Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that ~~Mr.~~/Miss Shreya Malwade

of IT Department, Semester 4 with

Roll No. 55 has completed a course of the necessary

experiments in the subject NL under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2024 - 2025

Teacher In-Charge 16/4/2025

Head of the Department

Date April 16, 2025

Principal

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

CONTENTS

SE-IT-S1-S13-55

ASSIGNMENT NO. 1

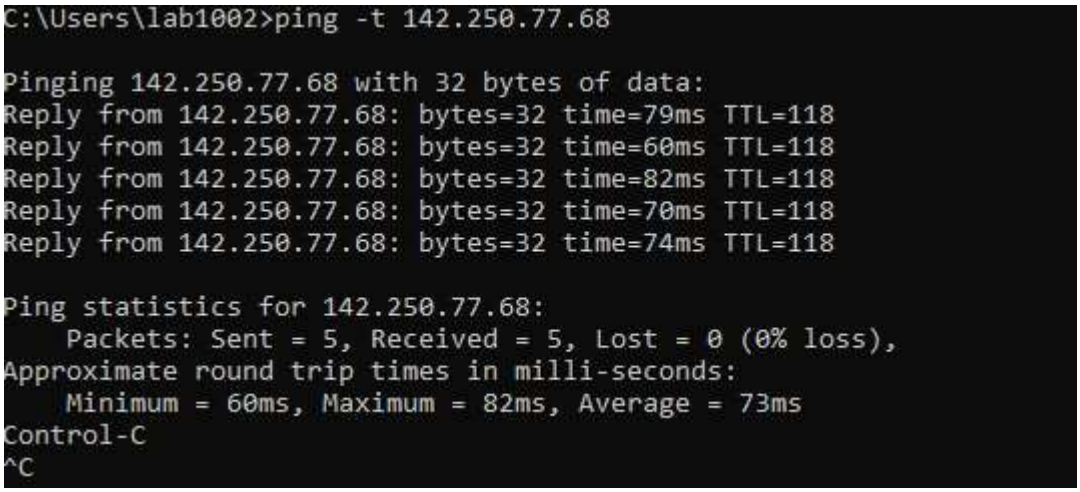
1. PING:

Ping (Packet Internet Groper) is a method for determining communication latency between two networks or ping is a method of determining the time it takes for data to travel between two devices or across a network. As communication latency decreases, communication effectiveness improves. A low ping time is critical in situations where the timely delivery of data is more important than the quantity and quality of the desired information.

options:

1. ping - t :

It continuously pings a specific IP address or Web site. This process continues until it stops. Press Ctrl + C to prevent/end the continuous ping command.



2. ping - a:

This command displays the computer name of the computer with the IP address you are pinging.

```
C:\Users\lab1002>ping -a 142.250.77.68

Pinging bom07s27-in-f4.1e100.net [142.250.77.68] with 32 bytes of data:
Reply from 142.250.77.68: bytes=32 time=14ms TTL=118
Reply from 142.250.77.68: bytes=32 time=33ms TTL=118
Reply from 142.250.77.68: bytes=32 time=53ms TTL=118
Reply from 142.250.77.68: bytes=32 time=26ms TTL=118

Ping statistics for 142.250.77.68:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 53ms, Average = 31ms
```

3. ping -n:

This command assigns a specified number of Ping to an IP address and Web site. Its value is 4 by default. You can ping the number of times you specify by adding -n to the command.

```
C:\Users\lab1002>ping 3 -n 142.250.77.68

Pinging 0.0.0.3 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 0.0.0.3:
    Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),
Control-C
^C
```

4.ping -l:

When you usually ping to a destination, the packet size sent is 32 Bytes. You can increase this value with the ping -l command. The ping packet size limit is 65,500 bytes.

```
C:\Users\lab1002>ping -a 142.250.77.68

Pinging bom07s27-in-f4.1e100.net [142.250.77.68] with 32 bytes of data:
Reply from 142.250.77.68: bytes=32 time=14ms TTL=118
Reply from 142.250.77.68: bytes=32 time=33ms TTL=118
Reply from 142.250.77.68: bytes=32 time=53ms TTL=118
Reply from 142.250.77.68: bytes=32 time=26ms TTL=118

Ping statistics for 142.250.77.68:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 53ms, Average = 31ms
```

5. ping -l flood Command

This command sends the packet size you specify without destroying the destination.

```
C:\Users\lab1002>ping -l 1245 -f 142.250.77.68

Pinging 142.250.77.68 with 1245 bytes of data:
Reply from 142.250.77.68: bytes=1245 time=83ms TTL=118
Reply from 142.250.77.68: bytes=1245 time=117ms TTL=118
Reply from 142.250.77.68: bytes=1245 time=96ms TTL=118
Reply from 142.250.77.68: bytes=1245 time=45ms TTL=118

Ping statistics for 142.250.77.68:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 45ms, Maximum = 117ms, Average = 85ms
```

6. ping -c :

Specify the number of packets to be sent.

```
C:\Users\lab1002>ping -c 142.250.77.68
Access denied. Option -c requires administrative privileges.
```

command

2. Traceroute:

The traceroute command is a network diagnostic tool used to track the path that data packets take from your computer to a specified destination, such as a website or an IP address. It helps identify network problems by showing each hop (router) the packets pass through and the time it takes to reach each hop.

```
C:\Users\lab1002>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.
```

option:

1. tracert -d :

tracert will list the IP addresses of each hop. When it doesn't have to resolve hostnames, the command also becomes much faster.

```
Tracing route to 142.250.77.68 over a maximum of 30 hops

  1     3 ms     3 ms     6 ms  192.168.0.1
  2     7 ms     9 ms     6 ms  203.212.25.1
  3     7 ms     5 ms     7 ms  203.212.24.53
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     2 ms     2 ms     3 ms  175.100.188.22
  7     2 ms     3 ms     2 ms  216.239.47.175
  8     3 ms     3 ms     3 ms  142.250.238.197
  9     2 ms     4 ms     3 ms  142.250.77.68

Trace complete.
```

2. tracert -h maximum_hops:

set the maximum number of hops in the path to search for target. Default is 30 hops.

```
C:\Users\lab1002>tracert -h 6 142.250.77.68

Tracing route to bom07s27-in-f4.1e100.net [142.250.77.68]
over a maximum of 6 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.0.1
  2    14 ms    14 ms    13 ms  1.0/25.212.203.fxwirelesssol.com [203.212.25.1]
  3     2 ms     1 ms     1 ms  1.0/24.212.203.fxwirelesssol.com [203.212.24.53]
  4     3 ms     *        *     53.177.100.175.mipl.com [175.100.177.53]
  5     7 ms     7 ms     *     172.16.2.202
  6     2 ms     2 ms     2 ms  22.188.100.175.mipl.com [175.100.188.22]

Trace complete.
```

3. tracert -w <timeout>

Specifies the timeout in milliseconds to wait for each reply. Default is 4000 ms.

```
C:\Users\lab1002>tracert -w 2000 142.250.182.228

Tracing route to bom07s29-in-f4.1e100.net [142.250.182.228]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.0.1
  2    68 ms    88 ms    82 ms  1.0/25.212.203.fxwirelesssol.com [203.212.
  3    61 ms    51 ms    50 ms  1.0/24.212.203.fxwirelesssol.com [203.212.
  4     *       81 ms     *     10.10.226.153
  5    43 ms     *       85 ms  72.14.196.213
  6    38 ms    36 ms    30 ms  192.178.84.175
  7    53 ms    58 ms    64 ms  142.250.214.105
  8    71 ms    86 ms    38 ms  bom07s29-in-f4.1e100.net [142.250.182.228]

Trace complete.
```

4. tracert -R <route_option>

This option routes each hop in reverse, from destination back to the source.

```
C:\Users\lab1002>tracert -R 142.250.182.228
Unable to resolve target system name 142.250.182.228.
```

5. tracert -4 <destination>

Forces Tracert to use IPv4 for the trace.

```
C:\Users\lab1002>tracert -4 www.google.com

Tracing route to www.google.com [142.250.182.228]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.0.1
  2    74 ms    65 ms    86 ms  1.0/25.212.203.fxwirelesssol.com [203.212.25.1]
  3    58 ms    74 ms    79 ms  1.0/24.212.203.fxwirelesssol.com [203.212.24.53]
  4     *        *        *     Request timed out.
  5    84 ms    55 ms    33 ms  72.14.196.213
  6    24 ms    20 ms    20 ms  192.178.84.175
  7    49 ms    48 ms    49 ms  142.250.214.105
  8    82 ms    84 ms    63 ms  bom07s29-in-f4.1e100.net [142.250.182.228]

Trace complete.
```

2.command : netstat

The ’netstat’ (network statistics) command is a powerful utility used primarily by system administrators and network engineers to monitor network connections and performance on a computer running Windows. It provides detailed information about network connections, protocol statistics, and topology, making it essential for diagnosing and resolving network issues. The command can be used to display active connections, listening ports, routing tables, and much more.

```
C:\Users\lab1002>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.0.231:49752    20.198.119.84:https    ESTABLISHED
  TCP    192.168.0.231:54059    151.101.38.172:http    TIME_WAIT
  TCP    192.168.0.231:54651    a23-193-114-57:https   ESTABLISHED
  TCP    192.168.0.231:55528    151.101.38.172:http    ESTABLISHED
  TCP    192.168.0.231:55593    server-18-66-38-57:https  ESTABLISHED
  TCP    192.168.0.231:55595    server-18-172-78-46:https  TIME_WAIT
  TCP    192.168.0.231:55596    server-18-172-78-124:https  TIME_WAIT
  TCP    192.168.0.231:55597    207.65.33.78:https     ESTABLISHED
```

options :

1. netstat -a

The ‘-a’ argument tells ’netstat’ to display all active TCP connections as well as the TCP and UDP ports the machine is currently listening to.

```
C:\Users\lab1002>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            DESKTOP-G2AEGUV:0      LISTENING
  TCP    0.0.0.0:445            DESKTOP-G2AEGUV:0      LISTENING
  TCP    0.0.0.0:2869           DESKTOP-G2AEGUV:0      LISTENING
  TCP    0.0.0.0:5040           DESKTOP-G2AEGUV:0      LISTENING
```

2. netstat -e

The '-e' option instructs 'netstat' to provide byte count statistics for network adapters, showing you the total bytes and packets sent and received, and errors.

```
C:\Users\lab1002>netstat -e
Interface Statistics

                           Received            Sent

Bytes                     948448560       125179924
Unicast packets              907636          745056
Non-unicast packets          158220            1364
Discards                          0               0
Errors                            0               0
Unknown protocols                 0
```

3. netstat -n

The '-n' argument prevents 'netstat' from attempting to resolve hostnames and displays all IP addresses numerically. This avoids any delay or errors related to DNS lookups, ensuring the settings and outputs are processed quickly and reflect actual IP address relationships.

```
C:\Users\lab1002>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.0.231:2869     192.168.0.1:38140      TIME_WAIT
  TCP    192.168.0.231:2869     192.168.0.1:38141      TIME_WAIT
  TCP    192.168.0.231:7680     192.168.0.115:61160    TIME_WAIT
  TCP    192.168.0.231:49752    20.198.119.84:443      ESTABLISHED
  TCP    192.168.0.231:57058    151.101.38.172:80      ESTABLISHED
  TCP    192.168.0.231:58698    184.26.168.225:80      TIME_WAIT
  TCP    192.168.0.231:58918    192.168.0.1:1900       TIME_WAIT
  TCP    192.168.0.231:58919    192.168.0.1:1900       TIME_WAIT
  TCP    192.168.0.231:58933    20.50.73.11:443        TIME_WAIT
```

4. netstat -o

With '-o', 'netstat' adds the process ID (PID) of each connection to the output. The PID helps link open connections directly back to the process that initiated them, thereby combining networking information with process management for administrator scrutiny.

```
C:\Users\lab1002>netstat -o

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    192.168.0.231:49752    20.198.119.84:https    ESTABLISHED     3512
  TCP    192.168.0.231:57058    151.101.38.172:http    TIME_WAIT       0
  TCP    192.168.0.231:59159    151.101.38.172:http    ESTABLISHED     3156
  TCP    192.168.0.231:59585    192.168.0.1:ssdp       TIME_WAIT       0
  TCP    192.168.0.231:59598    192.168.0.1:ssdp       TIME_WAIT       0
  TCP    192.168.0.231:59605    169:http               TIME_WAIT       0
  TCP    192.168.0.231:59627    DESKTOP-3VKAP1G:wsd    TIME_WAIT       0
```

5. netstat -r

The '-r' switch provides a view of the system's IP routing table, showing the network destinations, gateway addresses, and interface lists. This exposes how system-tied routes are managed and resolved across connected networks.

```
\Users\lab1002>netstat -r
===========================================================================
nterface List
2...e0 9d 31 e5 a2 c7 ......Intel(R) Dual Band Wireless-AC 3168
3...e0 9d 31 e5 a2 c8 ......Microsoft Wi-Fi Direct Virtual Adapter
7...e2 9d 31 e5 a2 c7 ......Microsoft Wi-Fi Direct Virtual Adapter #2
5...18 60 24 b0 35 34 ......Realtek PCIe GbE Family Controller
3...e0 9d 31 e5 a2 cb ......Bluetooth Device (Personal Area Network)
1...........................Software Loopback Interface 1
===========================================================================

Pv4 Route Table
===========================================================================
tive Routes:
twork Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.231     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      192.168.0.0    255.255.255.0         On-link     192.168.0.231    291
    192.168.0.231  255.255.255.255         On-link     192.168.0.231    291
    192.168.0.255  255.255.255.255         On-link     192.168.0.231    291
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link     192.168.0.231    291
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link     192.168.0.231    291
===========================================================================
ersistent Routes:
 None
```

```
Pv6 Route Table
===========================================================================
ctive Routes:
If Metric Network Destination      Gateway
 1    331 ::1/128                  On-link
 5    291 fe80::/64                On-link
 5    291 fe80::a134:2541:ed15:a23e/128
                                   On-link
 1    331 ff00::/8                 On-link
 5    291 ff00::/8                 On-link
===========================================================================
ersistent Routes:
 None
```

