University of Dhaka

CSE:3111 Computer Networking Lab

Lab Report

Lab exercises on LAN configuration and troubleshooting tools

Shariful Islam Rayhan
Roll: 41
Md Sakib Ur Rahman
Roll: 37

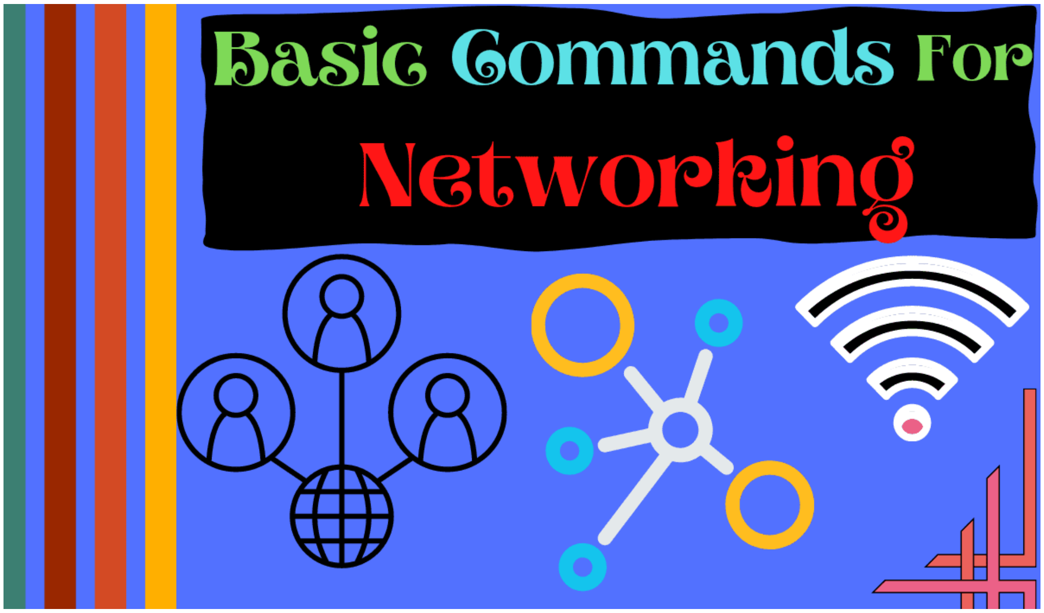
Submitted to:

Dr. Md. Abdur Razzaque
Dr. Muhammad Ibrahim
Dr. Md. Redwan Ahmed Rizvee
Dr. Md. Mamun Or Rashid

Submitted on: 2024-01-26

1 Introduction

Networking commands in the terminal are essential tools for managing and troubleshooting network configurations. These commands provide users with the ability to inspect, configure, and troubleshoot various aspects of network connectivity. In this lab report, we explore key networking commands and their applications, highlighting their significance in maintaining a robust and efficient network infrastructure.



2 Objectives

- To be familiar with different networking commands
- To know about different information related networking
- Visualising how actually commands work by executing them throug terminal

3 LAN configuration exercises

3.1 PING

he ping command is like saying ”Are you there?” to a computer. It sends a small message to a destination (like a website) and waits for a reply. It tells you how long the message took to go and come back. If the destination is reachable, you get a response; if not, there’s an issue. It’s a basic tool to check if a computer or website is connected and responsive.

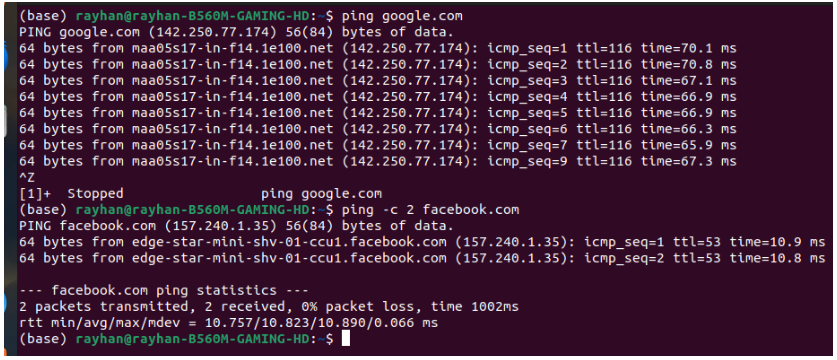
- ping google.com
- ping -c 2 google.com

