44:59 [ether] PERM on eno1
? (172.16.112.58) at 00:03:0f:1a:fd:00 [ether] on eno1
? (172.16.112.34) at 00:03:0f:1d:ab:fe [ether] on eno1
? (172.16.112.37) at 00:03:0f:1d:ab:32 [ether] on eno1
? (172.16.112.39) at 00:03:0f:1a:ff:c4 [ether] on eno1
? (172.16.112.33) at 00:03:0f:1d:ab:58 [ether] on eno1
? (172.16.112.35) at 00:03:0f:1d:ab:1a [ether] on eno1
? (172.16.112.46) at 00:03:0f:1d:ac:10 [ether] on eno1
? (172.16.112.42) at 00:03:0f:1d:ab:f6 [ether] on eno1
? (172.16.112.44) at 00:03:0f:1d:ac:0c [ether] on eno1
? (172.16.112.45) at 00:03:0f:1d:ac:0e [ether] on eno1
? (172.16.112.47) at 00:03:0f:1d:ab:36 [ether] on eno1
? (172.16.112.43) at 00:03:0f:1a:fd:06 [ether] on eno1
? (172.16.112.66) at 00:03:0f:1a:fc:38 [ether] on eno1
? (172.16.118.133) at 84:a9:3e:8a:8b:b0 [ether] on eno1
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

**To add an ARP entry, we need to know about the following entries:**

- (i) Interface. It specifies which interface the IP and MAC address pair should be associated with.
- (ii) MAC address. The MAC address of the device you are looking to add an entry for.
- (iii) IP address. The IP address of the device you are looking to add an entry for.
- (iv) Expiry: The entry period should remain in the ARP table. For manual additions, this period would typically be indefinite.

**Deleting an ARP entry:**

To delete an ARP entry on the command line, most systems support a command like “arp -d” to remove a specific entry. Command: `sudo arp -d`

- c) **What are the parameters that determine how long the entries in the cache of the ARP module of the kernel remain valid and when they get deleted from the cache? Describe a trial-and-error method to discover the timeout value for the ARP cache entries.**

The parameters that determine how long the entries in the cache of the ARP module of the kernel remain valid are as follows:

**Time to Live:** It represents the amount of time an entry is considered valid. When this expires, the entry from the cache is being removed

**ARP Aging Time:** Various operating system has aging timers which checks for expired arp cache entries and remove them from the cache

**trial-and-error method:** The trial-and-error method is used to check timeout value of Arp cache. We can check for default timeout value of arp cache entries by running the following command

**Command:** `cat /proc/sys/net/ipv4/neigh/default/gc_stale_time`

```
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$ cat /proc/sys/net/ipv4/neigh/default/gc_stale_time
60
aditya@aditya-HP-ProDesk-600-G4-PCI-MT:~$
```

**d) What will happen if two IP addresses map to the same Ethernet address? Be specific on how all hosts on the subnet operate.**

If two IP Addresses map to the same Ethernet Address, then it would create an IP Address conflict resulting in communication issues. If there is an IP address conflict, then it leads to confusion between the hosts. Hosts within a subnet may experience unpredictable behaviour. Some of them communicate with the hosts sharing mac addresses whereas others may communicate with others.

### Q7. Local network analysis:

In order to understand the dynamics of online hosts within our LAN, we conducted a series of network scans using the nmap command. The specific command used was: `nmap -n -sP 172.16.116.126/21`. This command was executed repeatedly at different times of the day to capture variations in the number of online hosts within the specified subnet.

The experiment involved running the nmap command a total of six times, with intervals spanning across the day. Each scan focused on the subnet **172.16.116.126/21**, capturing information about the online hosts in the LAN.

Datapoint 1.

```
akshz@LittleGhost3: ~
akshz@LittleGhost3:~$ nmap -n -sP 172.16.116.126/21
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-23 19:29 IST
Nmap scan report for 172.16.112.1
Host is up (0.0074s latency).
Nmap scan report for 172.16.112.2
```

```
akshz@LittleGhost3: ~
Host is up (0.00074s latency).
Nmap scan report for 172.16.118.180
Host is up (0.00028s latency).
Nmap done: 2048 IP addresses (309 hosts up) scanned in 46.19 seconds
akshz@LittleGhost3:~$
```



Datapoint 2.

```
akshz@LittleGhost3: ~  
akshz@LittleGhost3:~$ nmap -n -sP 172.16.116.126/21  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-23 21:42 IST  
Nmap scan report for 172.16.112.1  
Host is up (0.0038s latency).  
Nmap scan report for 172.16.112.2
```

```
akshz@LittleGhost3: ~  
Host is up (0.0011s latency).  
Nmap scan report for 172.16.118.180  
Host is up (0.00091s latency).  
Nmap done: 2048 IP addresses (302 hosts up) scanned in 29.73 seconds  
akshz@LittleGhost3:~$
```

Datapoint 3.

```
akshz@LittleGhost3: ~  
akshz@LittleGhost3:~$ nmap -n -sP 172.16.116.126/21  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-23 23:15 IST  
Nmap scan report for 172.16.112.1  
Host is up (0.0036s latency).  
Nmap scan report for 172.16.112.2
```

```
akshz@LittleGhost3: ~  
Host is up (0.00098s latency).  
Nmap scan report for 172.16.118.180  
Host is up (0.00056s latency).  
Nmap done: 2048 IP addresses (290 hosts up) scanned in 42.95 seconds  
akshz@LittleGhost3:~$
```



Datapoint 4.

```
akshz@LittleGhost3: ~  
akshz@LittleGhost3:~$ nmap -n -sP 172.16.116.126/21  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 10:29 IST  
Nmap scan report for 172.16.112.1  
Host is up (0.0039s latency).  
Nmap scan report for 172.16.112.2
```

```
akshz@LittleGhost3: ~  
Host is up (0.00062s latency).  
Nmap scan report for 172.16.118.180  
Host is up (0.00072s latency).  
Nmap done: 2048 IP addresses (289 hosts up) scanned in 62.72 seconds  
akshz@LittleGhost3:~$
```

Datapoint 5

```
akshz@LittleGhost3: ~  
akshz@LittleGhost3:~$ nmap -n -sP 172.16.116.126/21  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 12:19 IST  
Nmap scan report for 172.16.112.1  
Host is up (0.0059s latency).  
Nmap scan report for 172.16.112.2
```

```
akshz@LittleGhost3: ~  
Host is up (0.0011s latency).  
Nmap scan report for 172.16.118.180  
Host is up (0.00048s latency).  
Nmap done: 2048 IP addresses (306 hosts up) scanned in 45.49 seconds  
akshz@LittleGhost3:~$
```

## Datapoint 6

```
akshz@LittleGhost3: ~  
akshz@LittleGhost3:~$ nmap -n -sP 172.16.116.126/21  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 17:25 IST  
Nmap scan report for 172.16.112.1  
Host is up (0.0041s latency).  
Nmap scan report for 172.16.112.2
```

```
akshz@LittleGhost3: ~  
Host is up (0.00069s latency).  
Nmap scan report for 172.16.118.180  
Host is up (0.00036s latency).  
Nmap done: 2048 IP addresses (297 hosts up) scanned in 17.30 seconds  
akshz@LittleGhost3:~$
```

The data was plotted on a graph to visualize the hourly trends. The x-axis represents the time of day, while the y-axis indicates the number of online hosts. The resulting graph provides a clear representation of when computers are switched on or off within the LAN throughout the day.

The Nmap graph for various timeslot