4. command : ipconfig

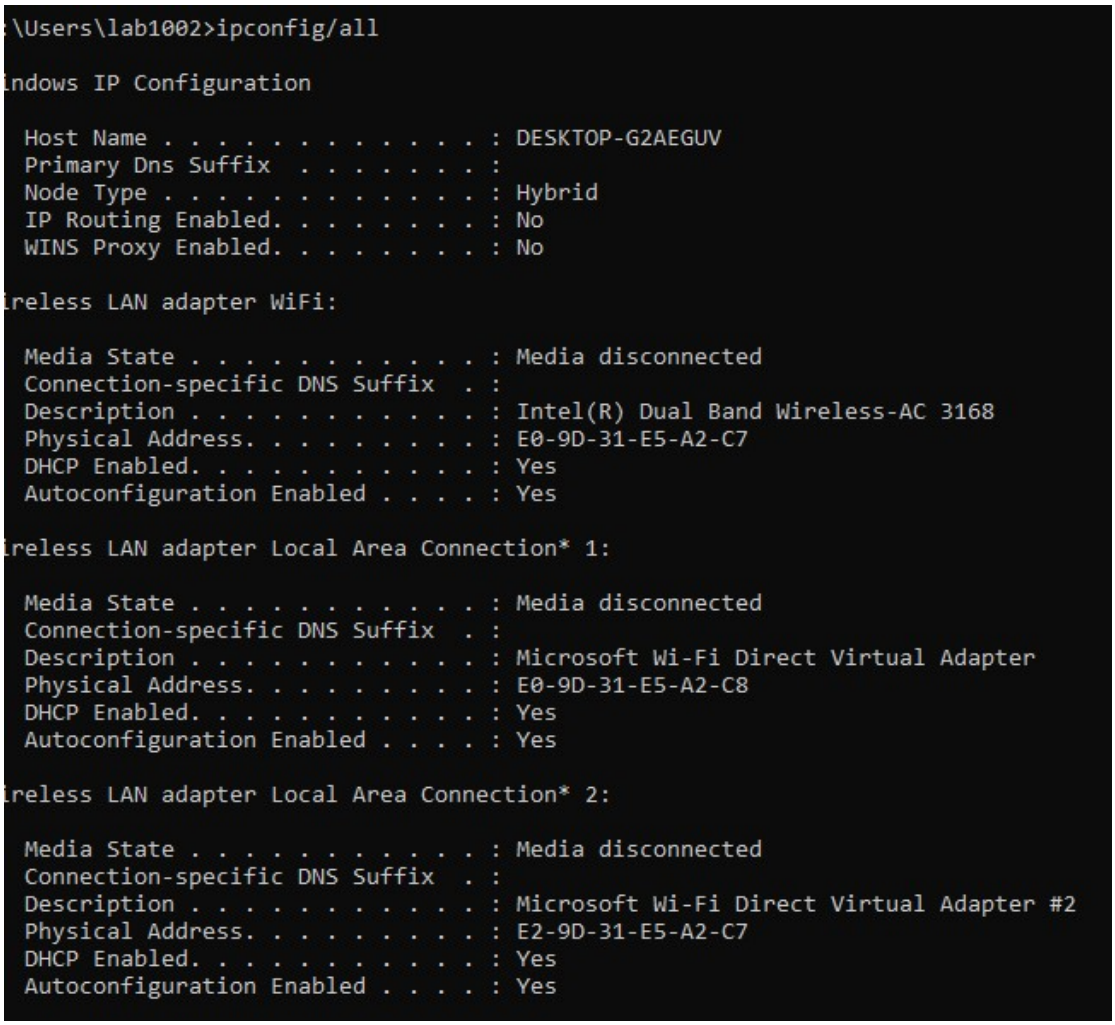
The ipconfig command is a powerful tool in Windows that displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings.

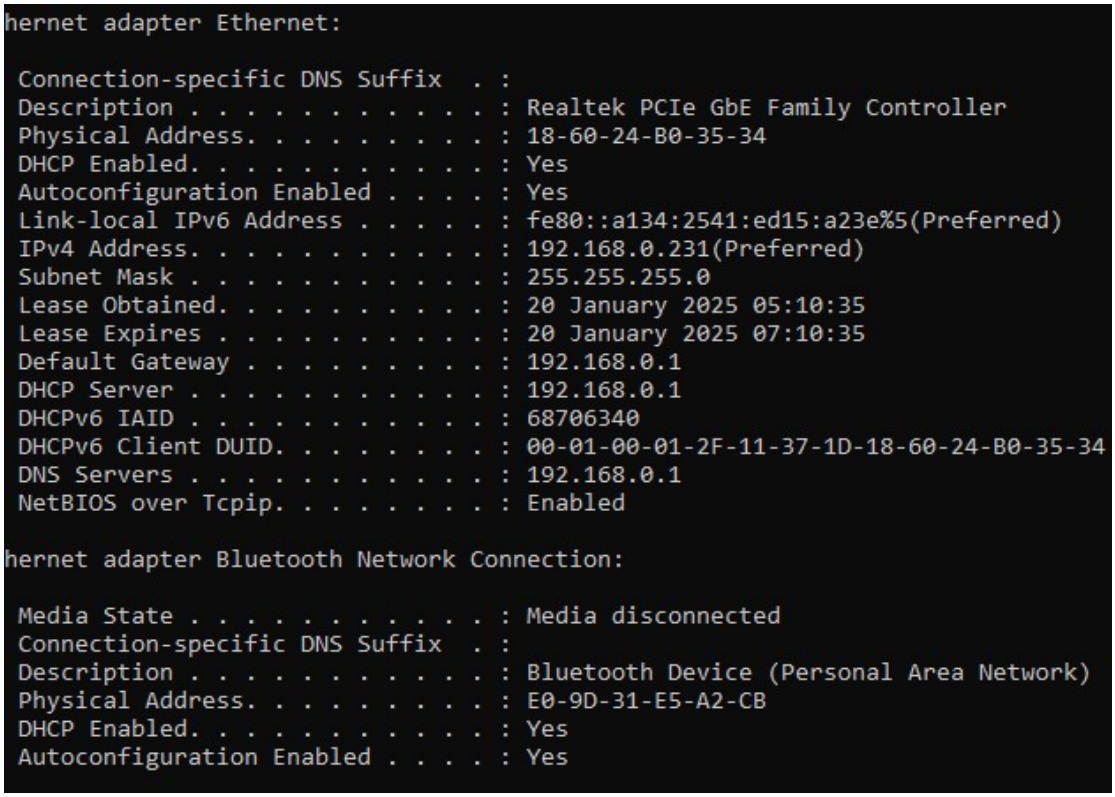
```
:\Users\lab1002>ipconfig

indows IP Configuration


ireless LAN adapter WiFi:

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

ireless LAN adapter Local Area Connection* 1:

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

ireless LAN adapter Local Area Connection* 2:

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

thernet adapter Ethernet:

  Connection-specific DNS Suffix  . :
  Link-local IPv6 Address . . . . . : fe80::a134:2541:ed15:a23e%5
  IPv4 Address. . . . . . . . . . . : 192.168.0.231
  Subnet Mask . . . . . . . . . . . : 255.255.255.0
  Default Gateway . . . . . . . . . : 192.168.0.1

thernet adapter Bluetooth Network Connection:

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

options:

1. ipconfig /all

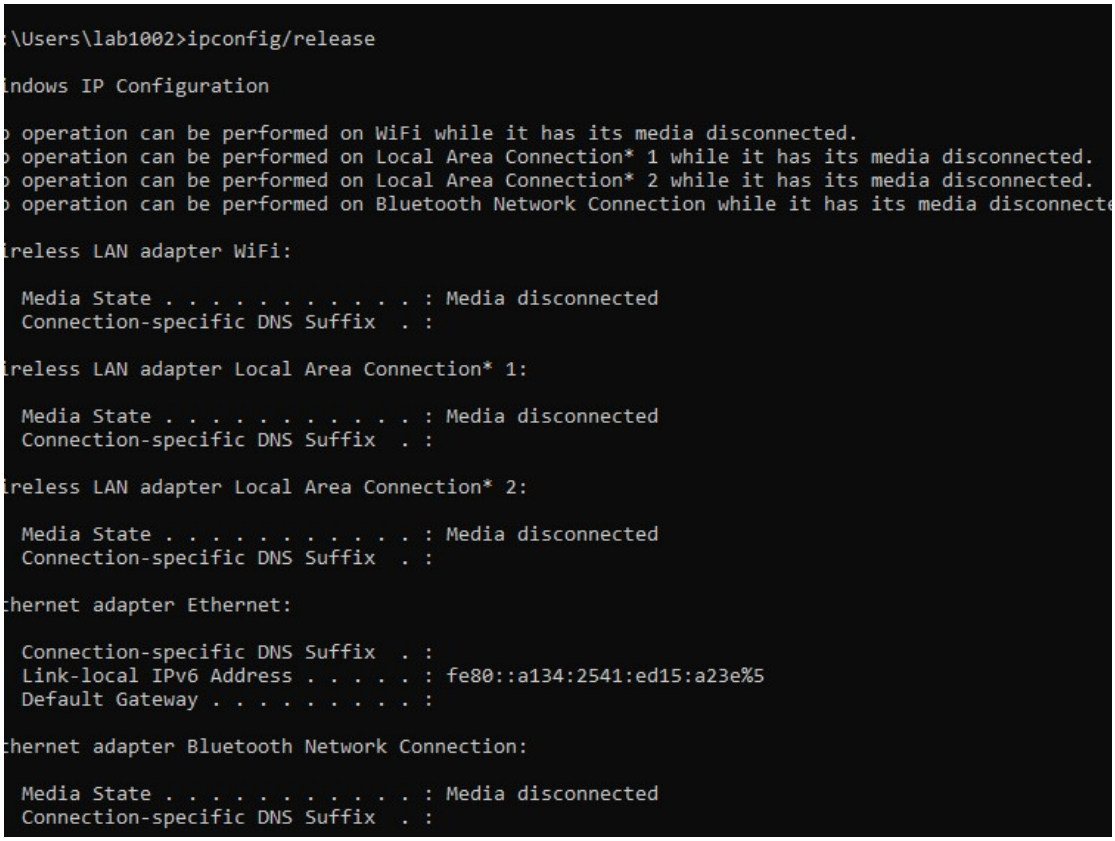
/all: Displays the full TCP/IP configuration for all adapters.





2.ipconfig /release

/release: Releases the current DHCP configuration and discards the IP address configuration for all adapters or a specific adapter.



3. ipconfig /renew

/renew: Renews the DHCP configuration for all adapters or a specific adapter

```
\Users\lab1002>ipconfig/renew

ndows IP Configuration

 operation can be performed on WiFi while it has its media disconnected.
 operation can be performed on Local Area Connection* 1 while it has its media disconnected.
 operation can be performed on Local Area Connection* 2 while it has its media disconnected.
 operation can be performed on Bluetooth Network Connection while it has its media disconnect

reless LAN adapter WiFi:

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

reless LAN adapter Local Area Connection* 1:

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

reless LAN adapter Local Area Connection* 2:

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

hernet adapter Ethernet:

 Connection-specific DNS Suffix  . :
 Link-local IPv6 Address . . . . . : fe80::a134:2541:ed15:a23e%5
 IPv4 Address. . . . . : 192.168.0.231
 Subnet Mask . . . . . : 255.255.255.0
 Default Gateway . . . . . : 192.168.0.1

hernet adapter Bluetooth Network Connection:

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

4.ipconfug/review

```
:\Users\lab1002>ipconfig/review

rror: unrecognized or incomplete command line.

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

here
   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.


he default is to display only the IP address, subnet mask and
```

5.ipconfig /flushdns

/flushdns: Flushes and resets the contents of the DNS client resolver cache.

```
\Users\lab1002>ipconfig/flushdns

ndows IP Configuration

ccessfully flushed the DNS Resolver Cache.
```

5. Command: nslookup:

The NsLoookup command is used to perform network troubleshooting, fetch DNS information, and verify domain configuration.

```
\Users\lab1002>nslookup
fault Server:  UnKnown
dress:  192.168.0.1
```

option:

1. NsLoookup <domain-name>

To lookup the IP of a domain name (domain name to IP address)

```
:\Users\lab1002>nslookup google.com
erver:  UnKnown
ddress:  192.168.0.1

on-authoritative answer:
ame:    google.com
ddresses:  2404:6800:4009:81c::200e
          142.250.77.46
```

2. nslookup -type = mx

View Mail Exchange server information.

```
C:\Users\lab1002>nslookup -type=mx google.com
Server:  UnKnown
Address:  192.168.0.1

Non-authoritative answer:
google.com      MX preference = 10, mail exchanger = smtp.google.com

smtp.google.com internet address = 74.125.24.26
smtp.google.com internet address = 74.125.24.27
smtp.google.com internet address = 142.251.10.27
smtp.google.com internet address = 142.251.175.27
smtp.google.com internet address = 142.251.175.26
smtp.google.com AAAA IPv6 address = 2404:6800:4003:c03::1a
smtp.google.com AAAA IPv6 address = 2404:6800:4003:c03::1b
smtp.google.com AAAA IPv6 address = 2404:6800:4003:c1c::1b
smtp.google.com AAAA IPv6 address = 2404:6800:4003:c1c::1a
```

3. nslookup -type=ns

Name Server (NS) records store names of the domain's name servers.

```
C:\Users\lab1002>nslookup -type=ns google.com
Server:  UnKnown
Address:  192.168.0.1

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

ns3.google.com  internet address = 216.239.36.10
ns3.google.com  AAAA IPv6 address = 2001:4860:4802:36::a
ns2.google.com  internet address = 216.239.34.10
ns2.google.com  AAAA IPv6 address = 2001:4860:4802:34::a
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  AAAA IPv6 address = 2001:4860:4802:32::a
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  AAAA IPv6 address = 2001:4860:4802:38::a
```

4. nslookup -type=soa [domain-name]

Start of Authority (SOA) records provide authoritative information about the domain and the server, such as the email address of the administrator, serial number, refresh interval, query expiration time, etc.

```
C:\Users\lab1002>nslookup -type=soa google.com
Server:  UnKnown
Address:  192.168.0.1

Non-authoritative answer:
google.com
        primary name server = ns1.google.com
        responsible mail addr = dns-admin.google.com
        serial  = 717200457
        refresh = 900 (15 mins)
        retry   = 900 (15 mins)
        expire  = 1800 (30 mins)
        default TTL = 60 (1 min)

google.com      nameserver = ns1.google.com
google.com      nameserver = ns3.google.com
google.com      nameserver = ns4.google.com
google.com      nameserver = ns2.google.com
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  AAAA IPv6 address = 2001:4860:4802:38::a
ns2.google.com  internet address = 216.239.34.10
ns2.google.com  AAAA IPv6 address = 2001:4860:4802:34::a
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  AAAA IPv6 address = 2001:4860:4802:32::a
ns3.google.com  internet address = 216.239.36.10
ns3.google.com  AAAA IPv6 address = 2001:4860:4802:36::a
```

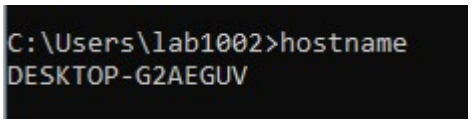
5. nslookup -type=txt [domain-name]

TXT records contain important information for users outside of the domain.

```
\Users\lab1002>nslookup -type=txt google.com
erver:  UnKnown
ddress:  192.168.0.1

n-authoritative answer:
oogle.com      text =

      "facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
oogle.com      text =

      "apple-domain-verification=30afIBcvSuDV2PLX"
oogle.com      text =

      "docusign=1b0a6754-49b1-4db5-8540-d2c12664b289"
oogle.com      text =

      "globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="
oogle.com      text =

      "cisco-ci-domain-verification=479146de172eb01ddee38b1a455ab9e8bb51542ddd7f1fa298557dfa7b22d96
oogle.com      text =

      "google-site-verification=wD8N7i1JTNTkezJ49swvWW48f8_9xveREV4oB-0Hf5o"
```

6. Hostname:

The hostname command is a versatile tool used to display or set the system's hostname. A hostname is a unique identifier assigned to a computer connected to a network.



