Assignment 1 for Tut1 for CSE232

Notes on Command Line Functionalities

Palak Bhardwaj (2022344)
Yashovardhan Singhal (2022591)

20 August 2024

Abstract :

This report provides an overview of command-line utilities introduced in TUT-1 Computer Network course. It covers 6 tools : ifconfig, ping, traceroute, netstat, nslookup, and dig. . The report includes the following topics -

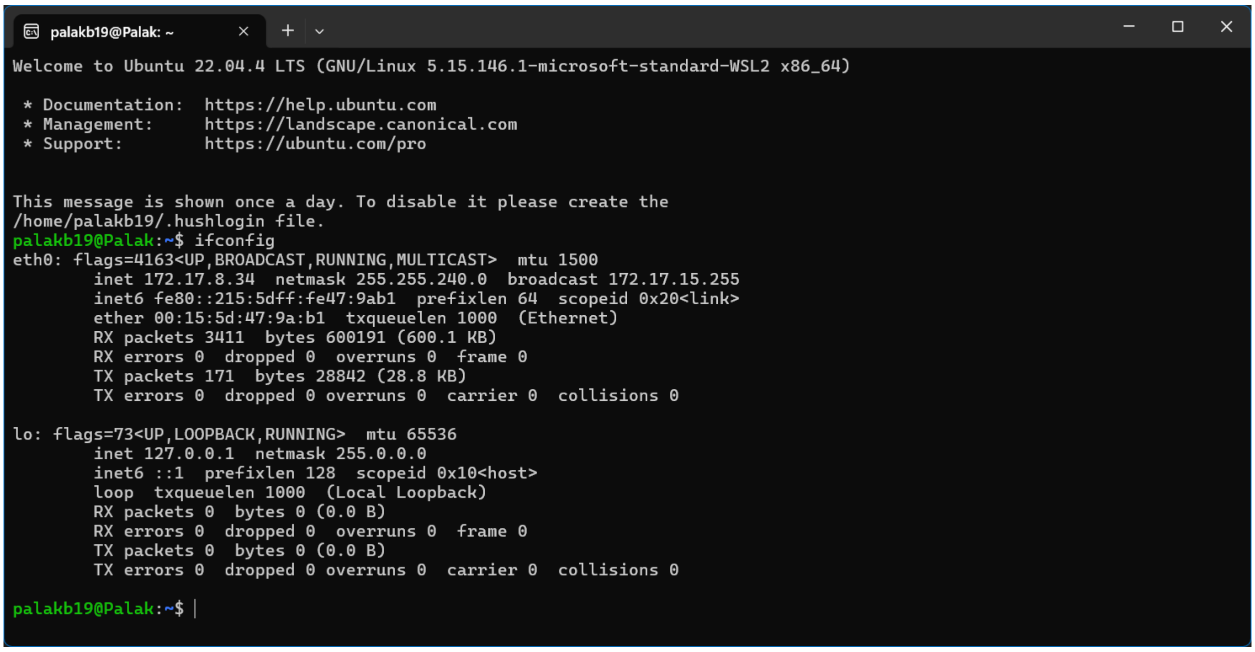
— ifconfig primarliy used for network interface configuration.
— ping used for testing network connectivity.
— traceroute for mapping network paths.
— netstat for displaying network statistics and connections.
— nslookup and dig for DNS-related queries.

1 Command Line Utilities for network debugging :

1.1 ifconfig

— ifconfig is used to assign an adress to a network inerface. It can be used to display information of a specific network interface.
— It can be used to display the current configuration for a network interface.
— The -a flag can be used instead of an interface name. This flag instructs ifconfig to display information about all interfaces in the system.
— The -d flag displays interfaces that are down.
— The -u flag displays interfaces that are up.This flag specifies protocols such as tcp, udp, tcp6, udp6, icmp, and icmp6..

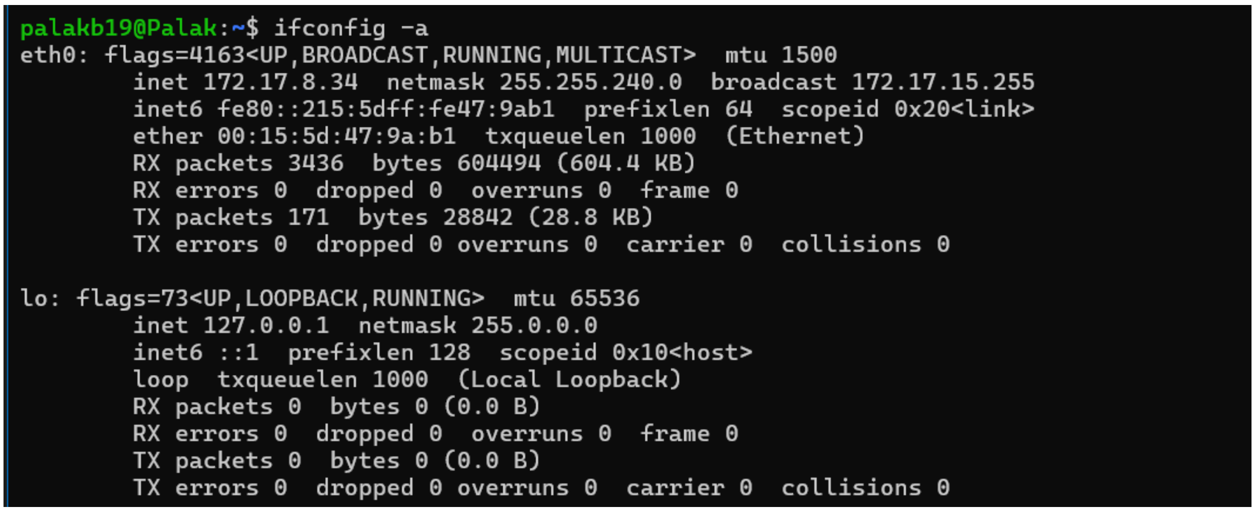
Purpose : Used to configure network interfaces.
Usage : Primarily for debugging and system tuning



```
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro


This message is shown once a day. To disable it please create the
/home/palakb19/.hushlogin file.
palakb19@Palak:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 3411  bytes 600191 (600.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 171  bytes 28842 (28.8 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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$ |
```

FIGURE 1 – Shows active interfaces

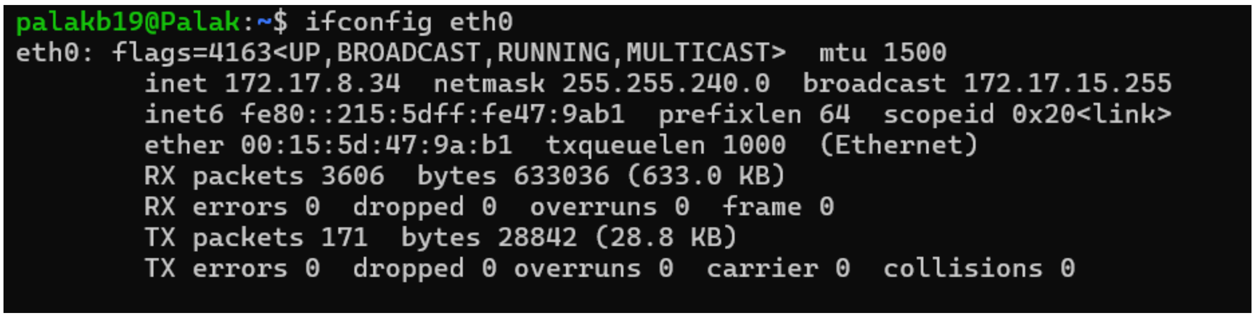
— The IP address is displayed after inet.
— UP : Indicates that the interface is active.
— BROADCAST : Supports broadcasting.
— RUNNING : The interface is up and running.
— MULTICAST : Supports multicast communication.
— MTU 1500 Maximum Transmission Unit : the largest packet size that can be sent over the network interface.



```
palakb19@Palak:~$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 3436  bytes 604494 (604.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 171  bytes 28842 (28.8 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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 2 – Shows all interfaces including inactive ones

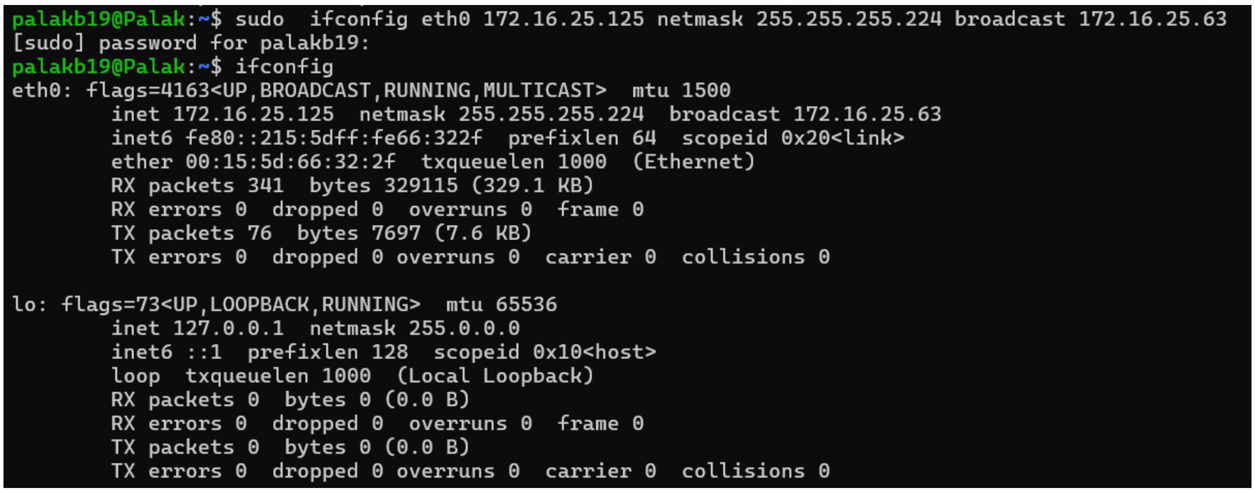
- All interfaces are active since there is no difference between ifconfig and ifconfig-a
- Uses the ifconfig -a command to display all interfaces.
- Includes both active and inactive interfaces in the output.
- Useful for identifying interfaces that are currently down.
- Provides a comprehensive view of the system's network configuration.
- **Interface Identification :** The output includes both loopback (lo) and physical interfaces (eth0), highlighting their different roles in networking. The loopback interface is used for internal testing and communication within the system, while eth0 typically represents an Ethernet connection.
- **Hardware and Link Information :** Includes hardware (MAC) addresses and link states, which help in identifying physical devices and diagnosing hardware-related issues.
- **Encapsulation Types :** Displays the type of encapsulation (e.g., Ethernet), which is important for understanding the type of network and data link protocols being used



```
palakb19@Palak:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 3606  bytes 633036 (633.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 171  bytes 28842 (28.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 3 – Displays Statistics of the given interface