Command Execution

To use PING, open the terminal and execute the following command:
$ping < destinationHost >$

Sample Output

Here is a sample output of a PING command:



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ ping google.com
PING google.com (142.250.77.174) 56(84) bytes of data.
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=1 ttl=116 time=70.1 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=2 ttl=116 time=70.8 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=3 ttl=116 time=67.1 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=4 ttl=116 time=66.9 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=5 ttl=116 time=66.9 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=6 ttl=116 time=66.3 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=7 ttl=116 time=65.9 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=9 ttl=116 time=67.3 ms
^Z
[1]+  Stopped                 ping google.com
(base) rayhan@rayhan-B560M-GAMING-HD:~$ ping -c 2 facebook.com
PING facebook.com (157.240.1.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-01-ccu1.facebook.com (157.240.1.35): icmp_seq=1 ttl=53 time=10.9 ms
64 bytes from edge-star-mini-shv-01-ccu1.facebook.com (157.240.1.35): icmp_seq=2 ttl=53 time=10.8 ms

--- facebook.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 10.757/10.823/10.890/0.066 ms
(base) rayhan@rayhan-B560M-GAMING-HD:~$ 
```

Analysis

The PING command is used in the sample output to transmit and receive ICMP (Internet Control Message Protocol) packets to the host that is supplied (in this case, `google.com`). The output contains general statistics and details like the round-trip time (rtt) for every packet.

You may evaluate the network performance and connectivity to the designated destination by examining the PING result.

Limiting the number of PING requests

By default, the PING command sends an unlimited number of requests to the destination. To limit the number of requests, use the `-c` option:

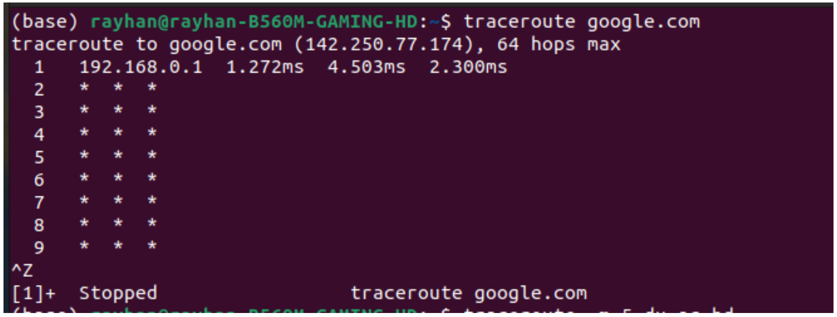
3.2 TRACEROUTE

The traceroute command is like a map for internet data. It shows the path your data takes to reach a destination. It reveals each ”stop” or ”hop” along the way, including the time it takes to get there. This helps identify where any delays or connection issues might be happening.Some versions of this commnad are itemized below:

- tracroute google.com
- tracroute -d google.com
- traceroute -N 5 google.com
- traceroute -m 5 google.com
- traceroute -q 2 google.com
- tracerote -n google.com

Command Execution

To use TRACEROUTE, open the terminal and execute the following command: *traceroute* $< destinationHost >$



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ traceroute google.com
traceroute to google.com (142.250.77.174), 64 hops max
  1   192.168.0.1  1.272ms  4.503ms  2.300ms
  2   *  *  *
  3   *  *  *
  4   *  *  *
  5   *  *  *
  6   *  *  *
  7   *  *  *
  8   *  *  *
  9   *  *  *
^Z
[1]+  Stopped                 traceroute google.com
(base) rayhan@rayhan-B560M-GAMING-HD:~$ traceroute -m 5 du.ac.bd
```

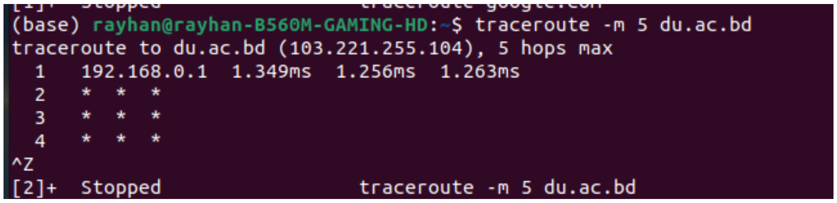
Number of hops is 30. It indicates the time to live (TTL) of the packet. The TTL value is decremented by one each time the packet is forwarded by a router. When the TTL value reaches zero, the packet is discarded and an ICMP error message is sent to the source. The source can use the ICMP error message to determine the IP address of the router that discarded the packet. The source can then use the IP address to determine the router name.

