University of Dhaka

Department of Computer Science and Engineering

CSE-3111 : Computer Networking Lab

Lab Report 1 :

Lab exercises on LAN configuration and troubleshooting tools (PING, Traceroute, ARP, RARP, ifconfig, nslook)

Submitted By:

Name : MD. Farhan Tanvir

Roll No : 45

Name : Asef Sami Chowdhury

Roll No : 53

Submitted On :

January 26, 2024

Submitted To :

Dr. Md. Abdur Razzaque

Dr. Md. Mamun Or Rashid

Dr. Md. Muhammad Ibrahim

Md. Redwan Ahmed Rizvee

1 Introduction

The suite of essential networking commands, including Ping, Traceroute, IFCONFIG, ARP, RARP, NSLOOKUP, and NETSTAT, forms the backbone of network diagnostics in a Unix/Linux environment. These commands collectively empower users to assess and troubleshoot network connectivity, resolve address mappings, inspect interface configurations, and scrutinize active network connections. Ping is utilized for basic reachability tests, Traceroute unveils the network path to a destination, IFCONFIG manages network interfaces, ARP resolves IP addresses to MAC addresses, RARP aids in reverse address resolution, NSLOOKUP handles domain name queries, and NETSTAT provides insights into active connections. Mastery of these commands is foundational for network administrators, enabling them to analyze and optimize network performance seamlessly.

1.1 Objectives

The primary objective of this report is to document and analyze the laboratory exercises focused on Local Area Network (LAN) configuration and the utilization of essential troubleshooting tools. The exercises center around practical applications of tools such as PING, Traceroute, ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol), ifconfig, and nslookup.

1.1.1 LAN Configuration Exploration

Investigate and comprehend the fundamental principles of LAN configuration. Gain hands-on experience in setting up and configuring a Local Area Network. Explore the nuances of LAN architecture, including IP addressing, subnetting, and network topology.

1.1.2 Troubleshooting Tools Application

Develop proficiency in using PING as a fundamental tool for network connectivity testing. Explore Traceroute to track real-time pathways and identify potential issues in data packet routes. Understand the role of ARP in resolving IP addresses to physical MAC addresses and troubleshoot related problems. Investigate RARP as a tool for reverse address resolution, mapping MAC addresses to IP addresses.

1.1.3 Network Interface Configuration with ifconfig

This segment involves learning the essentials of ifconfig, a utility for system and network administration in Unix/Linux environments. The emphasis is on acquiring practical skills in configuring, managing, and querying network interface parameters through the command-line interface.

1.1.4 DNS Query and Troubleshooting with nslookup

Here, the objective is to gain insights into the functionality of nslookup as a versatile tool for querying the Domain Name System (DNS). The focus extends to understanding how nslookup facilitates obtaining domain name or IP address mappings, as well as its role in troubleshooting DNS-related issues.

1.1.5 Comprehensive Troubleshooting Skills

The final phase aims at developing a holistic understanding of troubleshooting methodologies for LANs. This involves leveraging a combination of tools to identify and efficiently address common network issues. The acquired knowledge and practical skills from the entire exploration contribute to a comprehensive troubleshooting skill set.

2 Theory

2.1 PING

PING (Packet Internet Groper) is a fundamental tool for testing network connectivity. It sends ICMP(Internet Control Message Protocol) echo requests to a target device and measures the response time, aiding in identifying connectivity problems.The ping command sends a request over the network to a specific device. A successful ping results in a response from the computer that was pinged back to the originating computer.

PING is primary Internet software programs that enable users to check and verify whether specific destination IP addresses exist and are capable of accepting requests in computer networks.By sending a ping packet using ICMP(Internet Control Message Protocol) we can estimate how long it will take to attain target device/ host.The ping command works by sending a small packet of data called an ICMP Echo Request to a target device, and then waiting for a response called an ICMP Echo Reply.

2.2 Traceroute

Traceroute is a network diagnostic tool that tracks real time pathways taken by a packet on an IP network from source to destination. After, it reports the IP addresses of all the routers it pinged in between. This tool also records each hop's time to make packets during its route to the destination.

Traceroute's key features include pathway tracking, real-time monitoring, response time measurement, variable TTL usage, multiple queries for accuracy, and platform integration into operating systems like Windows and Linux.

In terms of operation, Traceroute utilizes the Time to Live (TTL) field in IP packet headers to trigger TTL exceeded messages from each hop. It initiates with a low TTL value, sending packets incrementally until reaching the destination. The tool records IP addresses and response times for each hop, measuring Round Trip Time (RTT) for accuracy.

Traceroute is an essential tool for real-time network troubleshooting and optimization. By leveraging TTL values and ICMP messages, it provides detailed insights into network pathways, empowering users to identify and address potential issues efficiently.

2.3 ifconfig

ifconfig, short for "interface configuration," is a vital tool in Unix/Linux operating systems for system and network administration. It enables the configuration, management, and querying of network interface parameters via the command-line interface or system configuration scripts.

It serves key functions such as displaying current network configurations, enabling or disabling interfaces, assigning IP addresses, defining netmask and broadcast addresses, configuring Maximum Transmission Unit (MTU), managing promiscuous mode for packet analysis, creating aliases for interfaces, and modifying MAC addresses. Despite its deprecation in modern Linux distributions in favor of the ip command, ifconfig remains essential for managing network interfaces, especially on legacy systems.

ifconfig, although deprecated, remains essential for configuring and managing network interfaces, particularly on legacy systems. Its versatile functionality makes it valuable for network administrators dealing with diverse network configurations.

2.4 ARP

arp command manipulates the System's ARP cache. It also allows a complete dump of the ARP cache. ARP stands for Address Resolution Protocol. The primary function of this protocol is to resolve the IP address of a system to its mac address, and hence it works between level 2(Data link layer) and level 3(Network layer).