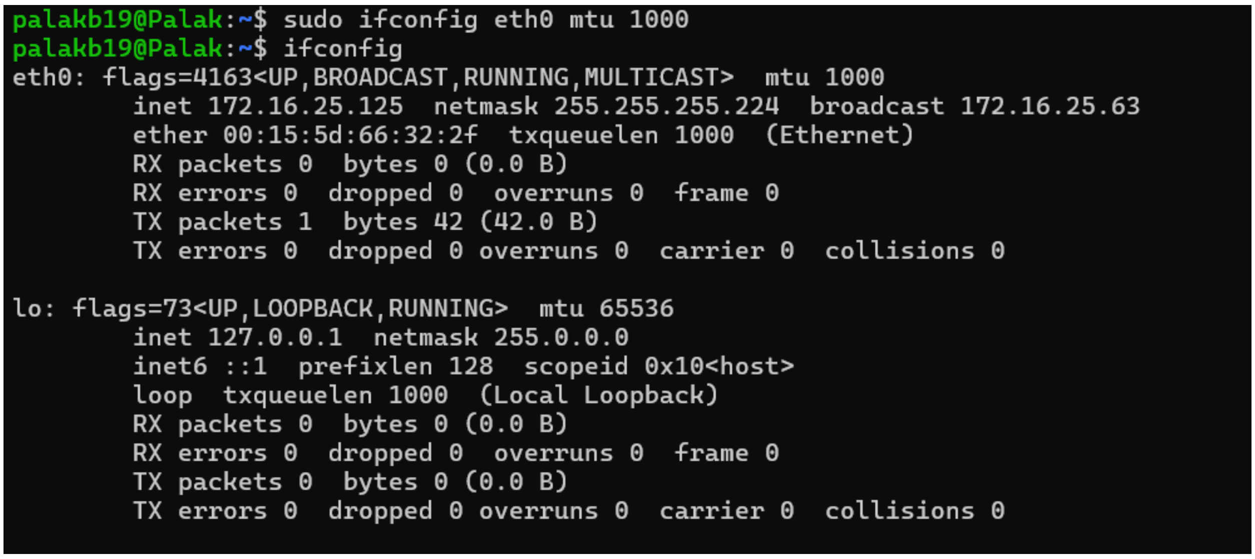
- Statistics include transmitted and received packets, errors, dropped packets, and collisions, which are useful for diagnosing network issues.
- **Transmission/Reception Data :** The figure shows the number of packets transmitted (TX packets) and received (RX packets), along with the number of bytes, giving a quantitative measure of the interface's activity.
- **Error Diagnosis :** Displays error-related information, such as frame errors, dropped packets, and overruns, which are critical for diagnosing network performance issues and identifying potential causes like hardware failures or misconfigurations.
- **Collisions and Carrier Issues :** Includes statistics on collisions (common in older Ethernet networks) and carrier signals, which can affect data integrity and transmission quality.



```
palakb19@Palak:~$ sudo  ifconfig eth0 172.16.25.125 netmask 255.255.255.224 broadcast 172.16.25.63
[sudo] password for palakb19:
palakb19@Palak:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.25.125  netmask 255.255.255.224  broadcast 172.16.25.63
        inet6 fe80::215:5dff:fe66:322f  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:66:32:2f  txqueuelen 1000  (Ethernet)
        RX packets 341  bytes 329115 (329.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 76  bytes 7697 (7.6 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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 4 – This command configures the interface eth0 with the IP address 172.16.25.125, a subnet mask of 255.255.255.224, and a broadcast address of 172.16.25.63.

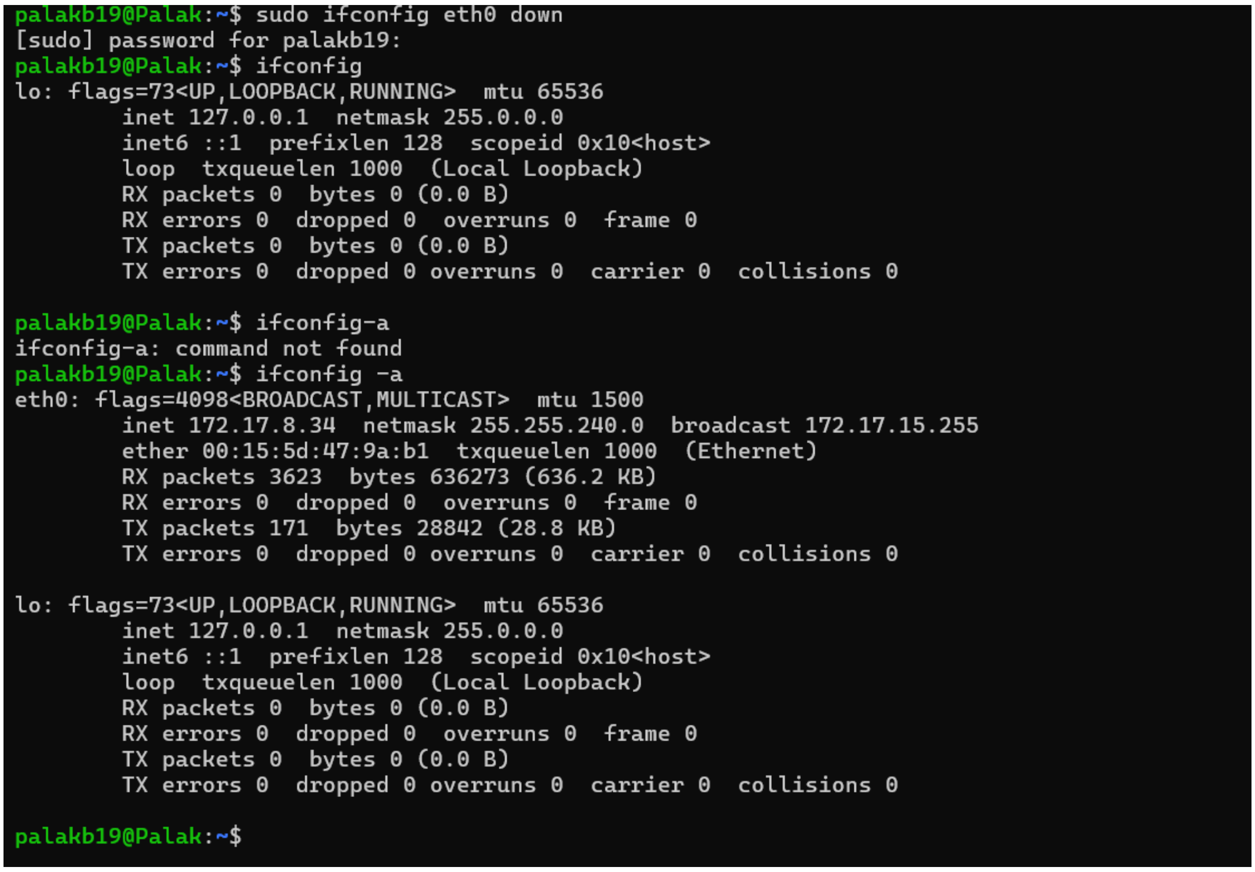
— The subnet mask (netmask) and broadcast address are configured along with the IP address. The subnet mask determines the network's size, while the broadcast address is used for broadcasting messages to all devices on the network.
— Static IP Assignment : This configuration represents a static IP assignment, which is critical in scenarios where dynamic IP assignment (via DHCP) is not preferable, such as for servers or devices requiring a consistent IP address.
— Temporary vs. Persistent Configuration : Note that this IP assignment is temporary and will be lost upon reboot unless it is added to the network configuration files for persistence.



```
palakb19@Palak:~$ sudo ifconfig eth0 mtu 1000
palakb19@Palak:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1000
        inet 172.16.25.125  netmask 255.255.255.224  broadcast 172.16.25.63
        ether 00:15:5d:66:32:2f  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1  bytes 42 (42.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

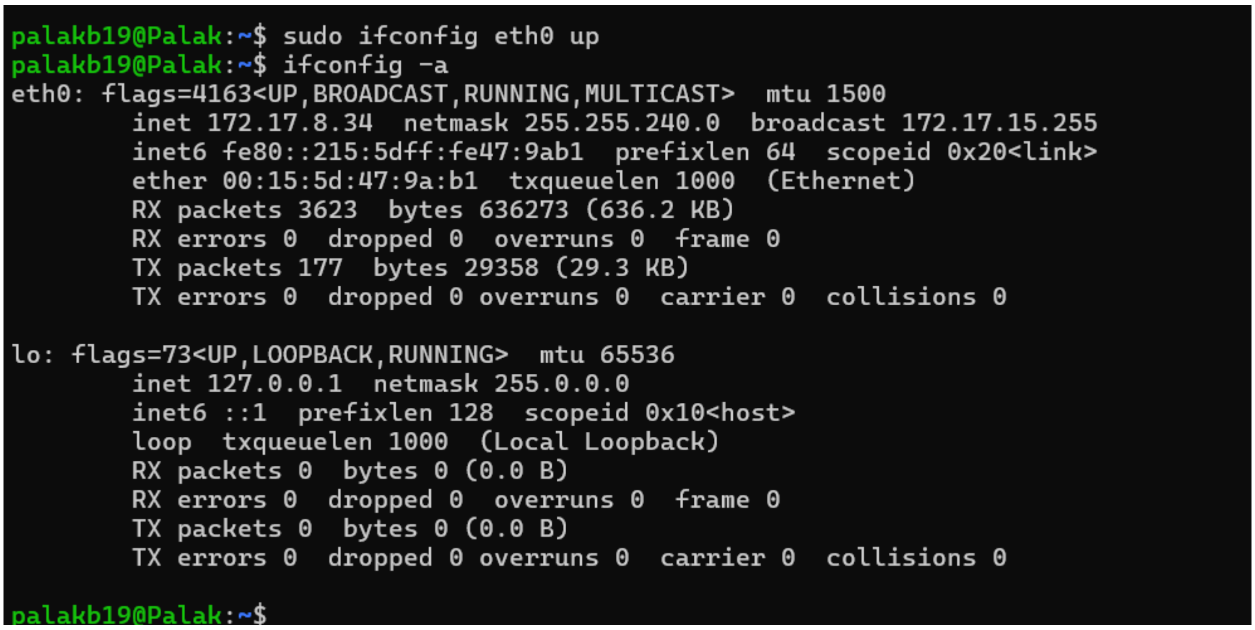
FIGURE 5 – This command is used to configure the Maximum Transmission Unit (MTU) of the network interface eth0.