Limiting the number of hops

By default, the TRACEROUTE command sends packets with a TTL value of 1 and increments the TTL value by 1 for each subsequent packet. To limit the number of hops, use the `-m` option:

Sample Output

$traceroute - m < num >< destinationHost >$



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ traceroute -m 5 du.ac.bd
traceroute to du.ac.bd (103.221.255.104), 5 hops max
  1   192.168.0.1  1.349ms  1.256ms  1.263ms
  2   *  *  *
  3   *  *  *
  4   *  *  *
^Z
[2]+  Stopped                 traceroute -m 5 du.ac.bd
```

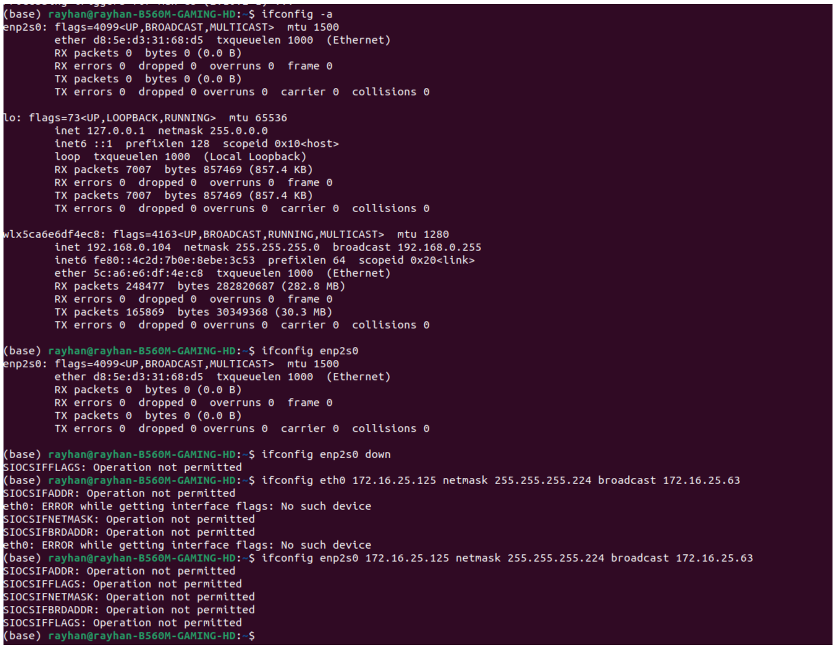
We use 5 hops. It indicates the time to live (TTL) of the packet. The TTL value is decremented by one each time the packet is forwarded by a router. When the TTL value reaches zero, the packet is discarded and an ICMP error message is sent to the source. The source can use the ICMP error message to determine the IP address of the router that discarded the packet. The source can then use the IP address to determine the router name.

3.3 IFCONFIG

The ifconfig command is like looking at the ID card of computer. It shows information about network interfaces, like computer's address and how it's connected to the internet. It's handy for checking and configuring network settings.

This command displays details about your network interfaces, such as IP addresses, MAC (physical) addresses, and current status. It gives you a snapshot of how your computer is connected to the network.

Command Execution



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ ifconfig -a
enp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d8:5e:d3:31:68:d5  txqueuelen 1000  (Ethernet)
        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

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 7007  bytes 857469 (857.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7007  bytes 857469 (857.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlx5ca6e6df4ec8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1280
        inet 192.168.0.104  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::4c2d:7b0e:8ebe:3c53  prefixlen 64  scopeid 0x20<link>
        ether 5c:a6:e6:df:4e:c8  txqueuelen 1000  (Ethernet)
        RX packets 248477  bytes 282820687 (282.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 165869  bytes 30349368 (30.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

(base) rayhan@rayhan-B560M-GAMING-HD:~$ ifconfig enp2s0
enp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d8:5e:d3:31:68:d5  txqueuelen 1000  (Ethernet)
        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

(base) rayhan@rayhan-B560M-GAMING-HD:~$ ifconfig enp2s0 down
SIOCSIFFLAGS: Operation not permitted
(base) rayhan@rayhan-B560M-GAMING-HD:~$ ifconfig eth0 172.16.25.125 netmask 255.255.255.224 broadcast 172.16.25.63
SIOCSIFADDR: Operation not permitted
eth0: ERROR while getting interface flags: No such device
SIOCSIFNETMASK: Operation not permitted
SIOCSIFBRDADDR: Operation not permitted
eth0: ERROR while getting interface flags: No such device
(base) rayhan@rayhan-B560M-GAMING-HD:~$ ifconfig enp2s0 172.16.25.125 netmask 255.255.255.224 broadcast 172.16.25.63
SIOCSIFADDR: Operation not permitted
SIOCSIFFLAGS: Operation not permitted
SIOCSIFNETMASK: Operation not permitted
SIOCSIFBRDADDR: Operation not permitted
SIOCSIFFLAGS: Operation not permitted
(base) rayhan@rayhan-B560M-GAMING-HD:~$
```