The Address Resolution Protocol (ARP) command is a network utility used for managing the ARP cache, which contains mappings of IP addresses to physical MAC addresses on a local network. Commonly available on Unix/Linux and Windows systems, the ARP command allows users to display the ARP cache, clear its entries, manually add or delete specific mappings, and obtain help on its usage. It is a crucial tool for troubleshooting network connectivity issues and ensuring accurate IP-to-MAC address resolution in local network communication.

2.5 RARP

The Reverse Address Resolution Protocol (RARP) is a networking protocol that is used to map a physical (MAC) address to an Internet Protocol (IP) address. It is the reverse of the more commonly used Address Resolution Protocol (ARP), which maps an IP address to a MAC address.

The Reverse Address Resolution Protocol (RARP) operates on the Network Access Layer and facilitates data transmission between network points. In a network, each participant possesses both an IP address (logical) and a MAC address (physical). The IP address is assigned by software, while the MAC address is embedded in the hardware. A RARP server, usually any computer in the network, must store mappings of MAC addresses to their respective IP addresses. When a RARP request is broadcasted, only these servers can respond, sending information packets at the lowest network layers simultaneously to all participants. The client initiates a RARP request with its Ethernet broadcast and physical addresses, and the server replies by providing the client with its corresponding IP address.

2.6 Nslookup

Nslookup, short for "Name Server Lookup," is a vital command used for querying the Domain Name System (DNS) and retrieving information from DNS servers. Functioning as a network administration tool, Nslookup facilitates the mapping of domain names to IP addresses or retrieves specific DNS records. It serves as a valuable resource for troubleshooting DNS-related issues, providing administrators with insights into the DNS infrastructure.

Administrators utilize the nslookup command to obtain essential information from DNS servers, aiding in the resolution of domain name or IP address mapping discrepancies. The command proves particularly beneficial for diagnosing and addressing DNS-related problems within a network. Various options such as -type=a, -type=any, -type=mx, -type=ns, -type=ptr, and -type=soa enhance the versatility of Nslookup, allowing administrators to retrieve specific types of DNS records.

Nslookup stands as a crucial tool for network administrators, offering a means to query DNS servers, troubleshoot DNS-related issues, and gain valuable insights into the DNS infrastructure. Its multifaceted functionality contributes to efficient problem resolution and maintenance of seamless DNS operations within a network.

2.7 Netstat

Netstat, short for "network statistics," is a powerful command in Linux designed to provide users with insights into how their computer interacts with the internet. Serving as a specialized tool, netstat offers a window into the connections, data paths, and technical details of network activities. It plays a crucial role in diagnosing networking issues, allowing users to monitor and understand the dynamics of their computer's internet connectivity.

The netstat command in Linux acts as a diagnostic tool, displaying a variety of network-related information. It reveals details about the connections established by the computer, the paths used for information exchange, and technical specifics such as the number of data packets transmitted or received. Essentially, netstat serves as a transparent window, offering users a real-time view of their computer's interactions with the internet.

Netstat is an invaluable command for Linux users, empowering them to monitor, analyze, and troubleshoot network-related issues effectively. Its versatility, coupled with practical examples and clear differentiation from other commands, makes it an essential tool for gaining insights into the intricacies of computer connectivity with the internet.

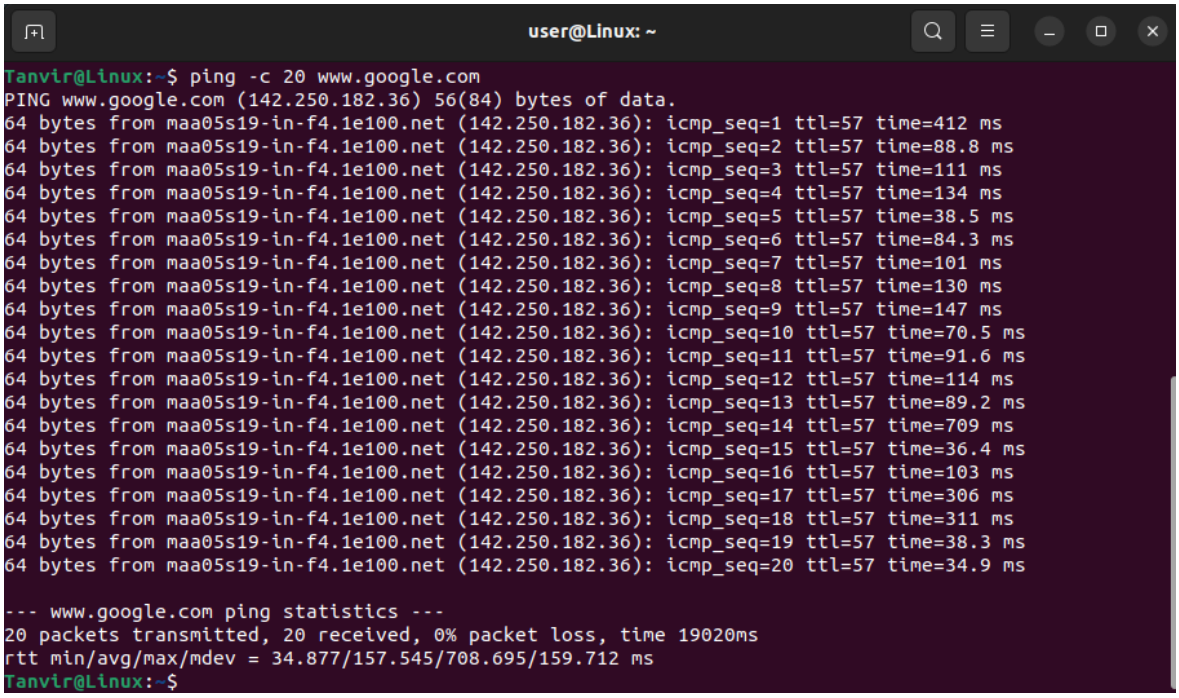
3 Methodology

Commenced with foundational knowledge, explored diverse switches and options, delved into relevant online resources, and analyzed a spectrum of network configurations and usage statistics