1.2 Flow of different ifconfig commands



```
palakb19@Palak:~$ sudo ifconfig eth0 down
[sudo] password for palakb19:
palakb19@Palak:~$ ifconfig
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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$ ifconfig-a
ifconfig-a: command not found
palakb19@Palak:~$ ifconfig -a
eth0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 3623  bytes 636273 (636.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 171  bytes 28842 (28.8 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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$
```

FIGURE 6 – Interfaces displayed after deactivating the eth0 interface

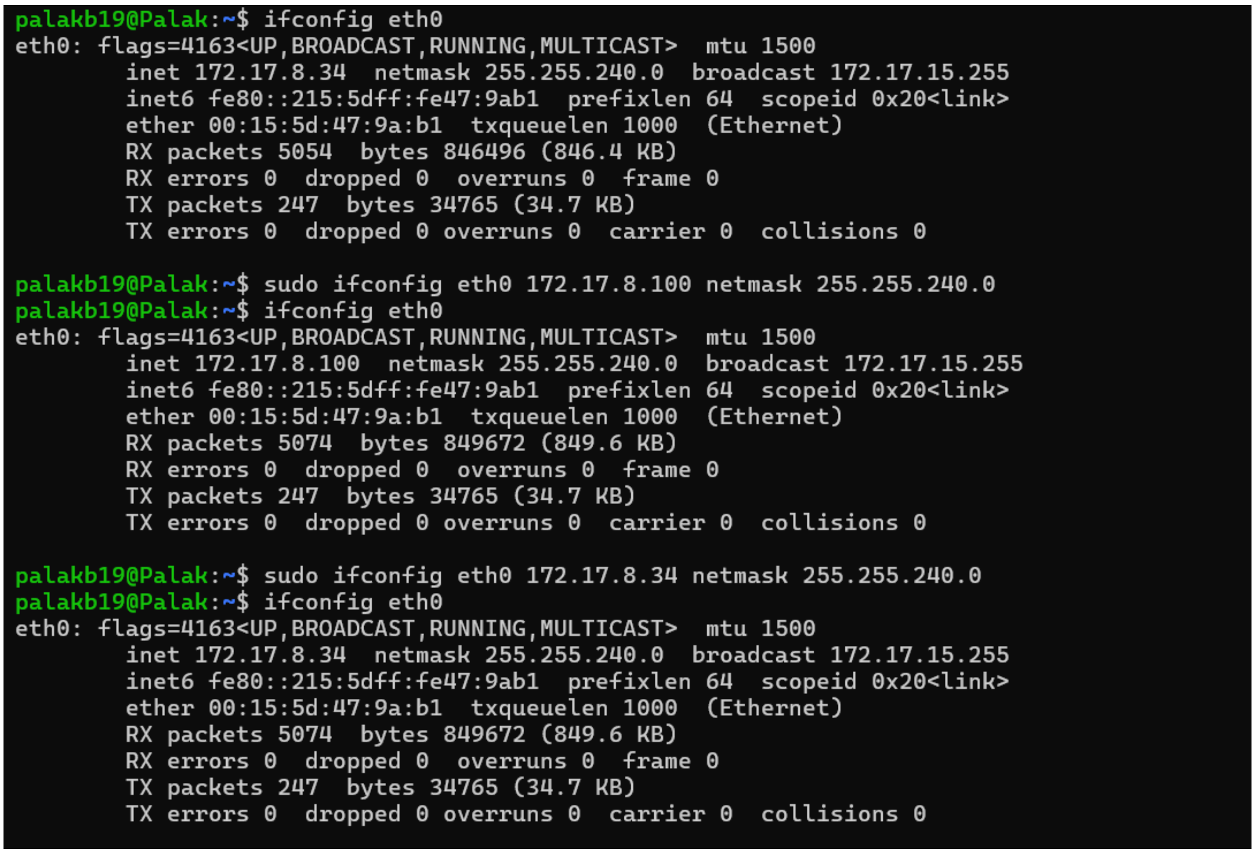


```
palakb19@Palak:~$ sudo ifconfig eth0 up
palakb19@Palak:~$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 3623  bytes 636273 (636.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 177  bytes 29358 (29.3 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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$
```

FIGURE 7 – Active interfaces displayed after activating the eth0 interface

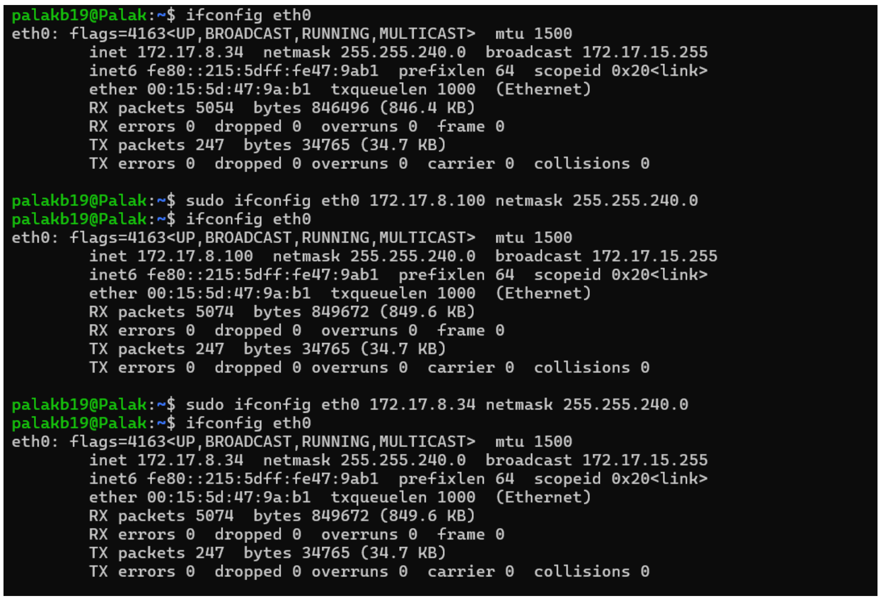
— The two active interfaces were loopback interface and eth0 interface.
— eth0 interface is used for external communication

— loopback interface is used for internal communication.



```
palakb19@Palak:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 5054  bytes 846496 (846.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 247  bytes 34765 (34.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$ sudo ifconfig eth0 172.17.8.100 netmask 255.255.240.0
palakb19@Palak:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.100  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 5074  bytes 849672 (849.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 247  bytes 34765 (34.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$ sudo ifconfig eth0 172.17.8.34 netmask 255.255.240.0
palakb19@Palak:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 5074  bytes 849672 (849.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 247  bytes 34765 (34.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

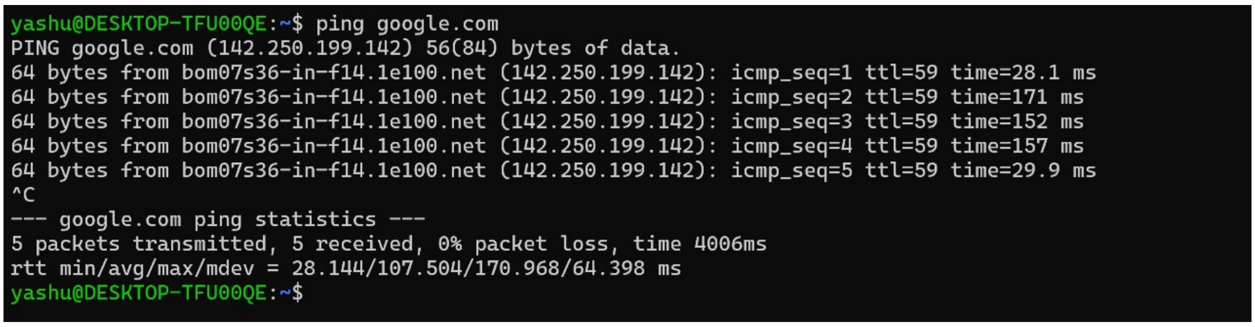
FIGURE 8 – Displays routing information about the network interface



```
palakb19@Palak:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 5054  bytes 846496 (846.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 247  bytes 34765 (34.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$ sudo ifconfig eth0 172.17.8.100 netmask 255.255.240.0
palakb19@Palak:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.100  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 5074  bytes 849672 (849.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 247  bytes 34765 (34.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

palakb19@Palak:~$ sudo ifconfig eth0 172.17.8.34 netmask 255.255.240.0
palakb19@Palak:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.8.34  netmask 255.255.240.0  broadcast 172.17.15.255
        inet6 fe80::215:5dff:fe47:9ab1  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:47:9a:b1  txqueuelen 1000  (Ethernet)
        RX packets 5074  bytes 849672 (849.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 247  bytes 34765 (34.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 9 – IP address being changed and reverted back

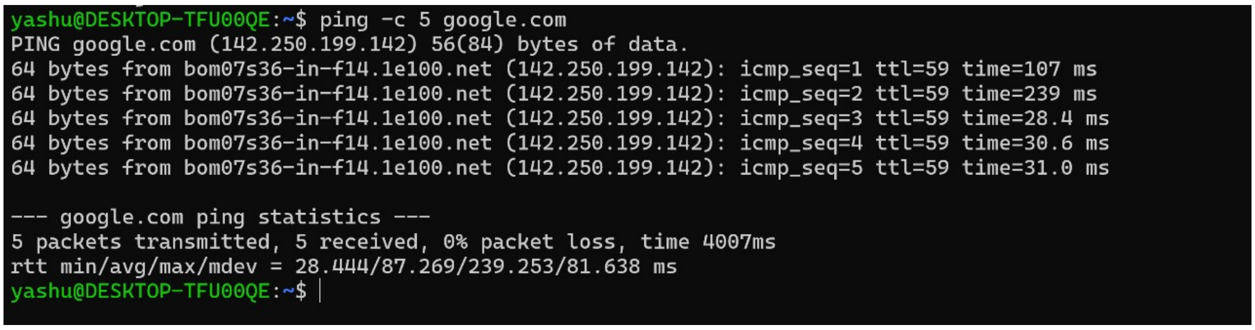
1.3 ping



```
yashu@DESKTOP-TFU00QE:~$ ping google.com
PING google.com (142.250.199.142) 56(84) bytes of data.
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=1 ttl=59 time=28.1 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=2 ttl=59 time=171 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=3 ttl=59 time=152 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=4 ttl=59 time=157 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=5 ttl=59 time=29.9 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 28.144/107.504/170.968/64.398 ms
yashu@DESKTOP-TFU00QE:~$
```

FIGURE 10 – Ping google.com

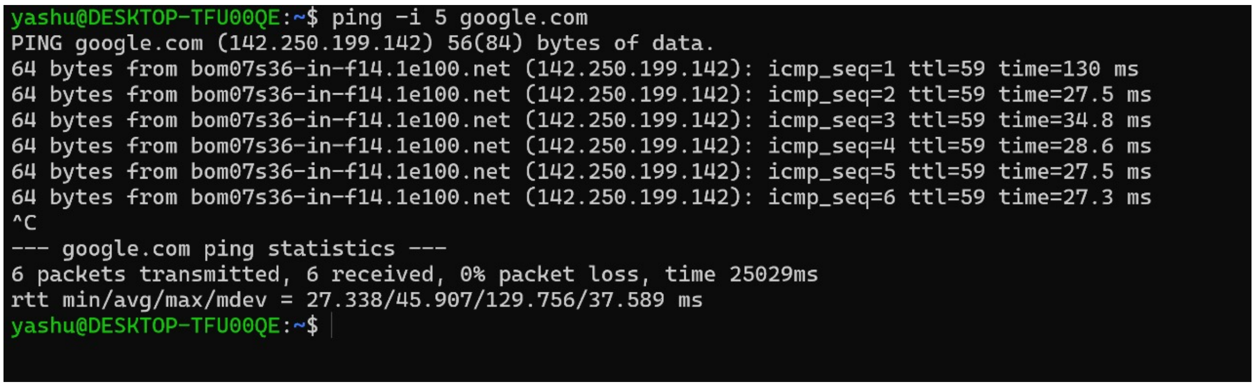
- The ping command was used to send 5 packets to google.com (IP : 142.250.199.142) to check network connectivity.
- All 5 packets were successfully received, with no packet loss.
- The round-trip times (RTT) varied, with a minimum of 28.4 ms, a maximum of 239.3 ms, and an average of 87.3 ms. The time it took for all operations was 4007 ms, indicating a stable but slightly variable connection to the server.



```
yashu@DESKTOP-TFU00QE:~$ ping -c 5 google.com
PING google.com (142.250.199.142) 56(84) bytes of data.
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=1 ttl=59 time=107 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=2 ttl=59 time=239 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=3 ttl=59 time=28.4 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=4 ttl=59 time=30.6 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=5 ttl=59 time=31.0 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 28.444/87.269/239.253/81.638 ms
yashu@DESKTOP-TFU00QE:~$
```

FIGURE 11 – ping -c 5 google.com

- The **ping -c 5 google.com** command checks connectivity to google.com by sending 5 ICMP echo request packets.
- It measures network latency, providing the round-trip time (RTT) for each packet sent and received.
- The command helps identify network issues like packet loss, with 0% packet loss indicating a stable connection.
- It is commonly used for basic network troubleshooting and performance assessment.