Analysis

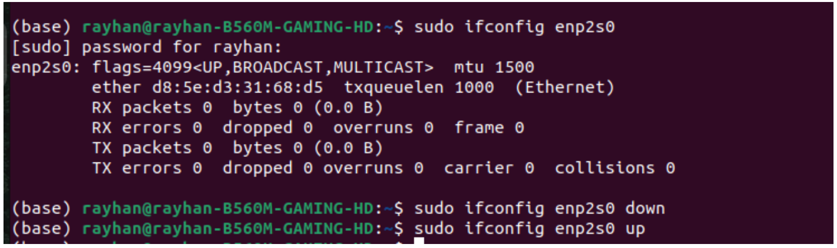
In the sample output, the IFCONFIG command displays the IP address, subnet mask, and default gateway for the device. The output also includes information such as the MAC address, MTU, and the number of packets transmitted and received.

Disable a network interface

To disable a network interface, use the `down` option:

Sample Output

$sudoifconfig < interfaceName > down$



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ sudo ifconfig enp2s0
[sudo] password for rayhan:
enp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d8:5e:d3:31:68:d5  txqueuelen 1000  (Ethernet)
        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

(base) rayhan@rayhan-B560M-GAMING-HD:~$ sudo ifconfig enp2s0 down
(base) rayhan@rayhan-B560M-GAMING-HD:~$ sudo ifconfig enp2s0 up
```

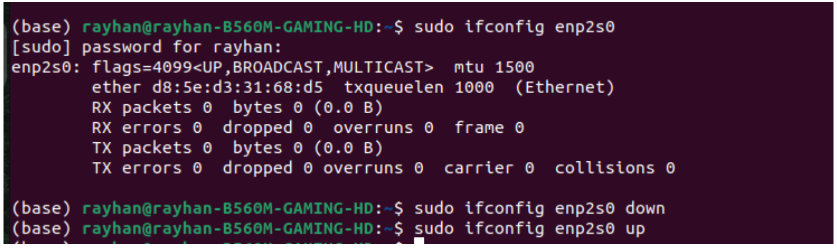
The network interface is down.

Enable a network interface

To enable a network interface, use the `up` option:

Sample Output

$sudoifconfig < interfaceName > up$



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ sudo ifconfig enp2s0
[sudo] password for rayhan:
enp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d8:5e:d3:31:68:d5  txqueuelen 1000  (Ethernet)
        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

(base) rayhan@rayhan-B560M-GAMING-HD:~$ sudo ifconfig enp2s0 down
(base) rayhan@rayhan-B560M-GAMING-HD:~$ sudo ifconfig enp2s0 up
```

The network interface is up.

