Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

CERTIFICATE

Certify that Mr./Miss ALPHONZ GEORGE VELACHERRY of I. T. Department, Semester 4 with Roll No. 05 has completed a course of the necessary experiments in the subject NETWORK LAB under my supervision in the **Thadomal Shahani Engineering College** Laboratory in the year 2023 - 2024

16/4/24

DR. KUMKUM SAXENA
Teacher In- Charge

Head of the Department

Date 16th April 2024

Principal

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

CONTENTS

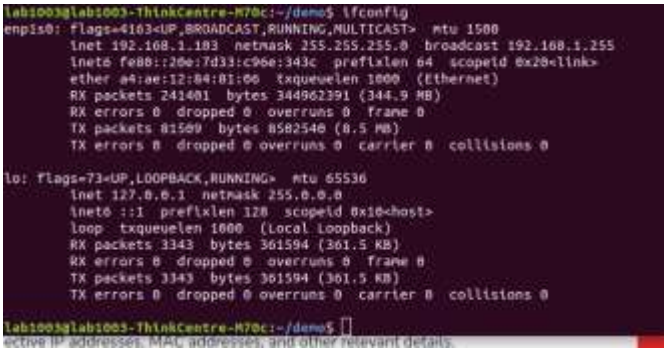
SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1	Understanding Basic Networking Commands		16/01/24 18/01/24	
2	Installation and Configuration of NS2 and implementation of TCL Hello Programming		25/01/24	
3	Implementation of Specific Network topology with respect to TCP		25/01/24	
4	Implementation of Specific Network topology with respect to UDP		30/01/24	
5	Simulation of Network with specific routing protocols (Distance Vector, Link State		08/02/24	
6	Socket Programming with C/Java- TCP Client, TCP Server		13/02/24	
7	Socket Programming with C/Java UDP Client, UDP Server		22/02/24	
8	Installation of WireShark and Analysis of Packet Headers- TCP-IP, UDP using Wireshark		29/02/24	
9	A Case Study to design and configure any organization network		05/03/24	
10	Written Assignment - I		11/03/24	
11	Written Assignment - II		11/03/24	
12	Analysis of Packet Headers- TCP, IP, UDP using TCPDUMP		19/03/24	

16/4/24

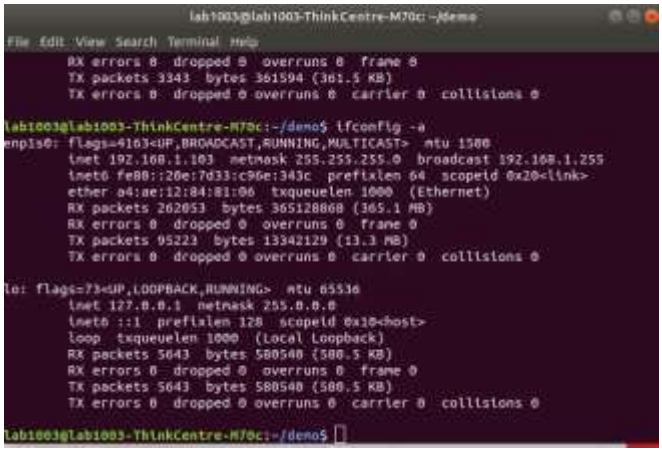
Alphonz George
S11-05

NL Assignment 1

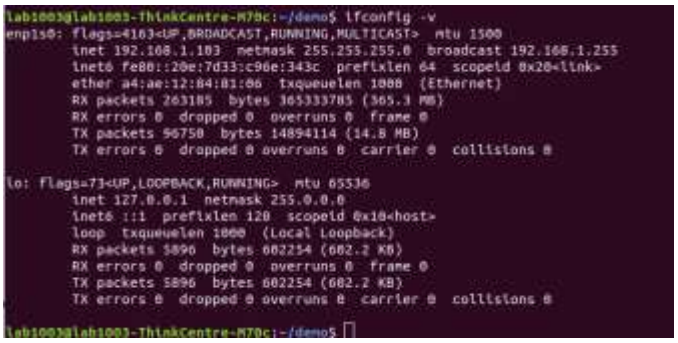
1. Ifconfig:
   ifconfig (interface configuration) command is used to configure the kernel-resident network interfaces. It is used at the boot time to set upthe interfaces as necessary. After that, it is usually used when needed during debugging or when you need system tuning. Also, thiscommand is used to assign the IP address and netmask to an interface or to enable or disable a given interface.



- **The ifconfig -a command** in Unix-like operating systems (suchas Linux) is used to display information about all network interfaces, whether they are currently active or not. This command provides details such as IP addresses, MAC addresses, and other network-related information for each interface on the system.



- **The ifconfig -s command** in Unix-like operating systems is used todisplay a short summary of network interfaces along with basic statistics. This command provides a condensed view of network interface information, including interface names, MTU (Maximum Transmission Unit), and various packet statistics.



2. Ip:
   Ip command in Linux is present in the net-tools which are used for performing several network administration tasks. IP stands for Internet Protocol. This command is used to show or manipulate routing, devices, and tunnels. It is similar to the *ifconfig* command, butit is much more powerful with more functions and facilities attached toit. *ifconfig* is one of the deprecated commands in the net-tools of Linux that has not been maintained for many years. The ip command is used to perform several tasks like assigning an address to a network interface or configuring network interface parameters. It can perform several other tasks like configuring and modifying the default and static routing, setting up a tunnel over IP, listing IP addresses andproperty information, modifying the status of the interface, and assigning, deleting, and setting up IP addresses and routes.

Alphonz George
S11-05

ip address: Display ip addresses of each interface.
Ip link: Display information of all interfaces.
ip route: Display routing table

3. Traceroute/tracert:
   traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know aboutthe route and about all the hops that a packet takes. Below image depicts how traceroute command is used to reach the Google(172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.

Alphonz George
S11-05

```
C:\Users\lab1003>tracert www.google.com

Tracing route to www.google.com [142.250.70.100]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.1.1
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     4 ms     6 ms     6 ms  175.100.188.22
  8     3 ms     3 ms     3 ms  142.251.225.9
  9     4 ms     4 ms     3 ms  192.178.86.241
 10     2 ms     2 ms     2 ms  pnbomb-ac-in-f4.1e100.net [142.250.70.100]

Trace complete.
```

- **traceroute -d**: The -d option is often used to prevent traceroute from attempting to resolve IP addresses to hostnames. This canspeed up the trace route process by avoiding DNS lookups.

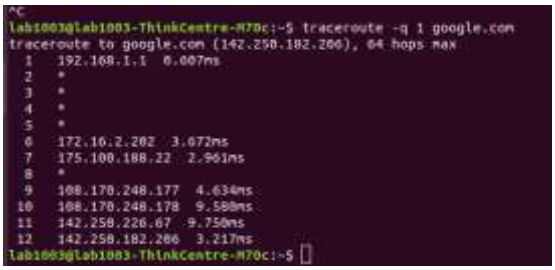
```
C:\Users\lab1003>tracert -d www.google.com

Tracing route to www.google.com [142.250.192.132]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.1.1
  2     1 ms    <1 ms    <1 ms  192.168.0.1
  3     1 ms    <1 ms    <1 ms  203.212.25.1
  4     1 ms    <1 ms     1 ms  203.212.24.53
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     1 ms     1 ms     1 ms  175.100.188.22
  8     7 ms     4 ms     4 ms  216.239.57.17
  9     2 ms     2 ms     2 ms  172.253.50.147
 10     2 ms   165 ms     2 ms  142.250.192.132

Trace complete.
```

- **tracert /?:** On Windows, the tracert command is used instead oftraceroute. The /? option is used to display the help message, showing a list of available options.

Alphonz George
S11-05

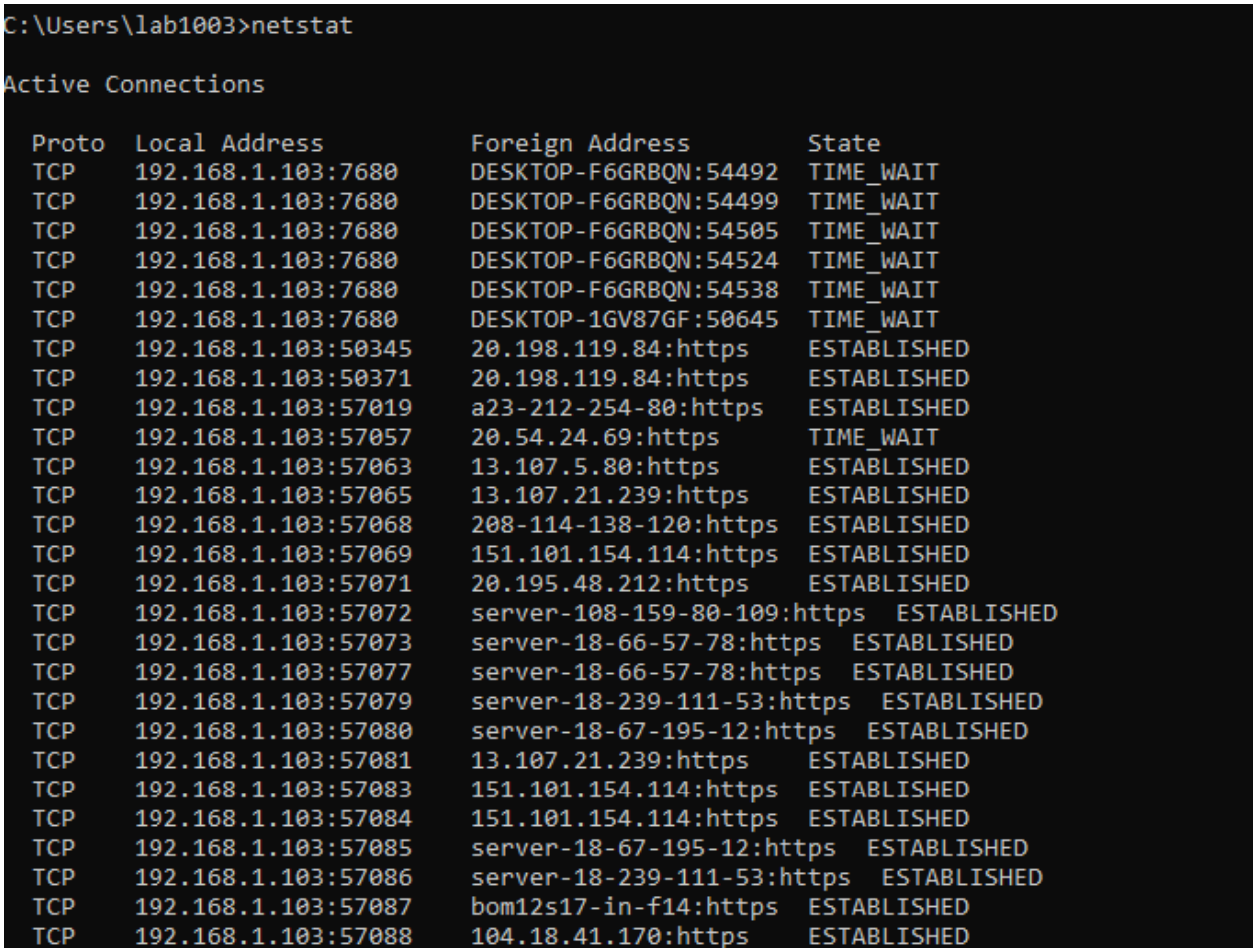
```
C:\Users\lab1003>tracert/?

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.
```

- **traceroute -q:** The -q option allows you to specify the numberof probes (packets) to send to each hop along the



route.

4. **Netstat**:
   The Network Statistics (netstat) tool has been around for a long time (including on Windows 10 and older versions), and it's a command-line utility you can use in Command Prompt to display statistics for all network connections. It allows you to understand open and connected ports to monitor and troubleshoot networking problems for systems or apps.
   The tool helps you to list active network (incoming and outgoing) connections and listening ports. You can view network adapter statistics and statistics for protocols (such as IPv4 and IPv6). You can even display the current routing table and much more.



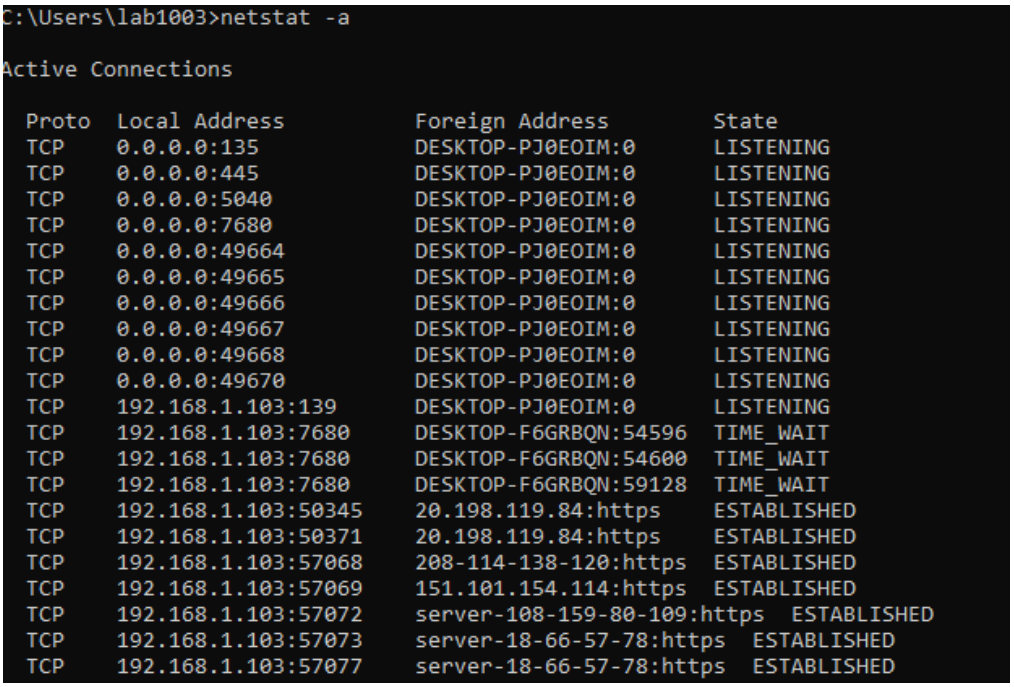
```
C:\Users\lab1003>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:54492  TIME_WAIT
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:54499  TIME_WAIT
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:54505  TIME_WAIT
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:54524  TIME_WAIT
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:54538  TIME_WAIT
  TCP    192.168.1.103:7680     DESKTOP-1GV87GF:50645  TIME_WAIT
  TCP    192.168.1.103:50345    20.198.119.84:https    ESTABLISHED
  TCP    192.168.1.103:50371    20.198.119.84:https    ESTABLISHED
  TCP    192.168.1.103:57019    a23-212-254-80:https   ESTABLISHED
  TCP    192.168.1.103:57057    20.54.24.69:https      TIME_WAIT
  TCP    192.168.1.103:57063    13.107.5.80:https      ESTABLISHED
  TCP    192.168.1.103:57065    13.107.21.239:https    ESTABLISHED
  TCP    192.168.1.103:57068    208-114-138-120:https  ESTABLISHED
  TCP    192.168.1.103:57069    151.101.154.114:https  ESTABLISHED
  TCP    192.168.1.103:57071    20.195.48.212:https    ESTABLISHED
  TCP    192.168.1.103:57072    server-108-159-80-109:https  ESTABLISHED
  TCP    192.168.1.103:57073    server-18-66-57-78:https  ESTABLISHED
  TCP    192.168.1.103:57077    server-18-66-57-78:https  ESTABLISHED
  TCP    192.168.1.103:57079    server-18-239-111-53:https  ESTABLISHED
  TCP    192.168.1.103:57080    server-18-67-195-12:https  ESTABLISHED
  TCP    192.168.1.103:57081    13.107.21.239:https    ESTABLISHED
  TCP    192.168.1.103:57083    151.101.154.114:https  ESTABLISHED
  TCP    192.168.1.103:57084    151.101.154.114:https  ESTABLISHED
  TCP    192.168.1.103:57085    server-18-67-195-12:https  ESTABLISHED
  TCP    192.168.1.103:57086    server-18-239-111-53:https  ESTABLISHED
  TCP    192.168.1.103:57087    bom12s17-in-f14:https  ESTABLISHED
  TCP    192.168.1.103:57088    104.18.41.170:https    ESTABLISHED
```

- **netstat -n:** The -n option displays numerical addresses instead of attempting to resolve them to hostnames. This can speed up the output and is useful when DNS resolution is not needed.

```
C:\Users\lab1003>netstat -n 5

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.103:7680     192.168.1.119:54596    TIME_WAIT
  TCP    192.168.1.103:7680     192.168.1.119:54600    TIME_WAIT
  TCP    192.168.1.103:7680     192.168.1.119:59128    TIME_WAIT
  TCP    192.168.1.103:50345    20.198.119.84:443      ESTABLISHED
  TCP    192.168.1.103:50371    20.198.119.84:443      ESTABLISHED
  TCP    192.168.1.103:57019    23.212.254.80:443      ESTABLISHED
  TCP    192.168.1.103:57068    120.138.114.208:443    ESTABLISHED
  TCP    192.168.1.103:57069    151.101.154.114:443    ESTABLISHED
  TCP    192.168.1.103:57072    108.159.80.109:443     ESTABLISHED
  TCP    192.168.1.103:57073    18.66.57.78:443        ESTABLISHED
  TCP    192.168.1.103:57077    18.66.57.78:443        ESTABLISHED
  TCP    192.168.1.103:57079    18.239.111.53:443      ESTABLISHED
  TCP    192.168.1.103:57080    18.67.195.12:443       ESTABLISHED
  TCP    192.168.1.103:57083    151.101.154.114:443    ESTABLISHED
  TCP    192.168.1.103:57084    151.101.154.114:443    ESTABLISHED
  TCP    192.168.1.103:57085    18.67.195.12:443       ESTABLISHED
  TCP    192.168.1.103:57086    18.239.111.53:443      ESTABLISHED
  TCP    192.168.1.103:57088    104.18.41.170:443      ESTABLISHED
  TCP    192.168.1.103:57090    18.172.64.56:443       ESTABLISHED
  TCP    192.168.1.103:57091    18.161.111.69:443      ESTABLISHED
  TCP    192.168.1.103:57093    18.172.64.72:443       ESTABLISHED
  TCP    192.168.1.103:57094    18.172.64.58:443       ESTABLISHED
  TCP    192.168.1.103:57096    18.66.41.11:443        ESTABLISHED
  TCP    192.168.1.103:57097    142.250.192.110:443    ESTABLISHED
  TCP    192.168.1.103:57098    18.172.64.78:443       ESTABLISHED
```

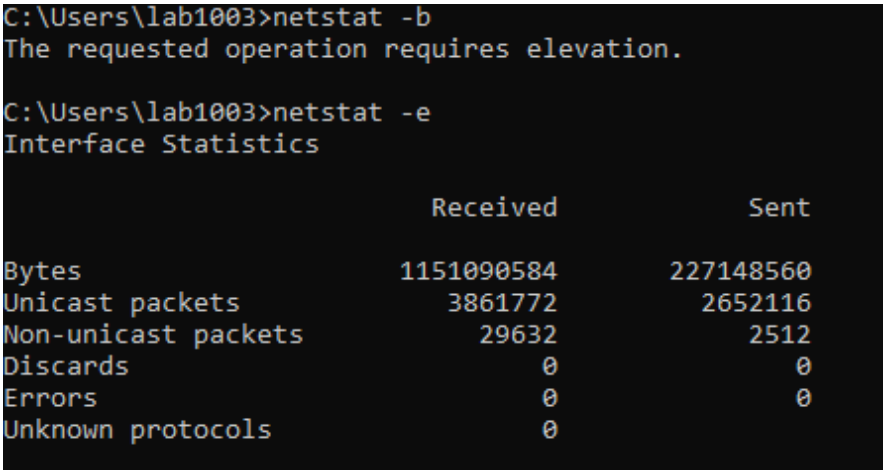
```
C:\Users\lab1003>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.103:7680     192.168.1.119:54546    TIME_WAIT
  TCP    192.168.1.103:7680     192.168.1.119:54565    TIME_WAIT
  TCP    192.168.1.103:7680     192.168.1.119:54587    TIME_WAIT
  TCP    192.168.1.103:7680     192.168.1.119:54592    TIME_WAIT
  TCP    192.168.1.103:7680     192.168.1.119:54596    TIME_WAIT
  TCP    192.168.1.103:50345    20.198.119.84:443      ESTABLISHED
  TCP    192.168.1.103:50371    20.198.119.84:443      ESTABLISHED
  TCP    192.168.1.103:57019    23.212.254.80:443      ESTABLISHED
  TCP    192.168.1.103:57068    120.138.114.208:443    ESTABLISHED
  TCP    192.168.1.103:57069    151.101.154.114:443    ESTABLISHED
  TCP    192.168.1.103:57071    20.195.48.212:443      ESTABLISHED
  TCP    192.168.1.103:57072    108.159.80.109:443     ESTABLISHED
  TCP    192.168.1.103:57073    18.66.57.78:443        ESTABLISHED
  TCP    192.168.1.103:57077    18.66.57.78:443        ESTABLISHED
  TCP    192.168.1.103:57079    18.239.111.53:443      ESTABLISHED
  TCP    192.168.1.103:57080    18.67.195.12:443       ESTABLISHED
  TCP    192.168.1.103:57083    151.101.154.114:443    ESTABLISHED
  TCP    192.168.1.103:57084    151.101.154.114:443    ESTABLISHED
  TCP    192.168.1.103:57085    18.67.195.12:443       ESTABLISHED
  TCP    192.168.1.103:57086    18.239.111.53:443      ESTABLISHED
  TCP    192.168.1.103:57087    142.250.192.110:443    TIME_WAIT
  TCP    192.168.1.103:57088    104.18.41.170:443      ESTABLISHED
  TCP    192.168.1.103:57089    18.172.64.78:443       TIME_WAIT
  TCP    192.168.1.103:57090    18.172.64.56:443       ESTABLISHED
  TCP    192.168.1.103:57091    18.161.111.69:443      ESTABLISHED
  TCP    192.168.1.103:57092    18.66.36.59:443        TIME_WAIT
  TCP    192.168.1.103:57093    18.172.64.72:443       ESTABLISHED
  TCP    192.168.1.103:57094    18.172.64.58:443       ESTABLISHED
```

- **netstat -a:**The -a option shows all sockets, including listeningand non-listening sockets.



```
C:\Users\lab1003>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:445            DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:5040           DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:7680           DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:49664          DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:49665          DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:49666          DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:49667          DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:49668          DESKTOP-PJ0EOIM:0      LISTENING
  TCP    0.0.0.0:49670          DESKTOP-PJ0EOIM:0      LISTENING
  TCP    192.168.1.103:139      DESKTOP-PJ0EOIM:0      LISTENING
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:54596  TIME_WAIT
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:54600  TIME_WAIT
  TCP    192.168.1.103:7680     DESKTOP-F6GRBQN:59128  TIME_WAIT
  TCP    192.168.1.103:50345    20.198.119.84:https    ESTABLISHED
  TCP    192.168.1.103:50371    20.198.119.84:https    ESTABLISHED
  TCP    192.168.1.103:57068    208-114-138-120:https  ESTABLISHED
  TCP    192.168.1.103:57069    151.101.154.114:https  ESTABLISHED
  TCP    192.168.1.103:57072    server-108-159-80-109:https  ESTABLISHED
  TCP    192.168.1.103:57073    server-18-66-57-78:https  ESTABLISHED
  TCP    192.168.1.103:57077    server-18-66-57-78:https  ESTABLISHED
```

- **netstat -b:** On Windows, the -b option is used with the netstatcommand to display the executable involved in creating each connection or listening port.