4 Experimental result

Here is some experimental result of PING, TRACEROUTE, IFCONFIG, ARP, RARP, NSLOOKUP, and NETSTAT I put here each of the creenshort for every task:

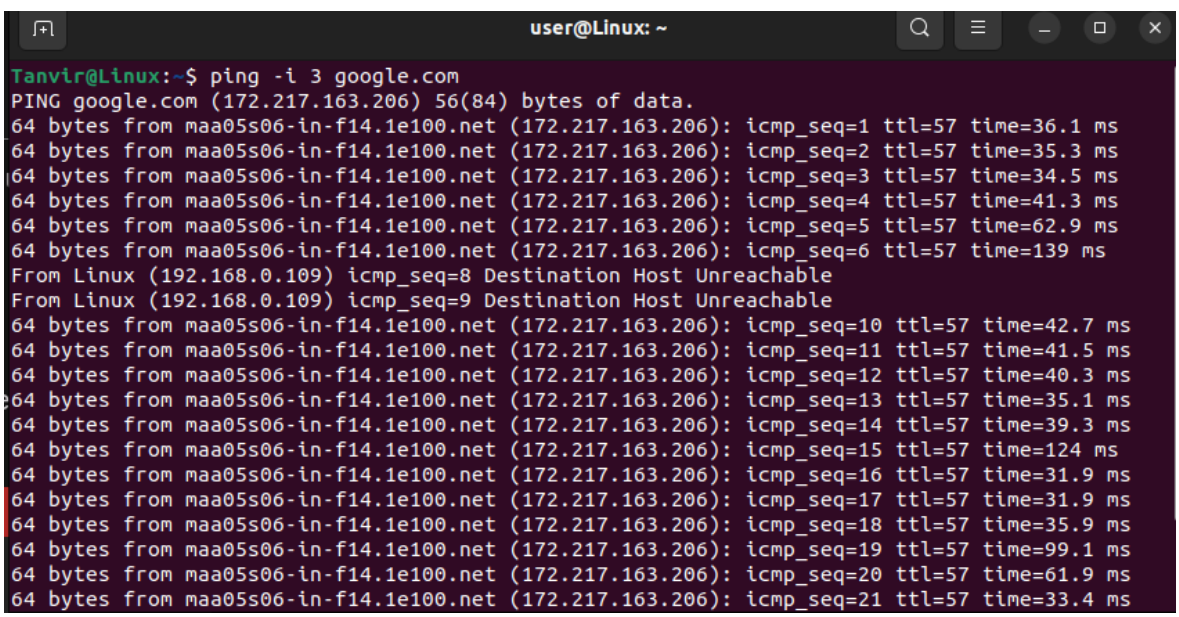
* **Ping Command**



```
user@Linux: ~
Tanvir@Linux:~$ ping -c 20 www.google.com
PING www.google.com (142.250.182.36) 56(84) bytes of data.
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=1 ttl=57 time=412 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=2 ttl=57 time=88.8 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=3 ttl=57 time=111 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=4 ttl=57 time=134 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=5 ttl=57 time=38.5 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=6 ttl=57 time=84.3 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=7 ttl=57 time=101 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=8 ttl=57 time=130 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=9 ttl=57 time=147 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=10 ttl=57 time=70.5 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=11 ttl=57 time=91.6 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=12 ttl=57 time=114 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=13 ttl=57 time=89.2 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=14 ttl=57 time=709 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=15 ttl=57 time=36.4 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=16 ttl=57 time=103 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=17 ttl=57 time=306 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=18 ttl=57 time=311 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=19 ttl=57 time=38.3 ms
64 bytes from maa05s19-in-f4.1e100.net (142.250.182.36): icmp_seq=20 ttl=57 time=34.9 ms

--- www.google.com ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19020ms
rtt min/avg/max/mdev = 34.877/157.545/708.695/159.712 ms
Tanvir@Linux:~$
```

Figure 1: Ping for google.com



```
user@Linux: ~
Tanvir@Linux:~$ ping -i 3 google.com
PING google.com (172.217.163.206) 56(84) bytes of data.
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=1 ttl=57 time=36.1 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=2 ttl=57 time=35.3 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=3 ttl=57 time=34.5 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=4 ttl=57 time=41.3 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=5 ttl=57 time=62.9 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=6 ttl=57 time=139 ms
From Linux (192.168.0.109) icmp_seq=8 Destination Host Unreachable
From Linux (192.168.0.109) icmp_seq=9 Destination Host Unreachable
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=10 ttl=57 time=42.7 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=11 ttl=57 time=41.5 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=12 ttl=57 time=40.3 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=13 ttl=57 time=35.1 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=14 ttl=57 time=39.3 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=15 ttl=57 time=124 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=16 ttl=57 time=31.9 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=17 ttl=57 time=31.9 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=18 ttl=57 time=35.9 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=19 ttl=57 time=99.1 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=20 ttl=57 time=61.9 ms
64 bytes from maa05s06-in-f14.1e100.net (172.217.163.206): icmp_seq=21 ttl=57 time=33.4 ms
```