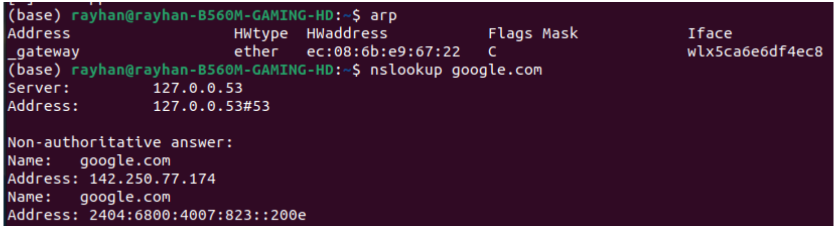
3.4 ARP

The arp command is like a phone book for your computer's network. It helps match IP addresses to physical MAC addresses. When your computer wants to talk to another device on the same network, it uses ARP to find the physical address associated with the IP address.

This command shows the ARP table, a list of IP addresses and their corresponding MAC addresses that your computer has recently communicated with. It's like a record of who your computer has been talking to on the network.

Command Execution

To use ARP, open the terminal and execute the following command: *arp*



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   ec:08:6b:e9:67:22   C                     wlx5ca6e6df4ec8
(base) rayhan@rayhan-B560M-GAMING-HD:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.77.174
Name:   google.com
Address: 2404:6800:4007:823::200e
```

Analysis

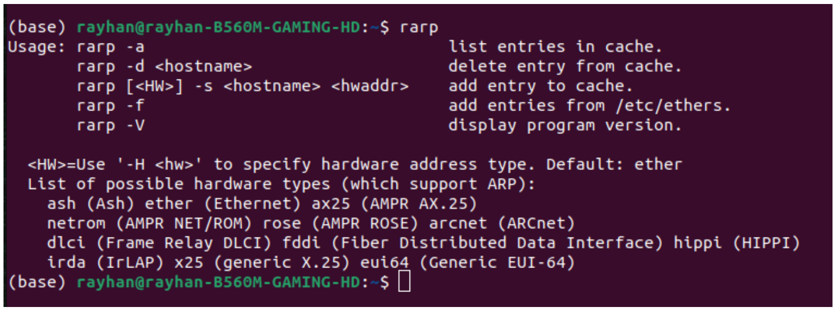
In the sample output, the ARP command displays the ARP cache for the device. The output includes information such as the IP address, MAC address, and type of each entry.

3.5 RARP

The rarp command is like asking, "Who am I?" to network. It's used to obtain the IP address associated with your computer's MAC address. While less common today, RARP historically helped diskless systems find their IP addresses.

Command Execution

To use RARP, open the terminal and execute the following command: *rarp*



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ rarp
Usage: rarp -a                               list entries in cache.
       rarp -d <hostname>                    delete entry from cache.
       rarp [<HW>] -s <hostname> <hwaddr>    add entry to cache.
       rarp -f                               add entries from /etc/ethers.
       rarp -V                               display program version.

  <HW>=Use '-H <hw>' to specify hardware address type. Default: ether
  List of possible hardware types (which support ARP):
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
    dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) x25 (generic X.25) eui64 (Generic EUI-64)
(base) rayhan@rayhan-B560M-GAMING-HD:~$ 
```

Analysis

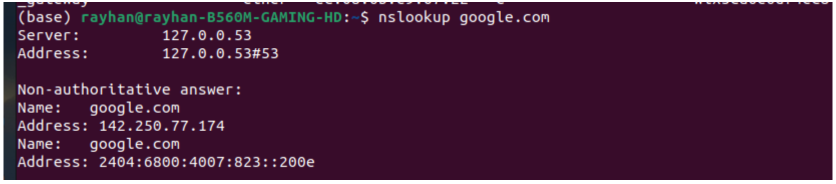
In the sample output, the RARP command displays the RARP cache for the device. The output includes information such as the IP address, MAC address, and type of each entry.

3.6 NSLOOKUP

Using the `nslookup` command, display the DNS cache for a device. Record the results.