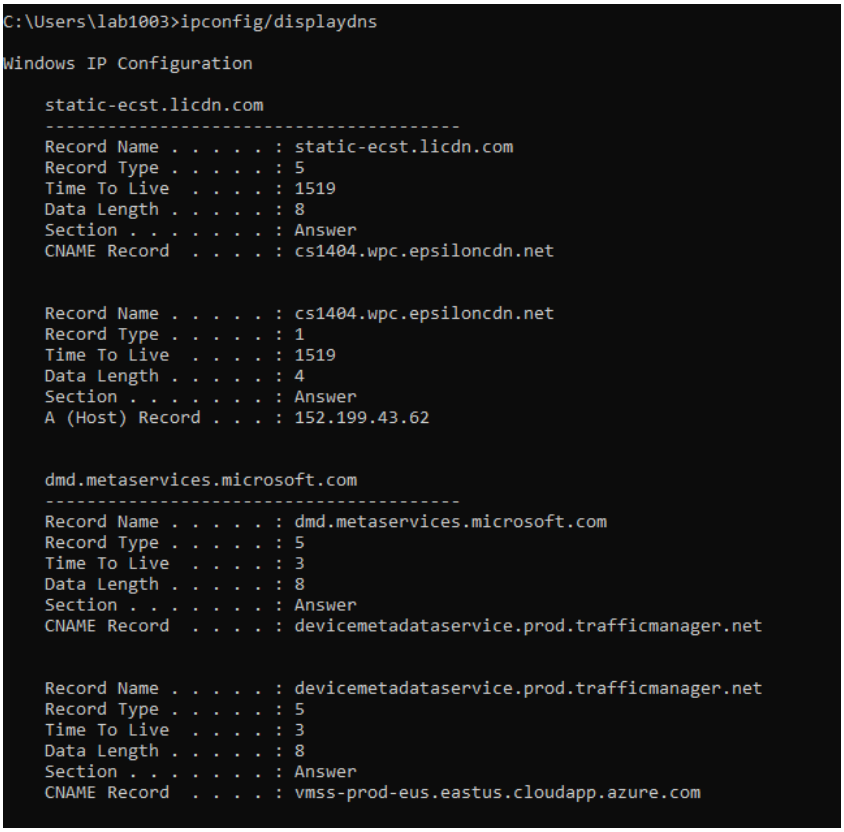


```
C:\Users\lab1003>netstat -b
The requested operation requires elevation.

C:\Users\lab1003>netstat -e
Interface Statistics

                           Received            Sent

Bytes                    1151090584       227148560
Unicast packets             3861772         2652116
Non-unicast packets           29632            2512
Discards                          0               0
Errors                            0               0
Unknown protocols                 0
```

5. ipconfig

It is one of the most used command-line tools for analyzing, configuring and troubleshooting your systems' network settings, bothin a home or enterprise environment. It has been one of those breadand butter tools that any sysadmin and network engineer will know and treasure.

- **ipconfig/displaydns:** The /displaydns option is used to show the contents of the DNS resolver cache. It displays a list of DNSresource records that have been resolved recently.



- **ipconfig/flushdns:** The /flushdns option is used to clear the contents of the DNS resolver cache. This can be useful if youwant to force the system to re-query DNS servers for updatedinformation.

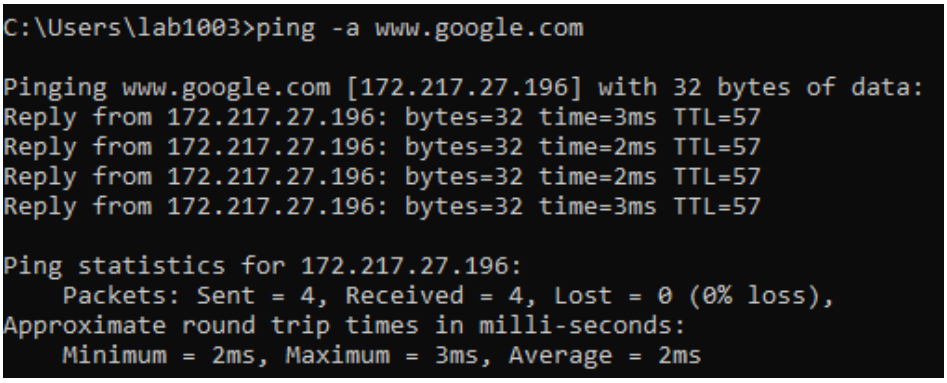
```
C:\Users\lab1003>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\lab1003>ipconfig/registerdns
The requested operation requires elevation.

C:\Users\lab1003>ipconfig/?

USAGE:
    ipconfig [/allcompartments] [/? | /all |
                                 /renew [adapter] | /release [adapter] |
                                 /renew6 [adapter] | /release6 [adapter] |
                                 /flushdns | /displaydns | /registerdns |
                                 /showclassid adapter |
                                 /setclassid adapter [classid] |
                                 /showclassid6 adapter |
                                 /setclassid6 adapter [classid] ]

where
    adapter             Connection name
                       (wildcard characters * and ? allowed, see examples)

    Options:
       /?               Display this help message
       /all             Display full configuration information.
       /release         Release the IPv4 address for the specified adapter.
       /release6        Release the IPv6 address for the specified adapter.
       /renew           Renew the IPv4 address for the specified adapter.
       /renew6          Renew the IPv6 address for the specified adapter.
       /flushdns        Purges the DNS Resolver cache.
       /registerdns     Refreshes all DHCP leases and re-registers DNS names
       /displaydns      Display the contents of the DNS Resolver Cache.
       /showclassid     Displays all the dhcp class IDs allowed for adapter.
       /setclassid      Modifies the dhcp class id.
       /showclassid6    Displays all the IPv6 DHCP class IDs allowed for adapter.
       /setclassid6     Modifies the IPv6 DHCP class id.
```

- **ipconfig/release:** The /release option is used to release the IPaddress obtained via DHCP (Dynamic Host Configuration Protocol). It informs the DHCP server that the IP address is nolonger in use.

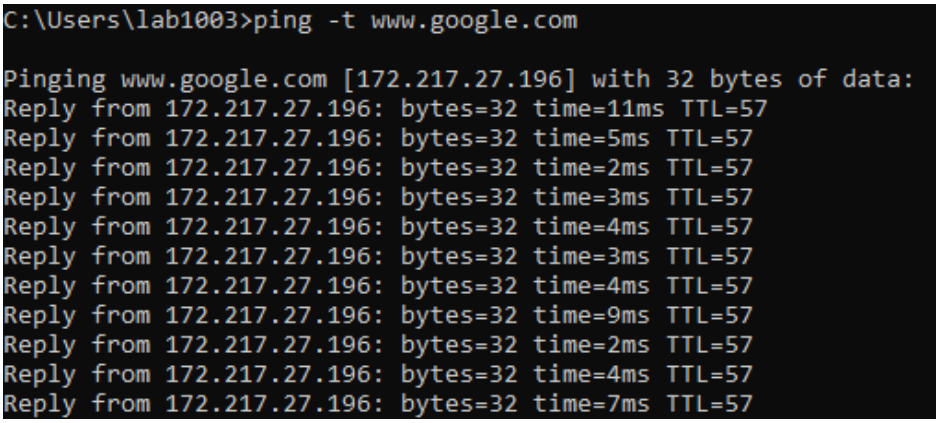
```
C:\Users\lab1003>ipconfig/release

Windows IP Configuration

No operation can be performed on WiFi while it has its media disconnected.
No operation can be performed on Local Area Connection* 1 while it has its media disconnected.
No operation can be performed on Local Area Connection* 2 while it has its media disconnected.
No operation can be performed on Bluetooth Network Connection while it has its media disconnected.

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2080:c5ec:ca42:62d1%10
   Default Gateway . . . . . . . . . :

Wireless LAN adapter WiFi:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

6. Ping:
   The ping command is a Command Prompt command used to test the ability of the source computer to reach a specified destination computer. It's a simple way to verify that a computercan communicate with another computer or network device.

- **ping -a:**The -a option in Windows triggers an audible ping.However, the -a option is not typically available in the Unix/Linux version of ping.



- **ping-t:** The -t option in Windows is used to ping the specified host indefinitely until manually stopped. In Unix/Linux, the equivalent behavior can be achieved by using the ping command without specifying a count ora timeout value.



- **ping-n:** The -n option in Windows allows you to specify the number of echorequests to send. In Unix/Linux, this can be achieved using the -c option.

```
C:\Users\lab1003>ping -n 5 www.google.com

Pinging www.google.com [172.217.27.196] with 32 bytes of data:
Reply from 172.217.27.196: bytes=32 time=2ms TTL=57
Reply from 172.217.27.196: bytes=32 time=7ms TTL=57
Reply from 172.217.27.196: bytes=32 time=2ms TTL=57
Reply from 172.217.27.196: bytes=32 time=2ms TTL=57
Reply from 172.217.27.196: bytes=32 time=2ms TTL=57

Ping statistics for 172.217.27.196:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 7ms, Average = 3ms
```

- **Ping -f:**The -f option in Windows is used to send a flood of ping requests.In Unix/Linux, you can use the -f option with the ping command to flood the network with ICMP Echo Requests.

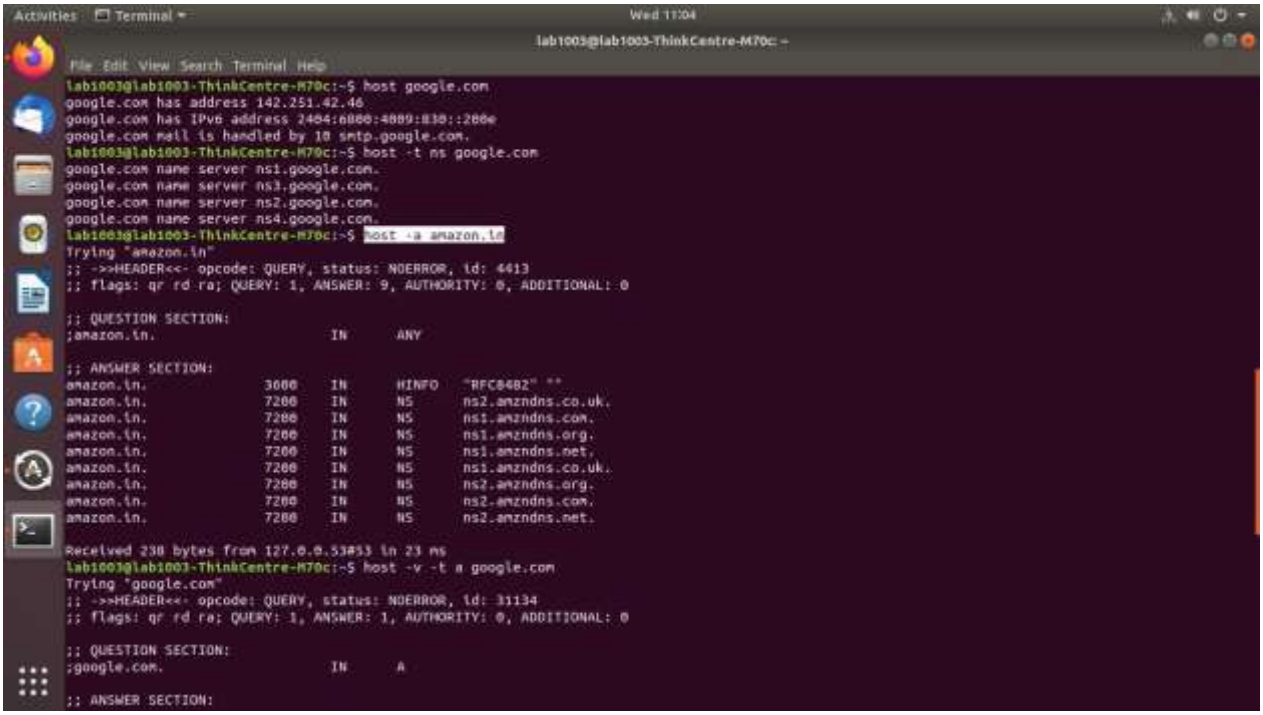
```
C:\Users\lab1003>ping -f www.google.com

Pinging www.google.com [172.217.27.196] with 32 bytes of data:
Reply from 172.217.27.196: bytes=32 time=2ms TTL=57
Reply from 172.217.27.196: bytes=32 time=5ms TTL=57
Reply from 172.217.27.196: bytes=32 time=2ms TTL=57
Reply from 172.217.27.196: bytes=32 time=3ms TTL=57

Ping statistics for 172.217.27.196:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 5ms, Average = 3ms
```

- **Ping -l:**The -l option in Windows allows you to set the size of the echorequest packet. In Unix/Linux, you can use the -s option for a similar purpose.

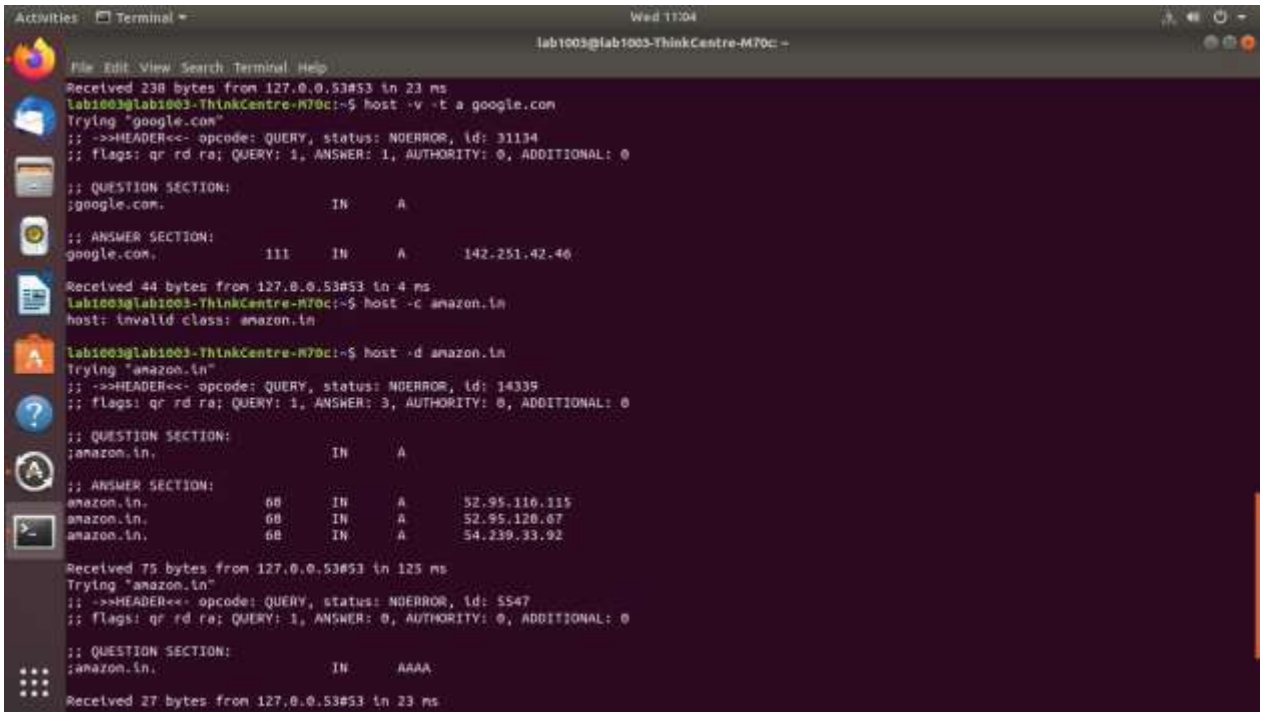
```
C:\Users\lab1003>ping -l 5 www.google.com

Pinging www.google.com [172.217.27.196] with 5 bytes of data:
Reply from 172.217.27.196: bytes=5 time=2ms TTL=57
Reply from 172.217.27.196: bytes=5 time=2ms TTL=57
Reply from 172.217.27.196: bytes=5 time=2ms TTL=57
Reply from 172.217.27.196: bytes=5 time=4ms TTL=57

Ping statistics for 172.217.27.196:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 2ms
```

7. Host:

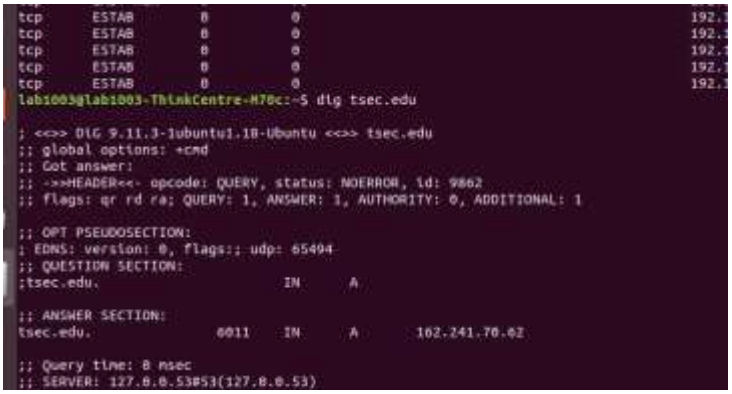
   The **host** command returns the Internet address of a host machine when the *HostName* parameter is specified and the name of the host when the *Address* parameter is specified. The **host** command also displays any aliases associated with the *HostName* paramete



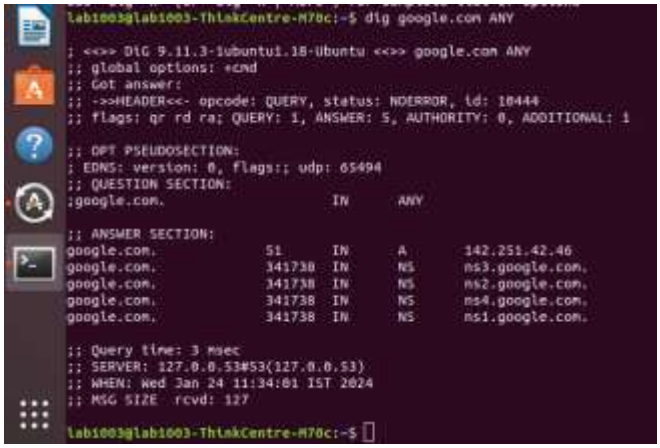
- The -a option in the host command is used to display allinformation about the specified domain.
- The -t option is used to specify the query type. You can use it to query for different types of DNS records, such as A (IPv4 address), AAAA (IPv6 address), MX (mail exchange), etc.
- The -v option is used to enable verbose output, providingadditional information about the DNS query.
- -c: Specifies the query class (e.g., -c IN for Internet).
- -d: Enables debugging output.



8. Dig command:
   The domain information groper (dig) is **a command line tool that can be used to gather information from the Domain Name System servers**. The dig command has two modes: simple interactive mode for a single query, and batch mode, which executes one query for each in a list of several query lines



- The ANY query type is used to request all records associatedwith the specified domain name.
- The +short option in the dig command is used to display a moreconcise output, providing only the essential information without additional details.

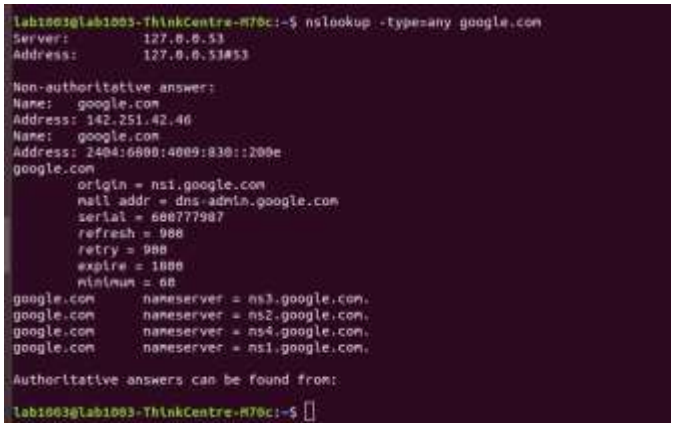


9. **nslookup command:**

Nslookup (stands for “Name Server Lookup”) is a useful commandfor getting information from the DNS server. It is a network administration tool for querying the Domain Name System (DNS) toobtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS-related

problems.

- The -type=any option in the nslookup command is used to query for all types of DNS records associated with a givendomain name.



- The -type=soa option is used to specifically query for the Startof Authority (SOA) record for

a domain.

- The -type=nx option is not a standard option in nslookup. If youwant to check for the non-existence of a domain, you typically look at the response from a regular query. An "NXDOMAIN" response indicates that the domain does not exist.

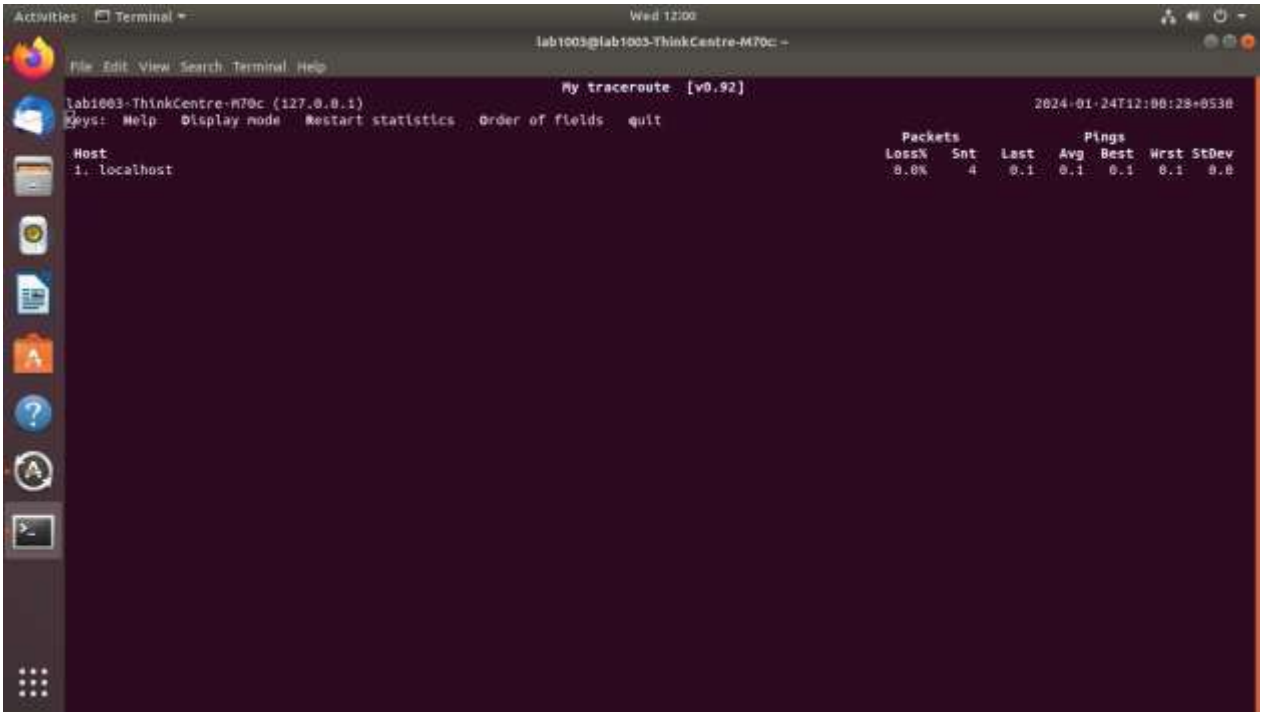
10. Arp command:

arp command manipulates the System's ARP cache. It also allows acomplete dump of the ARP cache. ARP stands for Address Resolution Protocol. The primary function of this protocol is to resolvethe 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 arp -a command** is used to display the current ARP (Address Resolution Protocol) cache on a system. ARP is used to map IP addresses to MAC addresses on a local network. The -a option in the arp command is used to show the ARP table, which contains mappings of IP addresses to MAC addresses.

11. Mtr command:

mtr is a networking tool that combines *ping* and *traceroute* to diagnose a network. Instead of using both tools separately, we could use only *mtr*. **The purpose of *mtr* is to analyze the network traffichop-to-hop using *ICMP* packets**.



12. Whois command:

Whois is a command-line utility used in Linux systems to retrieve information about domain names, IP addresses, and network devices registered with the Internet Corporation for Assigned Names and Numbers (ICANN). The data received by Whois consists of the name and contact information of the domain or IP address owner, the registration and expiration date, the domain registrar, and the server information. Whois command can be very useful for network administrators, web developers, and security professionals for achieving various tasks like checking network connectivity or troubleshooting. In this article, we will go through the usage of the Whois command on Linux (Ubuntu system).