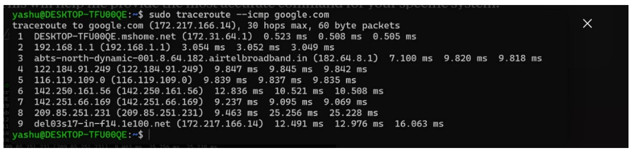


```
yashu@DESKTOP-TFU00QE:~$ ping -i 5 google.com
PING google.com (142.250.199.142) 56(84) bytes of data.
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=1 ttl=59 time=130 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=2 ttl=59 time=27.5 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=3 ttl=59 time=34.8 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=4 ttl=59 time=28.6 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=5 ttl=59 time=27.5 ms
64 bytes from bom07s36-in-f14.1e100.net (142.250.199.142): icmp_seq=6 ttl=59 time=27.3 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 25029ms
rtt min/avg/max/mdev = 27.338/45.907/129.756/37.589 ms
yashu@DESKTOP-TFU00QE:~$
```

FIGURE 12 – ping -i google.com

- Command Used : ping -i 5 google.com
- Purpose : This command also sends ICMP echo request packets to google.com, but with a different interval between each packet.
- Options Explained : -i 5 : This option specifies the interval between sending each ping packet. In this case, it waits for 5 seconds between each packet.
- Output : The output shows 6 packets transmitted with a 5-second interval between each, all successfully received with no packet loss. The RTT statistics provide a lower average RTT compared to the first image due to the different packet sending intervals.

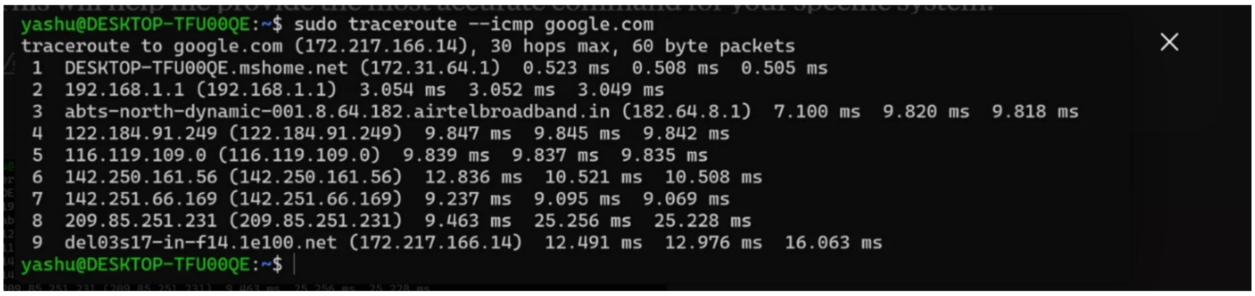
1.4 traceroute



```
yashu@DESKTOP-TFU00QE:~$ sudo traceroute --icmp google.com
traceroute to google.com (172.217.166.14), 30 hops max, 60 byte packets
 1  DESKTOP-TFU00QE.mshome.net (172.31.64.1)  0.523 ms  0.508 ms  0.505 ms
 2  192.168.1.1 (192.168.1.1)  3.054 ms  3.052 ms  3.049 ms
 3  abts-north-dynamic-001.8.64.182.airtelbroadband.in (182.64.8.1)  7.100 ms  9.820 ms  9.818 ms
 4  122.184.91.249 (122.184.91.249)  9.847 ms  9.845 ms  9.842 ms
 5  116.119.109.0 (116.119.109.0)  9.839 ms  9.837 ms  9.835 ms
 6  142.250.161.56 (142.250.161.56)  12.836 ms  10.521 ms  10.508 ms
 7  142.251.66.169 (142.251.66.169)  9.237 ms  9.095 ms  9.069 ms
 8  209.85.251.231 (209.85.251.231)  9.463 ms  25.256 ms  25.228 ms
 9  del03s17-in-f14.1e100.net (172.217.166.14)  12.491 ms  12.976 ms  16.063 ms
yashu@DESKTOP-TFU00QE:~$
```

FIGURE 13 – sudo traceroute ——icmp google.com

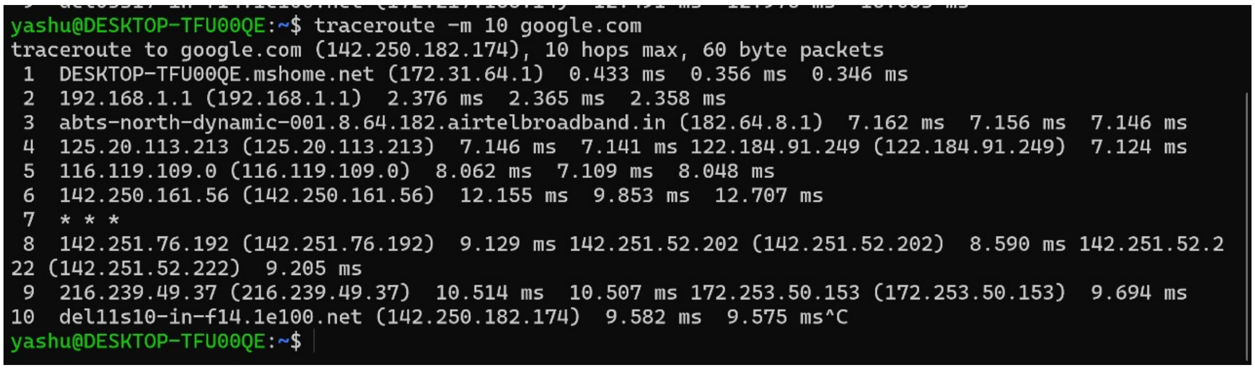
- Command : sudo traceroute –icmp google.com traces route using ICMP packets.
- Output : Shows 9 hops from your PC to Google's server (172.217.166.14).
- Local network → ISP → Internet backbone → Google server
- Increases from <1ms locally to 12-16ms at Google's server



```
yashu@DESKTOP-TFU00QE:~$ sudo traceroute --icmp google.com
traceroute to google.com (172.217.166.14), 30 hops max, 60 byte packets
 1  DESKTOP-TFU00QE.mshome.net (172.31.64.1)  0.523 ms  0.508 ms  0.505 ms
 2  192.168.1.1 (192.168.1.1)  3.054 ms  3.052 ms  3.049 ms
 3  abts-north-dynamic-001.8.64.182.airtelbroadband.in (182.64.8.1)  7.100 ms  9.820 ms  9.818 ms
 4  122.184.91.249 (122.184.91.249)  9.847 ms  9.845 ms  9.842 ms
 5  116.119.109.0 (116.119.109.0)  9.839 ms  9.837 ms  9.835 ms
 6  142.250.161.56 (142.250.161.56)  12.836 ms  10.521 ms  10.508 ms
 7  142.251.66.169 (142.251.66.169)  9.237 ms  9.095 ms  9.069 ms
 8  209.85.251.231 (209.85.251.231)  9.463 ms  25.256 ms  25.228 ms
 9  del03s17-in-f14.1e100.net (172.217.166.14)  12.491 ms  12.976 ms  16.063 ms
yashu@DESKTOP-TFU00QE:~$
```

FIGURE 14 – sudo traceroute ——icmp google.com

- Command : traceroute -m 10 google.com Sets maximum hops to 10 (-m 10) Shows route from local machine to Google's server (142.250.182.174).
- Displays IP addresses, hostnames, and response times for each hop Some hops (e.g., 7) show asterisks, indicating no response Final hop reaches Google's server (del11s10-in-f14.1e100.net)



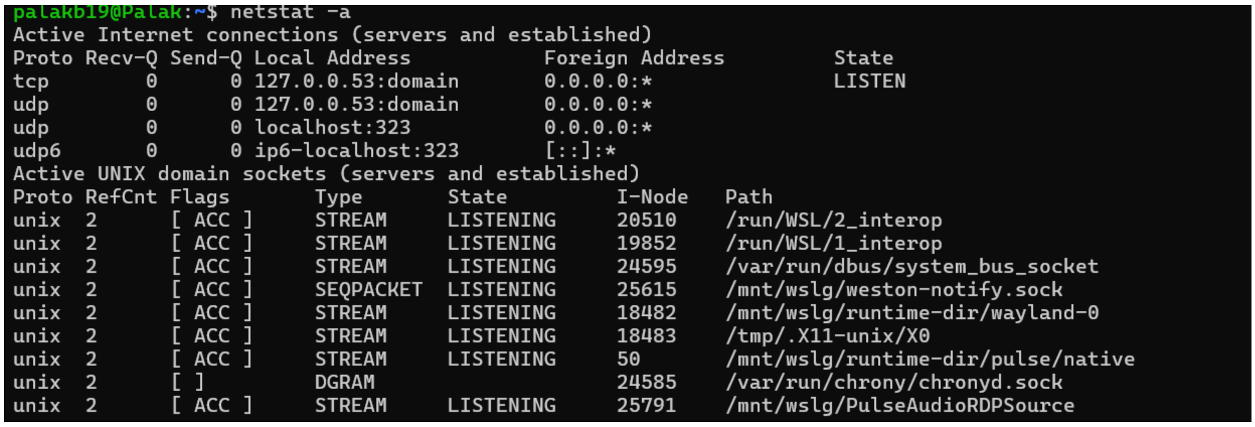
```
yashu@DESKTOP-TFU00QE:~$ traceroute -m 10 google.com
traceroute to google.com (142.250.182.174), 10 hops max, 60 byte packets
 1  DESKTOP-TFU00QE.mshome.net (172.31.64.1)  0.433 ms  0.356 ms  0.346 ms
 2  192.168.1.1 (192.168.1.1)  2.376 ms  2.365 ms  2.358 ms
 3  abts-north-dynamic-001.8.64.182.airtelbroadband.in (182.64.8.1)  7.162 ms  7.156 ms  7.146 ms
 4  125.20.113.213 (125.20.113.213)  7.146 ms  7.141 ms 122.184.91.249 (122.184.91.249)  7.124 ms
 5  116.119.109.0 (116.119.109.0)  8.062 ms  7.109 ms  8.048 ms
 6  142.250.161.56 (142.250.161.56)  12.155 ms  9.853 ms  12.707 ms
 7  * * *
 8  142.251.76.192 (142.251.76.192)  9.129 ms 142.251.52.202 (142.251.52.202)  8.590 ms 142.251.52.2
22 (142.251.52.222)  9.205 ms
 9  216.239.49.37 (216.239.49.37)  10.514 ms  10.507 ms 172.253.50.153 (172.253.50.153)  9.694 ms
10  del11s10-in-f14.1e100.net (142.250.182.174)  9.582 ms  9.575 ms^C
yashu@DESKTOP-TFU00QE:~$
```

Figure 15 – traceroute —m 10 google.com

- — Displays the usage and options for the traceroute command.
- — Provides brief explanations for each option, helping users customize their traceroute commands.