Command Execution

To use NSLOOKUP, open the terminal and execute the following command: *nslookup*



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.77.174
Name:   google.com
Address: 2404:6800:4007:823::200e
```

Analysis

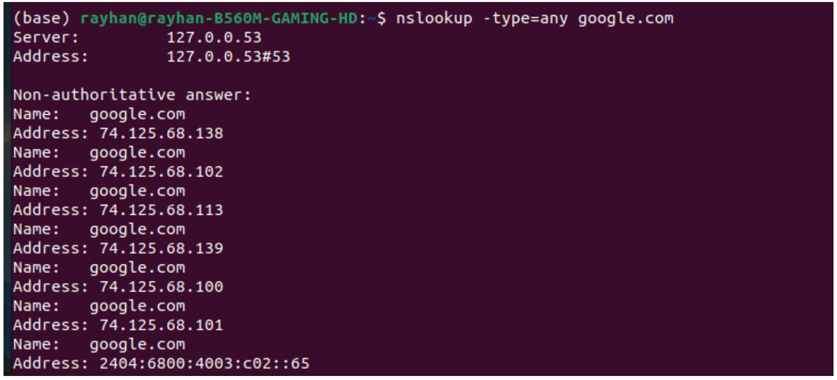
In the sample output, the NSLOOKUP command displays the DNS cache for the device. The output includes information such as the IP address, hostname, and type of each entry. nslookup followed by the domain name will display the “A Record” (IP Address) of the domain. Use this command to find the address record for a domain. It queries domain name servers and gets the details.

Using type any

To display all the information in the DNS cache, use the `set type=any` option:

Sample Output

$nslookup - type = any < domainName >$



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ nslookup -type=any google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 74.125.68.138
Name:   google.com
Address: 74.125.68.102
Name:   google.com
Address: 74.125.68.113
Name:   google.com
Address: 74.125.68.139
Name:   google.com
Address: 74.125.68.100
Name:   google.com
Address: 74.125.68.101
Name:   google.com
Address: 2404:6800:4003:c02::65
```

There are also available types of DNS records. The most common ones are:

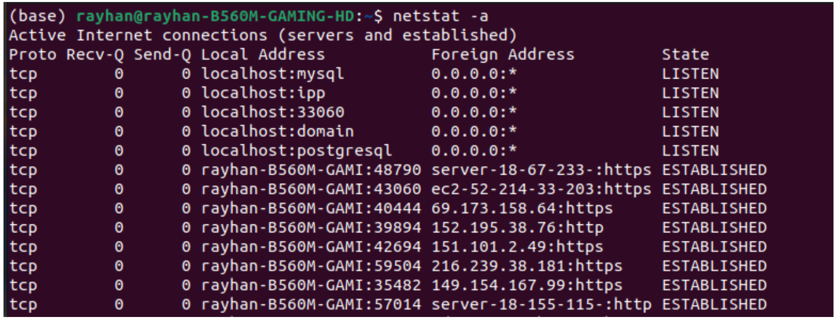
- A: Address record
- AAAA: IPv6 address record
- CNAME: Canonical name record
- MX: Mail exchange record
- NS: Name server record
- PTR: Pointer record
- SOA: Start of authority record
- TXT: Text record

3.7 NETSTAT

Using the `netstat` command, display the active TCP connections for a device. Record the results. The netstat command is like a special tool in Linux that helps you understand and check things about how your computer connects to the internet. It can tell you about the connections your computer is making, the paths it uses to send information, and even some technical details like how many packets of data are being sent or received. In simple terms, it's like a window that shows you what's happening with your computer and the internet. This article will help you learn how to use netstat, exploring different ways to get specific information and giving you a better idea of what's going on behind the scenes.