Options:

1. Display the alias name of the host: hostname -a

2. Display the DNS domain name: hostname -d

3. Display the IP address: hostname -i

4. Display the short hostname: hostname -s

5. Display the version information: hostname -V

7. ARP command

ARP stands for “Address Resolution Protocol” is a protocol for mapping an IP address to a physical MAC address on a local area network.

```
C:\Users\lab1002>arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physica
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each AR
                table are displayed.
  -g            Same as -a.
  -v            Displays current ARP entries in verbose mode.  All invalid
                entries and entries on the loop-back interface will be shown.
  inet_addr     Specifies an internet address.
  -N if_addr    Displays the ARP entries for the network interface specified
                by if_addr.
  -d            Deletes the host specified by inet_addr. inet_addr may be
                wildcarded with * to delete all hosts.
  -s            Adds the host and associates the Internet address inet_addr
                with the Physical address eth_addr.  The Physical address is
                given as 6 hexadecimal bytes separated by hyphens. The entry
                is permanent.
  eth_addr      Specifies a physical address.
  if_addr       If present, this specifies the Internet address of the
                interface whose address translation table should be modified.
                If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.
```

Option :

1. arp -a :

```
C:\Users\lab1002>arp -a

Interface: 192.168.0.231 --- 0x5
  Internet Address      Physical Address      Type
  192.168.0.1           ac-15-a2-b9-9e-29     dynamic
  192.168.0.100         a4-ae-12-84-80-e0     dynamic
  192.168.0.104         18-60-24-90-00-69     dynamic
  192.168.0.108         d4-be-d9-c7-87-19     dynamic
  192.168.0.115         a4-ae-12-84-7f-d8     dynamic
  192.168.0.116         18-60-24-8f-81-3c     dynamic
  192.168.0.119         f4-39-09-49-6c-af     dynamic
  192.168.0.121         18-60-24-84-fe-dc     dynamic
  192.168.0.122         48-9e-bd-9e-72-a3     dynamic
  192.168.0.126         48-9e-bd-9d-2e-ae     dynamic
  192.168.0.131         48-9e-bd-9e-73-91     dynamic
  192.168.0.133         18-60-24-8f-81-1b     dynamic
  192.168.0.139         18-60-24-8f-85-43     dynamic
```

8. CURL:

Curl is a command-line tool used to transfer data to and from a server using various protocols such as HTTP, HTTPS, FTP, and more.

```
C:\Users\lab1002>curl https://google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8"
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.com/">here</A>.
</BODY></HTML>
```

1. curl -o

This option specifies the output file for the response.

```
C:\Users\lab1002>curl -o renamedfile.txt https://example.com/myfile.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1256  100  1256    0     0    761      0  0:00:01  0:00:01 --:--:--   763
```

2. -v or --verbose:

This option enables verbose mode, which displays detailed information about the request and response.

```
\Users\lab1002>curl -v renamedfile.txt https://example.com/myfile.txt
Could not resolve host: renamedfile.txt
shutting down connection #0
rl: (6) Could not resolve host: renamedfile.txt
Host example.com:443 was resolved.
IPv6: (none)
IPv4: 23.215.0.138, 96.7.128.175, 96.7.128.198, 23.192.228.80, 23.192.228.84, 23.215.0.13
  Trying 23.215.0.138:443...
Connected to example.com (23.215.0.138) port 443
schannel: disabled automatic use of client certificate
ALPN: curl offers http/1.1
ALPN: server accepted http/1.1
using HTTP/1.x
GET /myfile.txt HTTP/1.1
Host: example.com
User-Agent: curl/8.9.1
Accept: */*
```

3. -s or --silent:

This option suppresses the output of the request and response.

```
\Users\lab1002>curl -v renamedfile.txt https://example.com/myfile.txt
Could not resolve host: renamedfile.txt
shutting down connection #0
rl: (6) Could not resolve host: renamedfile.txt
Host example.com:443 was resolved.
IPv6: (none)
IPv4: 23.215.0.138, 96.7.128.175, 96.7.128.198, 23.192.228.80, 23.192.228.84, 23.215.0.136
  Trying 23.215.0.138:443...
Connected to example.com (23.215.0.138) port 443
schannel: disabled automatic use of client certificate
ALPN: curl offers http/1.1
ALPN: server accepted http/1.1
using HTTP/1.x
GET /myfile.txt HTTP/1.1
Host: example.com
User-Agent: curl/8.9.1
Accept: */*

Request completely sent off
HTTP/1.1 404 Not Found
Accept-Ranges: bytes
Content-Type: text/html
ETag: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"
Last-Modified: Mon, 13 Jan 2025 20:11:20 GMT
Server: AkamaiNetStorage
Content-Length: 1256
Expires: Mon, 20 Jan 2025 06:46:59 GMT
Cache-Control: max-age=0, no-cache, no-store
Pragma: no-cache
Date: Mon, 20 Jan 2025 06:46:59 GMT
Connection: keep-alive

doctype html>
tml>
ead>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
  body {
```

4. -u or --user:

This option specifies the username and password for authentication.

```
\Users\lab1002>curl -u renamedfile.txt https://example.com/myfile.txt
ter host password for user 'renamedfile.txt':
doctype html>
tml>
ead>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
  body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "

  }
  div {
      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
  }
  a:link, a:visited {
      color: #38488f;
      text-decoration: none;
  }
  @media (max-width: 700px) {
      div {
          margin: 0 auto;
          width: auto;
      }
  }
  </style>
head>

ody>
iv>
```

ASSIGNMENT -2

Aim:

Installation and configuration of NS2 and implementation of TCL hello programming

Theory:

TCL Programming:

Tcl is a general purpose multi-paradigm system programming language. It is a scripting language that aims at providing the ability for applications to communicate with each other. Tcl is shortened form of Tool Command Language.

The features of Tcl are as follows —

- Reduced development time.
- Powerful and simple user interface kit with integration of TK.
- Write once, run anywhere. It runs on Windows, Mac OS X, and almost on every Unix platform.
- Quite easy to get started for experienced programmers; since, the language is so simple that they can learn Tcl in a few hours or days.
- You can easily extend existing applications with Tcl. Also, it is possible to include Tcl in C, C++, or Java to Tcl or vice versa.
- Have a powerful set of networking functions.
- Finally, it's an open source, free, and can be used for commercial applications without any limit.

Applications

Tcl is a general-purpose language and you can find Tcl everywhere. It includes, Scalable websites that are often backed by databases.

High performance web servers build with TclHttpd.

Tcl with CGI based websites.

Desktop GUI applications.

Embedded applications.

What is simulator:

A simulator refers to a software tool or environment used to mimic the behavior of hardware or a system for testing and development purposes without needing the actual physical hardware.

Simulators are commonly used in TCL programming:

1. Simulation of Hardware Systems:

If you're working with embedded systems, a simulator can help emulate hardware behavior, allowing you to test your TCL scripts without actually interacting with the real hardware.

2. Virtual Environments:

Simulators create virtual environments where the behavior of devices, networks, or systems can be replicated. For example, if you're writing TCL scripts to control a device or manage communication protocols, you might use a simulator to test how your script interacts with the simulated system.

3. Testing and Debugging:

It allows developers to test and debug their TCL programs in a controlled environment. This is especially useful when the real hardware is too expensive, unavailable, or difficult to access during early development stages.

4. Automated Testing:

Simulators can also be part of a larger automated testing framework, where you use TCL scripts to run predefined tests on the simulated system.

Nam file

a NAM file usually refers to a Network Animator (NAM) file, which is associated with the NS-2 or NS-3 simulators. These simulators are used for network simulation, and the NAM file is a visual representation of the simulation.

Key points about NAM files:

1. Purpose:

A NAM file is used to visualize the network simulations that have been run in NS-2 or NS-3. It contains details of the network's topology, events (like packet transmission), and the states of nodes and links over time.

2. Format:

A NAM file is typically a trace file, which records the events during a simulation run in the form of a text file. It contains details of events, such as packet arrival, packet drops, and the movement of nodes, etc.

3. Usage:

After running a simulation in NS-2 or NS-3, you can use the NAM tool to visualize the data from the NAM file. It provides an animation of how packets travel across the network, node interactions, and other relevant simulation details.

4. Features:

The NAM file allows users to see the simulation's progression in graphical form. You can observe how the network evolves over time, check node mobility (for mobile networks), and inspect packet flow and routing protocols.

5. Working with NAM Files:

In NS-2, the NAM tool can be used to create and visualize the simulation trace generated by NS-2. The process generally involves running a TCL script to set up the simulation, generating output in the form of a NAM file, and then loading it into NAM for visualization.

Installation steps:

1. Install TCL (if not already installed)

Ubuntu usually comes with TCL pre-installed, but if you don’t have it, you can easily install it. To check if TCL is installed, run:

bash: tclsh

If TCL is installed, it will open the interactive TCL shell. If it's not installed, you can install it using:

bash:

sudo apt update

sudo apt install tcl

2. Write a TCL Script

You can write a simple TCL script using any text editor of your choice. Here’s how to create a simple script.

Open a text editor like nano, vim, or gedit:

For example, using nano:

bash

nano hello.tcl

In the editor, write a basic TCL script, for example:

tcl

hello.tcl

puts "Hello, world!"

Save and exit the editor.

If using nano, press Ctrl + O to save, then Ctrl + X to exit.

3. Run the TCL Script

To run the TCL script you just wrote, use the tclsh command followed by the script name.

For example:

bash

tclsh hello.tcl

This will execute your script, and you should see the output:

Hello, world!

TCL script

```
  GNU nano 6.2                                                  hello.tcl
puts "Welcome to lab 1003"
```

```
sahil@1003-032:~$ nano hello.tcl
sahil@1003-032:~$ tclsh hello.tcl
Welcome to lab 1003
```

CONCLUSION:

Succesfully installed NS2 in computer and implemented a TCL script and learnt to run several commands.

LO Mapping: 02

Assignment 3

AIM:

Implementation of specific Network Topology with respect to TCP

THEORY:

In wireless network nodes communicate using a communication model that consists of TCP agent, TCP Sink agent and FTP application. The sender node is attached to TCP agent and the receiver is attached to TCP sink agent. The connection between TCP agent and TCP sink agent is established using a keyword 'connect'. Transport agent TCP and application FTP are connected using a keyword 'attach-agent'. On receiving the data packets TCP sink agent will transmit acknowledgement. It will also acknowledge the transmission rate. The lost packets are interpreted as congestion in the network.

CODE:

```
#Create a Simulator Object set

ns [new Simulator]


#0pen 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 finish procedure

proc finish {} {

global ns nf np $ns

flush-trace #Close

NAM Trace
```

SPECIFIC NETWORK PROTOCAL

```
close $nf #Execute NAM on
the tracefile exec nam
out.nam &
exit 0
}

#create two nodes
set n0 [$ns node] set
n1 [$ns node] set n2
[$ns node] set n3
[$ns node]

#Create Links between all nodes
$ns duplex-link $n0 $n1 2Mb 10ms DropTail #set Queue Size
$ns queue-limit $n0 $nl 5

#Monitor The queue for Link (n0-n1)
$ns duplex-link-op $n0 $n1 queuePos 0.5

#Set up a TCP connection set
tcp [new Agent/TCP] $ns
attach-agent $n0 $tcp set
sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
#Set Packet Colour
$tcp set fid_ 1

#Create Links between all nodes
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
```

SPECIFIC NETWORK PROTOCAL

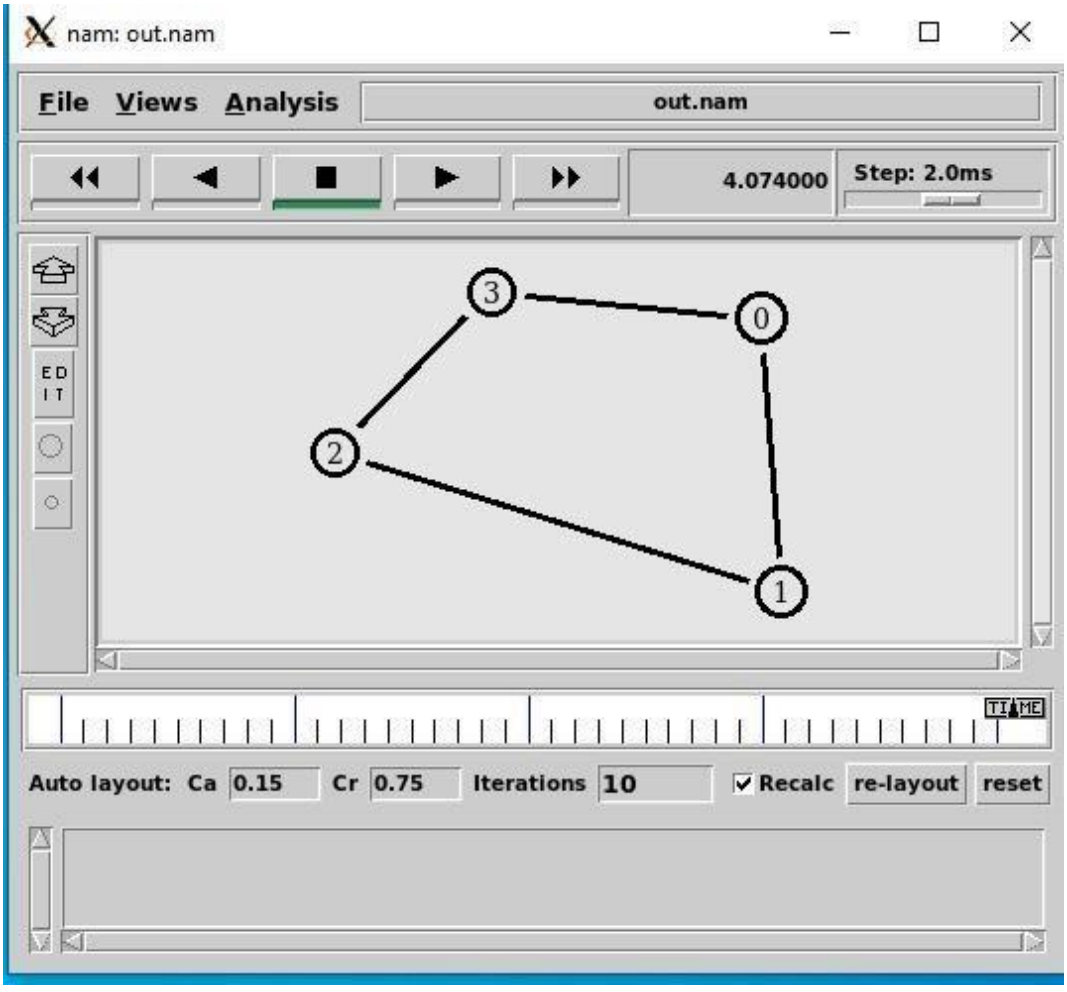
```
#set Queue Size
$ns queue-limit $n1 $n2 5

#Monitor The queue for link (n0-n1)
$ns duplex-link-op $nl $n2 queuePos 0.5

#Set up a TCP connection set
tcp [new Agent/TCP] $ns
attach-agent $n1 $tcp set
sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink
#Set Packet Colour
$tcp set fid_1
#Create Links between all nodes
$ns duplex-link $n2 $n3 2Mb 10ms DropTail

#set Queue Size
$ns queue-limit $n2 $n3 5

#Monitor The queue for link (n0-n1)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Set up a TCP connection set
tcp [new Agent/TCP] $ns
attach-agent $n2 $tcp set
sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
#Set Packet Colour
$tcp set fid_ 1
```