Figure 2: with an interval of 3 seconds to ping Google's domain

* **Traceroute Command**

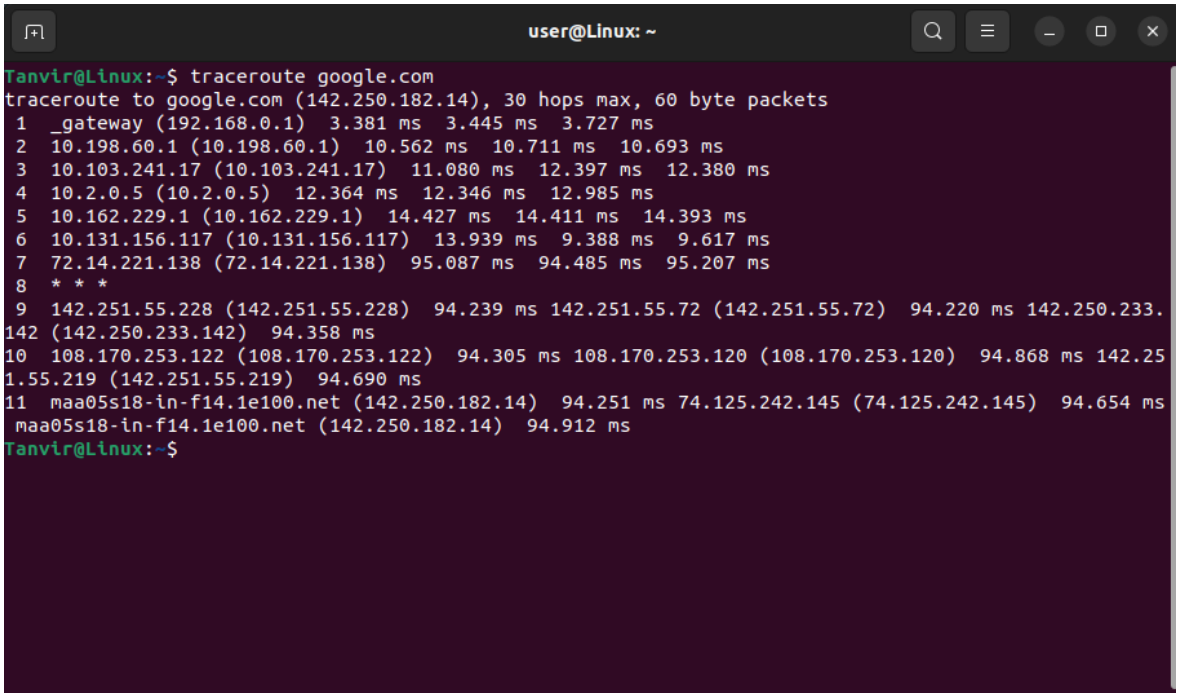


Figure 3: Traceroute

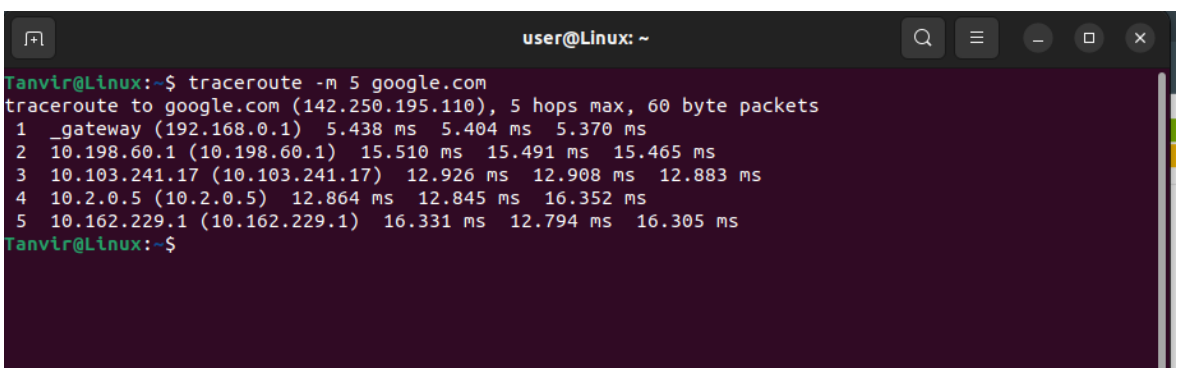
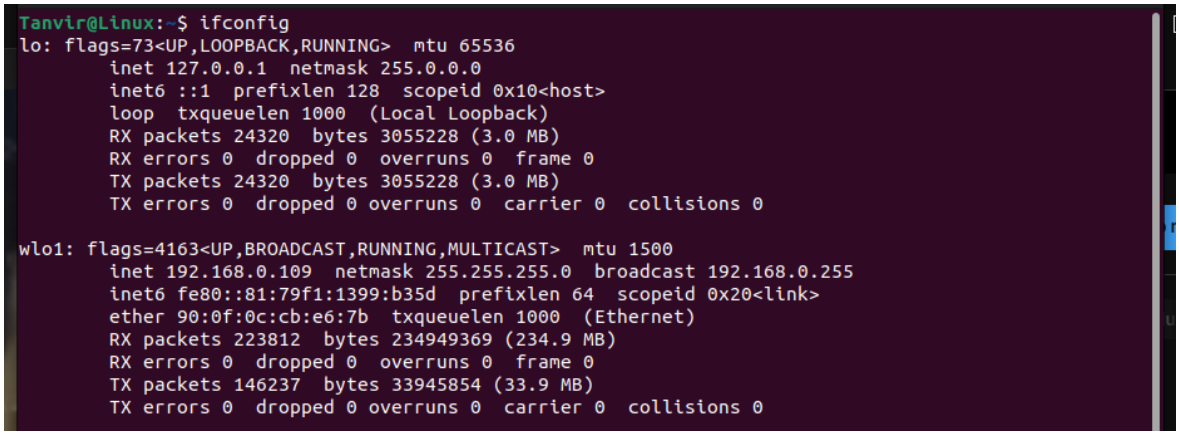