Command Execution

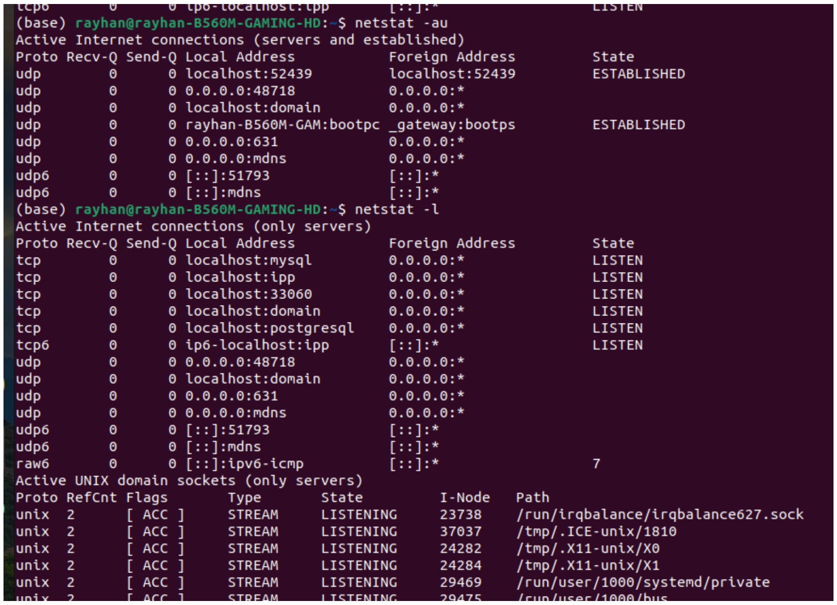
To use NETSTAT, open the terminal and execute the following command: *netstat*



```
(base) rayhan@rayhan-B560M-GAMING-HD:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:postgresql    0.0.0.0:*               LISTEN
tcp        0      0 rayhan-B560M-GAMI:48790 server-18-67-233-:https ESTABLISHED
tcp        0      0 rayhan-B560M-GAMI:43060 ec2-52-214-33-203:https ESTABLISHED
tcp        0      0 rayhan-B560M-GAMI:40444 69.173.158.64:https     ESTABLISHED
tcp        0      0 rayhan-B560M-GAMI:39894 152.195.38.76:http      ESTABLISHED
tcp        0      0 rayhan-B560M-GAMI:42694 151.101.2.49:https      ESTABLISHED
tcp        0      0 rayhan-B560M-GAMI:59504 216.239.38.181:https    ESTABLISHED
tcp        0      0 rayhan-B560M-GAMI:35482 149.154.167.99:https    ESTABLISHED
tcp        0      0 rayhan-B560M-GAMI:57014 server-18-155-115-:http ESTABLISHED
```

Analysis

In the sample output, the NETSTAT command displays the active TCP connections for the device. The output includes information such as the protocol, local address, foreign address, and state of each connection.



```
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
(base) rayhan@rayhan-B560M-GAMING-HD:~$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:52439         localhost:52439         ESTABLISHED
udp        0      0 0.0.0.0:48718           0.0.0.0:*
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 rayhan-B560M-GAM:bootpc _gateway:bootps         ESTABLISHED
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp6       0      0 [::]:51793              [::]:*
udp6       0      0 [::]:mdns               [::]:*
(base) rayhan@rayhan-B560M-GAMING-HD:~$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:postgresql    0.0.0.0:*               LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
udp        0      0 0.0.0.0:48718           0.0.0.0:*
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp6       0      0 [::]:51793              [::]:*
udp6       0      0 [::]:mdns               [::]:*
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     23738    /run/irqbalance/irqbalance627.sock
unix  2      [ ACC ]     STREAM     LISTENING     37037    /tmp/.ICE-unix/1810
unix  2      [ ACC ]     STREAM     LISTENING     24282    /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     24284    /tmp/.X11-unix/X1
unix  2      [ ACC ]     STREAM     LISTENING     29469    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     29475    /run/user/1000/bus
```

4 Experiences

- We have learnt how to run networking commands.
- We can modify ip address,netmask etc by this commands.
- We can see how many tcp/ip,udp,unix connection is available.
- Wc can find IP address and MAC address vice versa.
- We can find how many network interface is running on my machine

References

[1] PING : https://pimylifeup.com/ubuntu-ping/

[2] Ifconfig : https://www.tecmint.com/ifconfig-command-examples/

[3] Traceroute : https://cloudinfrastructureservices.co.uk/how-to-install-traceroute-and-run-on-ubuntu-20-04/

[4] ARP : https://www.geeksforgeeks.org/arp-command-in-linux-with-examples/

[5] RARP: https://www.geeksforgeeks.org/what-is-rarp/

[6] NSLookUP : https://www.geeksforgeeks.org/nslookup-command-in-linux-with-examples/

[7] Netstat :
https://www.geeksforgeeks.org/netstat-command-linux/