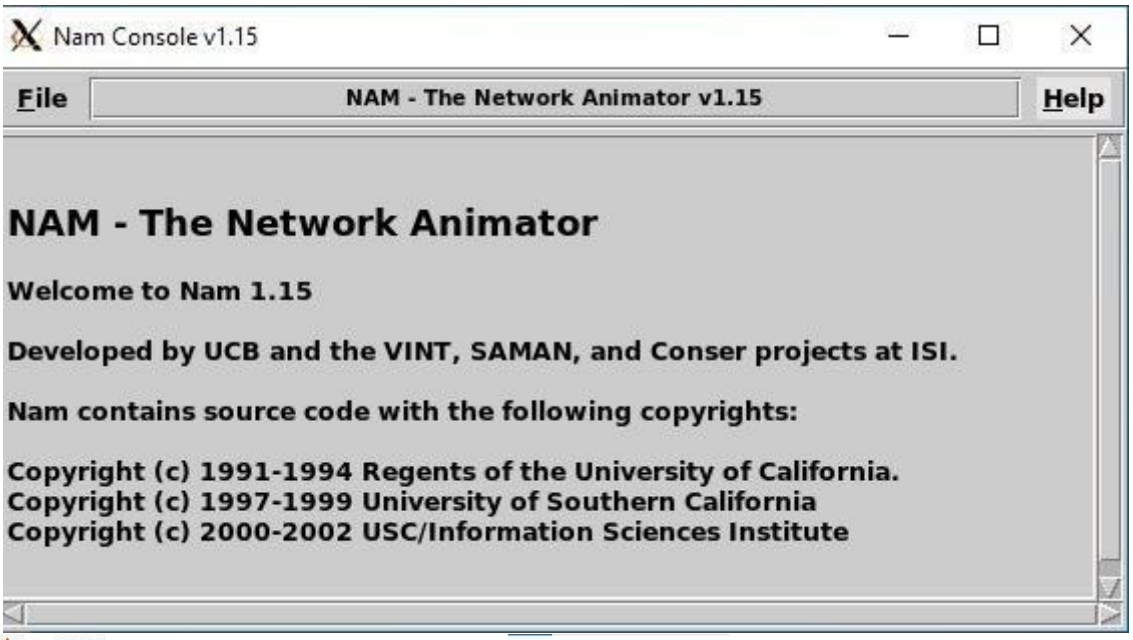
SPECIFIC NETWORK PROTOCAL

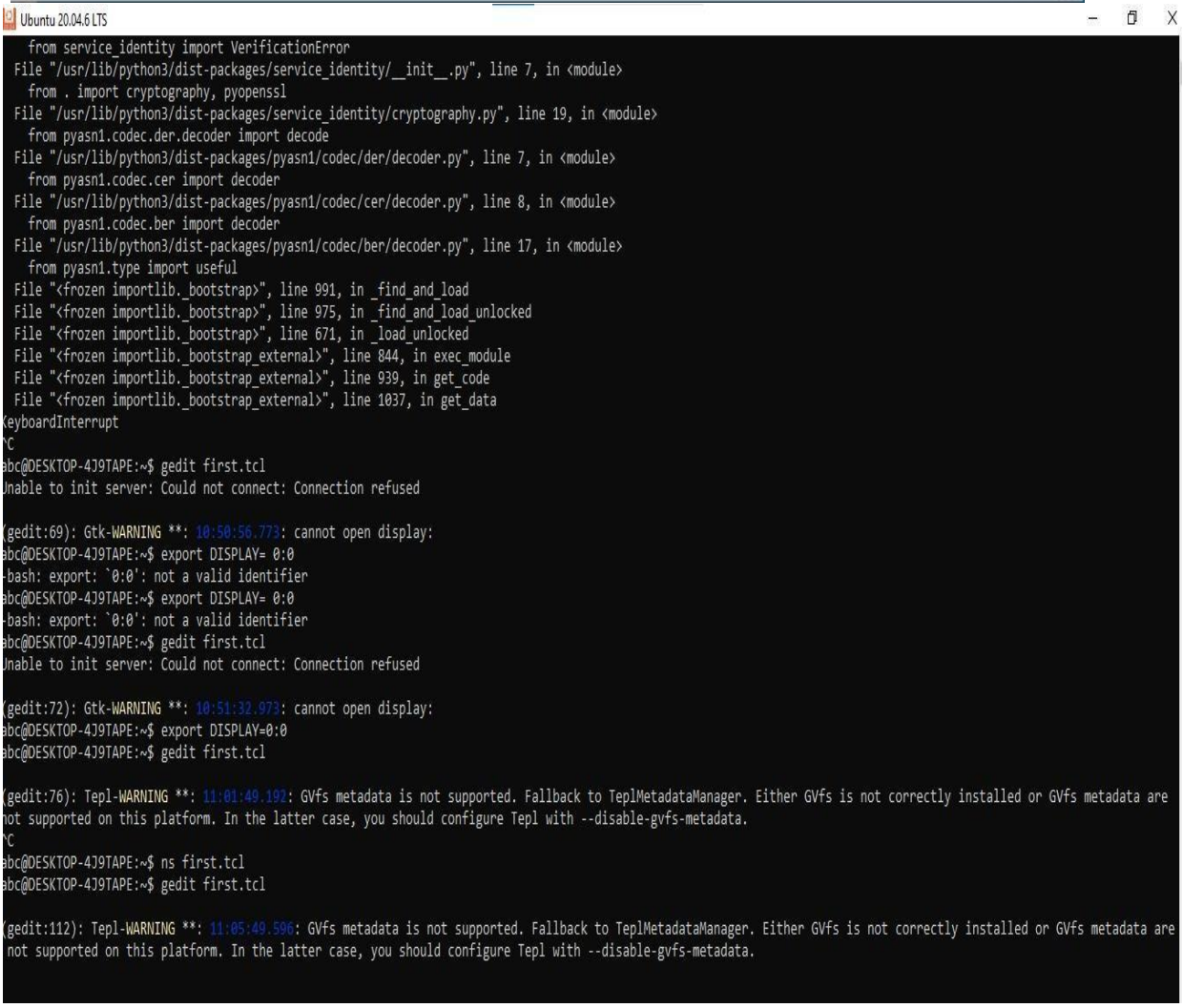
```
#Create links between all nodes
$ns duplex-link $n3 $n0 2Mb 10ms DropTail

#set Queue Size
$ns queue-limit $n3 $n0 5

#Monitor The queue for link (n0-n1)
$ns duplex-link-op $n3 $n0 queuePos 8.5

#Set up a TCP connection set
tcp [new Agent/ICI'] $ns
attach-agent $n3 $tcp set
sink [new Agent/TCPSink]
$ns attach-agent $n0 $sink
$ns connect Stop $sink
#Set Packet Colour
$tcp set fid_ 1

#Set up FTP Protocol (Application Layer) over TCP (Transport Layer) set
ftp [new Application/FTP]
$ftp attach-agent $tcp

#Schedule Events for FTP agents
$ns at 0.1 “$ftp start"
$ns at 4.0 “$ftp stop"

$ns at 5.0 “finish”

#Run Simulator
$ns run
```

SPECIFIC NETWORK PROTOCAL

OUTPUT:


nam: out.nam
File Views Analysis
out.nam
4.074000
Step: 2.0ms
ED IT
3
0
2
1
TIME
Auto layout: Ca 0.15 Cr 0.75 Iterations 10
Recalc
re-layout
reset





SPECIFIC NETWORK PROTOCAL

ASSSIGNMENT 10

AIM: Implement specific topology busing udp

Theory:

In NS2 (Network Simulator 2), topologies represent the structure of how network nodes are organized and how they communicate with each other during simulations. Some common topologies include linear, star, mesh, and tree.

For UDP simulations in NS2, communication is managed using agents. A UDP agent is attached to a node and linked to either a Null agent (used to simulate packet reception) or a CBR (Constant Bit Rate) agent that generates UDP traffic.

In a **linear topology**, nodes are connected in a sequential manner, which is often used in simulations to analyze parameters like end-to-end delay and packet loss during communication between nodes.

Other topologies like **bus**, **ring**, **star**, **tree**, and **mesh** are also used, depending on the type of network being simulated.

To measure the performance of UDP networks in NS2, key metrics such as **Packet Delivery Ratio (PDR)**, **throughput**, **delay**, and **jitter** are evaluated. These metrics are crucial for understanding how well the network performs, especially in real-time applications like voice or video communication.

NS2 simulations are scripted using **TCL** scripts, which define the nodes, connections, queues, and traffic flows in the network, allowing users to create realistic network models for performance analysis.

PROGRAM:

Topology using UDP

SEIT S13 55

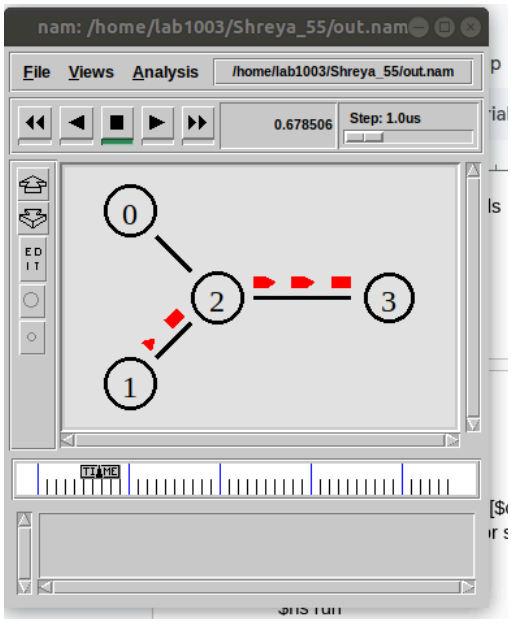
```
# Create a simulator object
set ns [new Simulator]

# Define colors for data flows
$ns color 1 Blue
$ns color 2 Red

# Open 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 $nf
        exec nam out.nam &
        exit 0
}

# Create nodes for bus topology
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

# Create links in bus topology
$ns duplex-link $n0 $n1 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
$ns duplex-link $n3 $n4 1.7Mb 20ms DropTail

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

# Positioning the nodes for NAM visualization
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient right
```

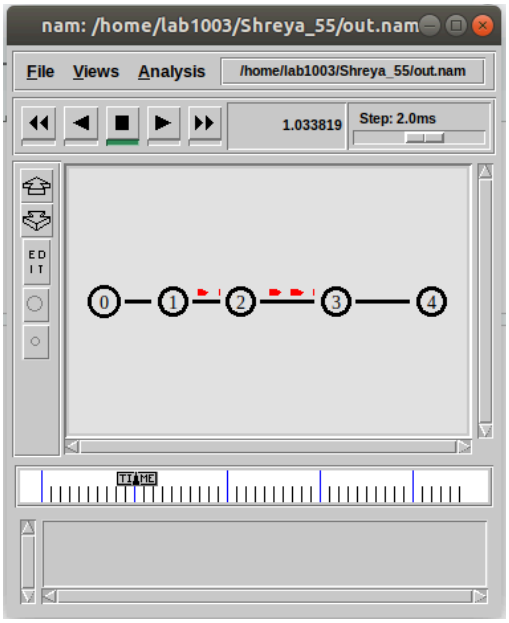
Topology using UDP

```
# Monitor the queue for link (n2-n3)
$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
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

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

# Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

# Setup a CBR over UDP
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

# Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

# Detach agents
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n4 $sink"

# Call finish procedure
$ns at 5.0 "finish"

# Print CBR packet details
```

Topology using UDP

```
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

# Run the simulation
$ns run
```





CONCLUSION:

UDP-based simulations in NS2 allow for the study of real-time network behavior across different topologies. The selected topology plays a significant role in influencing factors like latency, packet loss, and network congestion, making it essential for optimizing overall network performance. Because UDP does not include built-in error control, careful topology design is needed to ensure redundancy and multiple paths to maintain reliability. NS2 offers a robust platform for examining how UDP performs in various network configurations and understanding its effects on different network scenarios.

ASSIGNMENT 5 : L03,5

AIM: Simulation of network with Link State Routing algorithm
Theory:

Link State Routing

Link state routing is the second family of routing protocols. While distance-vector routers use a distributed algorithm to compute their routing tables, link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology. Based on this learned topology, each router is then able to compute its routing table by using the shortest path computation.

Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router i.e. the internet work. The three keys to understand the link state routing algorithm.

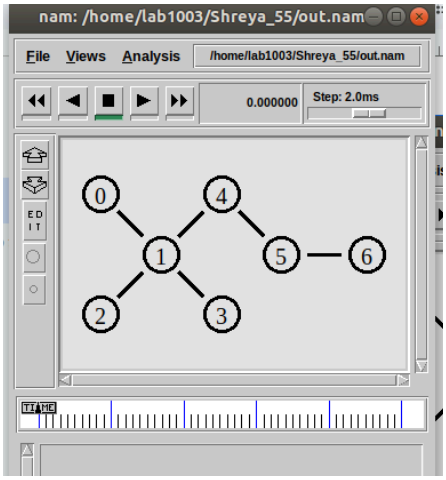
1. **Knowledge about the neighborhood** : Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.
2. **Flooding:** Each router sends the information to every other router on the internetwork except its neighbors. This process is known as flooding. Every router that receives the packet sends the copies to all the neighbors. Finally each and every router receives a copy of the same information.
3. **Information Sharing** : A router send the information to every other router only when the change occurs in the information.