1.5 Netstat

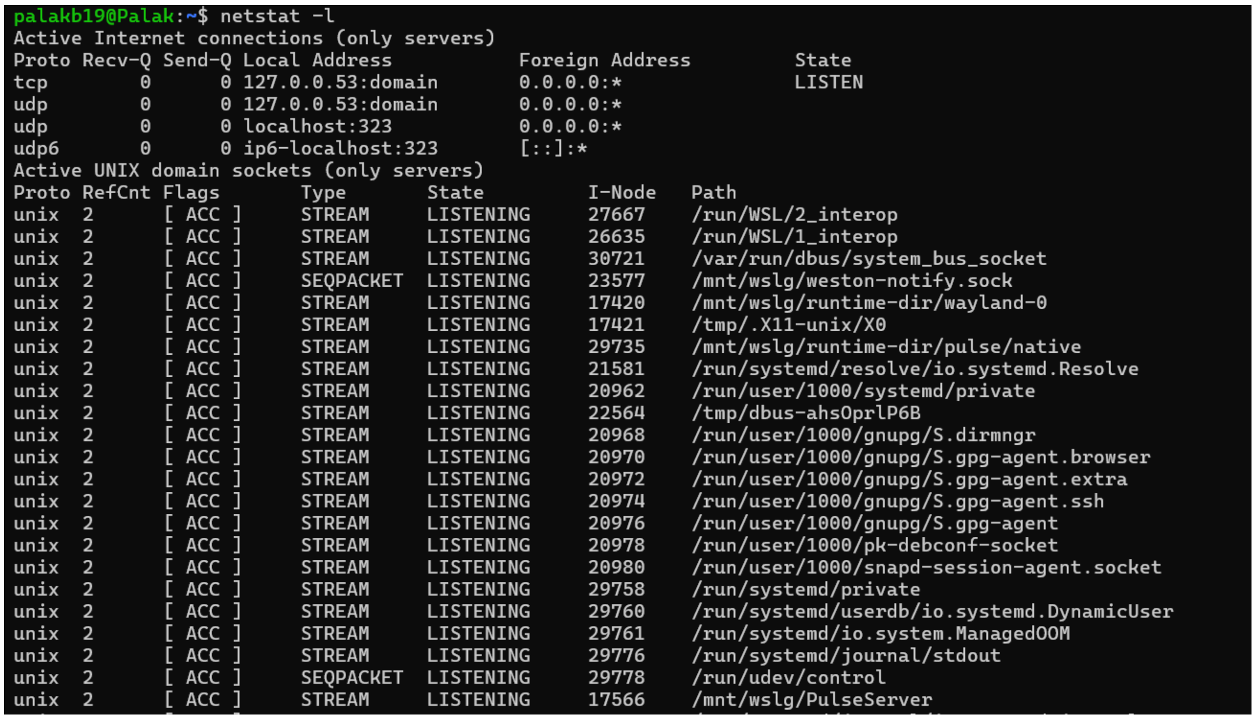
- Netstat commands are used to monitor network connections, including incoming and outgoing connection.
- Used for monitoring routing tables and network interface statistics.
- It is used for network diagnostics and system monitoring.



```
palakb19@Palak:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:domain       0.0.0.0:*               LISTEN
udp        0      0 127.0.0.53:domain       0.0.0.0:*
udp        0      0 localhost:323           0.0.0.0:*
udp6       0      0 ip6-localhost:323       [::]:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     20510    /run/WSL/2_interop
unix  2      [ ACC ]     STREAM     LISTENING     19852    /run/WSL/1_interop
unix  2      [ ACC ]     STREAM     LISTENING     24595    /var/run/dbus/system_bus_socket
unix  2      [ ACC ]     SEQPACKET  LISTENING     25615    /mnt/wslg/weston-notify.sock
unix  2      [ ACC ]     STREAM     LISTENING     18482    /mnt/wslg/runtime-dir/wayland-0
unix  2      [ ACC ]     STREAM     LISTENING     18483    /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     50       /mnt/wslg/runtime-dir/pulse/native
unix  2      [ ]         DGRAM                    24585    /var/run/chrony/chronyd.sock
unix  2      [ ACC ]     STREAM     LISTENING     25791    /mnt/wslg/PulseAudioRDPSource
```

FIGURE 16 – Displays all connections (listening and established)

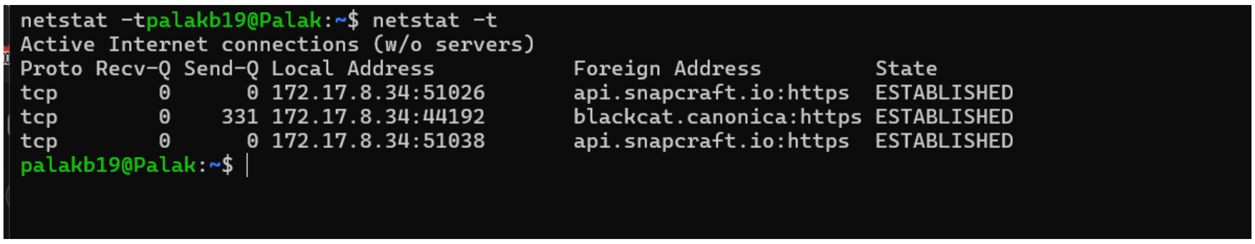
- This figure shows the output of the netstat command, displaying all active connections, including both listening and established ones. It lists TCP and UDP connections.
- Connections in the **"LISTEN"** state are waiting for inbound connections, while **"ESTABLISHED"** indicates active connections.
- **Local and Foreign Addresses :** The figure includes columns for local and foreign addresses, displaying the IP and port number on both ends of each connection.
- Process State Monitoring - This information is critical for monitoring real-time connection statuses, especially for network diagnostics and security assessments.



```
palakb19@Palak:~$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:domain       0.0.0.0:*               LISTEN
udp        0      0 127.0.0.53:domain       0.0.0.0:*
udp        0      0 localhost:323           0.0.0.0:*
udp6       0      0 ip6-localhost:323       [::]:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     27667    /run/WSL/2_interop
unix  2      [ ACC ]     STREAM     LISTENING     26635    /run/WSL/1_interop
unix  2      [ ACC ]     STREAM     LISTENING     30721    /var/run/dbus/system_bus_socket
unix  2      [ ACC ]     SEQPACKET  LISTENING     23577    /mnt/wslg/weston-notify.sock
unix  2      [ ACC ]     STREAM     LISTENING     17420    /mnt/wslg/runtime-dir/wayland-0
unix  2      [ ACC ]     STREAM     LISTENING     17421    /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     29735    /mnt/wslg/runtime-dir/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     21581    /run/systemd/resolve/io.systemd.Resolve
unix  2      [ ACC ]     STREAM     LISTENING     20962    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     22564    /tmp/dbus-ahsOprlP6B
unix  2      [ ACC ]     STREAM     LISTENING     20968    /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     20970    /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     20972    /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     20974    /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     20976    /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     20978    /run/user/1000/pk-debconf-socket
unix  2      [ ACC ]     STREAM     LISTENING     20980    /run/user/1000/snapd-session-agent.socket
unix  2      [ ACC ]     STREAM     LISTENING     29758    /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     29760    /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     29761    /run/systemd/io.system.ManagedOOM
unix  2      [ ACC ]     STREAM     LISTENING     29776    /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     29778    /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     17566    /mnt/wslg/PulseServer
```

FIGURE 17 – Displays listening connections

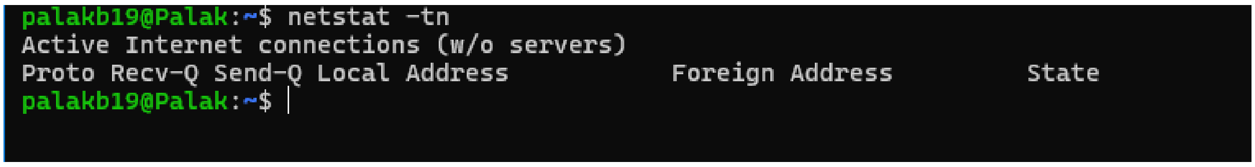
— Filtered Output : The netstat -l command is used to display only listening connections, filtering out established connections.
— Listening Ports : Focuses on ports that are open and waiting for incoming connections, which is important for server management and identifying open services.
— Connection States help in identifying services that are active on the server, which could be critical for troubleshooting or managing server loads.
— **Relevance :** Identifying listening ports is essential for security, as unnecessary open ports can be potential entry points for attacks.



```
netstat -tpalakb19@Palak:~$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 172.17.8.34:51026       api.snapcraft.io:https  ESTABLISHED
tcp        0    331 172.17.8.34:44192       blackcat.canonica:https ESTABLISHED
tcp        0      0 172.17.8.34:51038       api.snapcraft.io:https  ESTABLISHED
palakb19@Palak:~$ |
```

FIGURE 18 – Shows TCP connections and listening ports only.

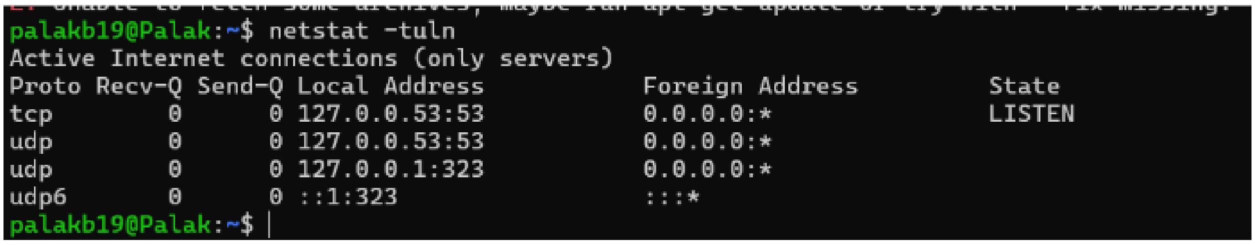
— This attached picture highlights the use of the netstat -t command to show only TCP connections, excluding other protocols like UDP.
— : TCP is a **connection-oriented protocol,** and this figure focuses on showing both active and listening TCP connections.
— **Network Diagnostics :** Understanding TCP connections is key to diagnosing connectivity issues, as TCP ensures reliable data transfer.
— Performance Monitoring : Monitoring TCP connections helps in assessing the load on the server and optimizing resource usage.



```
palakb19@Palak:~$ netstat -tn
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
palakb19@Palak:~$ |
```

FIGURE 19 – Displays Active TCP connections

— Unlike the other commands displayed in this document, this command displays active TCP connections that are established, meaning they are currently in use.
— Active TCP connections are crucial for monitoring network traffic, particularly in real-time applications.
— Security Implications : Keeping track of active connections can help in identifying unauthorized access or unusual activity on the network.

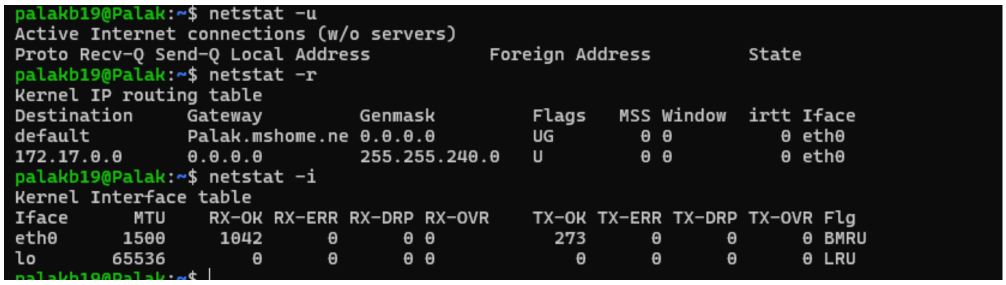


```
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
palakb19@Palak:~$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
udp        0      0 127.0.0.53:53           0.0.0.0:*
udp        0      0 127.0.0.1:323           0.0.0.0:*
udp6       0      0 ::1:323                 :::*
palakb19@Palak:~$ |
```

FIGURE 20 – Displays UDP connections, routing table, network interface statistics :

— This picture shows the output of the netstat -u command, which displays **UDP connections**, a connectionless protocol used for services like DNS and video streaming.