```
lab1003@lab1003-ThinkCentre-M70c:~$ whois google.com
   Domain Name: GOOGLE.COM
   Registry Domain ID: 2138514_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2019-09-09T15:39:04Z
   Creation Date: 1997-09-15T04:00:00Z
   Registry Expiry Date: 2028-09-14T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2086851750
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
   Name Server: NS4.GOOGLE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2024-01-24T06:33:52Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar.  Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
```

13. 9-curl

curl is a command-line tool to transfer data to or from a server, using anyof the supported protocols

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ curl tsec.edu
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://tsec.edu/">here</a>.</p>
</body></html>
```

NAME:Alphonz George Velacherry

ROLL NO:05

Lab Assignment No.: 02

Aim: To install and configure network simulator and learn basics of TCL scripting.

Lab Outcome Attained: LO – 2
Demonstrate the installation and configuration of network simulator.

Theory:

1) NS2

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

Features of NS2

1. It is a discrete event simulator for networking research.
2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.
3. It simulates wired and wireless network.
4. It is primarily Unix based.
5. Uses TCL as its scripting language.
6. Otcl: Object oriented support
7. Tclcl: C++ and otcl linkage
8. Discrete event scheduler

Basic Architecture
NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation

objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL

2) NAM

The network animator is one of the TCL animation tools that are deployed to view the network simulation traces and real time traces in packet data. Initially, the users have to generate a trace file along with topological information such as packet traces, links and nodes for the utilization of NAM.

After completing the trace file generation process, the Nam will real and create topology along with that is deployed to pop up the window and process with the required layout and pause the time of first packet based on the trace file. It is denoted as the provision of control through various aspects over the animation and user interface for the creation of blocks such as:

- Monitor
- Agent
- Used to separate protocol states from the node
- Packet
- Visualized as a block with an arrow
- Queue
- Associated with one edge of duplex link and is visualized as stacked packet
- Link
- Nam links are consist of two simplex links
- Node
- created from ‘n’ trace event in trace file

3) TCL

TCL is a high-level, general-purpose, interpreted, dynamic programming language. It was designed with the goal of being very simple but powerful. TCL casts everything into the mold of a command, even programming constructs like variable assignment and procedure definition. TCL supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural styles.

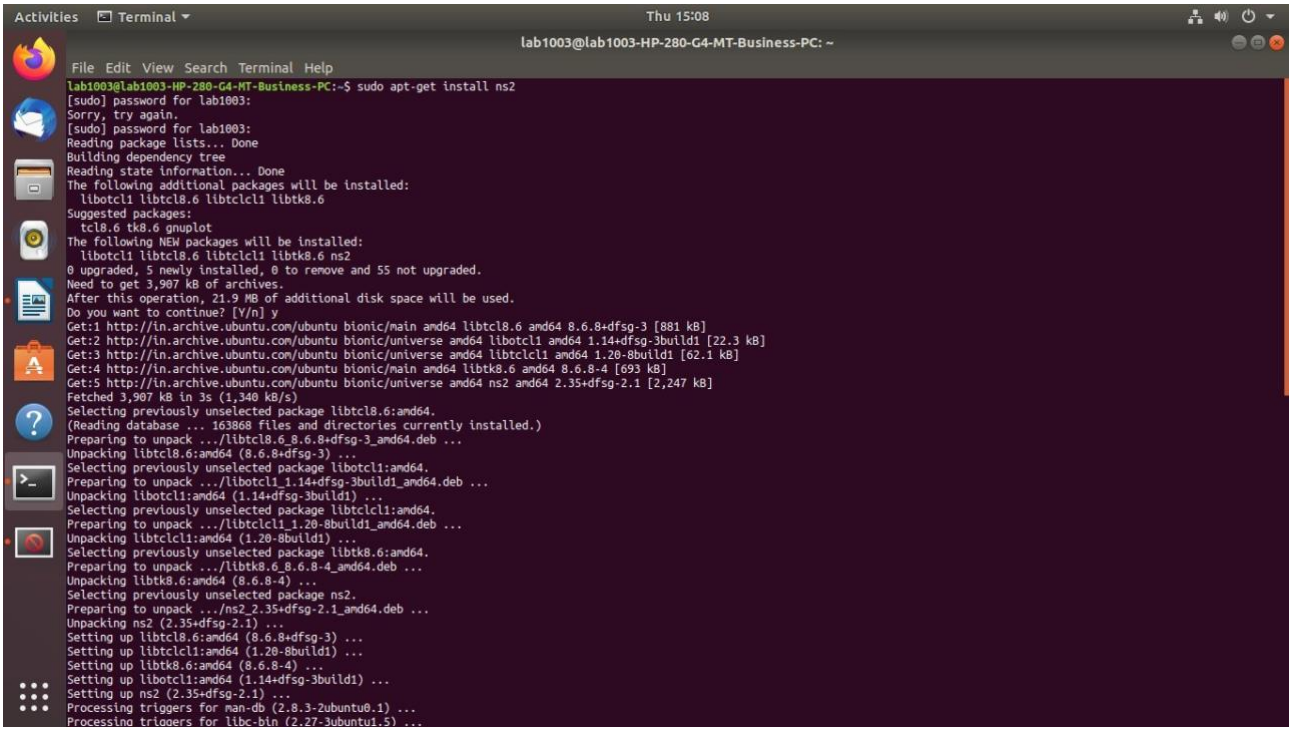
It is commonly used embedded into C applications, for rapid prototyping, scripted applications, GUIs, and testing. TCL interpreters are available for many operating systems, allowing TCL code to run on a wide variety of systems. Because TCL is a very compact language, it is used on embedded systems platforms, both in its full form and in several other small-footprint versions.

The popular combination of TCL with the Tk extension is referred to as TCL/Tk and enables building a graphical user interface (GUI) natively in TCL. TCL/Tk is included in the standard Python installation in the form of Tkinter.

Screenshots:

1) Installing ns2

sudo apt-get install ns2



2) Checking if ns2 is installed

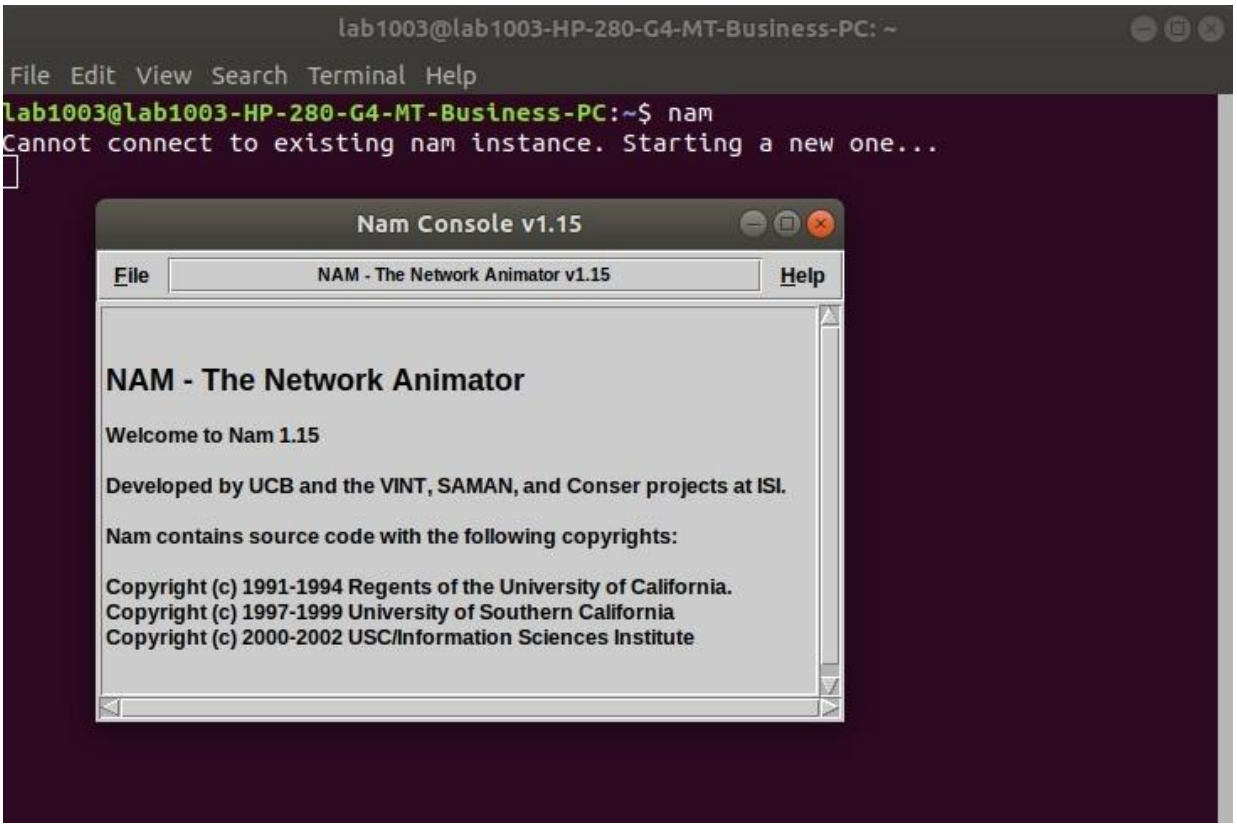
ns [% sign indicates it is installed]

3) Installing NAM

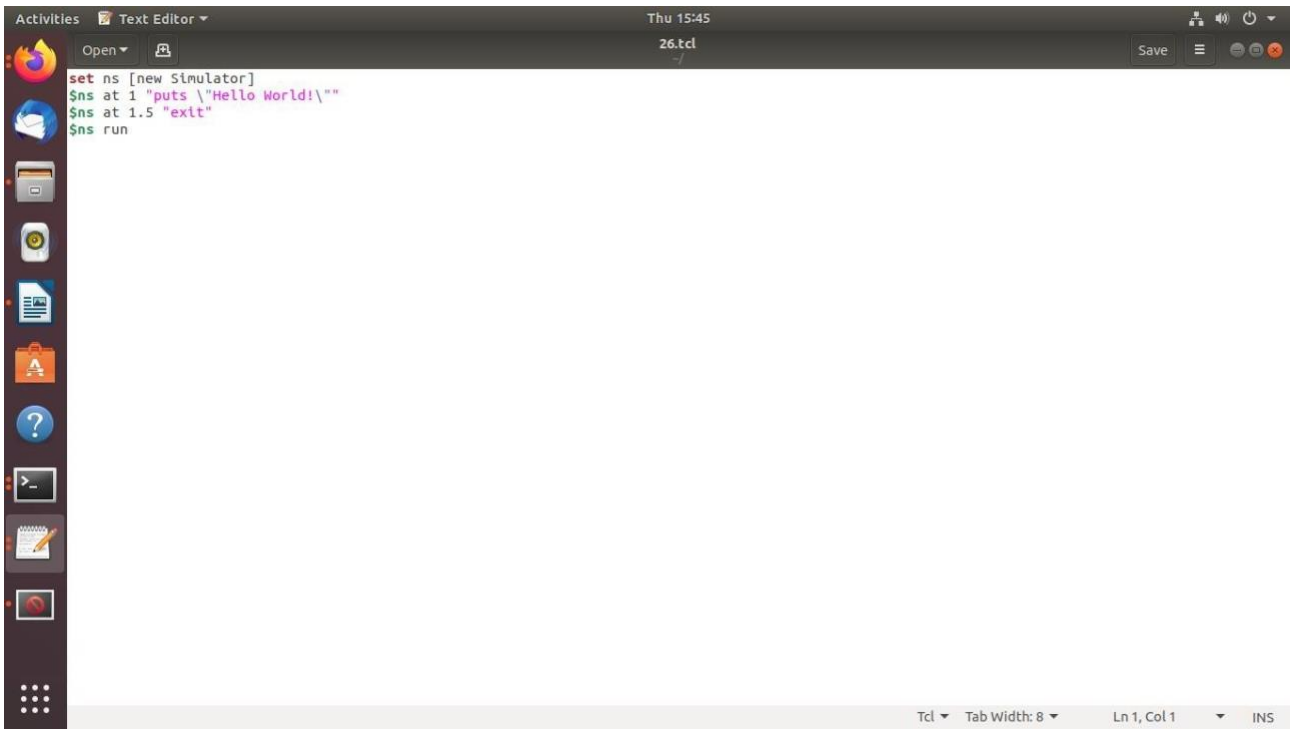
sudo apt-get purge nam
orsudo apt-get install nam

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo apt-get purge nam
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package 'nam' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 55 not upgraded.
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo apt-get install nam
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nam
0 upgraded, 1 newly installed, 0 to remove and 55 not upgraded.
Need to get 191 kB of archives.
After this operation, 683 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 nam amd64 1.15-4 [191 kB]
Fetched 191 kB in 1s (154 kB/s)
Selecting previously unselected package nam.
(Reading database ... 164239 files and directories currently installed.)
Preparing to unpack .../archives/nam_1.15-4_amd64.deb ...
Unpacking nam (1.15-4) ...
Setting up nam (1.15-4) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

4) Checking if NAM is installed



5) Running TCL file



Program:

```
set ns [new Simulator]
$ns at 1 "puts \"Hello World!\""
$ns at 1.5 "exit"
$ns run
```

Explanation:

1. set ns [new Simulator]:
   This line creates a new object called ns and assigns it an instance of the Simulator class. This suggests that the code is likely part of a network simulation scenario.

2. $ns at 1 "puts \"Hello World!\"":
   This line schedules an event at simulation time 1 (in some simulation time unit) to execute the TCL command puts "Hello World!". The puts command is used to print a message to the console. In this case, it prints the string "Hello World!".

3. $ns at 1.5 "exit":
   This line schedules another event at simulation time 1.5 to execute the TCL command exit. The exit command is used to terminate the TCL interpreter. So, the simulation will stop after printing "Hello World!".

4. $ns run:
   This command starts the simulation. The events scheduled with the at commandwill be executed in the specified order.

In summary, this program creates a network simulation environment using ns, schedules two events (printing "Hello World!" and exiting) at specific simulation times, and then runs the simulation. The simulation will print "Hello World!" at time 1 and exit at time 1.5.

Output:

Conclusion: LO – 2 (To install and configure network simulator and learn basics of TCL scripting) is achieved.

Name: Alphonz George Velacherry
Roll No.: 05

Lab Assignment No.: 03

What is TCP and UDP ?

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are both protocols used for sending data over the internet, but they have different characteristics:

1. TCP (Transmission Control Protocol):

- Provides reliable, ordered, and error-checked delivery of a stream of bytes.

- It's connection-oriented, meaning it establishes a connection before sending data and ensures that all data is received in the correct order.

- Generally slower than UDP due to the overhead of managing connections and ensuring reliability.

- Suitable for applications where data integrity and order are critical, such as web browsing, email, file transfer (FTP), and remote administration (SSH).

2. UDP (User Datagram Protocol):

- Provides a connectionless protocol for sending datagrams (packets) of data.

- It's faster and more efficient than TCP because it doesn't have the overhead of connection setup, teardown, and error checking.

- Does not guarantee delivery, ordering, or duplicate protection of packets.

- Suitable for applications where speed is more

Aim: To install and configure network simulator and learn basics of TCL scripting.

Lab Outcome Attained: LO – 2 Demonstrate the installation and configuration of network simulator.

Theory and Program :

Program:

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Purple
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set np [open out.tr w]
$ns trace-all $np

#Define a 'finish' procedure
proc finish {} {
global ns nf np
$ns flush-trace
#Close the NAM trace file
close $nf
#Execute NAM on the trace file
exec nam out.nam &
exit 0
}
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
$ns duplex-link $n4 $n1 2Mb 10ms DropTail

#Set Queue Size of link (n2-n3) to 10
```

```
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n1 $n4 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5


#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
#$ns attach-agent $n0 $tcp
$ns attach-agent $n4 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp

#Schedule events for the CBR and FTP agents
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n4 $tcp ; $ns detach-agent $n3 $sink"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
 $ns run
```

Working:

set ns [new Simulator]: This line creates a simulator object named ns'.

$ns color 1 Purple: This line sets the color of data flows with fid 1 to Purple in the NAM visualization tool.

$ns color 2 Red: This line sets the color of data flows with fid 2 to Red in the NAM visualization tool.

set nf [open out.nam w]: This line opens a file named out.nam for writing, which will be used to store the trace information for the NAM simulation.

$ns namtrace-all $nf : This line enables trace capturing for all events in the simulator and directs the output to the file out.nam'.

set np [open out.tr w]: This line opens a file named out.tr for writing, which will be used to store the trace information for the simulator.

$ns trace-all $np : This line enables trace capturing for all events in the simulator and directs the output to the file out.tr'.

proc finish {} {...}: This block defines a procedure named finish. It is called when the simulation finishes and performs tasks such as flushing traces, closing the NAM trace file, executing NAM on the trace file, and exiting the simulator.

set n'm' [$ns node]: This line creates a node object named 'n'm'' using the 'node' method of the simulator object.

$ns duplex-link $n0 $n2 2Mb 10ms DropTail: This line creates a duplex link between nodes n0 and n2 with a capacity of 2 Mbps, a delay of 10 ms, and a DropTail queue.

$ns duplex-link $n1 $n2 2Mb 10ms DropTail: This line creates a duplex link between nodes n1 and n2 with the same parameters as the previous line.

$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail: This line creates a duplex link between nodes n2 and n3 with a capacity of 1.7 Mbps, a delay of 20 ms, and a DropTail queue.

$ns duplex-link $n4 $n1 2Mb 10ms DropTail: This line creates a duplex link between nodes n4 and n1 with the same parameters as before.

$ns queue-limit $n2 $n3 100: This line sets the queue size of the link between nodes n2 and n3 to 100.

$ns duplex-link-op $n0 $n2 orient right-down: This line sets the direction of the link between nodes n0 and n2 in the NAM visualization tool.

$ns duplex-link-op $n1 $n2 orient right-up: This line sets the direction of the link between nodes n1 and n2 in the NAM visualization tool.

$ns duplex-link-op $n2 $n3 orient right: This line sets the direction of the link between nodes n2 and n3 in the NAM visualization tool.

$ns duplex-link-op $n1 $n4 orient right: This line sets the direction of the link between nodes n1 and n4 in the NAM visualization tool.

$ns duplex-link-op $n2 $n3 queuePos 0.5: This line monitors the queue position of the link between nodes n2 and n3 in the NAM visualization tool.

set tcp [new Agent/TCP]: This line creates a TCP agent object named 'tcp'.

$tcp set class_ 2: This line sets the class of the TCP agent to 2.

$ns attach-agent $n4 $tcp: This line attaches the TCP agent to node n4.

set sink [new Agent/TCPSink]: This line creates a TCP sink agent object named 'sink'.

$ns attach-agent $n3 $sink: This line attaches the TCP sink agent to node n3.

$ns connect $tcp $sink: This line establishes a TCP connection between the TCP agent and the TCP sink agent.

$tcp set fid_ 1: This line sets the flow ID of the TCP agent to 1.

set ftp [new Application/FTP]: This line creates an FTP application object named 'ftp'.

$ftp attach-agent $tcp: This line attaches the FTP application to the TCP agent.

Name: Alphonz George Velacherry

Roll No.: 05

Lab Assignment No.: 04

Aim: To install and configure network simulator and learn basics of TCL scripting.

Lab Outcome Attained: LO – 2 Demonstrate the installation and configuration of network simulator.

Program:

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Orange
$ns color 2 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set np [open out.tr w]
$ns trace-all $np

#Define a 'finish' procedure
proc finish {} {
      global ns nf np
      $ns flush-trace
      #Close the NAM trace file
      close $nf
      #Execute NAM on the trace file
         exec nam out.nam &
      exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 2ms DropTail
```

```
$ns duplex-link $n1 $n4 2Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n1 $n4 orient left

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n4 $udp

set null [new Agent/Null]
$ns attach-agent $n3 $null

$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp


# setting packet size
$cbr set packet_size_ 1000

#setting bit rate
$cbr set rate_ 1mb

# setting random false means no noise
$cbr set random_ false


#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 4.5 "$cbr stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation
$ns run
```

set udp [new Agent/UDP] : This line creates a new UDP agent object named 'udp'.

$ns attach-agent $n4 $udp : This line attaches the UDP agent to node n4.

set null [new Agent/Null] : This line creates a Null agent object named 'null'.

$ns attach-agent $n3 $null : This line attaches the Null agent to node n3.

$ns connect $udp $null : This line connects the UDP agent to the Null agent.

$udp set fid_ 2 : This line sets the flow ID of the UDP agent to 2.

set cbr [new Application/Traffic/CBR] : This line creates a CBR traffic application object named 'cbr'.

$cbr attach-agent $udp : This line attaches the CBR application to the UDP agent.

$cbr set packet_size_ 1000 : This line sets the packet size of the CBR application to 1000 bytes.

$cbr set rate_ 1mb : This line sets the bit rate of the CBR application to 1 Mbps.

$cbr set random_ false : This line sets the randomization of the CBR application to false, meaning there will be no variation in the packet sending rate.

$ns at 0.1 "$cbr start" : This line schedules the CBR application to start at 0.1 seconds in the simulation.

$ns at 4.5 "$cbr stop" : This line schedules the CBR application to stop at 4.5 seconds in the simulation.

$ns at 5.0 "finish" : This line schedules the 'finish' procedure to be called after 5.0 seconds of simulation time.

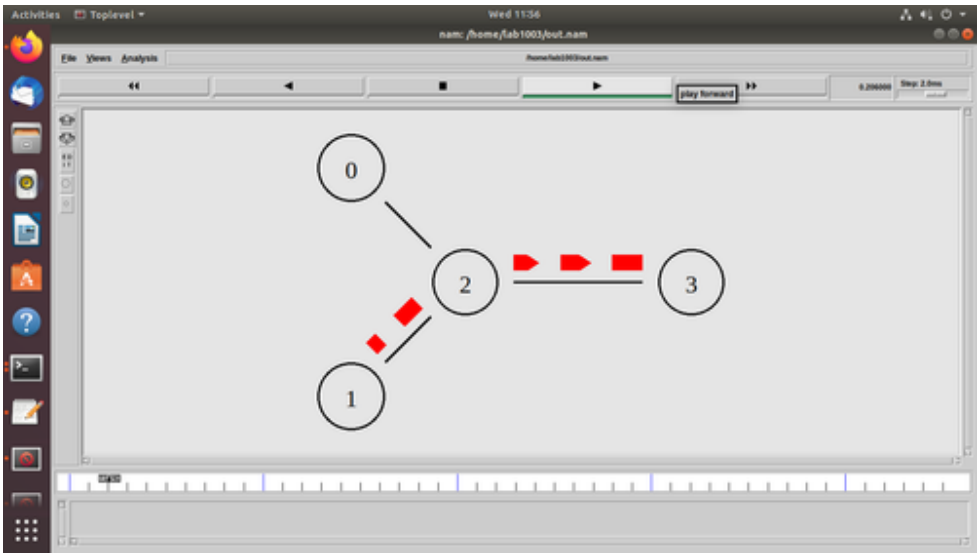
puts "CBR packet size = [$cbr set packet_size_]" : This line prints the packet size of the CBR application.