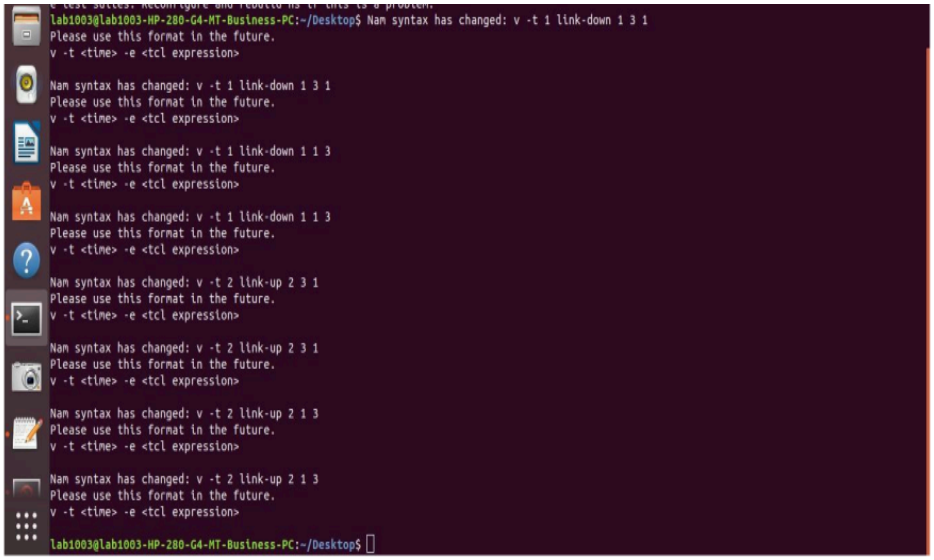
Output:set ns [new Simulator]

```
set nf [open out.nam w]
$ns namtrace-all $nf
set tr [open out.tr w]
$ns trace-all $tr

proc finish {} {
        global nf ns tr
        $ns flush-trace
        close $tr
        exec nam out.nam &
        exit 0
}

# Define 7 nodes
```

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]

# Define duplex links between the nodes
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n3 10Mb 10ms DropTail
$ns duplex-link $n2 $n1 10Mb 10ms DropTail
$ns duplex-link $n4 $n1 10Mb 10ms DropTail
$ns duplex-link $n5 $n4 10Mb 10ms DropTail
$ns duplex-link $n6 $n5 10Mb 10ms DropTail

# Set link orientations to avoid overlap
$ns duplex-link-op $n0 $n1 orient right-down
$ns duplex-link-op $n1 $n3 orient down-right
$ns duplex-link-op $n2 $n1 orient right-up
$ns duplex-link-op $n4 $n1 orient left-down
$ns duplex-link-op $n5 $n4 orient left-up
$ns duplex-link-op $n6 $n5 orient left

# Define TCP connection and FTP application
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set ftp [new Application/FTP]
$ftp attach-agent $tcp

# Define TCP Sink at node n3
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink

# Define UDP connection and CBR application
set udp [new Agent/UDP]
$ns attach-agent $n2 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
```

```
# Define Null agent at node n6
set null [new Agent/Null]
$ns attach-agent $n6 $null

# Connect agents
$ns connect $tcp $sink
$ns connect $udp $null

# Set routing protocol and events
$ns rtmodel-at 1.0 down $n1 $n3
$ns rtmodel-at 2.0 up $n1 $n3
$ns rtproto DV

# Start FTP and CBR applications
$ns at 0.0 "$ftp start"
$ns at 0.0 "$cbr start"

# Finish simulation at time 5.0
$ns at 5.0 "finish"

# Run the simulation
$ns run
```





Conclusion

Link State Routing in **NS-2** simulates the process by which each node shares its link state information with other nodes in the network. This helps nodes to build a complete view of the network topology and calculate the best path to forward packets. This simulation involves **LSA broadcasting**, **Link State Database maintenance**, and **Shortest Path Calculations** using Dijkstra's algorithm.

Assignment 6

Aim:

INSTALLATION OF WIRESHARK & ANALYSIS OF PACKET HEADERS- TCP, IP, UDP

Theory:

Wireshark – Network Traffic Analyzer

Wireshark: Your Network Inspector

Wireshark is a tool used to watch and capture the data that moves across a computer network. Imagine it as a magnifying glass that lets you look at every bit of internet traffic happening on your system.

Uses

- To troubleshoot network problems
- To understand how devices communicate
- For developers to see how data is sent/received
- To teach networking in schools and colleges

Features of Wireshark:

- Live capture of data packets
- Supports a wide range of network protocols
- Works on multiple platforms (Windows, Mac, Linux)
- Easy-to-use graphical interface
- Allows deep analysis of every captured packet

TCP (Transmission Control Protocol)

TCP is a reliable, connection-oriented protocol. It ensures that messages sent between computers arrive properly and in the correct order. It's like a delivery service that guarantees your parcel is safe and arrives exactly as intended.

Features:

- Assigns numbers to bytes for correct ordering
- Sends data as a smooth stream
- Guarantees safe delivery of data
- Manages how much data is sent to avoid congestion
- Supports two-way communication at the same time
- Allows multiple applications to talk over the same network

Structure of a TCP Segment:

1.Header – Holds control and tracking info

2. Data – The actual message being sent

TCP Header Fields Explained:

- **Source & Destination Ports:** Identify the sending and receiving apps
- **Sequence Number:** Tracks the order of bytes sent
- **Acknowledgment Number:** Confirms receipt of data
- **Header Length:** Size of the TCP header
- **Flags (Control Bits):**
  - **URG** – Urgent data
  - **ACK** – Acknowledgement
  - **PSH** – Push data immediately
  - **RST** – Reset the connection
  - **SYN** – Start connection
  - **FIN** – Finish connection
- **Window Size:** How much data the receiver can accept
- **Checksum:** Checks for errors
- **Urgent Pointer:** Points to urgent data in the stream
- **Options:** Extra configuration info

UDP (User Datagram Protocol)

UDP is a lightweight, faster alternative to TCP. It doesn't check if the data reaches or if it's in order. It's like tossing a paper airplane — it might reach, or it might not, but it's quick!

Features:

- No need to establish a connection
- Doesn't confirm delivery
- No error correction
- Small and simple
- Best for fast, real-time services (e.g., video calls, games)

UDP Packet Structure:

- **Source Port & Destination Port:** Identify apps on sender/receiver
- **Length:** Size of header + data
- **Checksum:** Optional error check

IP – Internet Protocol

IPv4 – Internet Protocol Version 4

IPv4 is the classic format for internet addresses, using 32-bit numbers (like 192.168.1.1). It can handle around 4.3 billion unique addresses.

IPv4 Header Fields:

- **Version:** Always 4 for IPv4
- **Header Length:** Size of the header
- **Type of Service:** Priority handling of packets

- **Total Length:** Total size of the packet
- **Identification:** Used to identify packet fragments
- **Flags:** Used for breaking and reassembling packets
- **Fragment Offset:** Location of fragment in original data
- **TTL (Time to Live):** Limits packet lifetime to avoid loops
- **Protocol:** Indicates if it's TCP, UDP, etc.
- **Header Checksum:** Detects errors in the header
- **Source & Destination IP:** Where it came from and where it's going
- **Options:** Rarely used, extra controls

IPv6 – Internet Protocol Version 6

IPv6 is the next-gen internet addressing system. It uses 128-bit addresses, which means more than enough unique IPs for every device on Earth (and beyond!).

Example:

2001:0db8:85a3:0000:0000:8a2e:0370:7334

Why IPv6?

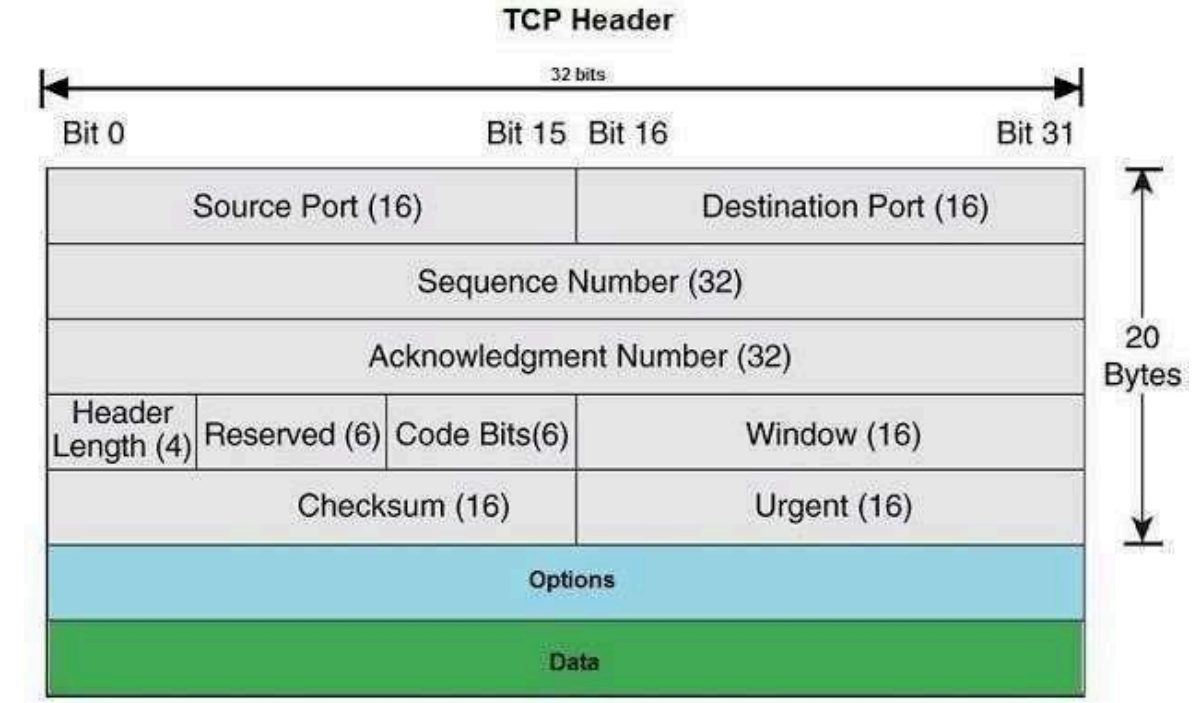
- Provides a huge number of addresses
- Improves network efficiency
- Has built-in security features
- Simpler and faster processing

IPv6 Header Fields:

- **Version:** Always 6
- **Traffic Class:** Packet priority
- **Flow Label:** Groups related packets
- **Payload Length:** Size of the data payload
- **Next Header:** What comes next (e.g., TCP or UDP)
- **Hop Limit:** Like TTL, limits how far the packet travels
- **Source & Destination Address:** Sender and receiver IPs

TCP HEADER

```
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 1333312426
0101 .... = Header Length: 20 bytes (5)
Flags: 0x014 (RST, ACK)
Window: 0
[Calculated window size: 0]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x9ee3 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
```

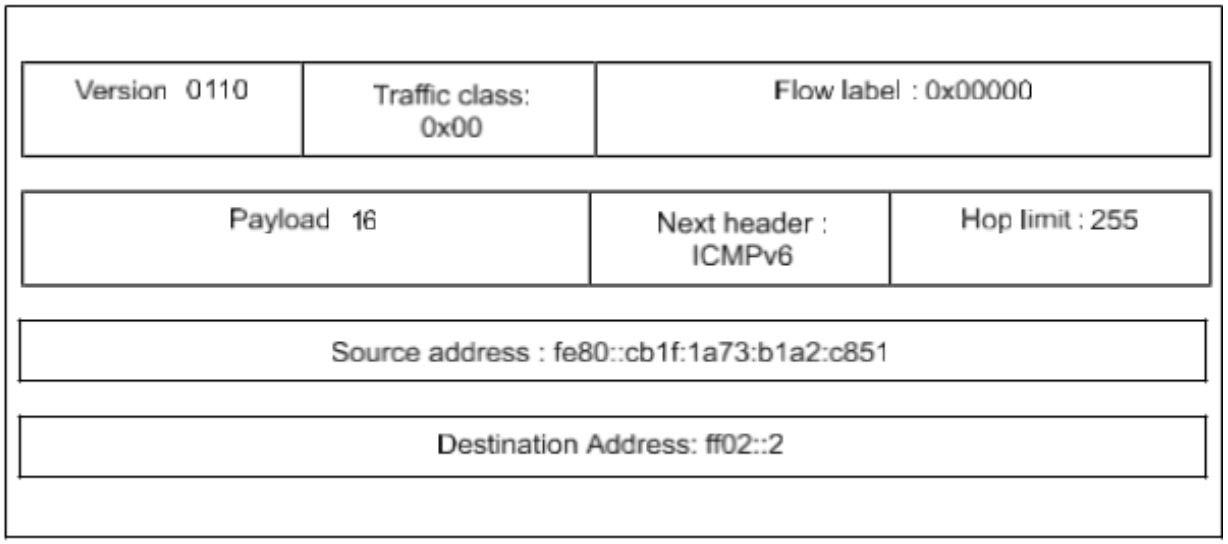


UDP HEADER

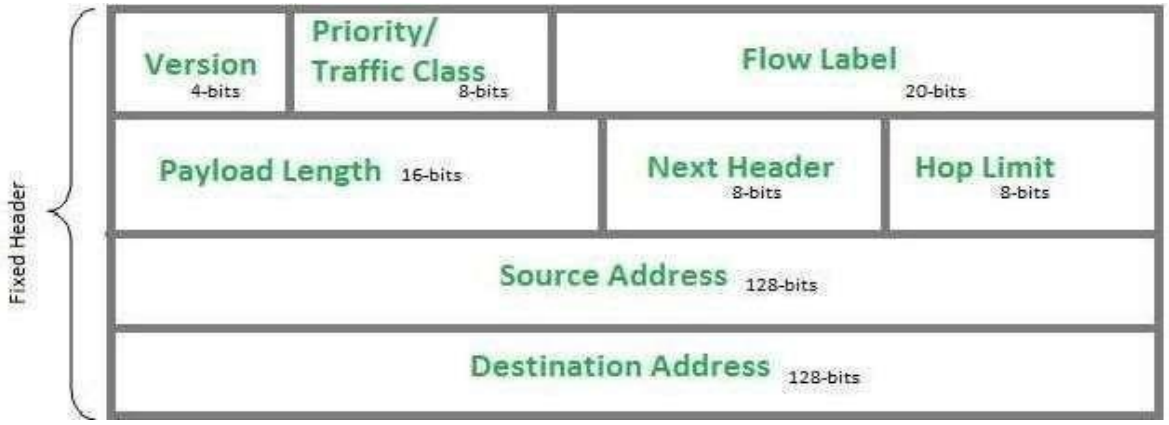
Source port : 59782	Destination port : 1900
Length : 145	Checksum : 0x6ae1 [unverified]
Data : —	

```
> Frame 134: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on interface \Devi
> Ethernet II, Src: HewlettPacka_49:07:d1 (f4:39:09:49:07:d1), Dst: IPv4mcast_7f:ff:fa (01:00
> Internet Protocol Version 4, Src: 192.168.0.55, Dst: 239.255.255.250
v User Datagram Protocol, Src Port: 59782, Dst Port: 1900
    Source Port: 59782
    Destination Port: 1900
    Length: 145
    Checksum: 0x6ae1 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 11]
  > [Timestamps]
    UDP payload (137 bytes)
```

IP HEADER