Figure 4: Traceroute with 5 probs

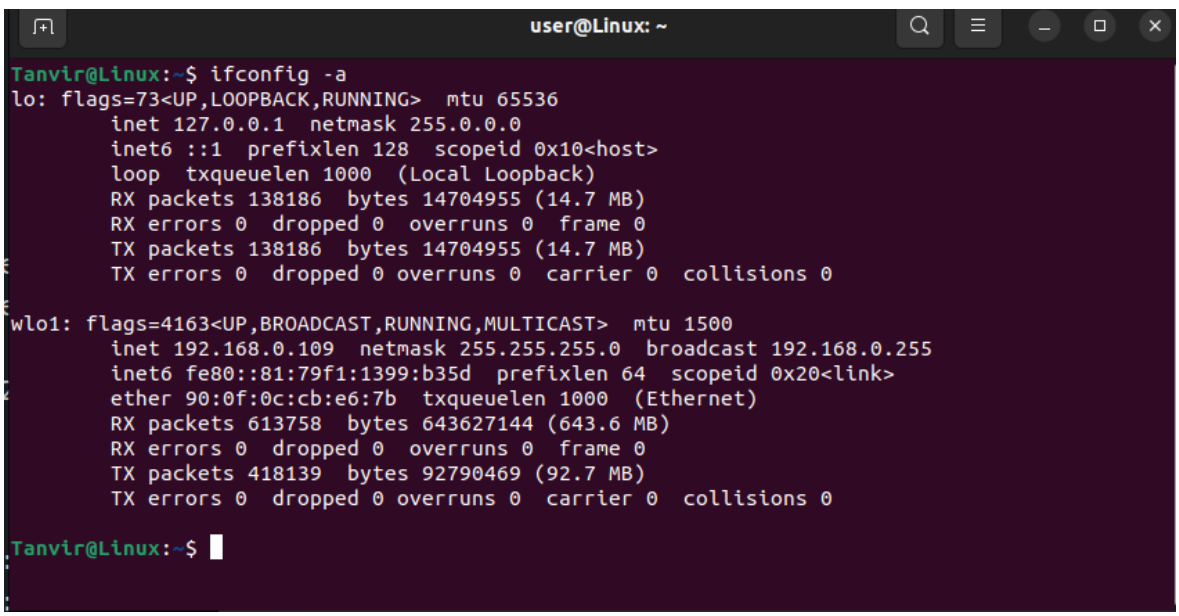
* **Ifconfig Command**



```
Tanvir@Linux:~$ 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 24320  bytes 3055228 (3.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24320  bytes 3055228 (3.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.109  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::81:79f1:1399:b35d  prefixlen 64  scopeid 0x20<link>
        ether 90:0f:0c:cb:e6:7b  txqueuelen 1000  (Ethernet)
        RX packets 223812  bytes 234949369 (234.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 146237  bytes 33945854 (33.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 5: Ifconfig



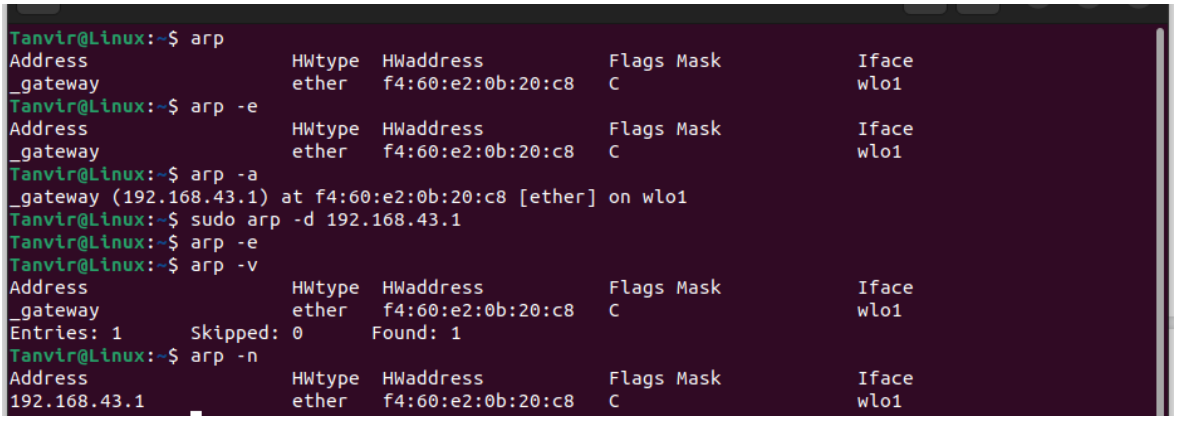
```
user@Linux: ~
Tanvir@Linux:~$ ifconfig -a
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 138186  bytes 14704955 (14.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 138186  bytes 14704955 (14.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.109  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::81:79f1:1399:b35d  prefixlen 64  scopeid 0x20<link>
        ether 90:0f:0c:cb:e6:7b  txqueuelen 1000  (Ethernet)
        RX packets 613758  bytes 643627144 (643.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 418139  bytes 92790469 (92.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

Tanvir@Linux:~$
```

Figure 6: argument will display information on all active or inactive network

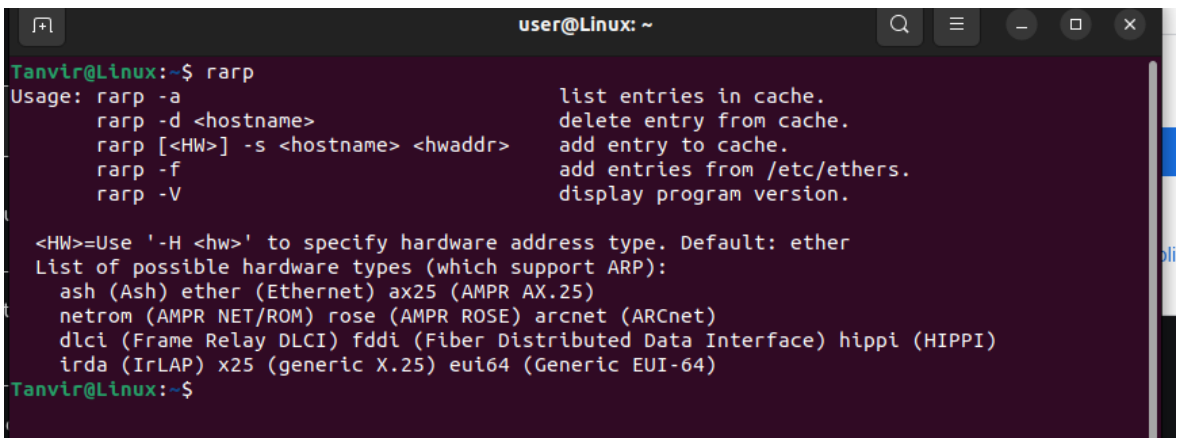
* **arp Command**