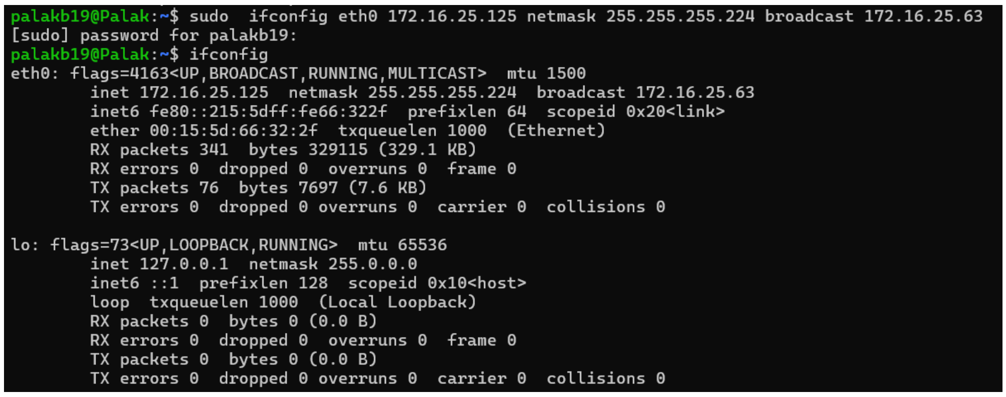
— Routing Table : The inclusion of the routing table shows how data is being directed across the network.
— Monitoring Performance : This data is useful for assessing overall network performance and diagnosing potential bottlenecks.



```
palakb19@Palak:~$ netstat -u
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
palakb19@Palak:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         Palak.mshome.ne 0.0.0.0         UG        0 0          0 eth0
172.17.0.0      0.0.0.0         255.255.240.0   U         0 0          0 eth0
palakb19@Palak:~$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500     1042      0      0 0           273      0      0      0 BMRU
lo       65536        0      0      0 0             0      0      0      0 LRU
palakb19@Palak:~$ |
```

FIGURE 21 – Display all connections with detailed info (PID, process name) etc.

— Process Information : Shows an advanced netstat command that includes **process IDs (PID)** and program names associated with each network connection or socket, useful for tracking which applications are using specific network resources.
— Detailed Network Usage : Knowing the process behind each connection helps in both performance tuning and security auditing.
— This detailed information is essential for identifying suspicious processes that may be initiating unauthorized network connections.
— R It also assists in understanding resource allocation across different processes and can help in load balancing or troubleshooting high resource usage.



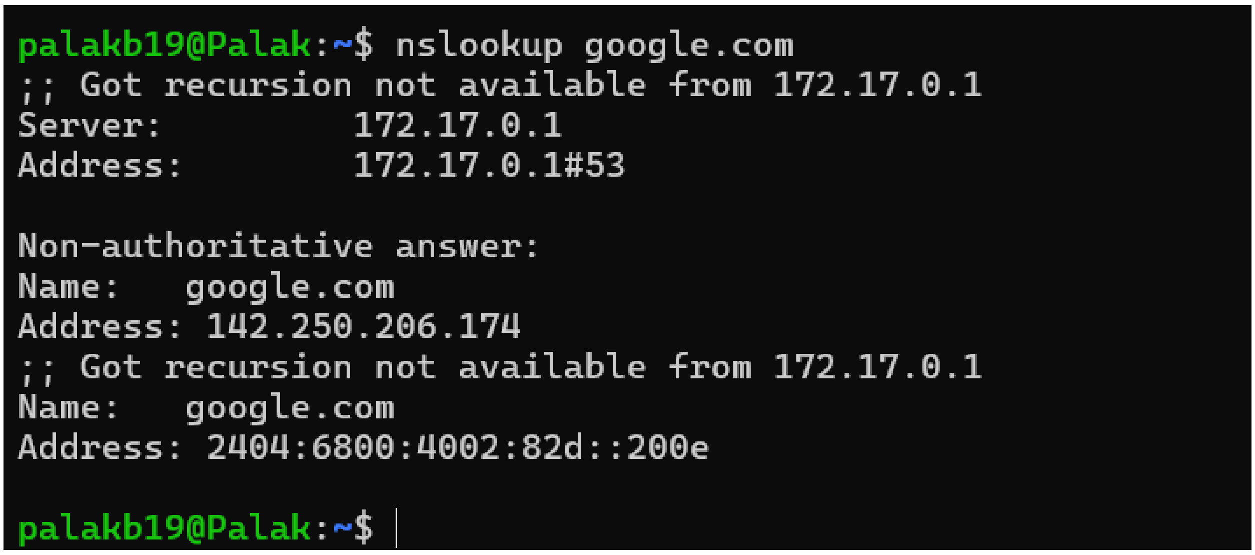
```
palakb19@Palak:~$ sudo  ifconfig eth0 172.16.25.125 netmask 255.255.255.224 broadcast 172.16.25.63
[sudo] password for palakb19:
palakb19@Palak:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.25.125  netmask 255.255.255.224  broadcast 172.16.25.63
        inet6 fe80::215:5dff:fe66:322f  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:66:32:2f  txqueuelen 1000  (Ethernet)
        RX packets 341  bytes 329115 (329.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 76  bytes 7697 (7.6 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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 22 – Displays the process ID (PID) and program name associated with each network connection or socket.

— Enhanced Detail : Similar to Figure 15, this figure focuses on linking network connections to their corresponding processes, giving an even more granular view of how network resources are being utilized.
— Application-Level Insight : By showing both the PID and the program name, administrators can quickly identify which software is responsible for each connection, crucial for both debugging and monitoring purposes.
— Connection Tracking : This is valuable for tracking application-level network usage, allowing for more targeted troubleshooting and optimization.
— Security and Auditing : This information is particularly relevant in security contexts, where knowing which processes are responsible for network traffic is key to preventing malicious activity.

1.6 DNS : Domain Name System

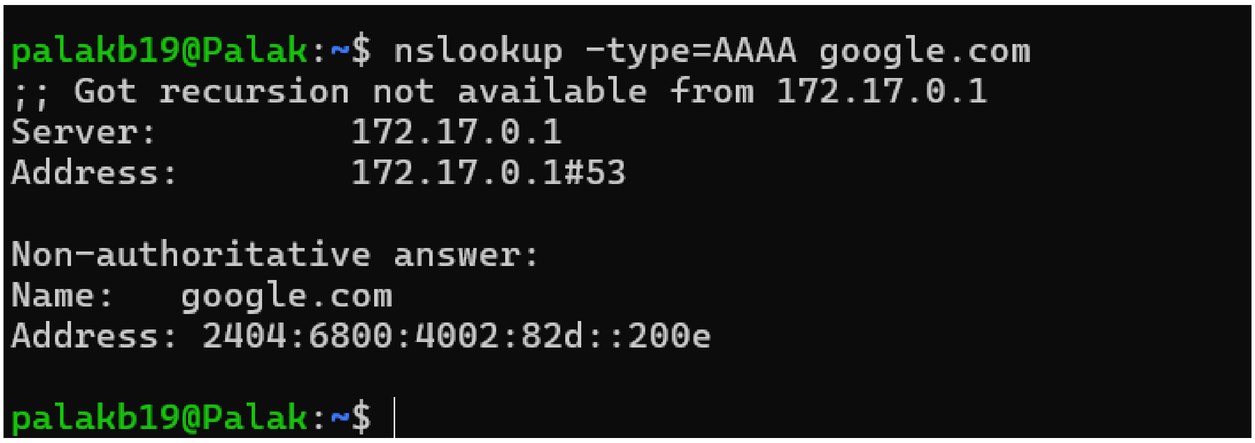
— DNS (Domain Name System) is used to translate or resolve domain names into IP addresses (like 142.250.182.174).
— It is like a "phone book" for the Internet, helping devices find each other using names.



```
palakb19@Palak:~$ nslookup google.com
;; Got recursion not available from 172.17.0.1
Server:         172.17.0.1
Address:        172.17.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.206.174
;; Got recursion not available from 172.17.0.1
Name:   google.com
Address: 2404:6800:4002:82d::200e

palakb19@Palak:~$
```

FIGURE 23 – Querying the DNS to resolve the domain name google.com to an IP address

— The Domain Name System (DNS) translates human-readable domain names like google.com into IP addresses, which are used for routing packets on the internet.
— Query Results : This figure demonstrates querying DNS for resolving a domain name (e.g., google.com) to its corresponding IP address.
— DNS Lookup Process : It visualizes the process where a client sends a query to the DNS server, which responds with the requested IP address.
— Internet Navigation : DNS is essential for navigating the internet, as it links domain names to their corresponding IP addresses, enabling users to reach websites using familiar names.
— Security Implications : Monitoring DNS queries can help in detecting and preventing DNS spoofing attacks, where malicious actors attempt to redirect traffic to fraudulent sites.

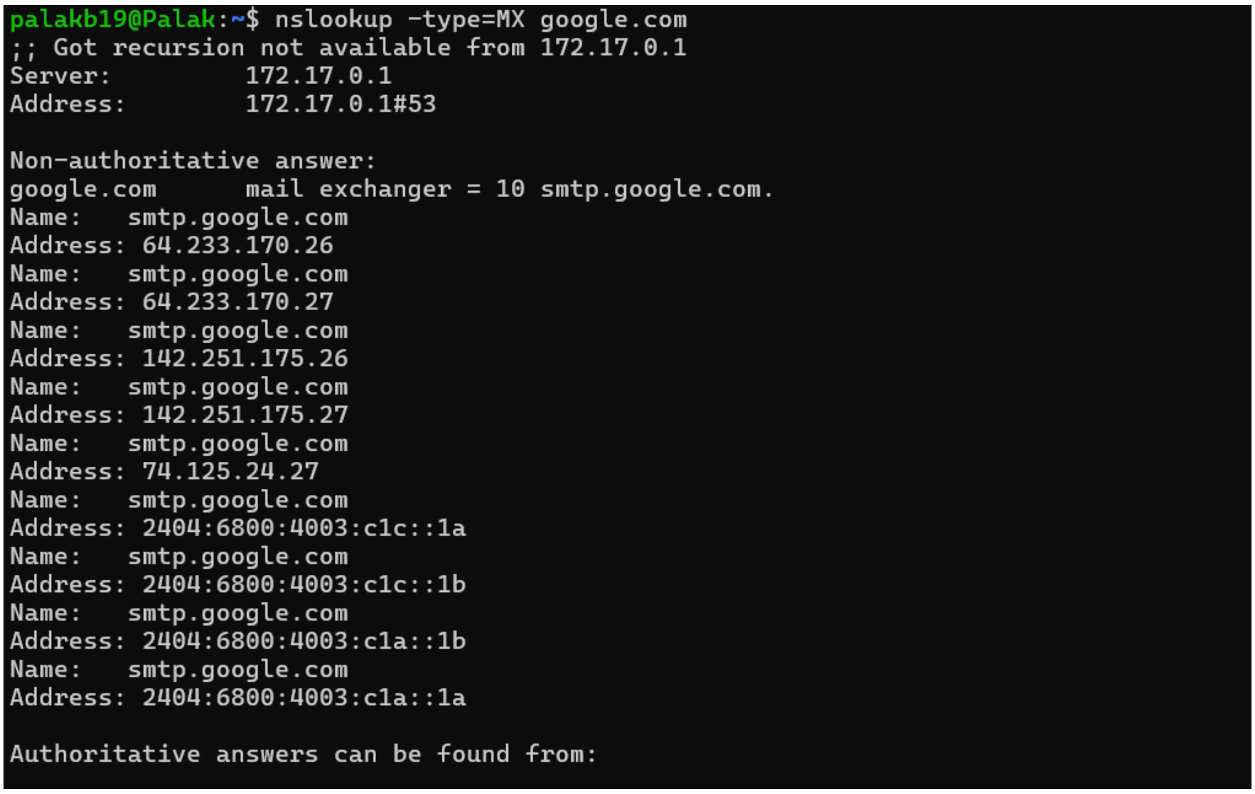