```
> Frame 133: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_
> Ethernet II, Src: HP_1a:60:a0 (04:0e:3c:1a:60:a0), Dst: IPv6mcast_02 (33:33:00:00:00:02)
v Internet Protocol Version 6, Src: fe80::cb1f:1a73:b1a2:c851, Dst: ff02::2
    0110 .... = Version: 6
  > .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... 0000 0000 0000 0000 0000 = Flow Label: 0x00000
    Payload Length: 16
    Next Header: ICMPv6 (58)
    Hop Limit: 255
    Source Address: fe80::cb1f:1a73:b1a2:c851
    Destination Address: ff02::2
> Internet Control Message Protocol v6
```



Conculsion :

The installation of Wireshark provides users with a powerful tool for network analysis and troubleshooting. By capturing and dissecting packet headers, particularly those of TCP, IP, and UDP protocols, users can gain valuable insights into network traffic patterns, identify potential issues, and optimize network performance.

Wireshark's intuitive interface and extensive protocol support make it a valuable asset for both novice and experienced network professionals. Through the analysis of packet headers, users can observe crucial information such as source and destination addresses, packet lengths, protocol types, and more, enabling them to diagnose network problems with precision.

LO5: Execute and evaluate network administration commands and demonstrate their use in different network scenarios

ASSSIGNMENT 7

AIM: Analysis of Packet header: TCP,UDP and IP using TCP

Theory:

TCPDUMP:

Tcpdump is a command line utility that allows you to capture and analyze network traffic going through your system. It is often used to help troubleshoot network issues, as well as a security tool.

A powerful and versatile tool that includes many options and filters, tcpdump can be used in a variety of cases. Because it's a command-line tool, it is ideal to run in remote servers or devices for which a GUI is not available to collect data that can be analyzed later. It can also be launched in the background or as a scheduled job using tools like cron.

TCPDUMP INSTALLATION:

Tcpdump is included with several Linux distributions, so chances are, you already have it installed. Check whether tcpdump is installed on your system with the following command:

$ which tcpdump

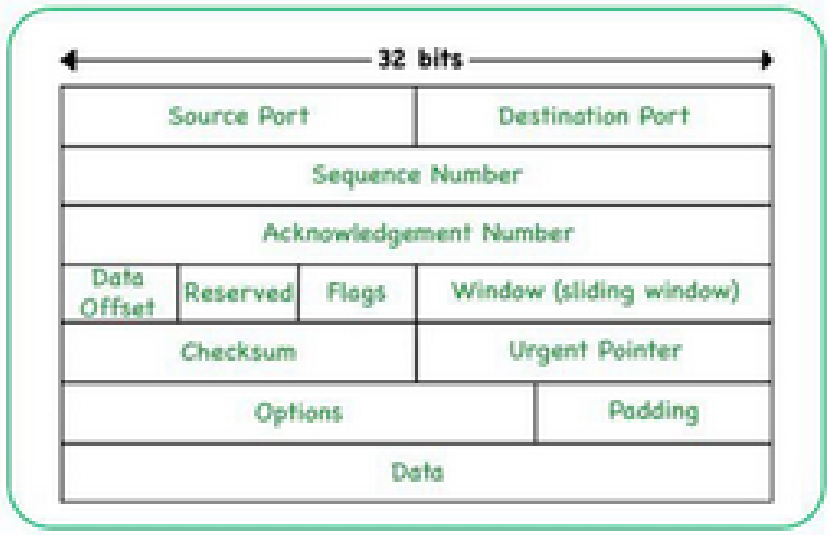
/usr/sbin/tcpdump

If tcpdump is not installed, you can install it but using your distribution's package manager. For example, on CentOS or Red Hat Enterprise Linux, like this:

$ sudo dnf install -y tcpdump

Tcpdump requires `libpcap`, which is a library for network packet capture. If it's not installed, it will be automatically added as a dependency.

TCP HEADER:



OPTIONS:

- `-i <interface>`: Specify the network interface to capture packets from (e.g., `eth0`, `wlan0`).
-

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any
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
11:40:11.748363 IP 192.168.0.228.netbios-ns > 192.168.0.255.netbios-ns: UDP, length 50
11:40:11.749567 IP localhost.39807 > localhost.domain: 51201+ [1au] PTR? 255.0.168.192.in-addr.arpa. (55)
11:40:11.750300 IP localhost.domain > localhost.39807: 51201 NXDomain 0/0/1 (55)
11:40:11.751390 IP localhost.35289 > localhost.domain: 16520+ [1au] PTR? 228.0.168.192.in-addr.arpa. (55)
11:40:11.752525 IP localhost.45392 > localhost.domain: 8309+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
11:40:12.486386 IP 192.168.0.228.netbios-ns > 192.168.0.255.netbios-ns: UDP, length 50
^C
6 packets captured
14 packets received by filter
4 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any tcp
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
11:40:23.585880 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [P.], seq 7742836
03:774284234, ack 2779811978, win 909, options [nop,nop,TS val 2515101084 ecr 2956672836], length 631
11:40:23.585899 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [.], seq 631:2031
, ack 1, win 909, options [nop,nop,TS val 2515101084 ecr 2956672836], length 1400
11:40:23.585900 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [.], seq 2031:343
1, ack 1, win 909, options [nop,nop,TS val 2515101084 ecr 2956672836], length 1400
11:40:23.585901 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [P.], seq 3431:45
16, ack 1, win 909, options [nop,nop,TS val 2515101084 ecr 2956672836], length 1085
11:40:23.588955 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [.], ack 631, win
 542, options [nop,nop,TS val 2956705402 ecr 2515101084,nop,nop,sack 1 {3431:4516}], length 0
11:40:23.588963 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [.], ack 631, win
 542, options [nop,nop,TS val 2956705402 ecr 2515101084], length 0
11:40:23.588964 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [.], ack 2031, wi
n 537, options [nop,nop,TS val 2956705402 ecr 2515101084,nop,nop,sack 1 {3431:4516}], length 0
11:40:23.588964 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [.], ack 4516, wi
n 528, options [nop,nop,TS val 2956705402 ecr 2515101084], length 0
11:40:23.720062 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [P.], seq 1:548,
ack 4516, win 544, options [nop,nop,TS val 2956705533 ecr 2515101084], length 547
11:40:23.720083 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [P.], seq 548:719
, ack 4516, win 544, options [nop,nop,TS val 2956705533 ecr 2515101084], length 171
11:40:23.720115 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [.], ack 548, win
 921, options [nop,nop,TS val 2515101218 ecr 2956705533], length 0
11:40:23.720127 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [.], ack 719, win
 933, options [nop,nop,TS val 2515101218 ecr 2956705533], length 0
11:40:23.721107 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [P.], seq 719:750
```

1. -i host:

```
0 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i enp4s0 -c5 -nn host 192.168.0.207
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:40:34.169177 IP 192.168.0.207.33090 > 54.229.31.114.443: Flags [P.], seq 521301721:521301761, ack 35757
01472, win 501, options [nop,nop,TS val 4046622398 ecr 2985579830], length 40
11:40:34.329746 IP 54.229.31.114.443 > 192.168.0.207.33090: Flags [.], ack 40, win 425, options [nop,nop,T
S val 2985609852 ecr 4046622398], length 0
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$
```

- `-n`: Don't resolve hostnames (i.e., display IP addresses instead).
- `-v`, `-vv`, `-vvv`: Increase the verbosity of the output. More `v`s provide additional details.
- `-c <count>`: Stop after receiving `count` packets.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i enp4s0 -c5 -nn host 192.168.0.207
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:40:34.169177 IP 192.168.0.207.33090 > 54.229.31.114.443: Flags [P.], seq 521301721:521301761, ack 3575701472, win
 501, options [nop,nop,TS val 4046622398 ecr 2985579830], length 40
11:40:34.329746 IP 54.229.31.114.443 > 192.168.0.207.33090: Flags [.], ack 40, win 425, options [nop,nop,TS val 2985
609852 ecr 4046622398], length 0
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

- `-A`: Print each packet in ASCII. Useful for seeing the contents of text-based protocols like HTTP.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -A
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:47:12.097998 00:9e:1e:15:44:53 (oui Unknown) > 34:db:fd:77:e4:61 (oui Unknown), ethertype Unknown (0xa0a0), lengt
h 60:
        0x0000:  000d 0101 0101 0101 0101 0101 0101 0101  ................
        0x0010:  0101 0101 0101 0101 0101 0101 0101 0101  ................
        0x0020:  0101 0101 0101 0101 0101 0101 0101       ..............
.............................................
11:47:12.308008 IP6 fe80::7eb0:6724:3002:f89e > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s)
, length 28
`....$..........~.g$0.....................:.........b.........................
11:47:12.309746 IP6 fe80::7eb0:6724:3002:f89e > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s)
, length 28
`....$..........~.g$0.....................:.........a.........................
11:47:12.309879 IP 192.168.0.107 > 224.0.0.252: igmp v2 report 224.0.0.252
F.. ."........k..........       ...................
11:47:12.309930 IP6 fe80::7eb0:6724:3002:f89e > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s)
, length 28
`....$..........~.g$0.....................:.........b.........................
11:47:12.309932 IP 192.168.0.107 > all-routers.mcast.net: igmp leave 224.0.0.252
F.. ...........k.............................
11:47:12.310090 IP6 fe80::7eb0:6724:3002:f89e > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s)
, length 28
```

- `-q`: Show less protocol information, making the output more compact.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -q
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:47:58.407393 IP 192.168.0.104.mdns > mdns.mcast.net.mdns: UDP, length 43
11:47:58.407787 IP6 fe80::c626:187b:9aca:8498.mdns > ff02::fb.mdns: UDP, length 43
11:47:58.409692 IP lab1003-HP-280-G4-MT-Business-PC.59026 > _gateway.domain: UDP, length 44
11:47:58.413197 IP _gateway.domain > lab1003-HP-280-G4-MT-Business-PC.59026: UDP, length 103
11:47:58.417182 IP lab1003-HP-280-G4-MT-Business-PC.49048 > _gateway.domain: UDP, length 90
11:47:58.419380 IP _gateway.domain > lab1003-HP-280-G4-MT-Business-PC.49048: UDP, length 149
11:47:58.550646 ARP, Request who-has 192.168.0.118 tell 192.168.0.131, length 46
11:47:58.551355 IP lab1003-HP-280-G4-MT-Business-PC.50330 > _gateway.domain: UDP, length 44
11:47:58.553781 IP _gateway.domain > lab1003-HP-280-G4-MT-Business-PC.50330: UDP, length 103
11:47:58.555315 IP lab1003-HP-280-G4-MT-Business-PC.51371 > _gateway.domain: UDP, length 44
11:47:58.557266 IP _gateway.domain > lab1003-HP-280-G4-MT-Business-PC.51371: UDP, length 103
11:47:58.609129 IP 192.168.0.196.57399 > 239.255.255.250.1900: UDP, length 137
11:47:58.610990 IP lab1003-HP-280-G4-MT-Business-PC.58772 > _gateway.domain: UDP, length 44
11:47:58.612800 IP _gateway.domain > lab1003-HP-280-G4-MT-Business-PC.58772: UDP, length 103
11:47:58.852786 00:9e:1e:15:44:53 (oui Unknown) > 34:db:fd:77:e4:61 (oui Unknown), Unknown Ethertype (0xa0a0), lengt
h 60:
11:47:59.050628 ARP, Request who-has _gateway tell 192.168.0.131, length 46
^C
16 packets captured
16 packets received by filter
0 packets dropped by kernel
```

- `-s <size>`: Define the snapshot length; that is, the amount of data captured from each packet. The default is sufficient for most protocols but can be increased if more of the packet content is needed.
- Port :

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump tcp port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:02:28.046379 IP lab1003-HP-280-G4-MT-Business-PC.38804 > a23-212-50-219.deploy.static.akamaitechnologies.com.http
: Flags [.], ack 3019206610, win 501, options [nop,nop,TS val 1553561921 ecr 3792284797], length 0
12:02:28.049957 IP a23-212-50-219.deploy.static.akamaitechnologies.com.http > lab1003-HP-280-G4-MT-Business-PC.38804
: Flags [.], ack 1, win 506, options [nop,nop,TS val 3792295037 ecr 1553480060], length 0
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump tcp port 443
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:03:00.217563 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [P.], seq 7749002
07:774900246, ack 2779945875, win 4460, options [nop,nop,TS val 2516457715 ecr 2958004023], length 39
12:03:00.219934 IP bom12s14-in-f14.1e100.net.https > lab1003-HP-280-G4-MT-Business-PC.57662: Flags [P.], seq 1:40, a
ck 39, win 544, options [nop,nop,TS val 2958062030 ecr 2516457715], length 39
12:03:00.219961 IP lab1003-HP-280-G4-MT-Business-PC.57662 > bom12s14-in-f14.1e100.net.https: Flags [.], ack 40, win
4460, options [nop,nop,TS val 2516457718 ecr 2958062030], length 0
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel
```

- Tcpdump -r: This command will now read the captured packets from the captured_packets.pcap file.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -r captured.pcap
reading from file captured.pcap, link-type EN10MB (Ethernet)
11:51:45.973576 IP lab1003-HP-280-G4-MT-Business-PC.51130 > 221.5.120.34.bc.googleusercontent.com.https: Flags [P.],
 seq 3364695807:3364695846, ack 3158291169, win 501, options [nop,nop,TS val 2730731590 ecr 1201740100], length 39
11:51:45.976977 IP 221.5.120.34.bc.googleusercontent.com.https > lab1003-HP-280-G4-MT-Business-PC.51130: Flags [P.],
 seq 1:40, ack 39, win 1050, options [nop,nop,TS val 1201799102 ecr 2730731590], length 39
11:51:45.977011 IP lab1003-HP-280-G4-MT-Business-PC.51130 > 221.5.120.34.bc.googleusercontent.com.https: Flags [.],
ack 40, win 501, options [nop,nop,TS val 2730731594 ecr 1201799102], length 0
11:51:46.149258 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 386
11:51:46.149262 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 395
11:51:46.149389 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 458
11:51:46.149545 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 454
11:51:46.149656 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 434
11:51:46.149820 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 466
11:51:46.149940 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 448
11:51:46.150068 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 450
11:51:46.150229 IP _gateway.42955 > 239.255.255.250.1900: UDP, length 450
11:51:46.204571 ARP, Request who-has 192.168.0.118 tell 192.168.0.223, length 46
11:51:47.017421 ARP, Request who-has 192.168.0.118 tell 192.168.0.223, length 46
```

- sudo tcpdump -w: This command will now output all the captures packets in a file named as captured_packets.pcap.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -w captured.pcap -i enp4s0
tcpdump: listening on enp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C14 packets captured
14 packets received by filter
0 packets dropped by kernel
```

- sudo tcpdump -D: This command will display all the interfaces that are available in the system.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -D
1.enp4s0 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth0 (Bluetooth adapter number 0)
5.nflog (Linux netfilter log (NFLOG) interface)
6.nfqueue (Linux netfilter queue (NFQUEUE) interface)
7.usbmon1 (USB bus number 1)
8.usbmon2 (USB bus number 2)
```

CONCLUSION:

`tcpdump` is a powerful and flexible tool for capturing and analyzing network traffic. It's especially useful for troubleshooting, debugging network issues, and monitoring network behavior. By using various filters and options, you can focus on specific types of traffic, whether it's based on protocols, IP addresses, ports, or even packet content.