```
Tanvir@Linux:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   f4:60:e2:0b:20:c8   C                     wlo1
Tanvir@Linux:~$ arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   f4:60:e2:0b:20:c8   C                     wlo1
Tanvir@Linux:~$ arp -a
_gateway (192.168.43.1) at f4:60:e2:0b:20:c8 [ether] on wlo1
Tanvir@Linux:~$ sudo arp -d 192.168.43.1
Tanvir@Linux:~$ arp -e
Tanvir@Linux:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   f4:60:e2:0b:20:c8   C                     wlo1
Entries: 1      Skipped: 0      Found: 1
Tanvir@Linux:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.43.1             ether   f4:60:e2:0b:20:c8   C                     wlo1
```

Figure 7: arp command

* **rarp Command**



```
user@Linux: ~
Tanvir@Linux:~$ 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)
Tanvir@Linux:~$
```

Figure 8: maps an IP address to a MAC address.

* **nslookup Command**

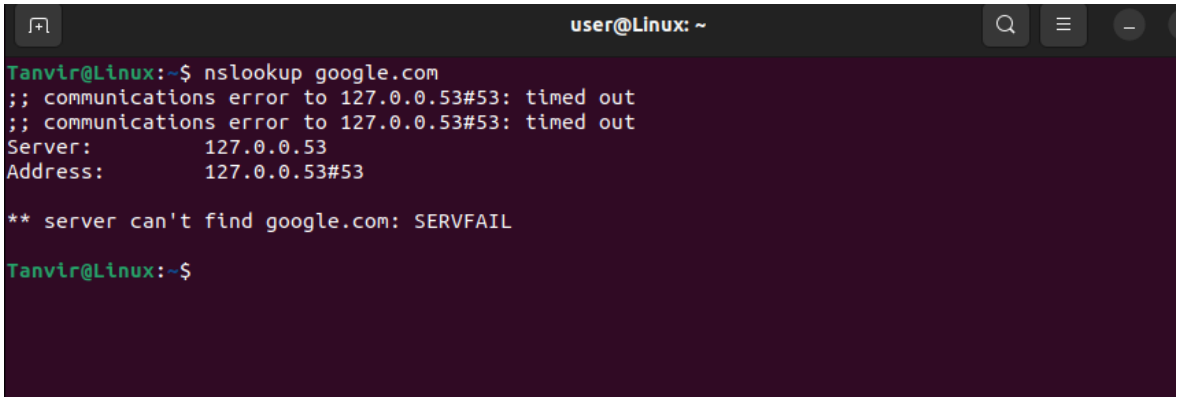


Figure 9: nslookup for google.com

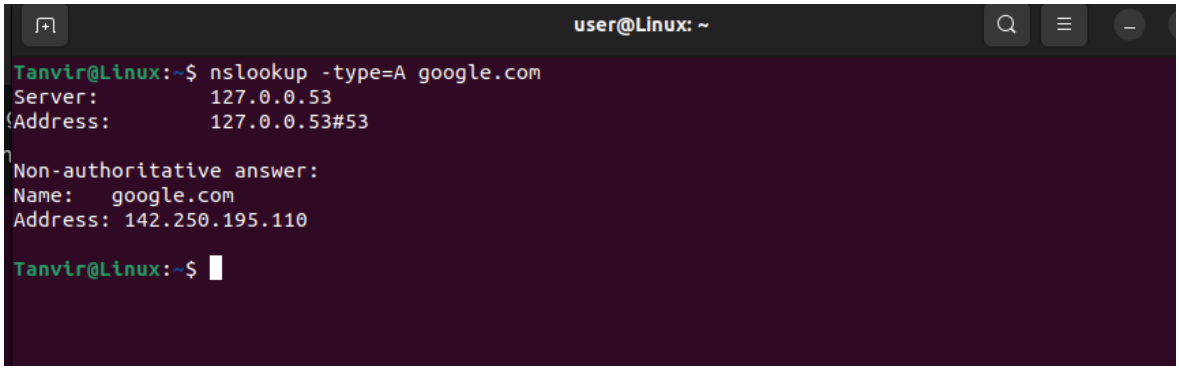


Figure 10: DNS records for a particular record like google

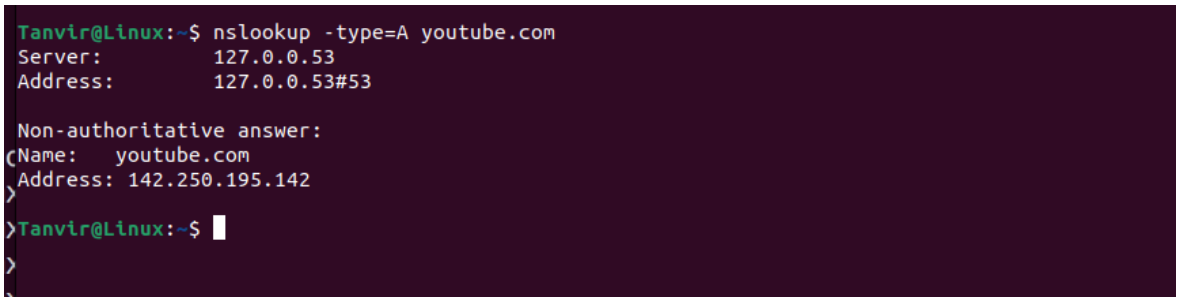


Figure 11: DNS records for a particular record like youtube

```
Tanvir@Linux:~$ nslookup -type=AAA google.com
unknown query type: AAA
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.195.110
Name:   google.com
Address: 2404:6800:4007:81b::200e

Tanvir@Linux:~$
```

Figure 12: find specefic address

```
Tanvir@Linux:~$ nslookup -type=soa google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
google.com
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 601396251
        refresh = 900
        retry = 900
        expire = 1800
        minimum = 60

Authoritative answers can be found from:

Tanvir@Linux:~$
```

Figure 13: provides the authoritative information about the domain

```
Tanvir@Linux:~$ nslookup -type=ns google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
google.com      nameserver = ns1.google.com.
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns4.google.com.