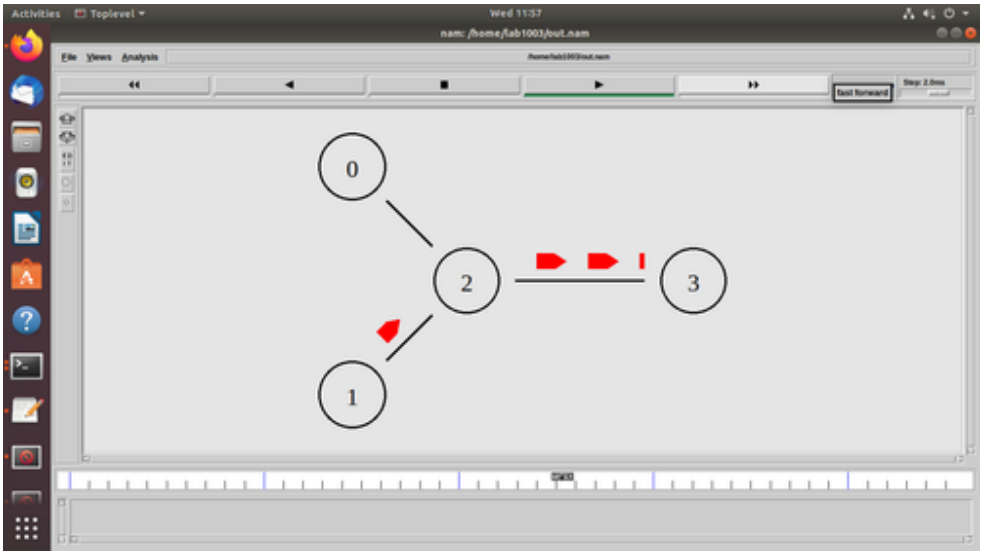
puts "CBR interval = [$cbr set interval_]" : This line prints the interval (rate) of the CBR application.

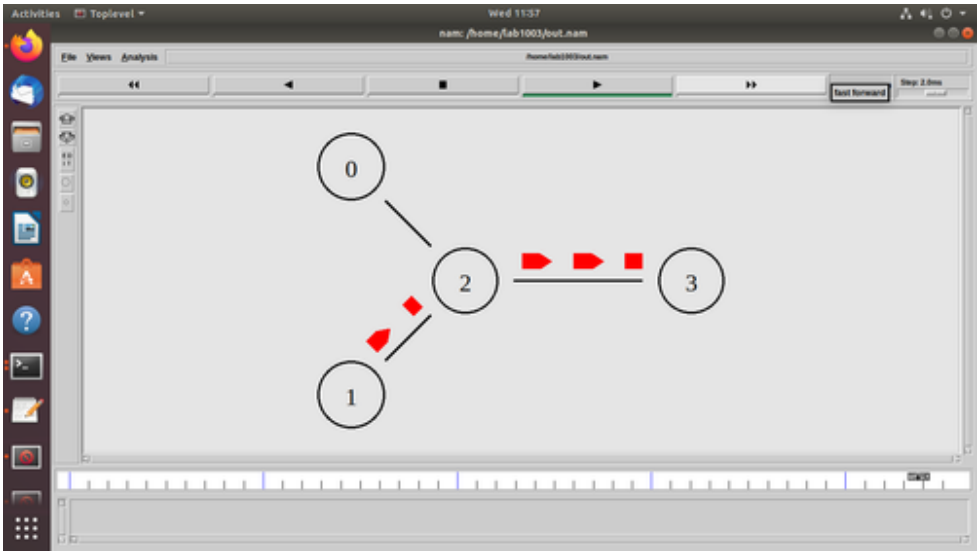
$ns run : This line starts the simulation.

Variations:

1. **We can change the color of the data**
2. **We can change the orientation of the Queue**
3. **We can change the shape , numbers and links between the nodes**
4. **We can change the speed and duration of the data**
5. **We can change the starting and the ending of the nodes as well.**







event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

r : receive (at to_node)
+ : enqueue (at queue)
- : dequeue (at queue)
d : drop (at queue)

src_addr : node.port (3.0)
dst_addr : node.port (0.0)

Trace :

<Event> <Time> <from> <to> <pktype> <size> --- <fid> <src> <dest> <seq> <pktid>

+ 0.1 1 2 CBR 1000 ------- 2 1.0 3.0 0 0
- 0.1 1 2 CBR 1000 ------- 2 1.0 3.0 0 0
+ 0.108 1 2 CBR 1000 ------- 2 1.0 3.0 1 1
- 0.108 1 2 CBR 1000 ------- 2 1.0 3.0 1 1
R 0.114 1 2 CBR 1000 ------- 2 1.0 3.0 0 0

+ 0.114 2 3 CBR 1000 ------- 2 1.0 3.0 0 0
- 0.114 2 3 CBR 1000 ------- 2 1.0 3.0 0 0
+ 0.116 1 2 CBR 1000 ------- 2 1.0 3.0 2 2
- 0.116 1 2 CBR 1000 ------- 2 1.0 3.0 2 2
R 0.122 1 2 CBR 1000 ------- 2 1.0 3.0 1 1
+ 0.122 2 3 CBR 1000 ------- 2 1.0 3.0 1 1
- 0.122 2 3 CBR 1000 ------- 2 1.0 3.0 1 1
+ 0.124 1 2 CBR 1000 ------- 2 1.0 3.0 3 3
- 0.124 1 2 CBR 1000 ------- 2 1.0 3.0 3 3
R 0.13 1 2 CBR 1000 ------- 2 1.0 3.0 2 2
+ 0.13 2 3 CBR 1000 ------- 2 1.0 3.0 2 2
- 0.13 2 3 CBR 1000 ------- 2 1.0 3.0 2 2
+ 0.132 1 2 CBR 1000 ------- 2 1.0 3.0 4 4
- 0.132 1 2 CBR 1000 ------- 2 1.0 3.0 4 4
R 0.138 1 2 CBR 1000 ------- 2 1.0 3.0 3 3
+ 0.138 2 3 CBR 1000 ------- 2 1.0 3.0 3 3
- 0.138 2 3 CBR 1000 ------- 2 1.0 3.0 3 3
R 0.138706 2 3 CBR 1000 ------- 2 1.0 3.0 0 0
+ 0.14 1 2 CBR 1000 ------- 2 1.0 3.0 5 5
- 0.14 1 2 CBR 1000 ------- 2 1.0 3.0 5 5
R 0.146 1 2 CBR 1000 ------- 2 1.0 3.0 4 4
+ 0.146 2 3 CBR 1000 ------- 2 1.0 3.0 4 4
- 0.146 2 3 CBR 1000 ------- 2 1.0 3.0 4 4

Name: Alphonz George Velacherry

Roll No.: 05

Lab Assignment No.: 04

Aim: To install and configure network simulator and learn basics of TCL scripting.

Lab Outcome Attained: LO – 2 Demonstrate the installation and configuration of network simulator.

Program:

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Orange
$ns color 2 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set np [open out.tr w]
$ns trace-all $np

#Define a 'finish' procedure
proc finish {} {
      global ns nf np
      $ns flush-trace
      #Close the NAM trace file
      close $nf
      #Execute NAM on the trace file
         exec nam out.nam &
      exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 2ms DropTail
```

```
$ns duplex-link $n1 $n4 2Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n1 $n4 orient left

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n4 $udp

set null [new Agent/Null]
$ns attach-agent $n3 $null

$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp


# setting packet size
$cbr set packet_size_ 1000

#setting bit rate
$cbr set rate_ 1mb

# setting random false means no noise
$cbr set random_ false


#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 4.5 "$cbr stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation
$ns run
```

set udp [new Agent/UDP] : This line creates a new UDP agent object named 'udp'.

$ns attach-agent $n4 $udp : This line attaches the UDP agent to node n4.

set null [new Agent/Null] : This line creates a Null agent object named 'null'.

$ns attach-agent $n3 $null : This line attaches the Null agent to node n3.

$ns connect $udp $null : This line connects the UDP agent to the Null agent.

$udp set fid_ 2 : This line sets the flow ID of the UDP agent to 2.

set cbr [new Application/Traffic/CBR] : This line creates a CBR traffic application object named 'cbr'.

$cbr attach-agent $udp : This line attaches the CBR application to the UDP agent.

$cbr set packet_size_ 1000 : This line sets the packet size of the CBR application to 1000 bytes.

$cbr set rate_ 1mb : This line sets the bit rate of the CBR application to 1 Mbps.

$cbr set random_ false : This line sets the randomization of the CBR application to false, meaning there will be no variation in the packet sending rate.

$ns at 0.1 "$cbr start" : This line schedules the CBR application to start at 0.1 seconds in the simulation.

$ns at 4.5 "$cbr stop" : This line schedules the CBR application to stop at 4.5 seconds in the simulation.

$ns at 5.0 "finish" : This line schedules the 'finish' procedure to be called after 5.0 seconds of simulation time.

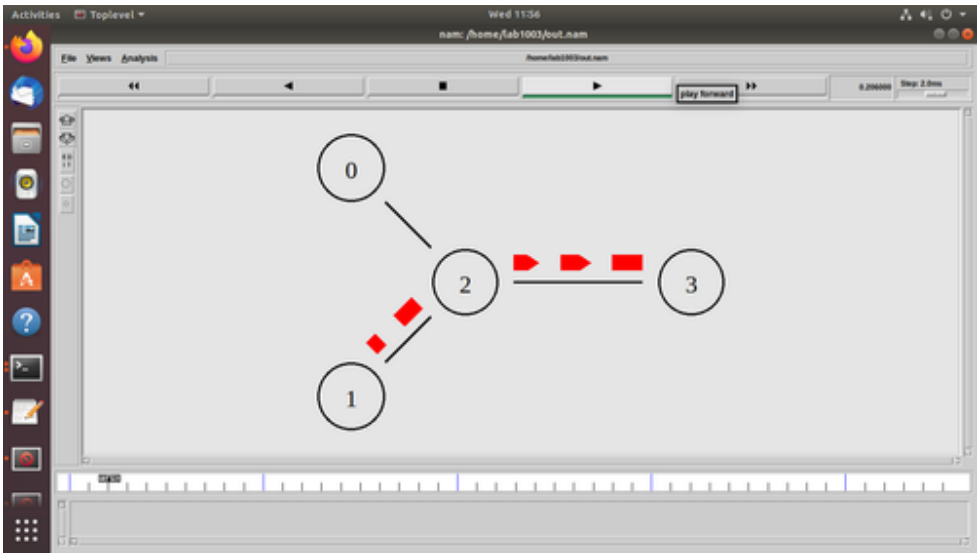
puts "CBR packet size = [$cbr set packet_size_]" : This line prints the packet size of the CBR application.

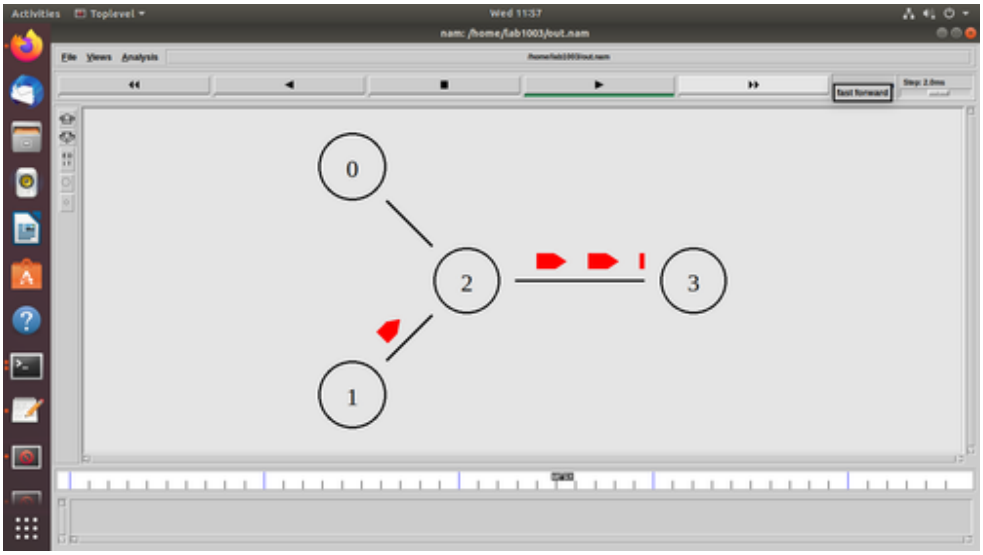
puts "CBR interval = [$cbr set interval_]" : This line prints the interval (rate) of the CBR application.

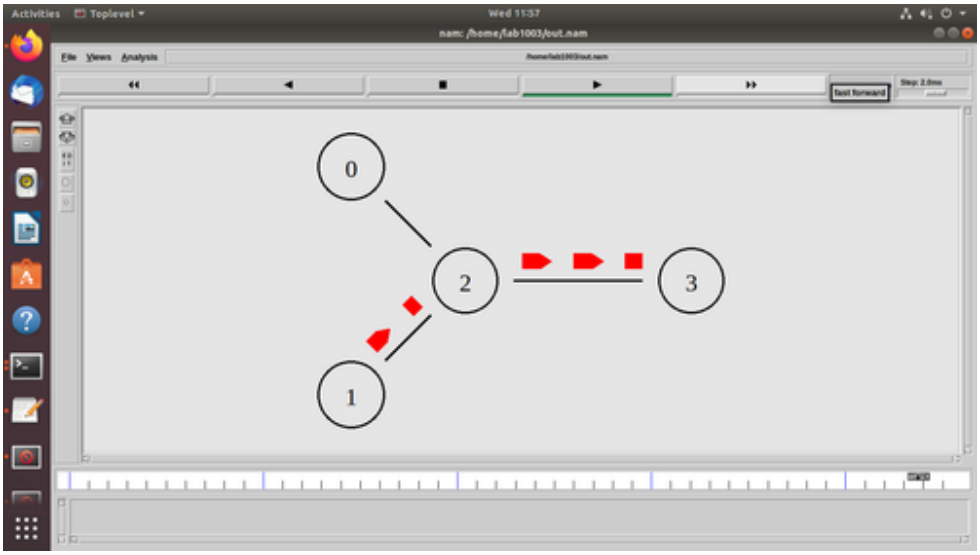
$ns run : This line starts the simulation.

Variations:

1. **We can change the color of the data**
2. **We can change the orientation of the Queue**
3. **We can change the shape , numbers and links between the nodes**
4. **We can change the speed and duration of the data**
5. **We can change the starting and the ending of the nodes as well.**







event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

r : receive (at to_node)
+ : enqueue (at queue)
- : dequeue (at queue)
d : drop (at queue)

src_addr : node.port (3.0)
dst_addr : node.port (0.0)

Trace :

<Event> <Time> <from> <to> <pktype> <size> --- <fid> <src> <dest> <seq> <pktid>

+ 0.1 1 2 CBR 1000 ------- 2 1.0 3.0 0 0
- 0.1 1 2 CBR 1000 ------- 2 1.0 3.0 0 0
+ 0.108 1 2 CBR 1000 ------- 2 1.0 3.0 1 1
- 0.108 1 2 CBR 1000 ------- 2 1.0 3.0 1 1
R 0.114 1 2 CBR 1000 ------- 2 1.0 3.0 0 0

+ 0.114 2 3 CBR 1000 ------- 2 1.0 3.0 0 0
- 0.114 2 3 CBR 1000 ------- 2 1.0 3.0 0 0
+ 0.116 1 2 CBR 1000 ------- 2 1.0 3.0 2 2
- 0.116 1 2 CBR 1000 ------- 2 1.0 3.0 2 2
R 0.122 1 2 CBR 1000 ------- 2 1.0 3.0 1 1
+ 0.122 2 3 CBR 1000 ------- 2 1.0 3.0 1 1
- 0.122 2 3 CBR 1000 ------- 2 1.0 3.0 1 1
+ 0.124 1 2 CBR 1000 ------- 2 1.0 3.0 3 3
- 0.124 1 2 CBR 1000 ------- 2 1.0 3.0 3 3
R 0.13 1 2 CBR 1000 ------- 2 1.0 3.0 2 2
+ 0.13 2 3 CBR 1000 ------- 2 1.0 3.0 2 2
- 0.13 2 3 CBR 1000 ------- 2 1.0 3.0 2 2
+ 0.132 1 2 CBR 1000 ------- 2 1.0 3.0 4 4
- 0.132 1 2 CBR 1000 ------- 2 1.0 3.0 4 4
R 0.138 1 2 CBR 1000 ------- 2 1.0 3.0 3 3
+ 0.138 2 3 CBR 1000 ------- 2 1.0 3.0 3 3
- 0.138 2 3 CBR 1000 ------- 2 1.0 3.0 3 3
R 0.138706 2 3 CBR 1000 ------- 2 1.0 3.0 0 0
+ 0.14 1 2 CBR 1000 ------- 2 1.0 3.0 5 5
- 0.14 1 2 CBR 1000 ------- 2 1.0 3.0 5 5
R 0.146 1 2 CBR 1000 ------- 2 1.0 3.0 4 4
+ 0.146 2 3 CBR 1000 ------- 2 1.0 3.0 4 4
- 0.146 2 3 CBR 1000 ------- 2 1.0 3.0 4 4

Name :Alphonz George Velacherry
Roll no:05

Lab Assignment No.: 06

Aim: Implementation of Socket Programming Using TCP

Lab Outcome Attained: LO – 2
Demonstrate the installation and configuration of network simulator.

Theory:

Socket programming using TCP (Transmission Control Protocol) is a method of establishing communication channels between devices or processes over a network. TCP is a reliable, connection-oriented protocol that ensures data integrity and ordered delivery.

Socket Creation: The first step is to create a socket, which is an endpoint for communication. In TCP, both the client and server applications create sockets.

Binding: For a server application, the next step is to bind the socket to a specific port on the host machine. This allows clients to connect to the server using the specified port number.

Listening for Connections: Once the server socket is bound to a port, it enters a passive listening state, waiting for incoming connection requests from clients.

Connection Establishment: A client initiates a connection by specifying the IP address and port number of the server it wants to connect to. The server accepts the connection request, creating a new socket for communication with that specific client.

Data Exchange: With the connection established, both the client and server can send and receive data through their respective sockets. TCP ensures that data is delivered reliably and in order, handling any packet loss or network congestion.

Connection Termination: When either the client or server wants to end the communication, they send a termination request, and the connection is closed gracefully.
Socket programming with TCP provides a robust foundation for building networked applications that require reliable, ordered data transmission. It's widely used in various scenarios such as web servers, email clients, file transfer protocols, and more. By leveraging TCP's features for error recovery and flow control, developers can create applications that are resilient to network fluctuations and ensure the integrity of transmitted data.

Program:

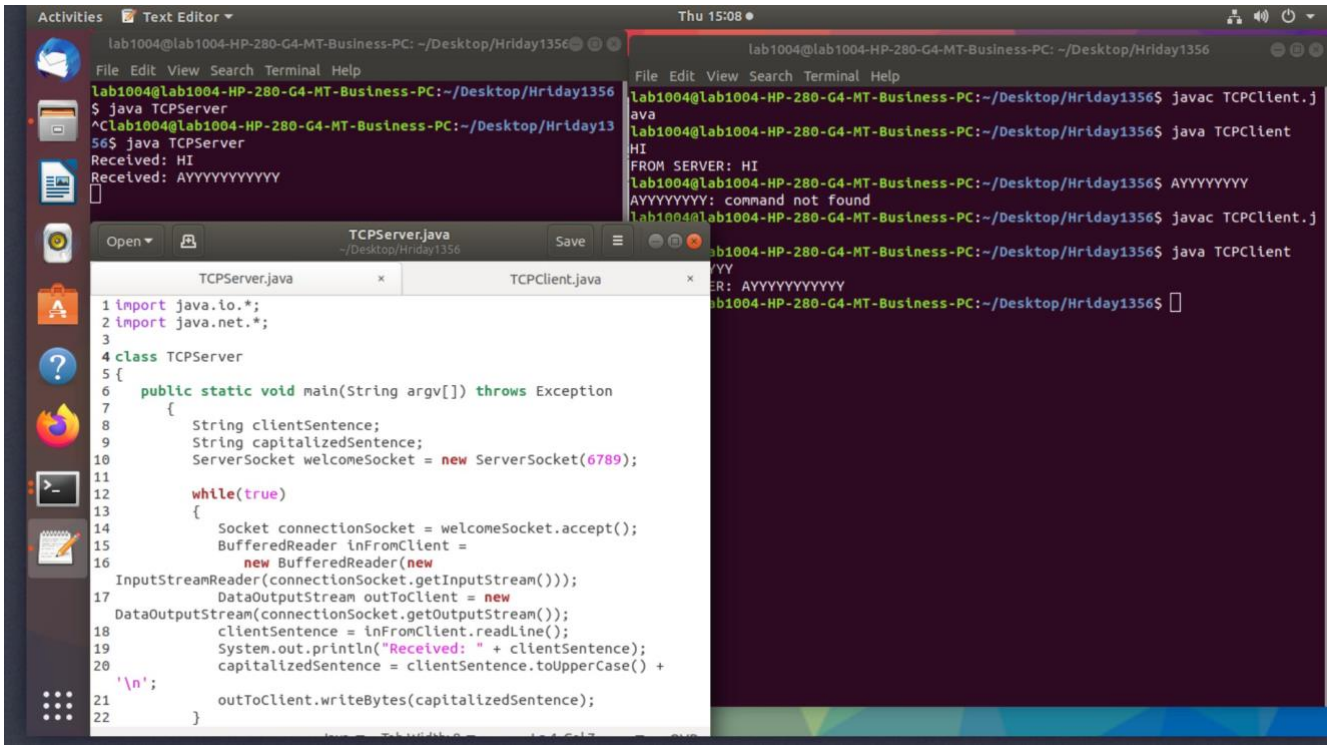
Client Side Server

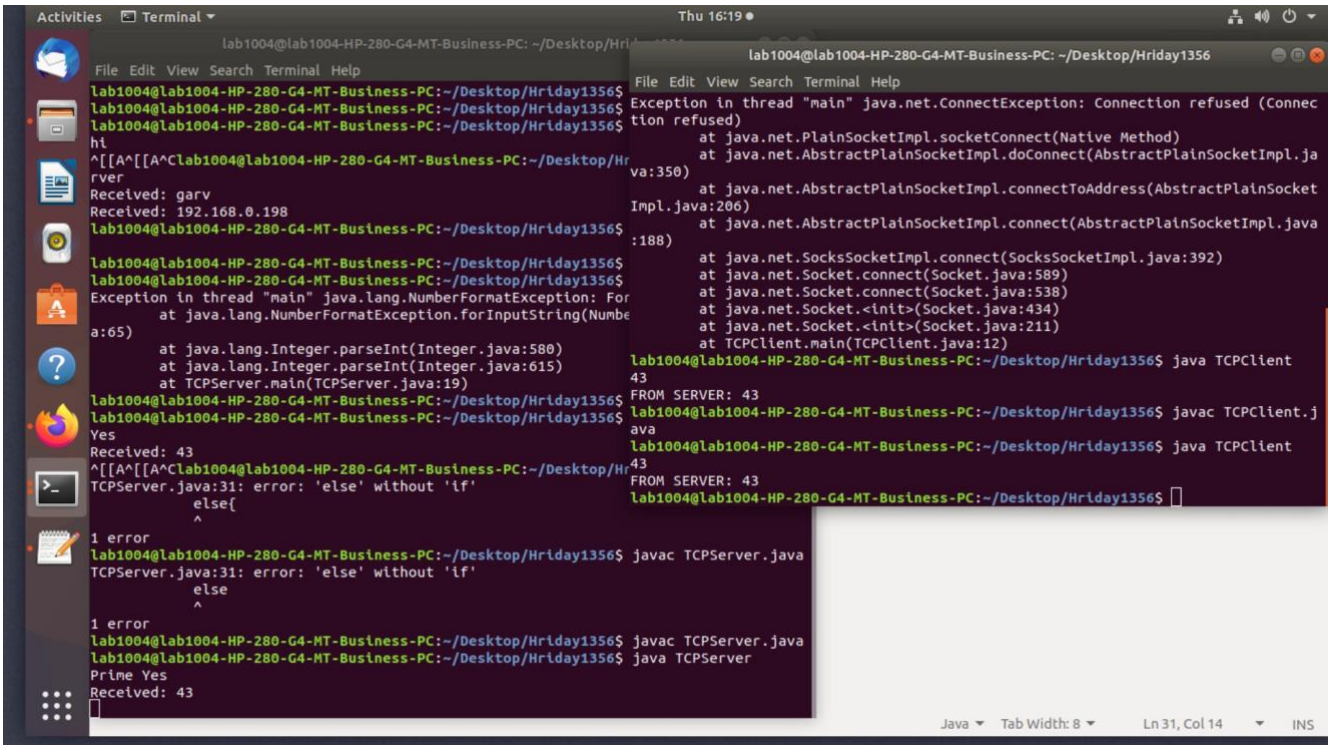
```
import java.io.*;
import java.net.*;

class TCPClient
{
 public static void main(String argv[]) throws Exception
 {
  String sentence;
  String modifiedSentence;
  BufferedReader inFromUser = new BufferedReader( new InputStreamReader(System.in));
  Socket clientSocket = new Socket("localhost", 6789);
  DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
BufferedReader inFromServer = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
  sentence = inFromUser.readLine();
  outToServer.writeBytes(sentence + '\n');
  modifiedSentence = inFromServer.readLine();
  System.out.println("FROM SERVER: " + modifiedSentence);
  clientSocket.close();
 }
}
```

Server Side Program for printing prime:

```
import java.io.*;
import java.net.*;

import java.io.*;
import java.net.*;

class TCPServer {
   public static void main(String argv[]) throws Exception {
      ServerSocket welcomeSocket = new ServerSocket(6789);

      while (true) {
         Socket connectionSocket = welcomeSocket.accept();
         BufferedReader inFromClient =
               new BufferedReader(new
InputStreamReader(connectionSocket.getInputStream()));
         DataOutputStream outToClient = new
DataOutputStream(connectionSocket.getOutputStream());
         String clientSentence = inFromClient.readLine();

         int number = Integer.parseInt(clientSentence);
         boolean isPrime = isPrime(number);
```

```
            outToClient.writeBytes(isPrime + "\n");
        }
    }

    private static boolean isPrime(int number) {
        if (number <= 1)
            return false;
        for (int i = 2; i * i <= number; i++) {
            if (number % i == 0)
                return false;
        }
        return true;
    }
}
```




Activities
Terminal
Thu 16:19
File Edit View Search Terminal Help
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Desktop/Hriday1356$
hi
rver
Received: garv
Received: 192.168.0.198
Exception in thread "main" java.lang.NumberFormatException: For
at java.lang.NumberFormatException.forInputString(Numbe
a:65)
at java.lang.Integer.parseInt(Integer.java:580)
at java.lang.Integer.parseInt(Integer.java:615)
at TCPServer.main(TCPServer.java:19)
Yes
Received: 43
TCPServer.java:31: error: 'else' without 'if'
else{
^
1 error
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Desktop/Hriday1356$ javac TCPServer.java
TCPServer.java:31: error: 'else' without 'if'
else
^
1 error
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Desktop/Hriday1356$ javac TCPServer.java
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Desktop/Hriday1356$ java TCPServer
Prime Yes
Received: 43
lab1004@lab1004-HP-280-G4-MT-Business-PC: ~/Desktop/Hriday1356
File Edit View Search Terminal Help
Exception in thread "main" java.net.ConnectException: Connection refused (Connection refused)
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:589)
at java.net.Socket.connect(Socket.java:538)
at java.net.Socket.<init>(Socket.java:434)
at java.net.Socket.<init>(Socket.java:211)
at TCPClient.main(TCPClient.java:12)
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Desktop/Hriday1356$ java TCPClient
43
FROM SERVER: 43
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Desktop/Hriday1356$ javac TCPClient.java
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Desktop/Hriday1356$ java TCPClient
43
FROM SERVER: 43
Java
Tab Width: 8
Ln 31, Col 14
INS

S11 05
Alphonz George
Velacherry

Assignment 7

Socket Programing using UDP

Client Side UDP Socket
:

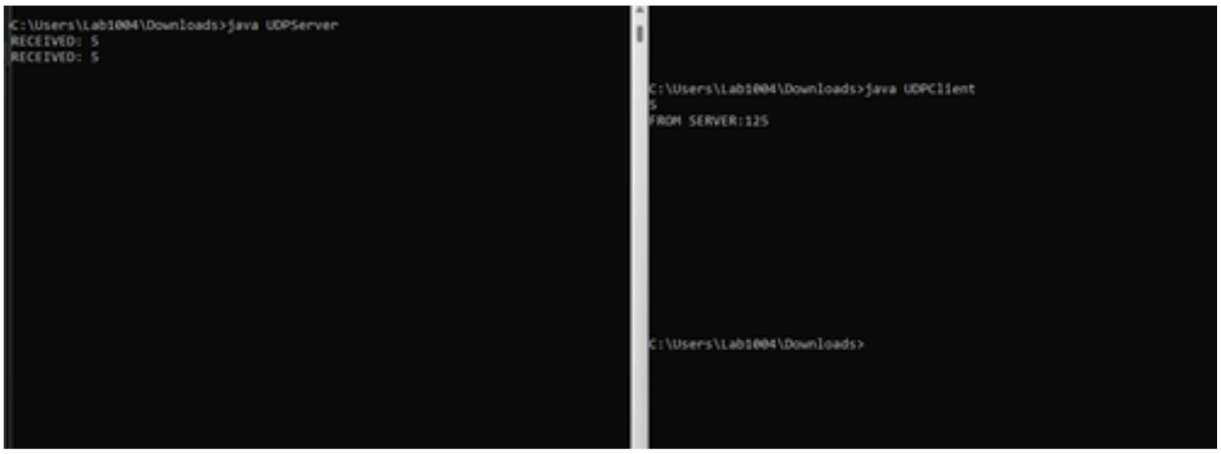
```
import java.io.*;
import java.net.*;

class UDPClient
{
   public static void main(String args[]) throws Exception
   {
      BufferedReader inFromUser =
         new BufferedReader(new InputStreamReader(System.in));
          DatagramSocket clientSocket = new DatagramSocket();
          InetAddress IPAddress = InetAddress.getByName("localhost");
          byte[] sendData = new byte[1024];
          byte[] receiveData = new byte[1024];
          String sentence = inFromUser.readLine();
          sendData = sentence.getBytes();
          DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
IPAddress, 9876);
 clientSocket.send(sendPacket);
 DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
 clientSocket.receive(receivePacket);
 String modifiedSentence = new String(receivePacket.getData());
 System.out.println("FROM SERVER:" + modifiedSentence);
 clientSocket.close();
   }
}
```

S11 05
Alphonz George
Velacherry

Server Side UDP Socket to print cube

```
public class UDPServer {

    public static void main(String args[]) throws Exception {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];

        while (true) {
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            serverSocket.receive(receivePacket);

            String sentence = new String(receivePacket.getData(), 0, receivePacket.getLength()); //
Use actual length
            System.out.println("RECEIVED: " + sentence);

            InetAddress IPAddress = receivePacket.getAddress();
            int port = receivePacket.getPort();

            String capitalizedSentence = sentence.toUpperCase();

            try {
                int num = Integer.parseInt(capitalizedSentence);
                num = num * num * num;
                capitalizedSentence = Integer.toString(num);
            } catch (NumberFormatException e) {
                capitalizedSentence = "Invalid input: Not a number";
            }

            sendData = capitalizedSentence.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
IPAddress, port);
            serverSocket.send(sendPacket);
        }
    }
}
```

S11 05
Alphonz George
Velacherry



Report for Lab 3-1: UDP

Name: Alphonz George Velacherry	Student ID: S11-05	

1	a. Source port number:53	b. Destination port number:53
	c. Total length of user diagram:68 bytes (payload only)	d. Length of data:68 bytes (same as total length as it's a UDP packet)
	e. Is the packet from client or server? Server	f. Application layer protocolDNS
	g. Is checksum calculated? No.Unverified	
2	Are answer in number 1 are verified by the information in the detail pane lane? Yes, the answers in question 1 seem to be verified by the information in the detail pane of Frame 71.	
3	Source and destination IP addresses in the query message: 192.168.1.138 (client) and 192.168.1.1 (router) (from Frame 67) Source and destination IP addresses in the response) message: Yes, the answers in question 1 seem to be verified by the information in the detail pane of Frame 71. Relation between IP addresses: They are on the same private network (192.168.1.x).	
4	Source and destination port number in the query message: 47681 (client) and 53 (DNS) (from Frame 67) Source and destination port number in the response message: 53 (DNS) and 47681 (client) (from Frame 71) Relation between port numbers:They are different ports: Each communication uses a dynamically assigned port (47681 in this case) on the client side and the well-known DNS port (53) on the server side. Which port number is well-known? 53	
5	The length of the first UDP packet: 1399 Packet 19 How many bytes of payload are carried by the first UDP packet? 455	
6	Number of bytes in the DNS message: 688 bits or 86 bytes Does the count agree with the answer to question 5? NO	
7	Is the checksum calculated for the first UDP packet? **Frame 67 (query):** UDP is connectionless and doesn't have a checksum at the protocol level. Wireshark shows it as unverified.	

	• **Frame 71 (response):** UDP is still connectionless, and the checksum information is unverified Value of the checksum: unverified

Report for LAB 3-2: TCP

Name: Alphonz George	**Student ID: S11-05**

Part I	
1	Socket addresses: **Client:** IP 192.168.1.138, Port 59992 **Server:** IP 142.250.192.132, Port 443
2	Set flags: **Frame 65:** SYN, ACK **Frame 66:** ACK
3	Sequence number and acknowledgement number: **Frame 65:** • Sequence number: 0 (server) • Acknowledgement number: 1 (server acknowledging client's SYN) **Frame 66:** • Sequence number: 1 (client) • Acknowledgement number: 1 (client acknowledging server's SYN-ACK)
4	Window size: Window size value: 502 (client)

Part II	
1	Set flag in HTTP GET message: Based on Frame 16222, the flags in the HTTP GET message are: • **PSH (Push):** Indicates this is the last segment of the HTTP GET message. • **ACK:** Acknowledges receiving data from the server.
2	Number of bytes transmitted by the HTTP GET message: The "TCP Segment Len" field in Frame 16222 shows 87 bytes. However, this only represents the size of this particular segment, not the entire message. The complete message size might be spread across multiple segments.
3	Acknowledgement frequency: TCP acknowledges received data segments, but the frequency isn't directly mentioned in the frame. It depends on various factors like window size, network conditions, and server configuration. Corresponding rule:
4	Number of bytes transmitted by each packet:

	Similar to point 2, the "TCP Segment Len" (87 bytes) only reflects the size of this specific segment. The actual number of bytes transmitted per packet depends on the network's MTU (Maximum Transmission Unit). Relation to sequence and acknowledgement Number:
5	Original window sizes: Frame 16222 shows a window size value of 502 bytes. However, the "Calculated window size" is 64256 bytes, indicating a scaling factor of 128 applied. It's likely the initial window size was smaller, and it increased based on previous acknowledgements from the receiver. Are these numbers expected? **Calculated window size of 64256 bytes:** This larger size is expected because it considers the **window scaling factor of 128**. This scaling allows for larger window sizes than the standard 2-byte field in the TCP header can represent. How window sizes change? TCP utilizes a dynamic window size mechanism to optimize data flow and prevent congestion. Here's an overview of how window sizes change: • **Initial Window Size:** This is set when the connection is established and depends on various factors. It is typically a small value to avoid overwhelming the receiver initially. • **Window Increase:** When the receiver acknowledges received data, the sender's window size increases. This allows the sender to send more data without overwhelming the receiver's buffer.
6	How the window size is used in flow control? **TCP utilizes window size to prevent overwhelming the receiver's buffer.** It acts as a credit system, allowing the sender to transmit data up to the advertised window size before needing an acknowledgement (ACK) from the receiver. • **Window size is dynamic and changes based on various factors:** • **Increases:** When the receiver acknowledges received data, the sender's window size increases, allowing it to send more data. • **Decreases:** This can happen due to congestion control mechanisms (e.g., fewer ACKs due to congestion) or receiver buffer limitations (receiver advertises a zero-window size when full).
7	Purpose of the HTTP OK message: The purpose of the HTTP OK (status code 200) message from the server is to indicate **a successful request**. It signifies that: • **The server understood the request:** The server successfully parsed and interpreted the HTTP request sent by the client (e.g., the browser). • **The resource was located:** The server was able to find the resource requested by the client (e.g., a specific web page, image, or file). • **The resource is being sent:** The server is actively sending the requested resource data

	back to the client. Therefore, the HTTP OK message acts as a confirmation and positive response to the client's request, informing them that their request was processed successfully and the desired information is being delivered.

Part III	
1	Number of TCP segments exchanged for connection termination:
1	Which end point started the connection termination phase?
2	Flags sets in each of the segments used for connection termination:

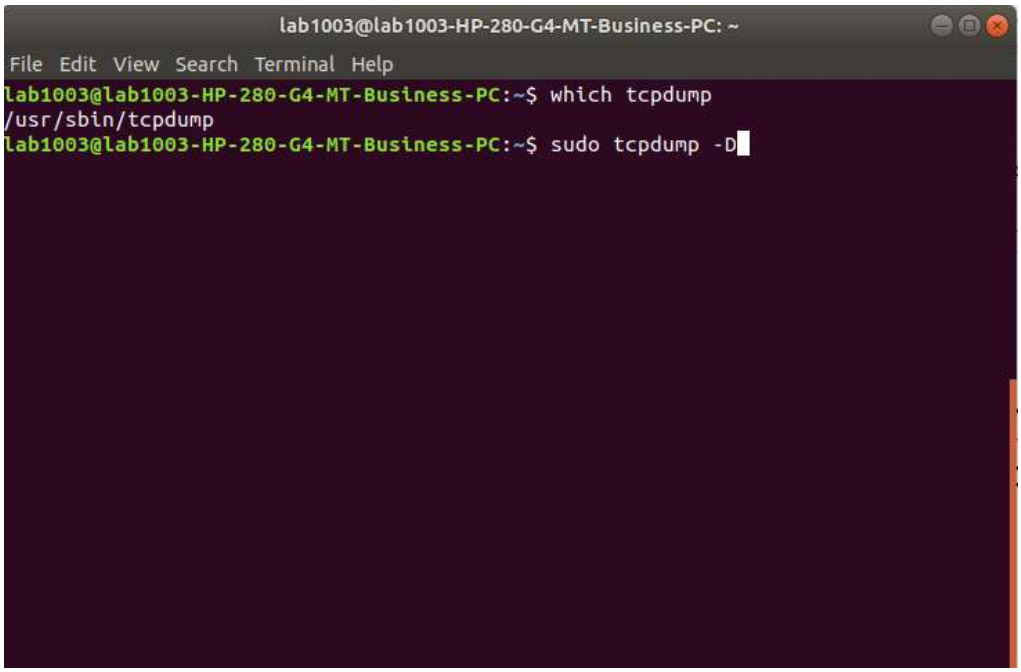
Part IV		
1	a. Source port number: 59992	b. Destination port number: 443
	c. Sequence number : 1	d. Acknowledgement number: 1
	e. Heather length: 32	f. Set flags: ACK
	g. Window size: 502	h. Urgent pointer:0
2	Are answer in the question number 1 verified by the information in the detail pane lane? Yes	
3	Does any of the TCP packet headers carry options? No potions Explain	
4	Size of a TCP packet with no option: 20 Bytes plus Data length Size of a TCP packet with options: 20 bytes + OPTION LENGTH + DATA LENGTH	
5	Is window size in any of the TCP packet zero? NO Explain:	

Alphonz George Velacherry
S11-05

Assignment - TCPDUMP

1.) **Install Tcpdump**

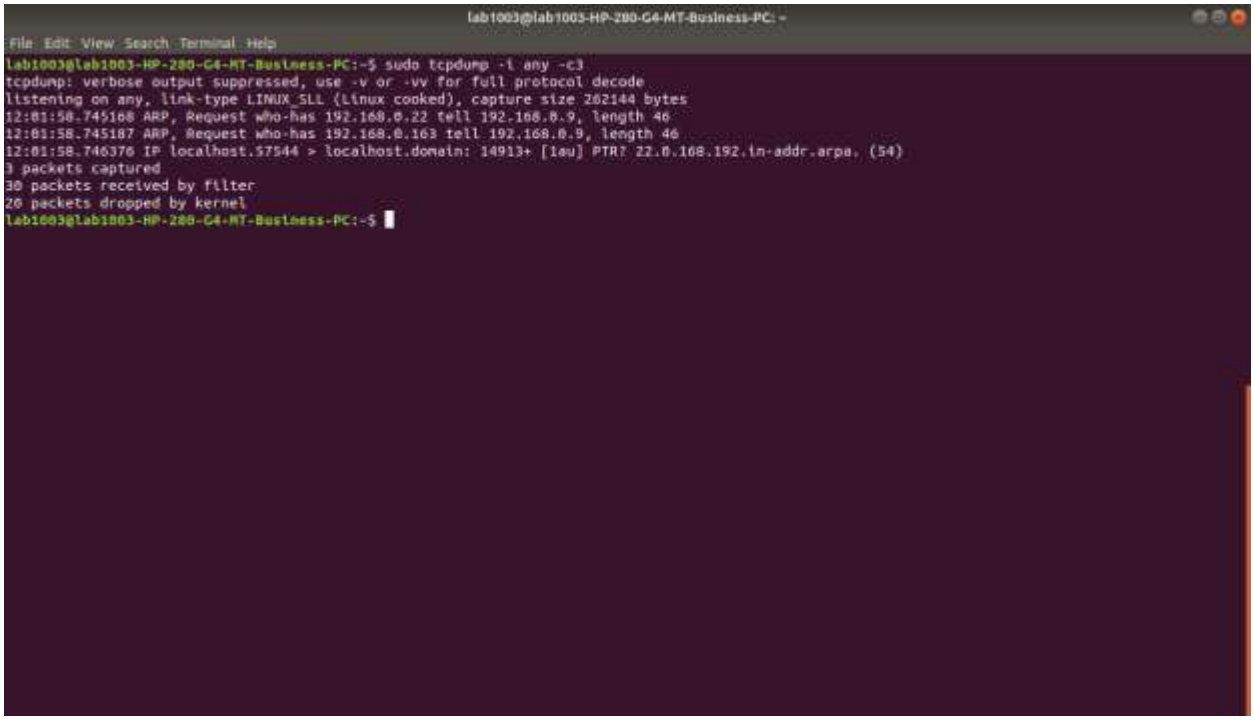
Command = which tcpdump



1.) **$sudo tcpdump -i any -c3**

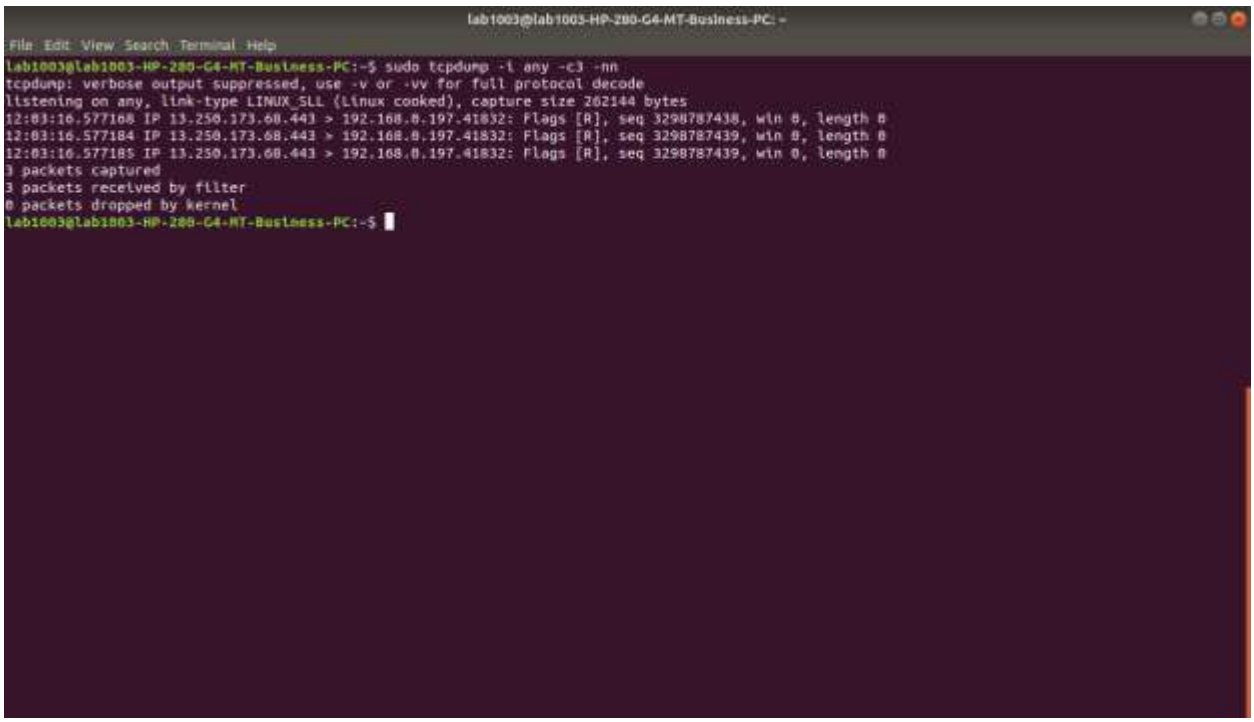
We can also limit the number of packets to be captured using the “-c” flag that signifies the “count.” To capture 3 packets, use:

Alphonz George Velacherry
S11-05



2.) sudo tcpdump -i any -c3 -nn

The "**tcpdump**" command captures packets with IP and port names by default but to clean-up, the mess and make the output easier to understand; the names can be disabled using "**-n**" and "**-nn**" for the port option:

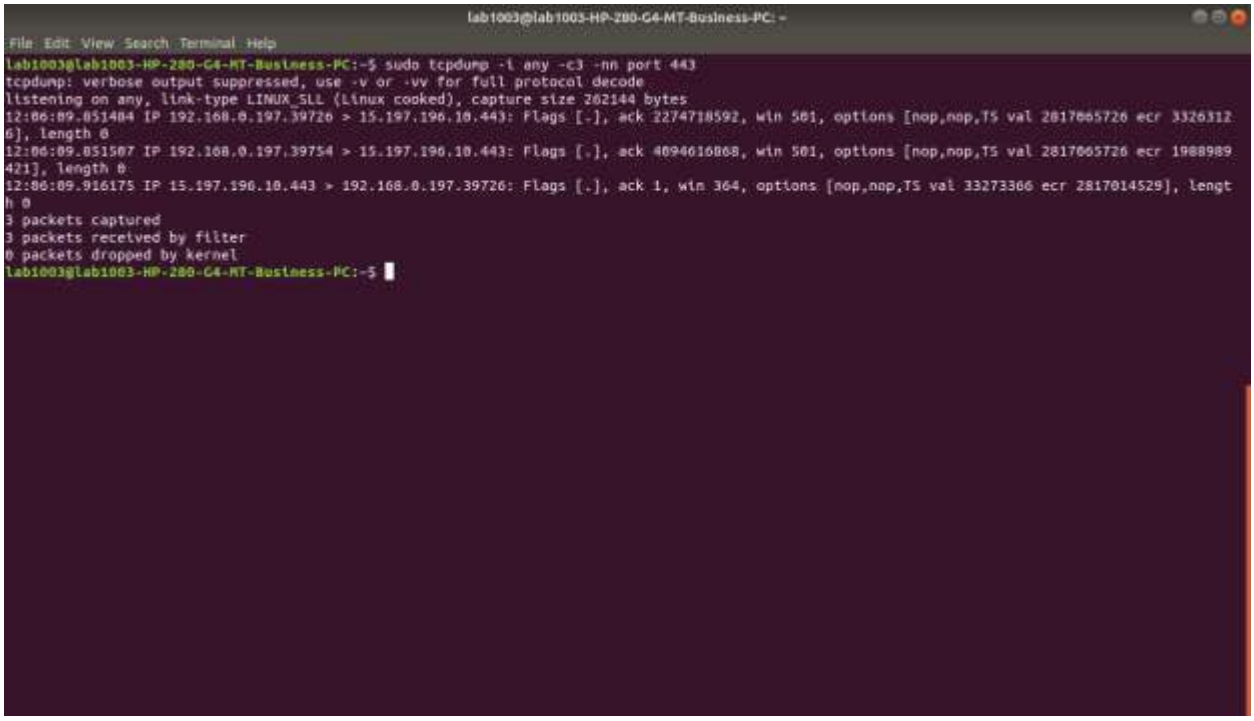


Alphonz George Velacherry
S11-05

How to capture a packet from tcpdump

A packet can have various fields, but the general ones are displayed above. The first field, "**09:48:18.960683,**" represents a time when the packet is received. Next comes IP addresses; the first IP [**216.58.209.130**] is the source IP and the second IP [**10.0.2.15.55812**] is the destination IP. Then you will get the flag [**P.**]; a list of typical flags are given below:

3.) sudo tcpdump -i any -c3 -nn port 443



Alphonz George Velacherry
S11-05

4.)sudo tcpdump -i any -c10 -nn host 192.168.0.164 and port 80