```
palakb19@Palak:~$ nslookup -type=AAAA google.com
;; Got recursion not available from 172.17.0.1
Server:         172.17.0.1
Address:        172.17.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 2404:6800:4002:82d::200e

palakb19@Palak:~$
```

FIGURE 24 – Queries the DNS server to return the AAAA record (IPv6 address) for the domain google.com

— IPv6 Address Query : This figure displays the result of querying the DNS server for an AAAA record, which provides the IPv6 address associated with a domain (e.g., google.com).

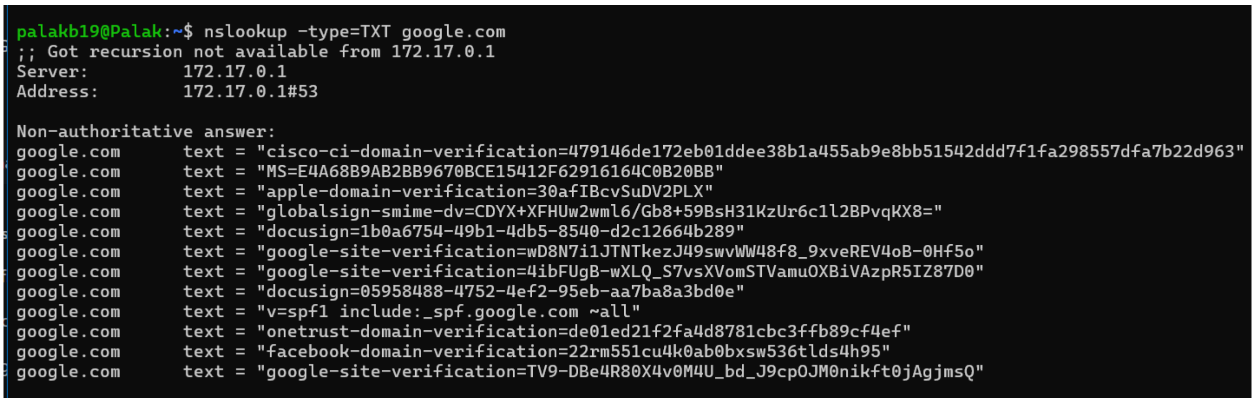
- DNS Record Type : The AAAA record is a DNS record used to map a domain name to its corresponding IPv6 address, as opposed to the A record used for IPv4 addresses.
- IPv6 Adoption : With the depletion of IPv4 addresses, IPv6 is increasingly becoming important, offering a vastly larger address space.
- Usage : The figure demonstrates the transition towards IPv6 and its growing relevance in modern networks, especially in regions with advanced networking infrastructure.
- Security Implications : IPv6 queries are crucial for maintaining connectivity in a world moving towards IPv6, with enhanced security features compared to IPv4.



```
palakb19@Palak:~$ nslookup -type=MX google.com
;; Got recursion not available from 172.17.0.1
Server:         172.17.0.1
Address:        172.17.0.1#53

Non-authoritative answer:
google.com      mail exchanger = 10 smtp.google.com.
Name:   smtp.google.com
Address: 64.233.170.26
Name:   smtp.google.com
Address: 64.233.170.27
Name:   smtp.google.com
Address: 142.251.175.26
Name:   smtp.google.com
Address: 142.251.175.27
Name:   smtp.google.com
Address: 74.125.24.27
Name:   smtp.google.com
Address: 2404:6800:4003:c1c::1a
Name:   smtp.google.com
Address: 2404:6800:4003:c1c::1b
Name:   smtp.google.com
Address: 2404:6800:4003:c1a::1b
Name:   smtp.google.com
Address: 2404:6800:4003:c1a::1a

Authoritative answers can be found from:
```

FIGURE 25 – Mail Exchange (MX) records for the domain google.com

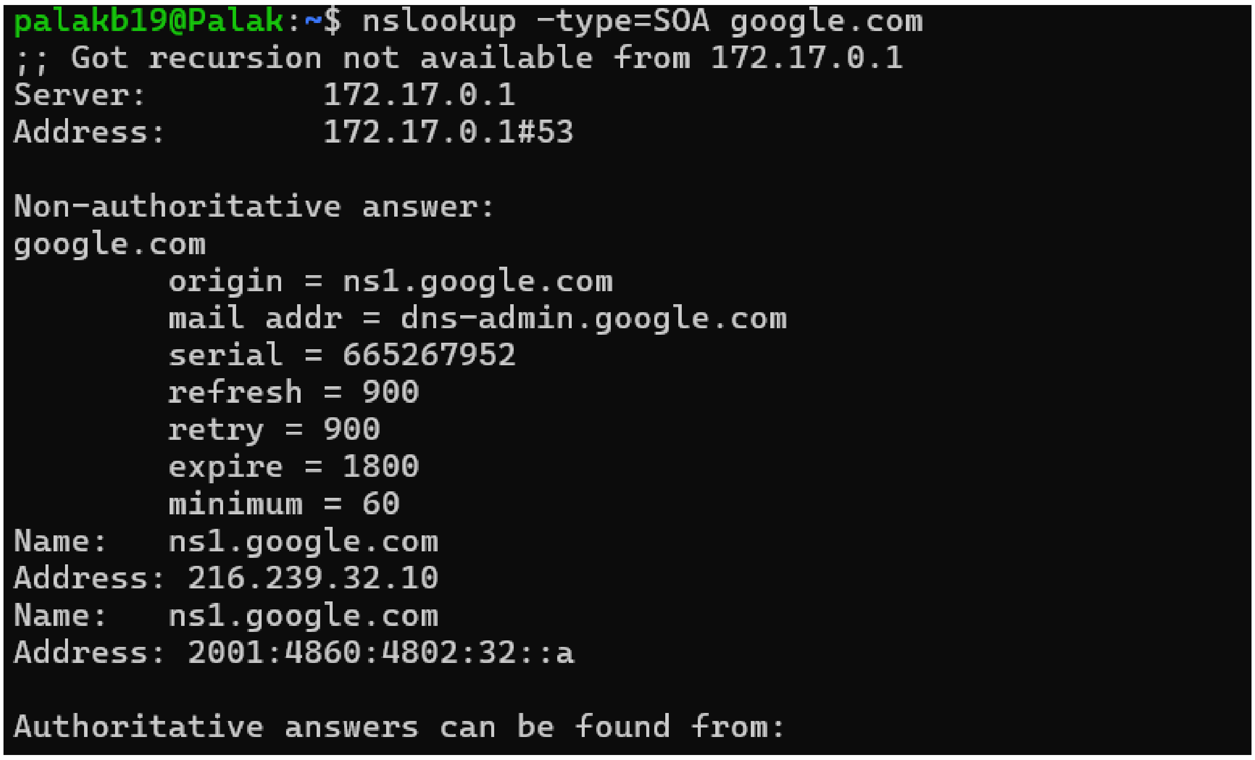
- This figure displays the result of querying the DNS for Mail Exchange (MX) records, which are used to identify the mail servers responsible for receiving emails for a domain (e.g., google.com).
- MX records determine how emails are routed within the network, ensuring that they reach the correct mail server.
- MX records often include a **priority value**, indicating the order in which mail servers should be contacted.
- Correct configuration of MX records is important for ensuring reliable email delivery and preventing disruptions in communication.
- Monitoring MX records helps in implementing security measures against spam and phishing attacks, as attackers often try to target the mail servers.



```
palakb19@Palak:~$ nslookup -type=TXT google.com
;; Got recursion not available from 172.17.0.1
Server:         172.17.0.1
Address:        172.17.0.1#53

Non-authoritative answer:
google.com      text = "cisco-ci-domain-verification=479146de172eb01ddee38b1a455ab9e8bb51542ddd7f1fa298557dfa7b22d963"
google.com      text = "MS=E4A68B9AB2BB9670BCE15412F62916164C0B20BB"
google.com      text = "apple-domain-verification=30afIBcvSuDV2PLX"
google.com      text = "globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="
google.com      text = "docusign=1b0a6754-49b1-4db5-8540-d2c12664b289"
google.com      text = "google-site-verification=wD8N7i1JTNTkezJ49swvWW48f8_9xveREV4oB-0Hf5o"
google.com      text = "google-site-verification=4ibFUgB-wXLQ_S7vsXVomSTVamuOXBiVAzpR5IZ87D0"
google.com      text = "docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
google.com      text = "v=spf1 include:_spf.google.com ~all"
google.com      text = "onetrust-domain-verification=de01ed21f2fa4d8781cbc3ffb89cf4ef"
google.com      text = "facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
google.com      text = "google-site-verification=TV9-DBe4R80X4v0M4U_bd_J9cpOJM0nikft0jAgjmsQ"
```

FIGURE 26 – Queries the DNS for TXT (Text) records associated with the domain google.com

— This figure shows the result of querying DNS for Text (TXT) records,.
— TXT records are commonly used for domain ownership verification, email security (e.g., SPF, DKIM).
— They play a significant role in email security, helps to prevent phishing by verifying the sender's domain.
— TXT records are flexible and can be used for different purposes, such as adding metadata to domains.
— A typical use of TXT records is to set up Sender Policy Framework (SPF) records,.



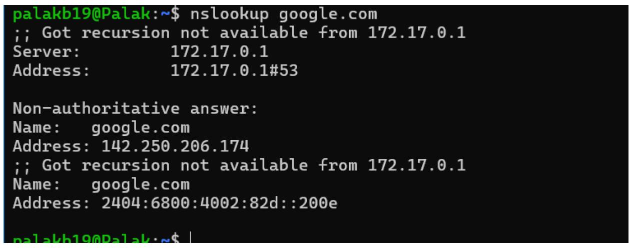
```
palakb19@Palak:~$ nslookup -type=SOA google.com
;; Got recursion not available from 172.17.0.1
Server:         172.17.0.1
Address:        172.17.0.1#53

Non-authoritative answer:
google.com
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 665267952
        refresh = 900
        retry = 900
        expire = 1800
        minimum = 60
Name:   ns1.google.com
Address: 216.239.32.10
Name:   ns1.google.com
Address: 2001:4860:4802:32::a