Authoritative answers can be found from:
ns3.google.com  internet address = 216.239.36.10
ns3.google.com  has AAAA address 2001:4860:4802:36::a
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  has AAAA address 2001:4860:4802:32::a
ns2.google.com  internet address = 216.239.34.10
ns2.google.com  has AAAA address 2001:4860:4802:34::a
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  has AAAA address 2001:4860:4802:38::a

Tanvir@Linux:~$
```

Figure 14: record maps a domain name to a list of DNS servers

* **netstat Command**

```
Tanvir@Linux:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 Linux:58764             82.221.107.34.bc.g:http ESTABLISHED
tcp        0      0 Linux:57794             maa03s42-in-f22.1:https TIME_WAIT
tcp        0      0 Linux:41444             91.108.56.141:https     ESTABLISHED
tcp        0      0 Linux:37180             207.65.33.78:https      TIME_WAIT
tcp        0      0 Linux:57526             93.243.107.34.bc.:https ESTABLISHED
tcp        0      1 Linux:39838             82.221.107.34.bc.g:http FIN_WAIT1
tcp        0      0 Linux:46564             maa05s19-in-f22.1:https ESTABLISHED
tcp        0      1 Linux:34782             sc-in-f188.1e100.:https FIN_WAIT1
tcp        0      0 Linux:43064             104.19.159.19:https     TIME_WAIT
tcp        0      0 Linux:43192             69.173.158.64:https     TIME_WAIT
tcp        0      0 Linux:54568             sc-in-f188.1e100.n:5228 ESTABLISHED
tcp        0      0 Linux:40052             91.108.23.100:https     ESTABLISHED
tcp        0    174 Linux:38992             maa05s22-in-f3.1e:https FIN_WAIT1
tcp        0      0 Linux:41948             1.80.190.35.bc.go:https TIME_WAIT
tcp        0      0 Linux:60494             172.67.74.110:https     TIME_WAIT
tcp        0      0 Linux:42510             maa05s20-in-f14.1:https ESTABLISHED
tcp        0      0 Linux:59014             168.81.95.34.bc.g:https TIME_WAIT
tcp        0      0 Linux:43188             69.173.158.64:https     TIME_WAIT
tcp        0      0 Linux:51426             91.108.56.141:https     FIN_WAIT2
tcp        0      1 Linux:39852             82.221.107.34.bc.g:http FIN_WAIT1
tcp        0      0 Linux:40898             sin01-convex-floa:https TIME_WAIT
tcp        0   1970 Linux:52064             maa03s42-in-f14.1:https FIN_WAIT1
tcp        0      1 Linux:37562             maa05s19-in-f3.1e:https LAST_ACK
tcp        0      0 Linux:57512             93.243.107.34.bc.:https ESTABLISHED
tcp        0      0 Linux:36044             ec2-13-214-94-186:https TIME_WAIT
tcp        0      0 Linux:51814             ade9ecc7904667038:https TIME_WAIT
tcp        0      0 Linux:46566             maa05s19-in-f22.1:https TIME_WAIT
tcp        0      0 Linux:43172             69.173.158.64:https     TIME_WAIT
tcp        0      0 Linux:52068             maa03s42-in-f14.1:https TIME_WAIT
tcp        0      0 Linux:43200             69.173.158.64:https     TIME_WAIT
tcp        0      0 Linux:59792             maa03s34-in-f3.1e:https ESTABLISHED
tcp        0      0 Linux:41952             64.52.120.34.bc.g:https ESTABLISHED
```

Figure 15: Show both listening and non-listening sockets

```
Tanvir@Linux:~$ netstat -st
IcmpMsg:
    InType0: 38
    InType3: 11389
    InType8: 289
    InType11: 122
    OutType0: 289
    OutType3: 10194
    OutType8: 76
Tcp:
    8304 active connection openings
    1 passive connection openings
    2977 failed connection attempts
    620 connection resets received
    14 connections established
    202762 segments received
    171551 segments sent out
    10079 segments retransmitted
    252 bad segments received
    3866 resets sent
UdpLite:
TcpExt:
    696 ICMP packets dropped because they were out-of-window
    1567 TCP sockets finished time wait in fast timer
    12 packetes rejected in established connections because of timestamp
    3397 delayed acks sent
    Quick ack mode was activated 2314 times
    18250 packet headers predicted
    22279 acknowledgments not containing data payload received
    14599 predicted acknowledgments
    TCPSackRecovery: 92
    Detected reordering 35 times using SACK
    Detected reordering 4 times using time stamp
    3 congestion windows fully recovered without slow start
    4 congestion windows partially recovered using Hoe heuristic
    TCPDSACKUndo: 35
    366 congestion windows recovered without slow start after partial ack
    TCPLostRetransmit: 5148
    TCPSackFailures: 8
```

Figure 16: displays statistics exclusively for TCP ports.