ASSSIGNMENT 8: LO4

Aim:

To implement Socket Programming in C/Java - TCPServer, TCPClient

Theory:

Socket Programming with TCP:
Socket Programming is a way to enable communication between two processes (applications) over a network. It provides the interface for network communication and is a fundamental aspect of networking in modern software development. In socket programming, TCP (Transmission Control Protocol) is one of the most commonly used protocols for reliable data transmission.

Key Concepts:
Socket:
A socket is a software endpoint that establishes communication between two machines. It is used for sending and receiving data over a network. A socket is identified by a unique combination of:

- **IP address**: The address of the machine.
- **Port number**: A unique identifier on the machine that helps differentiate between different services running on the same machine (e.g., web servers use port 80, FTP uses port 21, etc.).

TCP (Transmission Control Protocol):
TCP is a connection-oriented protocol that ensures reliable, ordered, and error-free communication between two computers. It provides:

- **Reliability**: TCP ensures that data is delivered accurately and in the same order it was sent.
- **Flow Control**: It manages the data transfer speed to avoid overwhelming the receiver.
- **Error Detection and Correction**: If packets are lost or corrupted, TCP ensures they are retransmitted.
- **Connection Management**: TCP establishes a connection before sending data and ensures both sides are ready to communicate.

Client-Server Model:

- **Server**: The server is a program that listens for incoming requests on a specific IP address and port. Once a client connects, the server accepts the connection and communicates with the client.
- **Client**: The client is a program that initiates a request to the server. It connects to the server using the server's IP address and port number.

Client-Server Communication Flow:

1. The **server** creates a socket and binds it to a specific port, then listens for incoming connections.
2. The **client** creates a socket and connects to the server by specifying the server's IP address and port.
3. Once the connection is established, the client and server can send and receive data in both directions.

Basic Operations in Socket Programming:

- **Create a Socket**: A socket must be created using the Socket (client) or ServerSocket (server) classes in Java.
- **Bind to a Port**: A server must bind its socket to a specific port number where it will listen for incoming connections.
- **Listen for Connections**: The server listens for client connections. Once a connection is established, it accepts the connection.
- **Data Exchange**: After the connection is established, both the server and the client can send and receive data using input/output streams (e.g., InputStream, OutputStream, BufferedReader, PrintWriter).
- **Close the Connection**: After the communication is complete, the server and client close their respective sockets to release resources.

TCP Socket Programming Steps:

1. **Server Side**:
    - **Create a ServerSocket object**: The server creates a ServerSocket bound to a specific port.
    - **Wait for client connections**: The server listens for client requests on the specified port.
    - **Accept the connection**: When a client connects, the server accepts the connection, creating a new socket for communication.
    - **Communicate**: After the connection is established, the server communicates with the client by sending and receiving data.
    - **Close the connection**: Once the communication is over, the server closes the socket.
2. **Client Side**:
    - **Create a Socket object**: The client creates a socket and connects to the server using the server's IP address and port number.
    - **Send data to the server**: The client sends data (e.g., a request) to the server.
    - **Receive data from the server**: The client waits for a response from the server and processes it.
    - **Close the connection**: After communication is finished, the client closes the socket

Advantages of TCP Socket Programming:

1. **Reliable**: TCP guarantees the delivery of data packets in the correct order.
2. **Error Handling**: TCP automatically handles retransmission of lost packets and ensures error-free transmission.
3. **Flow Control**: Helps manage the speed of data transfer between client and server to avoid network congestion.
4. **Connection-Oriented**: TCP provides an established connection, ensuring that the client and server are ready to communicate.

Use Cases for TCP Socket Programming:

- **Web Servers**: TCP is the backbone of HTTP (HyperText Transfer Protocol) for communication between web browsers (clients) and web servers.
- **FTP Servers**: File Transfer Protocol (FTP) uses TCP for reliable file transfer.
- **Chat Applications**: TCP is used for real-time messaging between clients and servers.
- **Remote Access Systems**: Services like SSH (Secure Shell) and RDP (Remote Desktop Protocol) rely on TCP for secure and reliable communication.

Program and it's output:

TCPServer.java

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

public class TCPServer {
   public static void main(String[] args) {
      try {
         // Create a ServerSocket to listen on port 1234
         ServerSocket serverSocket = new ServerSocket(1234);
         System.out.println("Server is listening on port 1234...");

         // Wait for client connection
         Socket clientSocket = serverSocket.accept();
         System.out.println("Client connected: " + clientSocket.getInetAddress().getHostAddress());

         // Get input and output streams
         BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
         PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

         // Ask for first number
         out.println("Please enter the first number:");
         String num1String = in.readLine();
```

```
        // Ask for second number
        out.println("Please enter the second number:");
        String num2String = in.readLine();

        try {
            // Parse the numbers and perform multiplication
            double num1 = Double.parseDouble(num1String);
            double num2 = Double.parseDouble(num2String);
            double result = num1 * num2;

            // Send the result back to the client
            out.println("Result of multiplication: " + result);
        } catch (NumberFormatException e) {
            out.println("Error: Invalid number format. Please enter valid numbers.");
        }

        // Close the connections
        in.close();
        out.close();
        clientSocket.close();
        serverSocket.close();

    } catch (IOException e) {
        System.err.println("Error: " + e.getMessage());
    }
  }
}
```

TCPClient.java

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

public class TCPClient {
  public static void main(String[] args) {
    try {
      // Connect to the server running on localhost and port 1234
      Socket socket = new Socket("localhost", 1234);
      System.out.println("Connected to server...");

      // Get input and output streams
      BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
```

```
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

        // Get user input for the two numbers
        BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));

        // Receive message asking for the first number
        System.out.println(in.readLine());
        String num1 = userInput.readLine(); // Read first number
        out.println(num1); // Send first number to server

        // Receive message asking for the second number
        System.out.println(in.readLine());
        String num2 = userInput.readLine(); // Read second number
        out.println(num2); // Send second number to server

        // Read the server's response
        String serverResponse = in.readLine();
        System.out.println("Server response: " + serverResponse);

        // Close the connections
        in.close();
        out.close();
        userInput.close();
        socket.close();

      } catch (IOException e) {
        System.err.println("Error: " + e.getMessage());
      }
    }
}
```

Output:

```
C:\Users\Lab1003\Desktop\Shreya_55>javac TCPServer.java

C:\Users\Lab1003\Desktop\Shreya_55>java TCPServer
Server is listening on port 1234...
Client connected: 127.0.0.1
```

```
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lab1003\Desktop\Shreya_55>javac TCPClient.java

C:\Users\Lab1003\Desktop\Shreya_55>java TCPClient
Connected to server...
Please enter the first number:
1
Please enter the second number:
2
Server response: Result of multiplication: 2.0

C:\Users\Lab1003\Desktop\Shreya_55>
```

Conclusion:

TCP socket programming facilitates reliable data exchange between a client and server over a network. Using the client-server architecture and the TCP protocol ensures secure, ordered, and error-free communication. Java's `ServerSocket` and `Socket` classes make it easy to implement network-based applications, such as web servers and file transfer systems, which require stable and dependable connections.

ASSIGNMENT 8 : LO4

Aim:
To implement Socket Programming using UDP in Java - UDPServer, UDPClient
Theory:

UDP (User Datagram Protocol) is a connectionless protocol in the Transport layer of the OSI model. Unlike TCP (Transmission Control Protocol), UDP does not establish a connection before sending data and does not guarantee message delivery. It's faster than TCP because it avoids the overhead of connection management and acknowledgment, but it may lose packets or deliver them out of order.

In Java, we can implement UDP socket programming using the `DatagramSocket` and `DatagramPacket` classes.

Key Components of UDP Socket Programming in Java

1. **DatagramSocket**:
    - This class is used for both sending and receiving UDP packets.
    - A `DatagramSocket` listens on a specified port and sends/receives datagrams (packets).
2. **DatagramPacket**:
    - This class represents a packet of data that can be sent or received. It contains information such as:
        - The data (in the form of a byte array).
        - The length of the data.
        - The destination address (IP address).
        - The port number on the destination machine.

Key Steps in UDP Socket Programming in Java

1. **Create a DatagramSocket**:
    - On the server side, you create a `DatagramSocket` that listens for incoming packets on a specific port.
    - On the client side, you create a `DatagramSocket` to send the packets to the server.
2. **Create a DatagramPacket**:
    - On the client side, you create a `DatagramPacket` that contains the data (e.g., a message or number) you want to send, along with the destination address and port.

    - On the server side, you create a `DatagramPacket` to receive incoming data.
3. **Send and Receive Data**:
    - The client sends a packet to the server using `DatagramSocket.send()`.
    - The server receives the packet using `DatagramSocket.receive()`.
    - The server then processes the data and sends a response back to the client.
4. **Process Data**:
    - The server processes the received data, such as performing a mathematical operation (e.g., addition or multiplication) or simply printing the data.
    - The server sends a response back to the client, which can be another packet.
5. **Close the DatagramSocket**:
    - Both the client and server should close their `DatagramSocket` objects when they are done to release system resources.

When to Use UDP:

UDP is generally used when:

- **Speed is more important** than reliability (e.g., streaming video, live audio, online gaming).
- **Real-time communication** is required, and occasional data loss can be tolerated.
- **Broadcasting or multicasting** is needed, such as in network management or IPTV.

When to Avoid UDP:

UDP should generally be avoided when:

- **Data reliability is critical** (e.g., file transfers, financial transactions).
- **Packet ordering** is important (e.g., database synchronization, ordered messaging systems).
- **Error handling and retransmission** are required by the application.

Program:

Server:

```
import java.net.*;

public class UDPServer {

    public static void main(String[] args) {
        DatagramSocket socket = null;

        try {
            // Create a DatagramSocket to listen on port 9876
            socket = new DatagramSocket(9876);
            byte[] receiveData = new byte[1024];

            System.out.println("Server is listening on port 9876...");

            while (true) {
                // Receive the packet from the client
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                socket.receive(receivePacket);

                // Extract data from the packet
                String message = new String(receivePacket.getData(), 0,
receivePacket.getLength());
                String[] parts = message.split(",");

                if (parts.length == 2) {
                    // Parse numbers for multiplication
                    try {
                        int num1 = Integer.parseInt(parts[0].trim());
                        int num2 = Integer.parseInt(parts[1].trim());

                        // Perform multiplication
                        int result = num1 * num2;

                        // Send the result back to the client
                        String responseMessage = "Multiplication Result: " + result;
                        byte[] sendData = responseMessage.getBytes();
                        InetAddress clientAddress = receivePacket.getAddress();
```

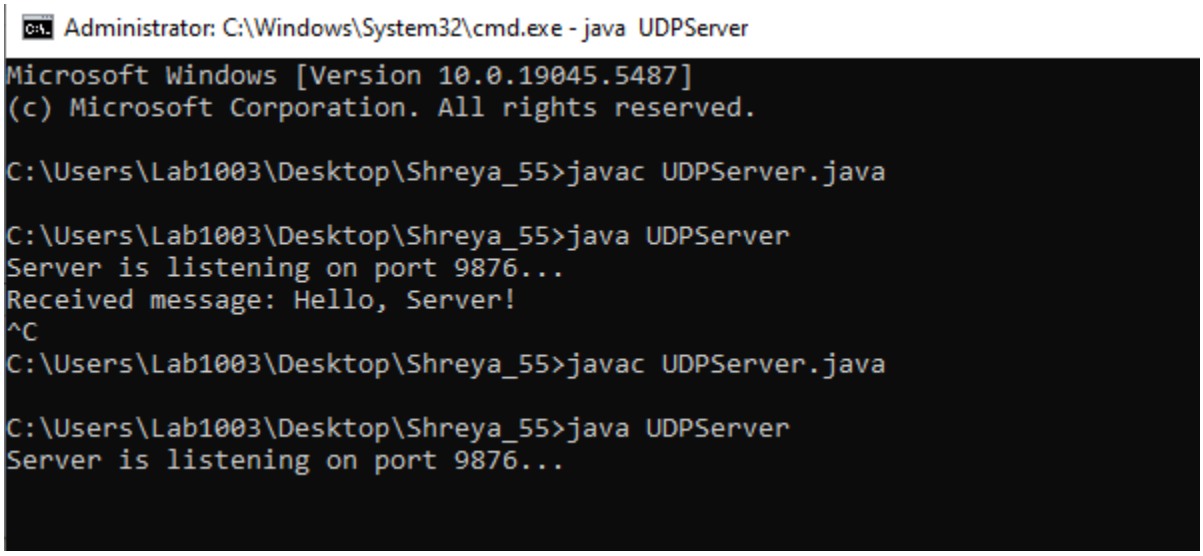
```
                int clientPort = receivePacket.getPort();
                DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
                socket.send(sendPacket);
            } catch (NumberFormatException e) {
                System.out.println("Error: Invalid numbers received.");
            }
        } else {
            System.out.println("Error: Invalid message format.");
        }
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (socket != null && !socket.isClosed()) {
        socket.close();
    }
}
}
```

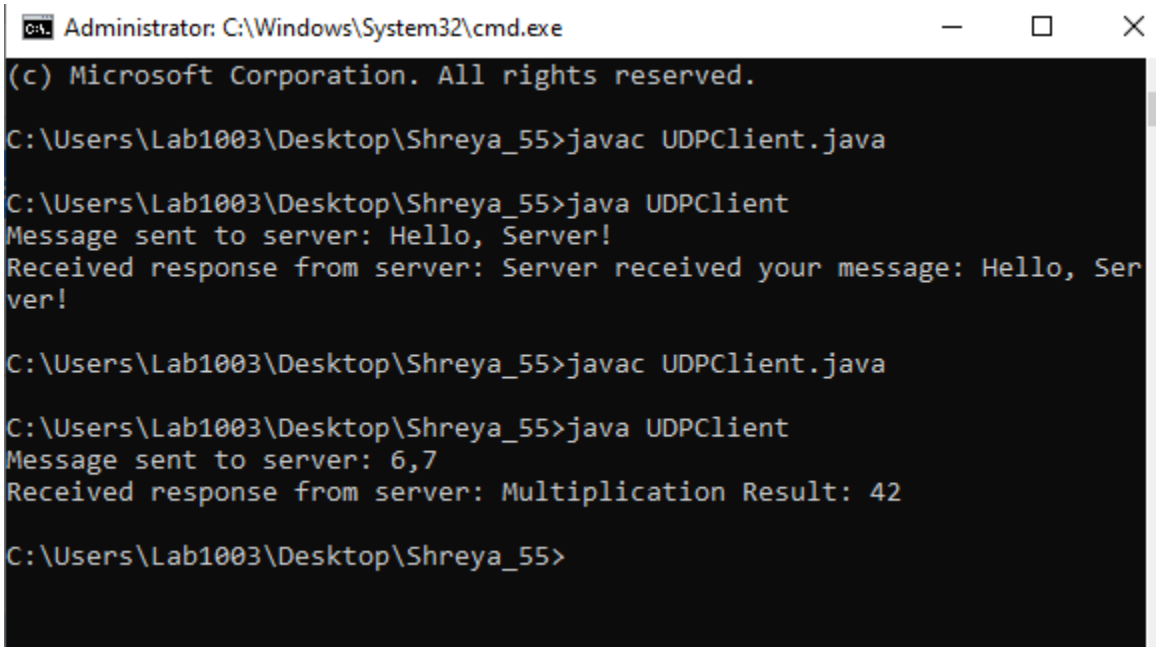
Client:

```
import java.net.*;

public class UDPClient {

    public static void main(String[] args) {
        DatagramSocket socket = null;

        try {
            // Create a DatagramSocket
            socket = new DatagramSocket();

            // Numbers to send for multiplication
            int num1 = 6;
            int num2 = 7;

            // Format the message as "num1,num2"
            String message = num1 + "," + num2;
            byte[] sendData = message.getBytes();
```

```
            // Specify the server address and port
            InetAddress serverAddress = InetAddress.getByName("localhost");
            int serverPort = 9876;

            // Create a DatagramPacket to send the message
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, serverPort);
            socket.send(sendPacket);
            System.out.println("Message sent to server: " + message);

            // Receive the response from the server
            byte[] receiveData = new byte[1024];
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            socket.receive(receivePacket);

            String responseMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("Received response from server: " + responseMessage);

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (socket != null && !socket.isClosed()) {
                socket.close();
            }
        }
    }
}
```

Output:





Conclusion:

UDP is a highly efficient and fast protocol suitable for scenarios where speed is more important than reliability. However, because it does not guarantee data delivery or order, and lacks mechanisms like flow control or error recovery, it is best suited for applications that can handle potential data loss or require real-time data transmission, like video streaming, gaming, or VoIP. For applications that require guaranteed delivery, such as file transfer or sensitive transactions, TCP would be a better choice

Assignment-10.