```
0 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any -c10 -nn host 192.168.0.164 and port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
12:22:13.754709 IP 192.168.0.164.39638 > 142.250.183.67.80: Flags [.], ack 4017781957, win 501, options [nop,nop,TS val 3568674142 ecr 2201067
97], length 0
12:22:13.758908 IP 142.250.183.67.80 > 192.168.0.164.39638: Flags [.], ack 1, win 269, options [nop,nop,TS val 228117038 ecr 3568602773], leng
th 0
12:22:15.120388 IP 192.168.0.164.44524 > 120.138.114.137.80: Flags [S], seq 4075068419, win 64240, options [mss 1460,sackOK,TS val 3002044950
ecr 0,nop,wscale 7], length 0
12:22:15.123689 IP 120.138.114.137.80 > 192.168.0.164.44524: Flags [S.], seq 1474589129, ack 4075068420, win 65160, options [mss 1440,sackOK,T
S val 1821334805 ecr 3002044950,nop,wscale 7], length 0
12:22:15.123638 IP 192.168.0.164.44524 > 120.138.114.137.80: Flags [.], ack 1, win 502, options [nop,nop,TS val 3002044953 ecr 1821334805], le
ngth 0
12:22:15.124392 IP 192.168.0.164.44524 > 120.138.114.137.80: Flags [P.], seq 1:424, ack 1, win 502, options [nop,nop,TS val 3002044954 ecr 182
1334805], length 423: HTTP: POST / HTTP/1.1
12:22:15.127629 IP 120.138.114.137.80 > 192.168.0.164.44524: Flags [.], ack 424, win 500, options [nop,nop,TS val 1821334810 ecr 3002044954],
length 0
12:22:15.129229 IP 120.138.114.137.80 > 192.168.0.164.44524: Flags [P.], seq 1:889, ack 424, win 500, options [nop,nop,TS val 1821334811 ecr 3
002044954], length 888: HTTP: HTTP/1.1 200 OK
12:22:15.129301 IP 192.168.0.164.44524 > 120.138.114.137.80: Flags [.], ack 889, win 501, options [nop,nop,TS val 3002044959 ecr 1821334811],
length 0
12:22:15.179091 IP 192.168.0.164.36274 > 10.67.196.194.80: Flags [S], seq 1329172547, win 64240, options [mss 1460,sackOK,TS val 2708771995 ec
r 0,nop,wscale 7], length 0
10 packets captured
10 packets received by filter
0 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$
```

Different filtering options can be combined using logical operators like “and/or”:

5.)sudo tcpdump -i any -c5 -w packetData.pcap

The extension of the file would be “.pcap,” which stands for “packet capture.” Once the capturing is done, the file would be saved in your local drive. This file cannot be opened or read using any text editor program. To read it, use the “**-r**” flag with “tcpdump”:

```
lab1003@lab1003-HP-280-G4-MT-Business-PC: ~
File  Edit  View  Search  Terminal  Help
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any -c5 -w packetData.pcap
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
5 packets captured
71 packets received by filter
0 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ tcpdump -r packetData.pcap
reading from file packetData.pcap, link-type LINUX_SLL (Linux cooked)
12:07:47.256919 IP ec2-13-250-173-68.ap-southeast-1.compute.amazonaws.com.https > lab1003-HP-280-G4-MT-Business-PC.33484: Flags [P.], seq 7508
14044:750814068, ack 3176335751, win 251, options [nop,nop,TS val 349019609 ecr 1689833886], length 24
12:07:47.256923 IP ec2-13-250-173-68.ap-southeast-1.compute.amazonaws.com.https > lab1003-HP-280-G4-MT-Business-PC.33484: Flags [F.], seq 24,
ack 1, win 251, options [nop,nop,TS val 349019609 ecr 1689833886], length 0
12:07:47.256963 IP lab1003-HP-280-G4-MT-Business-PC.33484 > ec2-13-250-173-68.ap-southeast-1.compute.amazonaws.com.https: Flags [.], ack 24, w
in 501, options [nop,nop,TS val 1689834473 ecr 349019609], length 0
12:07:47.257187 IP lab1003-HP-280-G4-MT-Business-PC.33484 > ec2-13-250-173-68.ap-southeast-1.compute.amazonaws.com.https: Flags [P.], seq 1:40
, ack 25, win 501, options [nop,nop,TS val 1689834473 ecr 349019609], length 39
12:07:47.257541 IP lab1003-HP-280-G4-MT-Business-PC.33484 > ec2-13-250-173-68.ap-southeast-1.compute.amazonaws.com.https: Flags [P.], seq 40:6
4, ack 25, win 501, options [nop,nop,TS val 1689834474 ecr 349019609], length 24
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$
```

ALphonz George

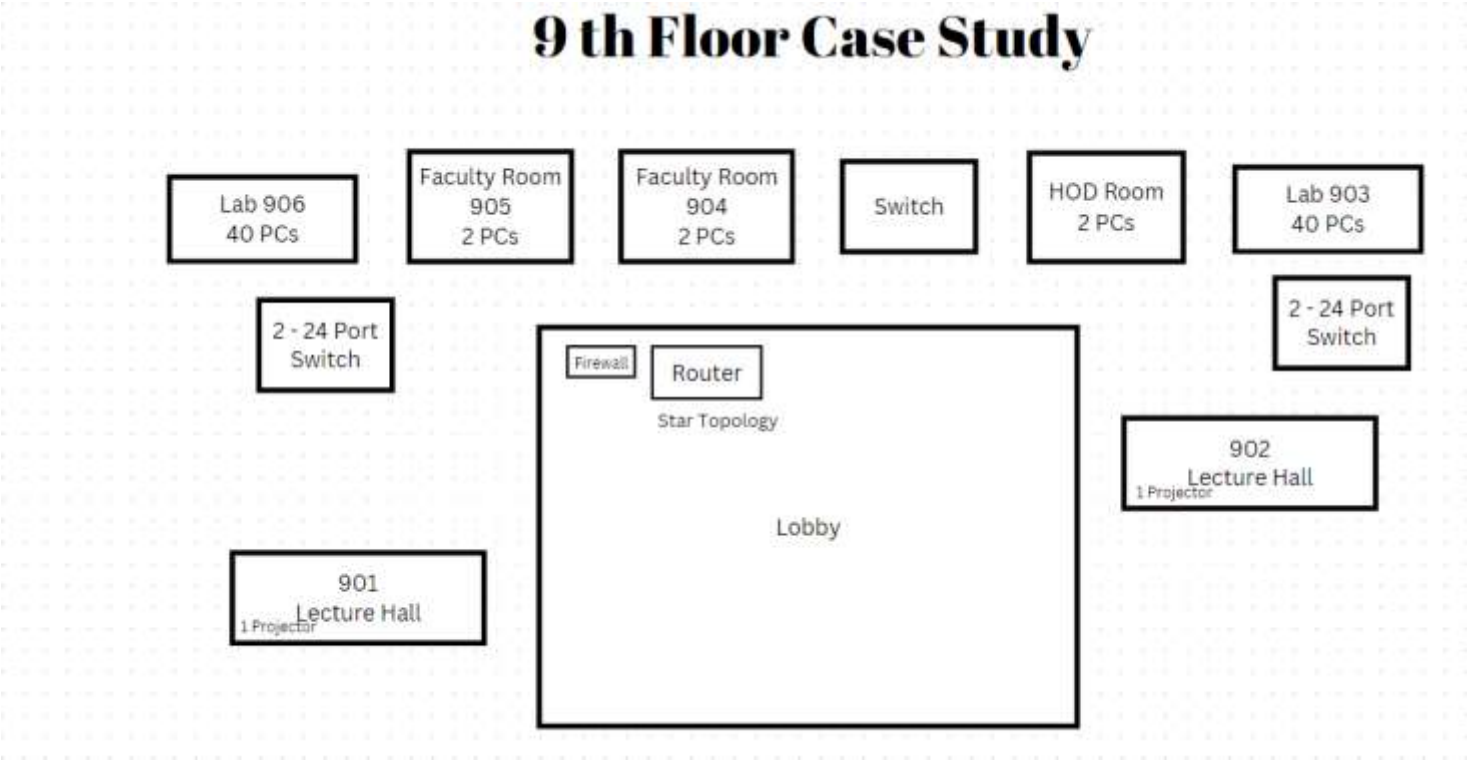
S11-05

12/03/2024

Lab Assignment No.10

Aim: A case study to design and configure any organization network.

Lab Outcome Attained: To design and configure a network for an organization.



Device Requirements & Cost to be invested

Product	Product Links	Qty	Cost	Total
Switches	**1)** https://amzn.eu/d/4S8AApL **2)** https://rb.gy/0qh0ge	**24 x 4** **8 x 1**	**7200/-** **5000/-**	**33,800/-**
Firewall	https://amzn.eu/d/ijuv1W9	**1**	**99,999/-**	**99,999/-**
Router	https://amzn.eu/d/0GiYtEH	**1**	**11,100/-**	**11,100/-**
RJ45 Connector	https://amzn.in/d/5DGhvQv	**400**	**2000/-**	**2000/-**
Ethernet Cable	https://rb.gy/0qh0ge	**250m x 2**	**2100/-**	**4200/-**
Total				**1,51,099/-**

Different types of devices used:

1) **Switch:**

- A network switch connects devices in a network to each other, enabling them to talk by exchanging data packets.
- Switches can be hardware devices that manage physical networks or software-based virtual devices.
- In (LAN) using Ethernet, a network switch determines where to send each incoming message frame by looking at the media access control (MAC) address.
- Switches maintain tables that match each MAC address to the port receiving the MAC address.

2) **Crimping Tool:**

- A crimping tool consists of two hinged handles with jaws or dies at the end.
- It is a handy device that allows you to create secure connections between wires and connectors.
- It works by deforming the connector onto the wire, ensuring a strong bond. With a crimping tool, you can easily join electrical wires, network cables, coaxial cables, and more.

3) **RJ 45 Connector:**

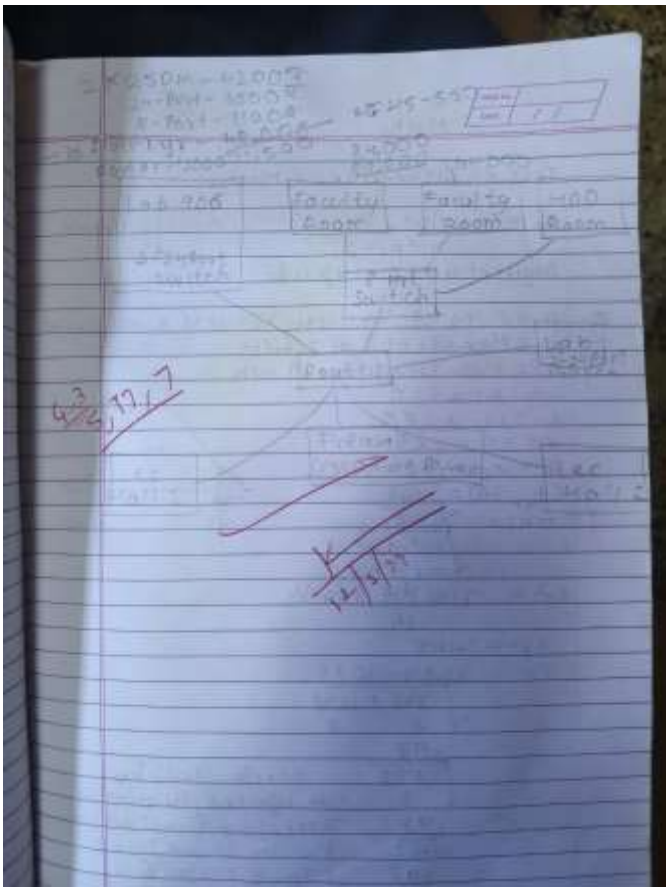
- RJ45 stands for Registered Jack 45 and is the most commonly used connector in wired networks.
- The jacks are mainly used to connect to the Local Area Network (LAN). It was earlier devised for telephones but is now widely used in Ethernet Networking.

4) **Firewall**:

- A Firewall is a network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies.
- It is essentially the barrier that sits between a private internal network and the public Internet.
- A firewall's main purpose is to allow non-threatening traffic in and to keep dangerous traffic out.

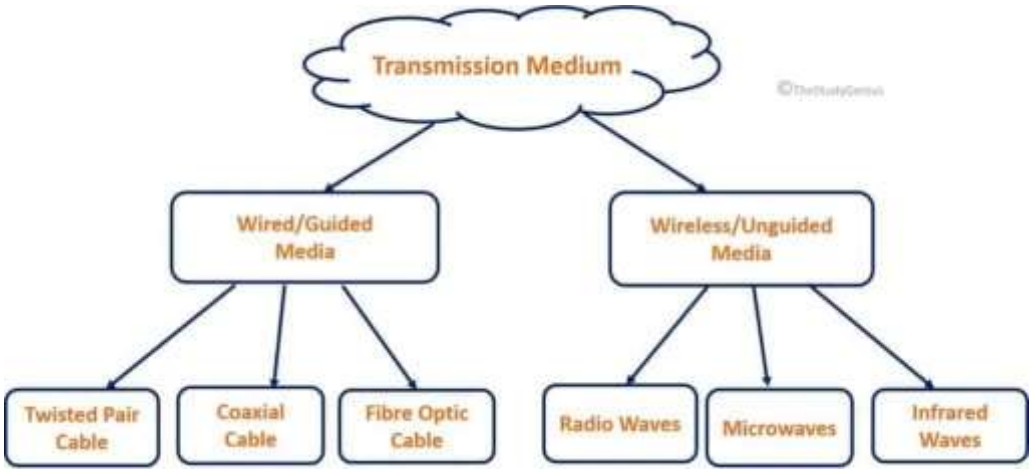
5) Router:

- A router is a hardware device that allows you to connect several computers and other devices to a single Internet connection, which is known as a home network.
- Routers are crucial devices in computer networks that serve as traffic coordinators, directing data packets to their intended destinations.
- They operate at the network layer of the OSI model and use IP addresses to determine the best path for data to travel.
- Routers play a vital role in connecting different networks, such as local area networks (LANs) or wide area networks (WANs), and ensure efficient data transmission. They are essential for internet connectivity, as they help data packets navigate through various networks to reach their final destination.



CN WRITTEN ASSIGNMENT 1

1. Explain guided and unguided transmission media with diagrams.



Wired/Guided Media

Wired Media or Guided Media is a physical medium of signal and data Transmission . It isalso referred to as Bounded Media . It is a narrow pathway to transmit signals using physical layer.
Types of Guided Media are:

- Twisted Pair Cable
- Coaxial Cable
- Fiber Optic Cable

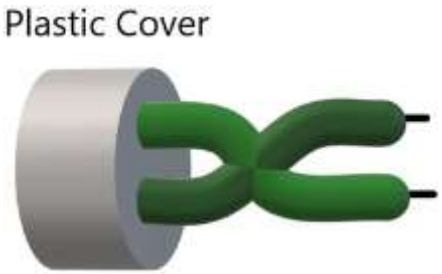
Twisted Pair Cable

These are a type of guided media. It was invented by Alexander Graham Bell.The two conductors of twisted pair cables are typically composed of copper and are insulated on both ends. Twisted pair cables derive their name from the way these two conductors are twisted together.
There are two types Of twisted paircable:

- Shielded Twisted Pair Cable
- Unshielded Twisted Pair Cable

Unshielded Twisted Pair Cable

Unshielded twisted pair cables are made up of two insulated copper wires twisted together without any additional shielding or insulation. Because of the insulation, they lessen influence from the outside world. Pairs of unshielded twisted pair cables are used so that an additional connection can be added as needed. Our homes feature one spare pairof telephone or DSL connections. The AT&T Corporation's 25-pair colour code specifies that when UTP are grouped in pairs, each pair is coded with a distinct



colour. Based on certain specifications, UTP is divided into 7 categories by the Electronic Industries Association. Cable quality is divided into seven categories, with 1 being the best quality and 7 representing the lowest.

Benefits

⇢ Most affordable

⇢ Simple to set up

⇢ Rapid speed capabilityDrawbacks:

⇢ Subject to outside intervention

⇢ Lower performance and capacity compared to STP

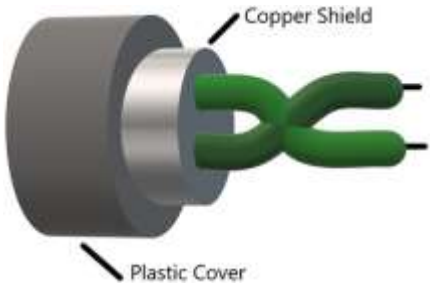
⇢ Transmission attenuation causes short-range transmission

Applications:

The shielded twisted pair type of cable is most frequently used in extremely cold climates, where the additional layer of outer covering makes it perfect for withstanding such temperatures or for shielding the interior components.

Sheilded Twisted Pair Cable

Copper braid coating serves as an additional layer of insulation or protection for the conductors in these kinds of cables. The cable's overall structure is strengthened by this covering. Additionally, it lessens signal interference and noise



in the connection. By circling the induced signal only around the shield, the shielding assures thatit cannot interfere with the primary propagating signal and may be returned to the source via ground. Given that separate colour pairs are needed for analogue and digital transmission, theSTP cables are likewise color-coded, just like theUTP cables. These wires are expensive and challenging to install.

Benefits

⇢ Superior performance when compared to UTP at a greater data rate

⇢ Removes intertext

⇢ Compared to UDP, faster

Drawbacks:

⇢ Relatively challenging to produce and install

⇢ More costly

⇢ Big

Applications:

Used in telephone connections and LAN networks

Coaxial Cable

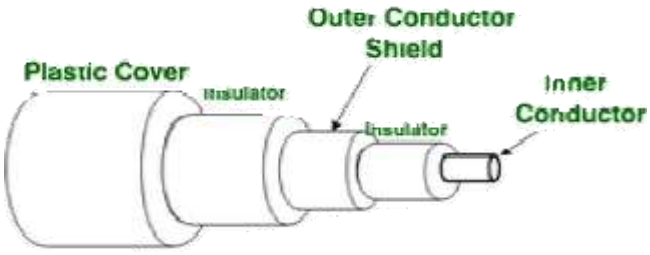


Figure of Coaxial Cable

Coaxial Cable is also called as Coax .It has two parallel conductors with insulated protective covers on each, and an outer plastic covering with an insulating layer composed of Teflon or PVC. Two kinds of information transmission are available with coaxial cables: broadband (where the cable bandwidth is divided into distinct ranges) and baseband (which uses dedicated cable capacity). Coaxial cables are extensively used by analogue television networks and cable TVs.

Benefits

⇢ High bandwidth is supported by coaxial wires.

⇢ Installing coaxial wires is simple.

⇢ Coaxial cables are more dependable and long-lasting because of their superior cut-through resistance.

⇢ Less susceptible to electromagnetic interference, noise, and crosstalk.

⇢ Multiple channels are supported with coaxial connections.Negative aspects

⇢ The cost of coaxial cables is high.

⇢ Crosstalk can only be avoided by grounding the coaxial connection.

⇢ A coaxial cable is quite large because of its many layers.

⇢ Hackers might potentially damage the coaxial wire and attach a "t-joint," jeopardising the data's security.

Coaxial cable Applications:

Coaxial cables are utilised in both MANs and Ethernet LANs.

⇢ Television: RG-6 coaxial wire with a 75 Ohm resistance is used for televisions.

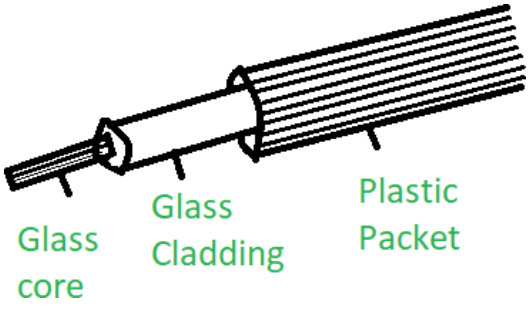
⇢ Internet: RG-6 cables are utilised to transmit internet communications overcoaxial cables.

⇢ CCTV: RG-59 AND RG-6 coaxial cables as well as coaxial cables are utilised in CCTV systems.

⇢ Video: Coaxial cables are also used for video transmission; RG-59 is used for lossless video transmission, and RG-6 is used for superior digital transmissions.

⇢ HDTV: Since RG-11 offers additional area for signal transmission, HDTVs use it.

Fiber Optics Cable



The technology that sends data as light pulses through glass or plastic fibre is known as fibre optics, or optical fibre. Glass fibre counts in fibre optic cables canrange from a few to several hundred. Theglass fibre core is encircled by a second

glass layer known as cladding. The cladding is shielded by the buffer tube layer, and the last line of defence for each individual strand is the jacket layer. Becausefibre optic cables have advantages over copper lines, they are frequently employed. A few of those advantages are faster transmit and bandwidth rates.
High-performance, long-distance data networking is facilitated by fibre optics. Additionally, it is frequently utilised in communications services including phone, television, and the internet.
Benefits

⇢ They are capable of supporting larger bandwidth capacities.
⇢ Light requires less of a signal booster to travel farther.
⇢ They are less vulnerable to electromagnetic interference and other forms of disturbance.
⇢ They are water-submersible.
⇢ Copper wire cables are bulkier, thicker, and heavier than fibre optic cables.
⇢ They don't require as much upkeep or replacement.

Negative aspects

⇢ Copper cable is frequently less expensive than fibre optics.
⇢ Compared to copper, glass fibre needs additional protection inside an outer wire.
⇢ New cabling installation requires a lot of work.
⇢ Fibre optic wires are more brittleFiber

Optic cable Applications:

⇢ Computer networking and broadcasting
⇢Internet and cable television
⇢Undersea environments
⇢Military and space
⇢Medical

Wireless/Unguided Media

Radio Waves

Electromagnetic impulses known as radio waves are employed in a variety of wireless communication technologies, including radio broadcasting, Bluetooth, and Wi-Fi. Radio waves are electromagnetic waves that typically have frequenciesbetween 3 kHz and 1 GHz. All directions are open to radio waves. Radio waves travel in all directions when they are sent by an antenna. This implies that alignment of the sending and receiving antennas is not required. Any receiving antenna can pick up waves sent by a sending antenna. There is a drawback to theomnidirectional characteristic as well. Antennas that emit signals on the same frequency or band as one another have the potential to interfere with the radio waves transmitted by the first antenna.

Micro Waves

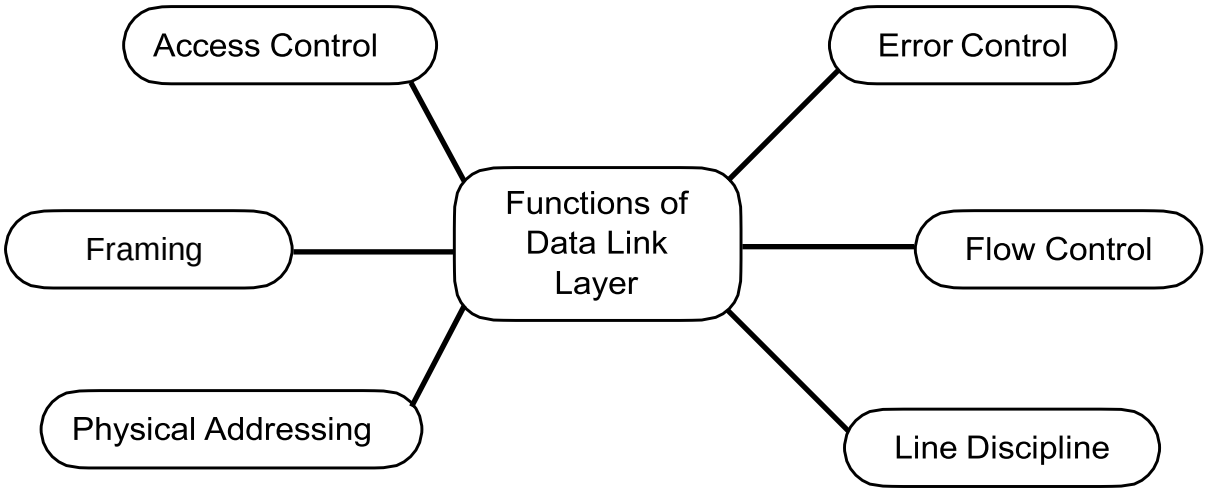
Since it is a line-of-sight transmission, the sending and receiving antennas must be correctly oriented in relation to one another. The height of the antenna directly relates to the signal's coverage distance. 1 GHz – 300 GHz is the frequency range. These are mostly utilised for television distribution and mobile phone communication.

Infrared Waves

For short-range communication, infrared waves with frequency between 300 GHzand 400 THz (wavelengths between 1 mm and 770 nm) can be employed. Due of their high frequencies, infrared rays are unable to pass through walls. Because of this beneficial feature, systems cannot interact with one another; for example, a short-range communication system in one room cannot be impacted by a systemin the adjacent room. We don't interfere with our neighbours' use of the remote when we use our infrared remote control. No application for infrared signals in long-distance communication. Furthermore, because the sun's rays contain infrared wavelengths that could disrupt communication, we are unable to employinfrared waves outside of buildings. It's found in wireless mouse, keyboard, printers, TV remotes, and more.

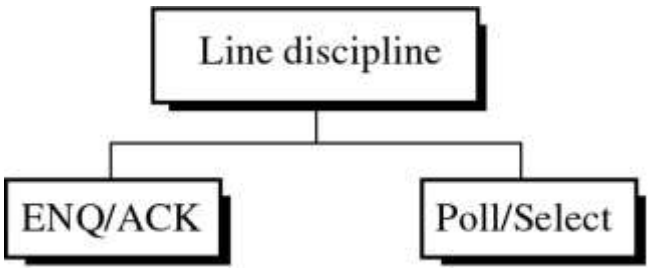
2. Explain various functions of data link layer.

In the network architectural model of OSI (Open System Interconnection), the data link layer is the second layer from the bottom. It is in charge of data delivery from node to node. Its primary responsibility is to provide error-free information transfer. Additionally, encoding, decoding, and organising incoming and outgoing data are under the purview of DLL. Because it keeps all of the hardware's underlying complexity hidden from the other levels above, this layer of the OSI model isregarded as the most sophisticated.



Line Discipline

Data link layer functions include line discipline. It only establishes and recognises the communication's direction. It is essentially the process of establishing and maintaining continuitybetween sender and recipient stations prior to data transfer, allowing data to be delivered in both ways on a data communication network, but not simultaneously. Additionally, it verifies if the recipient is prepared to accept or signal the sender to begin. Line discipline refers to the order in which different network operations are often carried out among devices. These operations typically involve thetransfer and reception of data, the management of transmission errors, the handling of message sets in sequence, and the assurance of data receipt confirmation.



There are Two ways of For carrying out Line Discipline as follows:

ENQ/ACK

ENQ/ACK stands for Enquiry/Acknowledgment. It is a line discipline technique that is typically used to find out whether device on a network is capable of initiating or starting data

or message transmission, as well as if the receiver is prepared and able to receive the data or not. Either the hosts or the stations may start the transmission procedure if their ranks are comparable and equal.

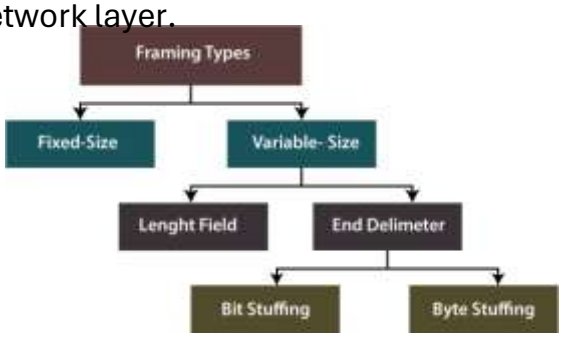
Typically, the starting device creates a session for both full-duplex and half-duplex transmission. Once a full-duplex session is established, both devices can send or transmit concurrently. However,in full-duplex, the responder just waits while the initiator sends data. When the initiator is done, theresponder may assume control of the link again or may have only asked for the response.

Poll/Select

In line discipline, the Poll/Select method essentially functions with topologies in which a primary station is one device and other devices are considered secondary stations. Rather than just two, multipoint connections can be observed; therefore, multipoint systems need to coordinate multiple nodes. Select mode is used when a primary station wants to send data downstream, or to a secondary station. Poll mode is used to request (ask for) transmission from secondary to primary, orupstream. The secondary device merely executes commands from the primary device, which essentially manages and controls the link or connection.

Framing

In the data link layer, the packet that is received from the network layer is referred to as a frame. DLL splits up packets from the network layer at the sender's end into smaller frames,which it then sends bit by bit to the physical layer. Additionally, it appends a few unique bitsto the header and end of the frame (for error control and addressing). DLL gathers bits fromthe Physical layer at the receiver's end, assembles them into a frame, and transmits them to the Network layer.



Fixed Size Framing

Because the frame is predetermined in size, itslength serves as a delimiter, negating the needfor additional borders.

Cons: If the data size is smaller than the framesize, internal fragmentation occurs.

Variable Size Framing

Variable size: To differentiate in this case, it's necessary to specify when one frame ends andthe next one starts. There are two ways to go about this:

Length field:

To show the frame's length, we can add a length field to the frame. utilised in 802.3

Ethernet. This has the drawback that the length field may occasionally become tainted.End Delimiter (ED):

To signal the conclusion of the frame, we can add an ED(pattern). utilised in the Ring of Tokens. The issue with this is that the data may contain ED. One way to resolve this is to:Bit Stuffing:

With this method, the flag byte, 01111110, is a unique bit pattern that starts and ends eachframe. It is possible that the data itself contains the specific pattern 01111110 in this technique as well. To address this issue, the sender data link layer automatically inserts a 0bit into the outgoing stream whenever it finds five consecutive ones in the data stream.

Before forwarding the data to the network layer, the receiver automatically destuffs the 0 bitwhen it observes five consecutive incoming ones followed by one.

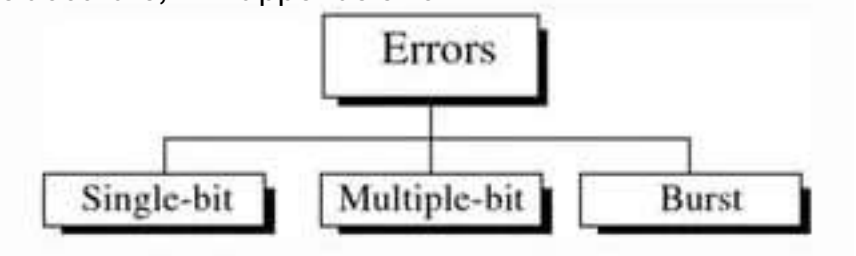
Byte Stuffing:

Framing delimiters, such as DLE STX (start of the frame) and DLE ETX (end of the frame),are represented by ASCII characters. However, there is a chance that the characters DLE STX and DLE ETX will appear in the data if binary data is being transmitted. Character stuffing is a technique used because this can disrupt the framing. An ASCII DLE character isinserted by the sender's data link layer right before the DLE character in the data. This DLE is eliminated by the receiver's data link layer prior to the data being sent to the network layer. Character stuffing, on the other hand, is strongly related to 8-bit characters, which presents a significant obstacle to sending characters of any size. When these character patterns show up in the data, there is an issue.

Other Methods are Character Count and Physical layer coding violations.

Error Control

Noise, attenuation, and other factors are among the many causes of data corruption. Therefore, the data link layer is in charge of identifying errors in transmitted data andmaking the necessary corrections using error detection and correction techniques, respectively. To enable the recipient to verify whether the data they have received is accurate, DLL appends error detection bits to the frame's header. Itincreases physical layer reliability byincluding



mechanisms to identify and retransmit lost or damaged frames.

Single Bit Errors

A single bit, or one binary digit, in a sent data unit can be changed during transmission to produce an incorrect or corrupted data unit. This kind of data transmission error is known asa single-bit error.

Multiple Bit Errors

An error type known as a multiple-bit error occurs when more than one bit during data transfer is impacted. Despite being less common than single-bit errors, multiple-bit errorscan nevertheless happen, especially in digital environments with high levels of noise or interference.

Burst Errors

A burst error occurs when multiple consecutive bits are inadvertently switched during digital transmission. A series of consecutively wrong values are produced by this error.Onetypical strategy for mistake detection is to add redundant bits that carry extra information. Various methods for identifying errors consist of:

Simple Parity Check

Two-dimensional Parity

CheckChecksum

Cyclic Redundancy Check (CRC)

Flow Control

In order to manage the rate of data transmission between two nodes in a network and avoiddata loss from buffer overflow or congestion, flow management in the data link layer is essential.Here's a succinct five-point explanation:

Goal: In order to prevent the sender from sending more data than the receiver can handle orstore, flow control techniques manage the data flow between the sender and the recipient.

Protocol for Sliding Windows: The sliding window protocol is one popular technique for flow control. Both the sender and the recipient of this protocol uphold an acceptable sequence number window. Only packets inside the window's size may be sent by the sender,and if packets are acknowledged by the recipient, the sender can dynamically resize the window.

Selective Repeat vs. Go-Back-N: Selective Repeat and Go-Back-N are the two primary sliding window methodology variants. With selective repeat, the sender can retransmit only the packets that were lost because the recipient acknowledges each correctly received packetseparately. When a timeout happens, the sender uses Go-Back-N to resend all unacknowledged packets.

Signals for Flow Control: Received packet status is communicated using flow control signals like ACK (acknowledgment) and NAK (negative acknowledgment). While NAK denotes the necessity for retransmission, ACK signifies successful reception.

Buffer Management: Both the sender and the recipient must practise appropriate buffer management for effective flow control. Incoming data is momentarily stored in buffers untilit can be processed, allowing data to flow smoothly even in the event of brief network congestion.

Physical Addressing

One of the core components of networking is physical addressing in the data link layer, which addresses hardware-level devices in a local area network (LAN). This is a succinctsynopsis:

Definition: Network interface controllers (NICs) and network adapters are given unique IDsas part of physical addressing, commonly referred to as MAC (Media Access Control) addressing. These IDs, which are used to distinguish devices on the same network segment,are usually hardcoded into the hardware of the device during production.

The format of a MAC address is XX:XX:XX:XX:XX:XX:XX, where each pair of hexadecimal digits represents one byte of the address. MAC addresses are 48-bit (6-byte)hexadecimal numbers.

Uniqueness: To guarantee effective communication, each device connected to a network needs a distinct MAC address. Due to the large number of available permutations, it is uncommon for two devices to have the same MAC address; nonetheless, conflicts may ariseif MAC addresses are duplicated.

Function in Data Transmission: A device addresses the data frame using the recipient's MAC address when it wishes to communicate with another device on the same network. Prior to being transmitted via the physical media, the data link layer encapsulates the packetwith its source and destination MAC addresses.

Protocol for Address Resolution (ARP): Devices in IP-based networks normally

communicate using IP addresses. IP addresses are mapped to MAC addresses using the ARPmechanism. A device looks for the matching MAC address in its ARP cache before attempting to communicate with another device on the same subnet. An ARP request is sentout to get the MAC address if it cannot be located in the cache.

Access Control

Controlling access to a shared transmission channel, like a local area network (LAN), is themain function of access control in the data link layer. This is done to make sure that many devices can connect effectively without crashing or corrupting data. The Media Access Control (MAC) protocol is the main access control mechanism utilised in the data connection layer. This is a synopsis:

The Media Access Control (MAC) protocol establishes guidelines and practices for the non-interfering access of devices on a shared medium to the transmission medium. Diverse MAC protocols provide various ways for controlling access, such as controlled and contention-based approaches.

Contending Access Methods: Devices that use carrier-sense multiple access (CSMA) listento the media before sending data. They wait for the medium to become idle before transmitting if it is busy.

Collision Detection, or CSMA/CD, is a feature of Ethernet networks that recognises collisions when they happen and starts a backoff process to send data again after an arbitraryamount of time.

Collision Avoidance Protocol/CSMA/CA: CSMA/CA is used in wireless networks such asWi-Fi and incorporates extra measures to prevent collisions, notably the Reserve the Medium and Request to Send (RTS) signals.

Controlled Access Methods:

Token passing: Using a unique token, devices communicate data in turn. Orderly access to the media is ensured by the fact that only the device holding the token can broadcast. This ishow Token Ring networks operate.

Polling: Each device on the network is given permission to send data one at a time by acentral controller. Upon completion of transmission, the controller proceeds to poll thesubsequent unit.

Addressing: At the data link layer, devices are uniquely identified by their MAC addresses.The source and destination of data frames are identified by a device's unique MAC addresson a network.

Frame Collision Handling: When several devices try to transmit at the same time using contention-based access techniques like CSMA/CD, collisions may happen. In order to handle collisions, MAC protocols start backoff procedures, which require devices to waitfor arbitrary amounts of time before trying to retransmit. Through the implementation of these access control techniques, the data link layerminimises the likelihood of collisions and maximises network performance while guaranteeing effective and equitable access to the shared transmission medium.

Explain the following headers

- **IPV4**
- **IPV6**
- **TCP**
- **UDP**

IPV4

Version (4 bits): The IPv4 header begins with a 4-bit field that, for IPv4, is usually set to 4 to indicate the version of the Internet Protocol being used.

Header Length: The length of the IPv4 header in 32-bit words is indicated by this 4-bit field. This field determines the beginning of the packet's data part, since theIPv4 header length can vary due to optional fields.

Type of Service (8 bits): Originally designed to specify the packet's quality of service (QoS), the Type of Service (ToS) field consists of 8 bits. Subfields for defining precedence, latency, throughput, and reliability are included. DifferentiatedServices, or diffServ, are used in the majority of contemporary networks instead.

Total Length (16 bits): The length of the IPv4 packet, including the header anddata, is indicated by this 16-bit field. This field can have a maximum value of 65,535 bytes.

Identification (16 bits): When transmission across networks with varying maximum transmission unit (MTU) sizes causes fragmentation of an original IP datagram, the Identification field is utilised to uniquely identify the fragments.

Fragment Offset (13 bits) and Flags (3 bits): The purpose of these fields is to control fragmentation. The Flags field has three bits: More Fragments (MF), Reserved, and Don't Fragment (DF). The location of the fragment within the originalIP datagram is indicated by the Fragment Offset field.

Time to Live (TTL) (8 bits): This field indicates how many hops (routers) a packet can go through before being dropped. When forwarding a packet, each router reduces its TTL value by a minimum of one; if the TTL value approaches zero, thepacket is deleted.

Protocol (8 bits): This field, which can be TCP (6), UDP (17), ICMP (1), or another protocol, indicates which protocol was used in the packet's data section. It enables the packet to be processed by the IP layer and then determine which protocol the receiving computer should use to handle it.

Header Checksum (16 bits): The IPv4 header's Header Checksum field is used to check for errors. By generating a checksum from the contents of the header and

appending it to the header, it guarantees the integrity of the header duringtransmission. The 32-bit source and destination IP addresses are as follows: The packet's source and destination IP addresses are specified in these fields, respectively. They are necessary for the packet to be routed throughout the network to reach itsdestination. In conclusion, the IPv4 header has a number of fields, including addressing, fragmentation, error checking, and protocol identification, that offer crucial information for packet delivery and routing over IP networks.

IPV6

The most recent version of the Internet Protocol, known as IPv6, was created to replace IPv4 due to the exhaustion of accessible IPv4 addresses and to get around some of its drawbacks.

Address Format: Unlike IPv4 addresses, which are 32 bits long, IPv6 addresses are 128 bits long. Usually, they are shown as eight groups of four hexadecimal numbers, like 2001:0db8:85a3:0000:0000:8a2e:0370:7334, with colons between each group.

Increased Address Space: With roughly 3.4 × 10^38 possible addresses, IPv6 offers a vast address space. The IPv4 address exhaustion issue is addressed bythis expansion.

Address Types: Unicast, multicast, and anycast addresses are among the various address types that IPv6 offers. Anycast addresses identify the closest interface among a collection of interfaces, multicast addresses identify many interfaces, andunicast addresses identify a single interface.

Simplified Header: By being less complex than the IPv4 header, the IPv6 header lowers overhead and increases routing efficiency. The addition of a flow label for quality of service, the relocation of choices to extension headers, and the elimination of header checksum are noteworthy modifications.

Autoconfiguration: Devices can automatically create IPv6 addresses with IPv6 thanks to stateless address autoconfiguration (SLAAC), which is supported by IPv6and eliminates the requirement for DHCP servers. Devices configure their IPv6 addresses using information from nearby routers.

Extension Headers: In addition to the basic IPv6 header, IPv6 adds extension headers that offer more capabilities. The Hop-by-Hop Options header, Routing header, Fragmentation header, Authentication header, and Encapsulating SecurityPayload header are a few examples of extension headers.

Neighbour Discovery Protocol: Neighbour Discovery messages from Internet Control Message Protocol version 6 (ICMPv6) and Address Resolution Protocol(ARP) are replaced by Neighbour Discovery Protocol (NDP) in IPv6. NDP makesneighbour reachability detection, router identification, and address resolution easier.
IPv6 Transition techniques: During the migration process, IPv4 and IPv6 networkscan coexist thanks to these techniques. These techniques include translation (likeNAT64), tunnelling (like IPv6-over-IPv4), and dual-stack operation.
Improvements in Security: IPv6 comes with built-in security features like IPsec (Internet Protocol Security), which offers IPv6 packets secrecy, integrity, and authentication. IPv6 hosts and networks can communicate securely by using IPsec.
Mobility and Quality of Service (QoS) Support: Mobile IPv6 is one of IPv6's mobility-supporting capabilities, which allows devices to stay connected even when they switch across networks. Furthermore, IPv6 facilitates quality of service(QoS) by utilising flow labels, which enable traffic to be prioritised according to itsneeds.
In addition to offering scalability, flexibility, and enhanced functionality for upcoming Internet communication needs, IPv6 is intended to rectify the limitationsof IPv4. The Internet cannot continue to expand and thrive without its acceptance.

TCP

TCP is a connection-oriented protocol, which implies that before any data is exchanged, a dependable connection is established between the sender and the recipient. Data delivery is guaranteed to be error-free and in the right order using this connection.
Reliability: Sequence numbers and acknowledgments (ACKs) are two ways that TCP provides dependable data transfer. A segment is sent, and the sender waits for the recipient to acknowledge it. Sender retransmits data if acknowledgement isnot received within the allotted period. At the receiver, out-of-order segments are rearranged using sequence numbers.