```
Tanvir@Linux:~$ netstat -su
IcmpMsg:
    InType0: 38
    InType3: 11390
    InType8: 289
    InType11: 122
    OutType0: 289
    OutType3: 10195
    OutType8: 76
Udp:
    441412 packets received
    3680 packets to unknown port received
    0 packet receive errors
    296381 packets sent
    0 receive buffer errors
    7 send buffer errors
    IgnoredMulti: 418
UdpLite:
IpExt:
    InMcastPkts: 4052
    OutMcastPkts: 3287
    InBcastPkts: 420
    OutBcastPkts: 7
    InOctets: 594398638
    OutOctets: 69401028
    InMcastOctets: 624706
    OutMcastOctets: 444876
    InBcastOctets: 267678
    OutBcastOctets: 4474
    InNoECTPkts: 658262
    InECT1Pkts: 6
    InECT0Pkts: 10
MPTcpExt:
Tanvir@Linux:~$
```

Figure 17: statistical information related to UDP ports.

```
Tanvir@Linux:~$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 Linux:44810             maa05s28-in-f10.1:https ESTABLISHED
udp        0      0 Linux:45212             yw-in-f94.1e100.n:https ESTABLISHED
udp        0      0 mdns.mcast.net:mdns     0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp        0      0 Linux:55387             maa05s28-in-f10.1:https ESTABLISHED
udp        0      0 Linux:55607             maa05s14-in-f10.1:https ESTABLISHED
udp        0      0 Linux:47563             maa05s19-in-f14.1:https ESTABLISHED
udp        0      0 Linux:56413             maa05s28-in-f10.1:https ESTABLISHED
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 Linux:bootpc            _gateway:bootps         ESTABLISHED
udp        0      0 Linux:57669             maa05s28-in-f10.1:https ESTABLISHED
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:58107           0.0.0.0:*
udp        0      0 Linux:58943             lcatla-aa-in-f3.1:https ESTABLISHED
udp        0      0 Linux:59166             maa05s28-in-f10.1:https ESTABLISHED
udp        0      0 Linux:34623             maa05s28-in-f10.1:https ESTABLISHED
udp6       0      0 [::]:53706              [::]:*
udp6       0      0 [::]:mdns               [::]:*
udp6       0      0 [::]:1716               [::]:*
Tanvir@Linux:~$
```

Figure 18: revealing details about UDP connections.

```
Tanvir@Linux:~$ netstat -lu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 mdns.mcast.net:mdns     0.0.0.0:*
udp        0      0 0.0.0.0:mdns            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 0.0.0.0:58107           0.0.0.0:*
udp6       0      0 [::]:53706              [::]:*
udp6       0      0 [::]:mdns               [::]:*
udp6       0      0 [::]:1716               [::]:*
Tanvir@Linux:~$
```

Figure 19: only the UDP ports that are actively listening.

```
Tanvir@Linux:~$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp6       0      0 [::]:1716               [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
tcp6       0      0 localhost:52829         [::]:*                  LISTEN
udp        0      0 mdns.mcast.net:mdns     0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp        0      0 0.0.0.0:55698           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 0.0.0.0:58107           0.0.0.0:*
udp6       0      0 [::]:53706              [::]:*
udp6       0      0 [::]:mdns               [::]:*
udp6       0      0 [::]:1716               [::]:*
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     19797    /run/systemd/fsck.progress
unix  2      [ ACC ]     STREAM     LISTENING     19808    /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     19810    /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     32688    /tmp/.ICE-unix/1860
unix  2      [ ACC ]     STREAM     LISTENING     37925    /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     37927    /tmp/.X11-unix/X1
unix  2      [ ACC ]     STREAM     LISTENING     21372    /run/systemd/journal/io.systemd.journal
unix  2      [ ACC ]     STREAM     LISTENING     33754    /tmp/zoG3Xn/s
unix  2      [ ACC ]     STREAM     LISTENING     35659    /tmp/fdm6fs1000
unix  2      [ ACC ]     STREAM     LISTENING     36905    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     36911    /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     36913    /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     36915    /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     76916    /tmp/.com.google.Chrome.R9MrbX/SingletonSo
t
unix  2      [ ACC ]     STREAM     LISTENING     36917    /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     36919    /run/user/1000/gnupg/S.gpg-agent.ssh
```

Figure 20: the ports that are actively listening for incoming connections

```
Tanvir@Linux:~$ netstat -s
Ip:
    Forwarding: 2
    650561 total packets received
    58 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    650277 incoming packets delivered
    474871 requests sent out
    27 outgoing packets dropped
    1430 dropped because of missing route
    2 fragments received ok
    4 fragments created
Icmp:
    11836 ICMP messages received
    1045 input ICMP message failed
    ICMP input histogram:
        destination unreachable: 11387
        timeout in transit: 122
        echo requests: 289
        echo replies: 38
    10557 ICMP messages sent
    0 ICMP messages failed
    OutRateLimitGlobal: 60
    OutRateLimitHost: 6
    ICMP output histogram:
        destination unreachable: 10192
        echo requests: 76
        echo replies: 289
IcmpMsg:
        InType0: 38
        InType3: 11387
        InType8: 289
        InType11: 122
        OutType0: 289
        OutType3: 10192
        OutType8: 76
Tcp:
    8300 active connection openings
```

Figure 21: statistical information for all ports,

5 Experience

1. Attempting to utilize the RARP but it doesn't work properly
2. testing of various terminal commands, including Ping, Traceroute, IFCONFIG, ARP, NSLOOKUP, and NETSTAT
3. Experimenting with the Ping command using a custom interval of 3 seconds to google.com (ping -i 3 google.com) showcased a dynamic approach to network testing.

References

[1] Computer networking : a top-down approach 8th ed.

[2] GeeksforGeeks : https://www.google.com/search?channel=fs&client=ubuntu-sn&q=gfg

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

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

[5] IFCONFIG: https://www.tecmint.com/ifconfig-command-examples/

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

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

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

[9] NETSTAT: https://www.geeksforgeeks.org/netstat-command-linux/

‘