Authoritative answers can be found from:
```

FIGURE 27 – Queries the DNS for the SOA (Start of Authority) record of the domain google.com

Non-authoritative : This means that the DNS server isn't the authoritative DNS server for google.com, but it is providing cached or secondary data.
Origin (ns1.google.com) : This is the primary DNS server for the domain google.com. It's the server that holds the authoritative DNS records.

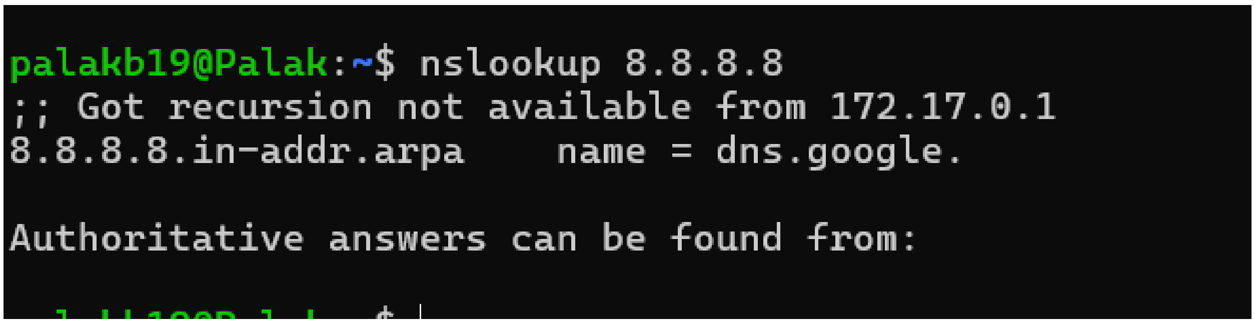
1.7 nslookup

ns lookup is also known as Name Server Lookup
Purpose : Queries DNS to obtain information about domain names and IP addresses.



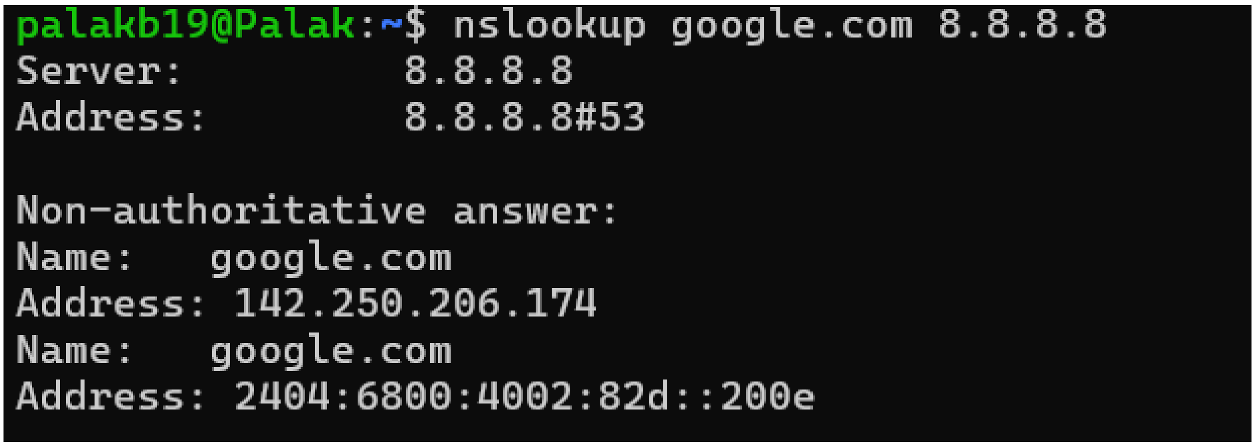
```
palakb19@Palak:~$ nslookup google.com
;; Got recursion not available from 172.17.0.1
Server:         172.17.0.1
Address:        172.17.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.206.174
;; Got recursion not available from 172.17.0.1
Name:   google.com
Address: 2404:6800:4002:82d::200e

palakb19@Palak:~$
```

FIGURE 28 – Queries the default DNS server to obtain the IP address of the domain google.com



```
palakb19@Palak:~$ nslookup 8.8.8.8
;; Got recursion not available from 172.17.0.1
8.8.8.8.in-addr.arpa    name = dns.google.

Authoritative answers can be found from:
```

FIGURE 29 – Queries the default DNS server to obtain the domain name associated with the IP address 8.8.8.8

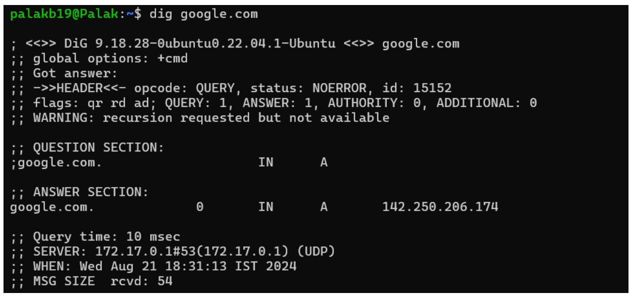


```
palakb19@Palak:~$ nslookup google.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.206.174
Name:   google.com
Address: 2404:6800:4002:82d::200e
```

FIGURE 30 – Queries the DNS server 8.8.8.8 (Google Public DNS) to obtain the IP address of example.com

- This figure demonstrates querying **Google's public DNS server (8.8.8.8)** to resolve the domain example.com.
- Public DNS services like Google's are widely used for their reliability, speed, and security features.
- Using a well-known public DNS can lead to faster domain resolution and improved internet browsing performance.
- Public DNS services often incorporate security features such as protection against DNS spoofing and cache poisoning.

— Besides Google, there are other public DNS providers like Cloudflare (1.1.1.1) and OpenDNS, each offering unique benefits for privacy and security.

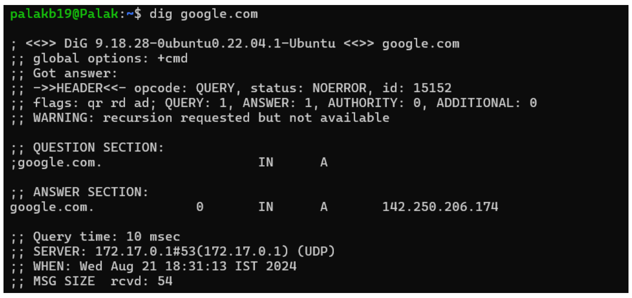


```
palakb19@Palak:~$ dig google.com

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15152
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             0       IN      A       142.250.206.174

;; Query time: 10 msec
;; SERVER: 172.17.0.1#53(172.17.0.1) (UDP)
;; WHEN: Wed Aug 21 18:31:13 IST 2024
;; MSG SIZE  rcvd: 54
```

FIGURE 31 – Queries the AAAA record for the domain google.com.

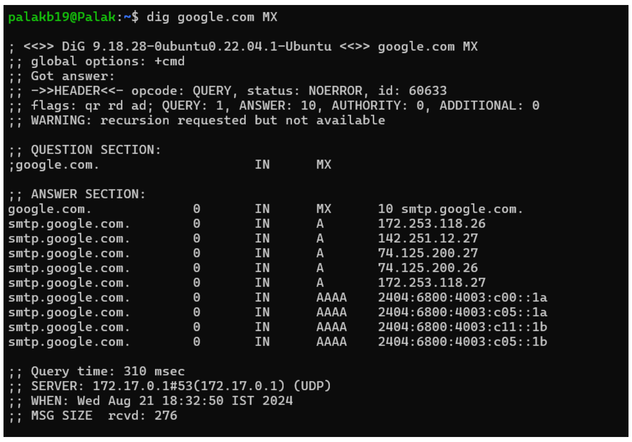
— Displays querying for an AAAA record to retrieve the IPv6 address of a domain (e.g., google.com).
— As the internet transitions to IPv6, DNS queries for AAAA records are becoming increasingly important.
— Expanded Addressing : IPv6 offers an expanded addressing capability, necessary for supporting the growing number of internet-connected devices.

1.8 dig



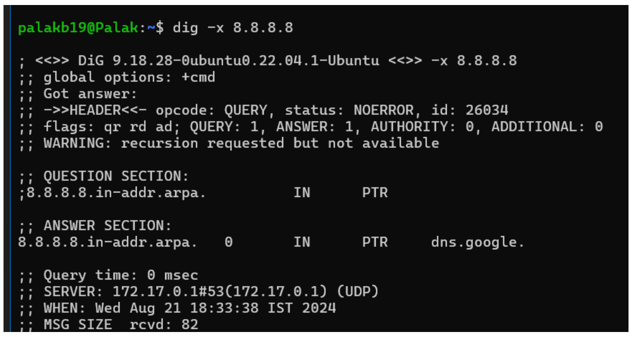
```
palakb19@Palak:~$ dig google.com

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15152
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             0       IN      A       142.250.206.174

;; Query time: 10 msec
;; SERVER: 172.17.0.1#53(172.17.0.1) (UDP)
;; WHEN: Wed Aug 21 18:31:13 IST 2024
;; MSG SIZE  rcvd: 54
```

FIGURE 32 – It shows that the domain google.com resolves to the IP address 142.250.206.174.



```
palakb19@Palak:~$ dig google.com MX

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> google.com MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60633
;; flags: qr rd ad; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;google.com.                    IN      MX

;; ANSWER SECTION:
google.com.             0       IN      MX      10 smtp.google.com.
smtp.google.com.        0       IN      A       172.253.118.26
smtp.google.com.        0       IN      A       142.251.12.27
smtp.google.com.        0       IN      A       74.125.200.27
smtp.google.com.        0       IN      A       74.125.200.26
smtp.google.com.        0       IN      A       172.253.118.27
smtp.google.com.        0       IN      AAAA    2404:6800:4003:c00::1a
smtp.google.com.        0       IN      AAAA    2404:6800:4003:c05::1a
smtp.google.com.        0       IN      AAAA    2404:6800:4003:c11::1b
smtp.google.com.        0       IN      AAAA    2404:6800:4003:c05::1b

;; Query time: 310 msec
;; SERVER: 172.17.0.1#53(172.17.0.1) (UDP)
;; WHEN: Wed Aug 21 18:32:50 IST 2024
;; MSG SIZE  rcvd: 276
```

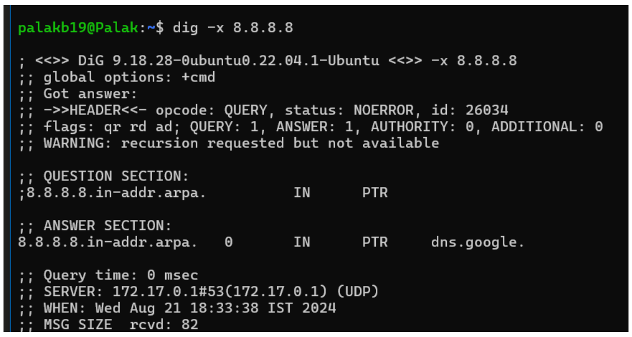
FIGURE 33 – Returns the mail servers configured for handling email for google.com.



```
palakb19@Palak:~$ dig -x 8.8.8.8

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> -x 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26034
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.   0       IN      PTR     dns.google.

;; Query time: 0 msec
;; SERVER: 172.17.0.1#53(172.17.0.1) (UDP)
;; WHEN: Wed Aug 21 18:33:38 IST 2024
;; MSG SIZE  rcvd: 82
```

FIGURE 34 – Displays the DNS records for twitter.com using the DNS server at IP address 8.8.8.8

— A detailed DNS query for twitter.com, using the Google public **DNS server (8.8.8.8)**, displaying various DNS records associated with the domain.
— The query returns multiple record types, such as A, AAAA, MX, and TXT records, providing a view of the domain's DNS configuration.
— Using Google's public DNS server ensures reliable and fast responses.
— DNS provides into a domain's infrastructure, which is important for administrators managing DNS configurations.
— Monitoring DNS records is essential for security, helping to detect unauthorized changes that might indicate a domain is under attack.



```
palakb19@Palak:~$ dig -x 8.8.8.8

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> -x 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26034
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.   0       IN      PTR     dns.google.

;; Query time: 0 msec
;; SERVER: 172.17.0.1#53(172.17.0.1) (UDP)
;; WHEN: Wed Aug 21 18:33:38 IST 2024
;; MSG SIZE  rcvd: 82
```

FIGURE 35 – reverse DNS lookup for the IP address 8.8.8.8.

— This figure demonstrates a reverse DNS lookup for the IP address 8.8.8.8, which is Google's public DNS server

2 Bibliography

Tut1 Slides, CSE232 Section B, Monsoon 2024