TCP uses flow control to make sure a quick sender doesn't overwhelm a sluggish recipient. Through the employment of a sliding window mechanism, the receiver notifies the sender of the amount of buffer space that is available. In order to prevent congestion and packet loss, the sender modifies its transmission rate in accordance with the receiver's declared window size.

Congestion regulate: To regulate network congestion and avoid network collapse,TCP uses congestion control methods. Slow start, congestion avoidance, quick retransmit, and fast recovery are some of the strategies it employs to dynamically adjust the transmission rate in response to network conditions.
Full-Duplex Communication: TCP enables full-duplex communication, which enables simultaneous data transmission in both directions. This indicates that datacan move freely in both directions and that every TCP connection consists of two endpoints.
Segmentation and Reassembly: Prior to transmission, TCP divides data into smaller components known as segments. To make reassembly easier at the recipient, each section has a sequence number. TCP can efficiently handle data ofdifferent sizes and adjust to network conditions thanks to segmentation.
Error Detection and Correction: To identify problems in the TCP header and payload, TCP employs a 16-bit checksum. The receiver can use selective repeat orgo-back-N techniques to request retransmission if an error is detected, after whichthe segment is discarded.
Multiplexing and Demultiplexing: TCP multiplexes and demultiplexes data at the transport layer using port numbers. Source and destination port numbers are included in every TCP segment, enabling simultaneous communication between several programmes over a single IP address.
Stateful Protocol: Throughout a connection's lifetime, TCP preserves its state. To handle connection formation, data transfer, and termination, it cycles through a number of states, including SYN_SENT, SYN_RECEIVED, ESTABLISHED, FIN_WAIT_1, FIN_WAIT_2, CLOSE_WAIT, LAST_ACK, and TIME_WAIT.
Application Support: TCP offers programmes a dependable, stream-oriented service, which makes it appropriate for applications like email, file transfer (FTP),web surfing, and remote shell (SSH) that need guaranteed delivery and ordereddata transmission.
In conclusion, TCP is one of the most extensively used protocols in the Internet protocol suite because it is a strong, connection-oriented protocol that guaranteesthe orderly, error-checked, and dependable transfer of data over IP networks.

UDP

In computer networking, UDP, or User Datagram Protocol, is a connectionless transport layer protocol. It offers a lightweight and easy-to-use method for sending datagrams between connected devices. This is a thorough description of UDP that

covers its benefits, characteristics, and use cases:

UDP is a connectionless protocol, which means that data is transmitted without first establishing a dedicated connection. A datagram, or UDP packet, is handled separately and can be sent without previous agreement.

Header Structure: When compared to other transport layer protocols such as TCP,the UDP header is rather straightforward. There are four fields in it:

Source Port: A 16-bit field that contains the sender's port number.

Destination Port: A 16-bit field that contains the receiver's port number.

Length: A 16-bit element that indicates the data and UDP header lengths.

Checksum: A 16-bit field that is used to identify errors in the data and UDP header.

Minimal Overhead: Because UDP does not have error recovery, flow management,or connection setup, it has less overhead than TCP. Because of this, UDP is appropriate for situations where speed and low latency are more important than reliability.

Unreliable Delivery: UDP lacks means for error recovery or packet resend, in contrast to TCP, which ensures the reliable and orderly delivery of data. The application layer is responsible for handling out-of-order delivery and data loss. Broadcast and Multicast Support: UDP enables the simultaneous transmission of asingle datagram to several recipients by supporting both broadcast and multicast communication. Applications like real-time communication, internet gaming, andstreaming video frequently leverage this feature.

Real-Time Applications: Because UDP can withstand the occasional packet loss and has minimal latency requirements, it is a good choice for real-time applications. Online gaming, live streaming, video conferencing, and voice over IP(VoIP) are a few examples.

Easy Implementation: Due to its simplicity, UDP is both efficient in terms of processing overhead and simple to implement. For applications where TCP's extracapabilities and complexity are not required, it is frequently chosen.

Use Cases: UDP is frequently utilised in situations where efficiency and speed takeprecedence above dependability, like:

DNS queries and answers for the Domain Name System

For network management, use the Simple Network Management Protocol (SNMP).File transfers using the Trivial File Transfer Protocol (TFTP)

Multimedia apps on a network, such as audio and video streaming

UDP is a stateless protocol, which means that it doesn't keep track of the status ofthe datagrams or retain any connection state. While network infrastructure is madesimpler and overhead is decreased, applications must handle any required state

management.

Datagram Format: A header and data are the first two parts of every UDP datagram. The maximum size of an IP packet less the size of the IP and UDP headers equals 65,535 bytes, which is the maximum length of the header and datacombined.

In conclusion, UDP is a lightweight and effective transport layer protocol that works well for programmes that value simplicity and speed over dependability. It isa good fit for multicast, broadcasting, and real-time communication because of itslow overhead and connectionless architecture. Nevertheless, as UDP lacks these capabilities, applications built on top of it must manage error recovery and data integrity at the application layer.

CN WRITTEN ASSIGNMENT 2

1. Explain VLAN and VPN in detail along with anexample.

VLAN

No matter where they are physically located, devices on a larger physical network can communicate with each other as if they were on the same physical network segment thanks to a logical grouping called a VLAN, or virtual local area network.In order to improve security, optimise network traffic, and streamline network administration, VLANs are a basic idea in network management. Here's a thoroughexplanation of VLANs with an illustration:

1. Purpose of VLANs:

Logical Segmentation: Network managers can split a big network into moremanageable, separated broadcast areas by using VLANs to enable logical segmentation of a physical network.

Enhanced Security: Organisations can improve network security by limiting whichdevices can connect with one another by isolating traffic within VLANs.

Optimised Traffic Flow: VLANs can reduce needless broadcast traffic and enhancenetwork performance by combining devices with comparable traffic patterns or requirements.

Simplified Network Management: Virtual local area networks, or VLANs, enablenetwork administrators to arrange devices logically, not according to physicalcloseness, but according to their purpose, department, or location.

2. VLAN Implementation:

Network switches are set up to allocate particular ports to various VLANs when VLANs are deployed, which is usually done at the switch level.

A VLAN ID, also called a VLAN tag, is assigned to each VLAN and is placed into the Ethernet frame header to indicate which VLAN the packet belongs to.

In order to properly isolate traffic inside each VLAN, VLAN-aware switches use thisVLAN tag to route traffic exclusively to ports assigned to the same VLAN.

3. VLAN Tagging:

Ethernet frames can be identified as belonging to a VLAN as they travel across the network by using a technique called VLAN tagging.

The industry standard for VLAN tagging is IEEE 802.1Q, which appends a 4-byte VLAN tag with the VLAN ID to the Ethernet packet header.
Proper traffic segregation is ensured by VLAN tagging, which makes VLAN-aware devices—like switches and routers—able to comprehend and handle VLAN information.
As an illustration, let's look at a medium-sized business that has several divisions,such as IT, HR, Sales, and Marketing. The organisation intends to optimise trafficflow and enhance network security by implementing VLANs.
The following is howVLANs can be set up:
Four VLANs are created by the network administrator: VLAN10 for sales, VLAN20for marketing, VLAN30 for human resources, and VLAN40 for information technology.
Assigning Switch Ports to VLANs: Each VLAN is assigned a distinct port by the network switches. As an illustration:
For sales, VLAN 10 is assigned ports 1 through 10.
VLAN20 has been designated as the marketing port
range.For HR, VLAN30 is given ports 21 through 30.
VLAN 40 is designated for IT traffic on ports 31–40.
Traffic Isolation: When VLANs are set up, devices in different departments can onlyspeak to other devices in their own VLAN. Devices in the Sales department, for instance, are limited to communicating with other devices in VLAN10 (Sales).
Security and Optimisation: By separating traffic between departments, VLANsimprove security by limiting unauthorised access to private data.
Because each VLAN functions as a distinct broadcast domain, they also optimize network flow by lowering broadcast traffic and congestion.

VPN

A technology known as a virtual private network, or VPN, makes it possible to communicate securely and privately over open networks like the internet. It encrypts all data traffic that travels over the private network connection, or tunnel, that it builds between a user's device and a distant server. VPNs provide many advantages, such as improved security, privacy, and the capacity to get around regional limitations. The components, functionality, and use of VPN technology areexplained in depth below, along with an example:

VPN components include:

Client: To create a secure connection, the user's device (such as a computer orsmartphone) connects to the VPN server. VPN clients can be physical devices, stand-alone programmes, or integrated operating system components.

VPN Server: The encrypted communication between the client and the target network or resource is managed by the VPN server, which is a distant server. While outgoing data is encrypted and sent to the client, incoming data from the client is decrypted and forwarded to the destination.

The process of encasing and encrypting data for safe internet transfer is defined by the tunnelling protocol. VPNs frequently employ OpenVPN, IPSec, L2TP/IPSec,and IKEv2/IPSec as tunnelling protocols.

Encryption: Data transmitted over the VPN connection is encrypted using algorithms like AES (Advanced Encryption Standard) or RSA (Rivest-Shamir-Adleman). This ensures secrecy and prevents unauthorised access to the transmitted data.

How a VPN operates:

Authentication: A client must go through an authentication process in order to prove its identity when it first connects to the VPN server. Usually, this entails supplying digital certificates, login credentials (password and username), or other authentication techniques.

Tunnel Establishment: Following authentication, the VPN client and server discussthe VPN tunnel's parameters, such as the encryption algorithm, tunnelling protocol, and other settings.

Data Transmission: All data transmission between the client and the server is encrypted and contained inside the VPN tunnel once it has been established. This guarantees that information sent via the internet is safe from unauthorised parties' interception or eavesdropping.

Access to the Destination Network: When a data packet arrives from the client, theVPN server decrypts it before sending it to the appropriate network or resource. Ina similar manner, the VPN server encrypts data coming from the destination network before sending it back to the client over the VPN tunnel.

Remote Access VPN: When employees need to safely access company resources from outside the office, remote access VPN is a popular use case for virtual privatenetworks. Let's say a worker has to access private company files

that are on the corporate network while they are working from a coffee shop. Through the use of aVPN client installed on their laptop, the employee creates a secure virtual network connection to the business network via the internet by connecting to the company's VPN server. Sensitive data is safeguarded from possible attackers and eavesdroppers on public Wi-Fi networks thanks to encryption of all data flow between the employee's device and the corporate network.

In conclusion, a virtual private network, or VPN, shields private information from prying eyes and allows users to safely access remote resources while guarding sensitive information from illegal access or interception. Virtual private networks, or VPNs, provide an adaptable way to communicate securely over the internet, whether for business purposes, individual privacy, or getting around geo-restrictions.

Example

The international corporation ABC Corporation maintains branch offices across the globe. ABC Corporation uses VPNs and VLANs to centralise network management and provide secure connectivity across its offices.

Execution:

Virtual Local Area Networks, or VLANs, are used by ABC Corporation to organise its network infrastructure into distinct VLANs according to departments or functional groups, such marketing, finance, and research and development (R&D).The organisation is able to logically divide its network and regulate traffic flow between departments since each VLAN is given a distinct VLAN ID.

Devices in the Marketing department are assigned to VLAN 20, whilst those in the Finance department are assigned to VLAN 10. Switches with VLAN configurationscan separate traffic within individual VLANs, improving network performance and security.

Virtual Private Network, or VPN: To enable safe online communication, ABC Corporation sets up VPN connections between its regional offices and the corporate office.

VPN routers or firewalls that create encrypted tunnels to the central VPN server atthe headquarters are installed in each branch office.

With VPN client software installed on their devices, employees may securely access corporate resources from any location, even when working

remotely ortravelling.

By offering a secure route of communication for data transfer between branch offices and central headquarters, VPN tunnels shield confidential data from outsideparties' interception or unauthorised access.

Combined Example

A worker in the marketing division of ABC Corporation's New York branch office need access to private financial information kept on servers at the corporate headquarters.

The employee's device is linked to the branch office's local network's Marketing VLAN (VLAN 20).

The employee uses the VPN client software that the corporation has installed on their laptop to establish a VPN connection in order to safely view the financial data.The VPN client creates an encrypted tunnel between the central VPN server at headquarters and the VPN router/firewall at the branch office.

The employee's device can safely send data to the central headquarters over theinternet once the VPN tunnel is built, avoiding any security risks or listening in onthe public network.

In this combined example, branch offices and central headquarters communicatesecurely via VPNs, which guarantee the security and integrity of data transferred over the internet. VLANs are utilised for network segmentation and traffic isolationwithin the branch office.

2. Compare Different Ethernet protocols (standard Ethernet, fast Ethernet, Gigabit Ethernet,10-GigabitEthernet).

Standard Ethernet

Standard Ethernet, sometimes referred to as 10BASE-T, employs coaxial connections and works at a data rate of 10 Mbps. Devices are usually connected to a single Ethernet bus in a bus architecture. Devices can identify and respond tocollisions thanks to the usage of the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access mechanism. Data, destination and source addresses, and a Frame Check Sequence (FCS) for error detection are among thefields that make up an Ethernet frame. In a typical Ethernet network segment, devices can be separated by up to 100 metres. In the early LAN implementations,standard Ethernet was commonly employed to enable simple networking featureslike printer and file sharing.

Fast Ethernet

With a data rate of 100 Mbps, fast Ethernet—also referred to as 100BASE-TX—offers ten times the speed of regular Ethernet. It supports both bus and star topologies and is primarily wired with twisted pair copper. Fast Ethernet maintainsa maximum distance restriction of 100 metres per segment and the Carrier SenseMultiple Access with Collision Detection (CSMA/CD) access mode. Because it provided more performance and bandwidth for LAN situations than regular Ethernet, it was widely adopted as an upgrade in the 1990s.

Gigabit Ethernet

Ten times faster than Fast Ethernet and one hundred times faster than ordinary Ethernet, gigabit Ethernet operates at 1 Gbps. It supports both bus and star topologies and transmits data mostly via twisted pair copper or fibre optic lines. Although Gigabit Ethernet operates at a fast data rate and in full duplex, it uses the CSMA/CD access technique, which makes collisions unlikely. Gigabit Ethernet is widely used in contemporary LANs and data centres to provide high-speed connectivity for demanding applications like big file transfers and video streaming.

10-Gigabit Ethernet

Operating at 10 Gbps, 10-Gigabit Ethernet (10GbE) provides ten times the speedof Gigabit Ethernet. In order to support bus and star topologies, it mainly makes use of twisted pair copper or fibre optic connections. Usually, 10GbE runs in full-duplex mode, which does away with the requirement for collision detection.

Copper cables are limited to a maximum distance of 100 metres per segment; fibre optic cables can accommodate larger distances. 10GbE is a widely used technology in data centres and enterprise networks. It offers fast connectivity that can handle enormous data volumes, allow virtualization, and enable real-time applications. Because of its adaptability and interoperability with current infrastructure, it is essential for modern networking settings that demand high-performance connectivity.

Differences

The main distinctions between Gigabit, 10-Gigabit, Fast, and regular Ethernet areas follows:

Data Rate :

10 Mbps is the standard Ethernet speed.

100 Mbps for fast Ethernet

1 Gbps for Gigabit Ethernet

10 Gbps for 10-Gigabit Ethernet

Media Type:

Ethernet Standard mostly coaxial cable

Copper twisted pair cable (Cat5 or higher) is used for fast Ethernet.

Gigabit Ethernet requires fibre optic cable and twisted pair copper wire (Cat5e or higher).

10-Gigabit Ethernet: Fibre optic cable and twisted pair copper cable (Cat6a orabove).

Topology:

Bus and star topologies are supported by all Ethernet protocols. In switched networks, gigabit and 10-gigabit Ethernet are frequently used.

Access method:

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is used by FastEthernet and Standard Ethernet. Gigabit and 10-Gigabit Ethernet frequently function without collision detection in full-duplex mode.

Frame Format:

The preamble, start frame delimiter, destination and source addresses, length/type, data, and frame check sequence (FCS) are all part of the same frame format that is

shared by all Ethernet protocols.

Distance Limit:

For twisted pair copper cables, the maximum distance allowed per segment in allEthernet protocols is 100 metres. Longer distances are supported by fibre optic cables.

Use Cases:

Common Ethernet: Initial LAN deployments

Quick Ethernet: improvements over conventional Ethernet, offering increased performance

Gigabit Ethernet: Higher bandwidth is needed for modern LANs, data centres, andenterprise networks.

10-Gigabit Ethernet: Enterprise networks, data centres, and clusters for high-performance computing all require even more speed and throughput.

3. Explain persistent and non-persistent HTTP.

Persistent HTTP

A protocol enhancement of the Hypertext Transfer Protocol (HTTP) called persistent HTTP, or HTTP/1.1, allows a client (such a web browser) and a server to reuse TCP connections for several request/response cycles. Compared to non-persistent HTTP, this connection persistence provides a number of benefits, suchas decreased latency, enhanced performance, and more effective resource use.

This is a thorough explanation of HTTP persistence:

Connection Reuse:

In persistent HTTP, the connection stays open even after the client and server establish a TCP connection in order to send and receive requests and responses.The persistent connection eliminates the requirement for the client to create new connections for every HTTP request, enabling the client to send more requests over the same TCP connection.

Through connection reuse, persistent HTTP lowers latency and improves speed forincoming requests by eliminating overhead related to connection formation (such as the TCP handshake) and deconstruction.

Pipelining:

Request pipelining, a method that lets the client send many HTTP requests to theserver without having to wait for individual answers, is supported by persistent HTTP.

With pipelining, the server can process and reply to requests in the order that they were received by the client by sending a sequence of HTTP requests across the persistent connection.

By overlapping the transmission of requests and responses, pipelining reduces thenegative effects of delay while increasing throughput and overall efficiency.

Multiplexing:

Multiplexing, a method that enables the sending and receiving of numerous HTTP requests and responses simultaneously over a single persistent connection, is alsosupported by persistent HTTP.

Multiplexing allows the server to process and reply to requests in parallel whileallowing the client to interleave several requests inside the same

connection.
By increasing concurrency and resource utilisation through multiplexing, networkresources can be used more effectively, and the likelihood of connection bottlenecks is decreased.

Keep-Alive Mechanism:
Persistent HTTP uses the "Keep-Alive" mechanism, in which the client and serverboth signal that they are prepared to keep the connection open for further requests, in order to sustain persistent connections.
HTTP requests and answers with headers like "Connection: keep-alive" indicatethat the connection will be reused for subsequent requests, which is part of the Keep-Alive mechanism.
Parameters like the maximum number of requests or the maximum amount of timethe connection should be open before being disconnected are specified in keep- alive headers.

Performance Benefits:
Because persistent HTTP minimises the overhead associated with connection formation and teardown, it delivers notable speed advantages over non-persistentHTTP.
Reusing connections and auxiliary methods like multiplexing and pipelining allowpersistent HTTP to reduce latency, maximise resource use, and enhance overallweb application responsiveness.

In conclusion, persistent HTTP improves online communication performance andefficiency by allowing clients and servers to reuse TCP connections for repeatedrequest/response cycles. In contemporary network contexts, persistent HTTP lowers latency, maximises resource usage, and improves the responsiveness of web applications through methods including pipelining, multiplexing, and the Keep-Alive mechanism.

Non Persistent HTTP

An older iteration of the Hypertext Transfer Protocol (HTTP) called non-persistent HTTP, or HTTP/1.0, is used to facilitate communication between web browsers (clients) and web servers. For every request/response cycle with non-persistent HTTP, a new TCP connection is made between the client and server. This is a thorough explanation of what non-persistent HTTP is.

Connection Establishment:

A client (such a web browser) opens a TCP connection to the server in order to obtain resources or web pages from it.

The desired resource (such as the URL) and any extra arguments are specified in an HTTP request message that the client sends to the server.

The requested resource (such as an HTML page, image, or document) is containedin an HTTP response message that the server generates after receiving the requestand processing it.

Connection Closure:

The server ends the communication session by cutting off the TCP connection after providing the client with the response.

After receiving the response, the client renders or shows the user the content.

The client disconnects from the server and closes the connection as soon as thematerial is presented.

Need for Persistent Connections:

Every request/response cycle in non-persistent HTTP involves the overhead of creating a new TCP connection, which includes the TCP handshake procedure(SYN, SYN-ACK, ACK).

Increased delay may arise from this burden, particularly for websites requiring multiple HTTP queries for resources like photos and scripts.

Limitations:

While non-persistent HTTP is easy to use and uncomplicated, it can result inhigher response times and wasteful use of network resources, particularly forcomplex content websites.

Performance can suffer when there is a requirement to create a new connection for every request/response cycle, especially in circumstances with high latency or hightraffic.

Usage and Evolution:

Early adopters of the World Wide Web made extensive use of non-persistent HTTP,which is still supported by web servers and clients.

But persistent HTTP (HTTP/1.1 and later versions) has essentially replaced it, allowing connections to be reused for several request/response cycles, whichlowers overhead and boosts performance.

To sum up, non-persistent HTTP causes more delay and overhead than

persistentHTTP since it creates a new TCP connection for every request/response cycle.

Non-persistent HTTP is easy to use and uncomplicated, but it is not as efficient or fast as it may be, especially for contemporary online applications and heavily visited websites.

4. Explain JPEG and GIF Compression Algorithms.

JPEG (Joint Photographic Experts Group)

A popular picture compression standard called JPEG makes it possible to transferand store digital photos quickly and effectively without sacrificing acceptable image quality. Below is a thorough breakdown of JPEG compression algorithms:

1. Visual Representation:

The input image is usually represented as an array of pixels before compression, with each pixel carrying colour information. The image could have a high pixel count and be in a high-resolution format, which would increase the file size.

2. Colour Space Conversion: To separate the luminance (brightness) component (Y) from the chrominance (colour) components, the image is transformed from its original colour space (such as RGB) to the YCbCr colour space.

The colour information is represented by the Cb and Cr components, whereas the grayscale or brightness information is represented by the Y component.

3. Discrete Cosine Transform (DCT):

The DCT is applied separately to each of the tiny blocks (usually 8 by 8 pixels) that make up the image.

By converting spatial information into frequency information, the DCT represents the content of a picture in terms of various frequencies and their corresponding magnitudes.

The contribution of different frequency components inside each block isrepresented by the DCT coefficients.

4. Quantization:

The DCT coefficients are quantized in order to lower the bit count needed toexpress them.

The DCT coefficients are quantized by dividing them by a quantization matrix thathas predetermined values in it.

Image quality is somewhat lost as a result of the quantization process' introductionof accuracy loss. Higher quantization settings result in reduced image quality but higher compression ratios.

5. Condensation:

Using entropy encoding methods like arithmetic or Huffman coding, the quantizedDCT coefficients are further compressed.
The quantized coefficients are given variable-length codes by Huffman coding,with shorter codes going to the coefficients that appear more frequently.

More effective compression is achieved when using arithmetic coding, whichencodes the whole stream of coefficients as a single fractional integer.

6. Encoding and Decoding:

The header data that describes the compression parameters is transferred orstored with the compressed image data.
The decoding procedure reverses the compression steps—entropy decoding, dequantization, inverse DCT, and colour space conversion—to rebuild the image. The amount of detail lost in the reconstructed image as a result of quantization andcompression is governed by the compression settings.
Example: Let's say you have an image with an 8x8 pixel block. The block's DCT coefficients are represented using a combination of quantized values and variable-length Huffman codes following the application of DCT, quantization, and entropy encoding. The compressed picture data is then transferred or stored with these codes included. In order to retrieve the pixel values of the reconstructed block, thecompressed data must first be processed in order to rebuild the original DCT coefficients. These coefficients are then inversely converted. The YCbCr components are then converted back to the RGB colour space using colour spaceconversion, producing the reconstructed image.

GIF (Graphics Interchange Format)

The famous picture file format GIF (Graphics Interchange Format) is well-known forsupporting transparency and animation. GIF pictures are frequently used in memes, web graphics, and basic animations. Several strategies are used in the compression of GIF images in order to minimise file size without sacrificing imagequality. The following describes the GIF compression algorithms:

LZW Compression:

LZW (Lempel-Ziv-Welch) is the primary compression algorithm used in GIF files. Using a lossless compression algorithm called LZW, redundant pixel patterns in a picture are found and replaced with shorter codes to minimise redundancy and filesize.

In order to represent the image data more effectively, the LZW algorithm creates a dictionary of frequently recurring pixel patterns and gives these patterns shorter codes.

To guarantee an accurate reconstruction of the original image, the dictionary is shared between the encoder and the decoder and is constructed dynamically during compression.

Colour Reduction:

GIF pictures use less data to depict the image because they can only have amaximum of 256 colours (8 bits per pixel).

A palette of up to 256 colours is chosen using colour reduction algorithms to best reflect the colours seen in the source image.

Common methods for reducing colour intensity include median cut, uniform quantization, and octree quantization. These methods assign representative colours from the palette and group colours that are comparable together.

Lossy Compression (Optional):

Although GIF is essentially a lossless compression technology, lossy compressionis an option if you want to further reduce file size at the sacrifice of image quality. In order to obtain greater compression ratios, lossy compression techniques like colour quantization and dithering add small flaws into the image.

While colour quantization lowers the amount of unique colours in the image tomatch the 256-color limit, dithering produces the illusion of more colours by blending pixels of different colours in patterns.

Interlacing (Optional):

This optional GIF compression technique increases the perceived speed at whichimages load by showing a low-resolution version of the image initially, then successively higher-resolution versions.

A brief preview of the image is shown to the viewer while the remaining information load in later passes thanks to interlacing, which splits the image into many passes.Compression Control:

GIF compression gives you the ability to adjust a number of settings, including colour palette size, dithering technique, and interlacing, to maximise image quality and minimise file size based on the particular needs of your application.

To summarise, the GIF compression methods, namely LZW compression, colour

reduction, interlacing, optional lossy compression, and compression control, collaborate to minimise file size without compromising image quality or effectively displaying drawings and animations in the GIF format.