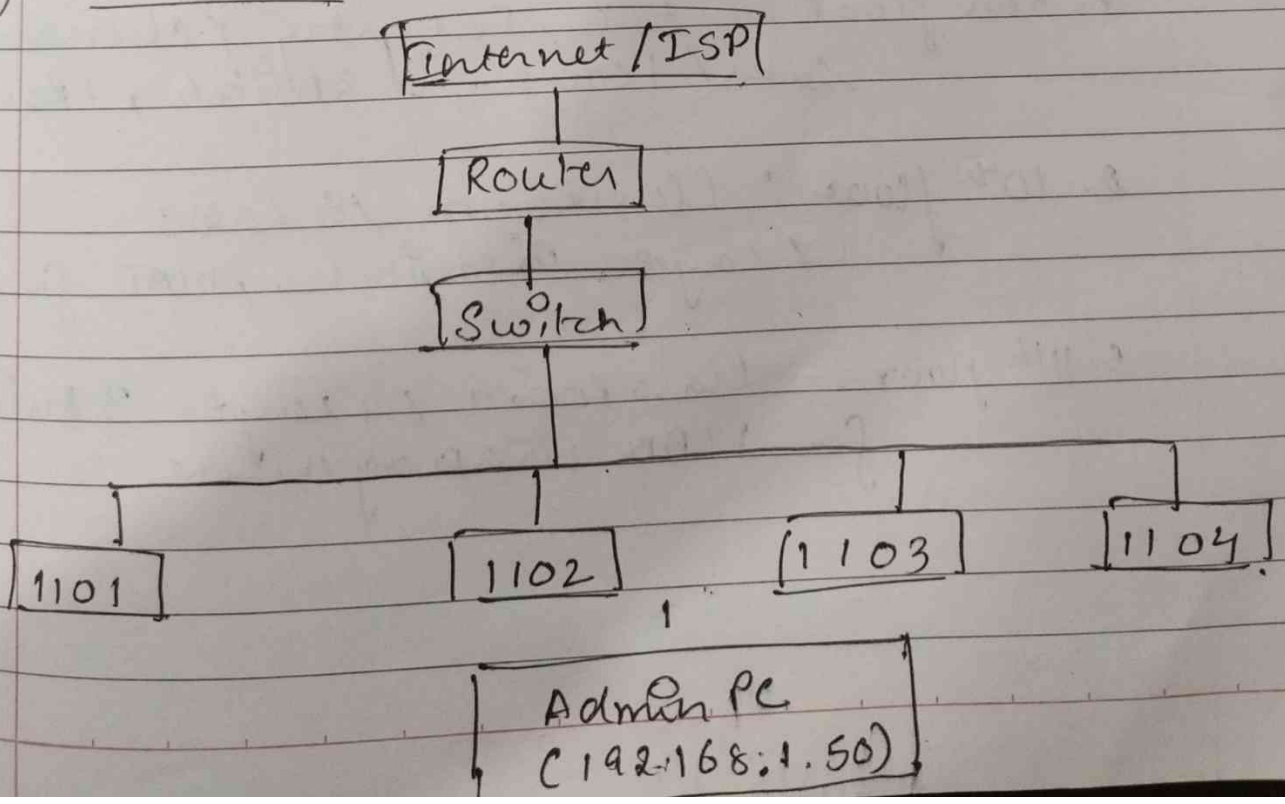
Q1] A case study to design & configure any organization Network.

objective :-
To design and configure the layout of network organization of IT Department, covering 9th, 10th & 11th floor.

Introduction :-
A network is constructed when two or more computers are connected to share information and Resources. Swapping of information is done by common convention called of Protocols over communication media.

Floor Plan :-

(i) 11th Floor :-



Layout

Floor	Room on:	
9th	903	HOD
	904, 905	faculty.
	906.	Server.
10th	1001	ClassRoom
	1002, 1003	
	1004, 1005	Labs.
	1005	
11th	1101, 1102,	ClassRooms.
	1103, 1104	

(a) Physical and Local. Network Structure

• Network devices on Each floor.

1. 9th floor - HOD, Faculty & ClassRoom, Labs (1 Layer. 3 Switch, 1 Router)

2. 10th floor: ClassRoom & 5 Labs (1 Layer 3 Switches, NAT for Labs)

3. 11th floor:- ClassRoom & 1 Layer 3 Switch for VLAN management.

Network Topology in 11th floor :-

Topology :- Star topology as all the computers are connected to central switch.

Hardware and Infrastructure.

1. Router :- Connects to the ISP & manages external Traffic

2. Switch :- Layer 2, 8-ports :- connecting projectors & admin PC.

3. Fire Wall :- Control network access.

4. Ethernet cables : CAT6 cables for wiring projectors.

5. Wireless :- Wifi connection

Shreya Malwade
S-B-SE-IT-55

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Written Assignment - 1 (CONTD)

Q1] ISO-OSI

The International Organisation for Standardization Open System Interconnection Model, is a conceptual fundamental framework used to understand and implement network communication protocols. It uses 7 layers.

1] Physical layer:-
This layer deals with the physical connection between devices. It includes the hardware technologies, such as cables, switches, & the electrical signals that transmit data.

2] Datalink Layer.
This layer is responsible for node to node data transfer and error detection and correction. It includes protocols like Ethernet & PPP. It also manages how data packets are placed on the physical medium.

3] Network layer
This layer handles the routing of the data packets across network. It determines the best path for the data to travel from source to the destination. Protocols like IP.

operates at this layer.

4] Transportation Layer:
Layer ensures complete data tra
and Error recovery. It manages
end to end communicatn & flow
control.
Protocols such as TCP & UDP are found
here.

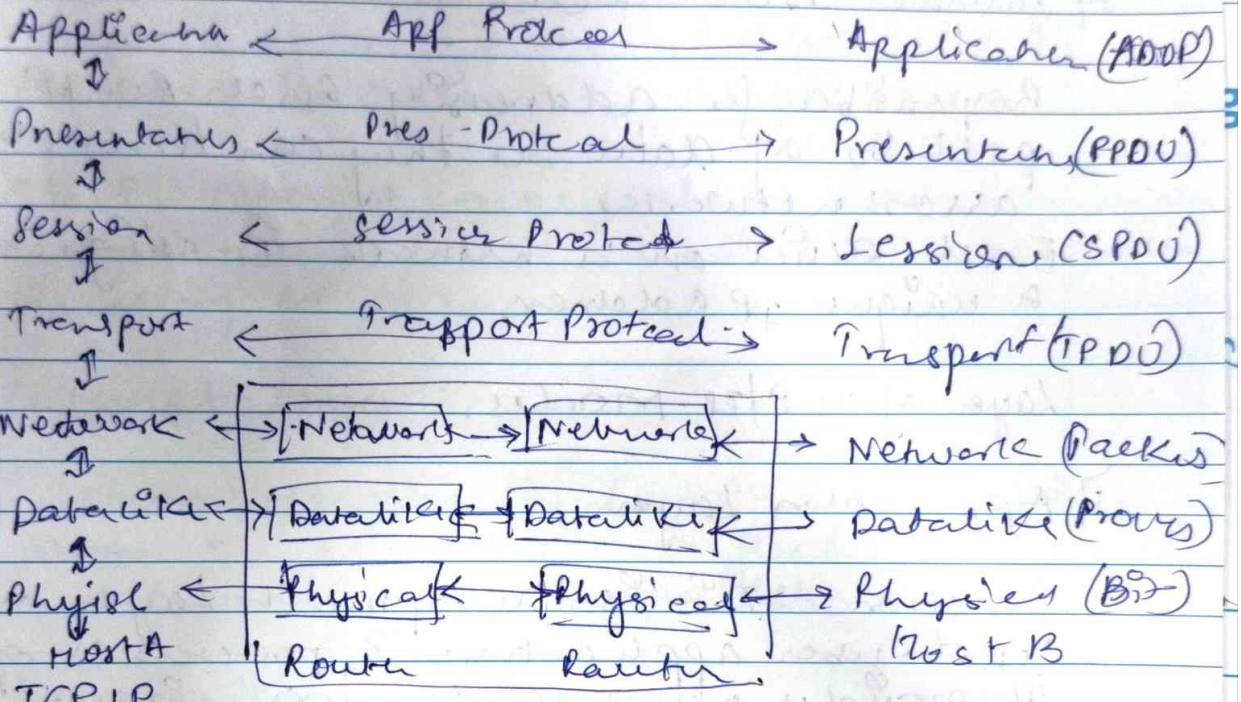
5] Session Layer
This layer manages sessions or connec
between applications. It establishes
maintains, terminates connections
ensuring that data is properly
synchronised and organized

6] Presentation Layer.
This layer translates data between
the application layer and the network
It is responsible for data formatting
encryption, & compression, ensuring
that data is in usable format
for the application layer

7] Application Layer:-
This is the topmost layer that
interacts directly with end-user
Application. It provides network servi
to applications & includes protocol like

HTTP / FTP & SMTP

Diagram.



Q2] TCP IP

Which stands for ~~Transport~~ Transmission Control Protocol, is a set of communication protocols used for the Internet and similar networks. It is the fundamental technology that enables devices to communicate over the Internet. Here's brief overview

Key protocols.

1. Transmission Control Protocol
Ensures Reliable transmission of data between devices. Break down messages

into destination in order without error. Provides flow control and congestion

2) Internet Protocol (IP)

Responsible for addressing and routing packets of data so they can travel across networks
Each device on a network is assigned a unique IP address.

Layer of TCP/IP Model.

1) Application Layer

function :- This is the topmost layer where end-user applications & processes operate. It provides network services directly to user applications

Protocols :- (i) HTTP/HTTPS :- used for web browsing
(ii) FTP :- File Transfer Protocol for transferring files
(iii) DNS :- Domain Name system

2) Transport Layer :-

Function Responsible for end to end comm. ensuring complete data transfer, error

recovery & flow control.

Protocol:-

- TCP: Provides reliable, connection-oriented communication. It ensures that data is delivered in order and without errors.

- UDP:- Offers a connectionless communication method that is faster but does not guarantee delivery or order.

3. Internet Layer:-

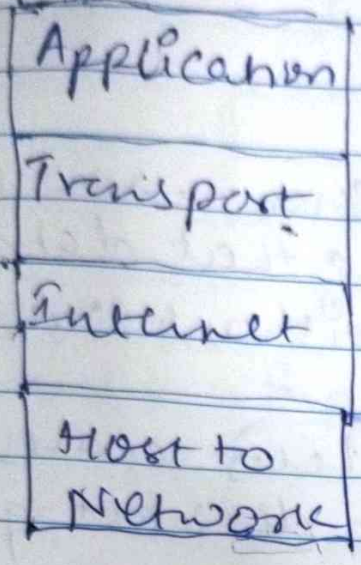
Function: Handles the routing of packets across the network. It defines the addressing and routing of data packets.

Protocols:-

- IP (Internet Protocol)
  Primary protocol for addressing & routing
  It includes
  IPV4 & IPV5

4) Link Layer or Host to network Layer:-

Function Responsible for the physical transmission of data over the network hardware. It deals with protocols that operate on the local network



Q2] Types of topologies:-

1] Bus Topology

All devices are connected to a single central cable, known as the bus
Data is transmitted in both directions along the bus.
Simple and cost-effective for small networks one can suffer from collisions and is difficult to trouble shoot

Advantage	Disadvantage
• Cost efficient	• Limited Cable lenght
• Easy to Install	• Collisions.

3] Ring Topology

- Each device is connected to two other devices, forming a circular pathway for data
- Data travels in one direction.
- Can be efficient but is vulnerable to a single point of failure

Advantage
- Orderly Data Transmission
- Predictable Performance

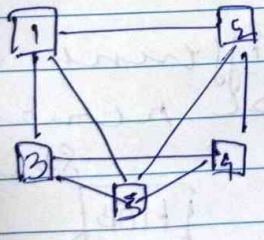
Disadvantage
- Single Point of failure
- Difficult Trouble shooting

4] Mesh Topology :-

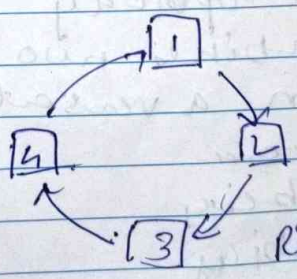
Every device is connected to every other device, providing multiple paths for data. Highly Reliable and fault tolerant.

Advantage
High Reliability

Disadvantage
Complexity



mesh



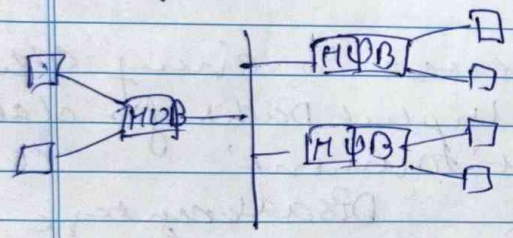
Ring

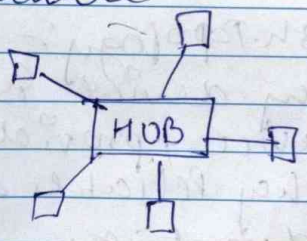
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

2] Star Topology
All devices are connected to a central hub or switch
Each device has a dedicated connection to the hub.
Easy to install and manage, if one connection fails it does not affect the others. However if the hub fails the entire network goes down.

Advantage
- Easy to mange
- fault Insolation

Disadvantage
- Central Point of failure

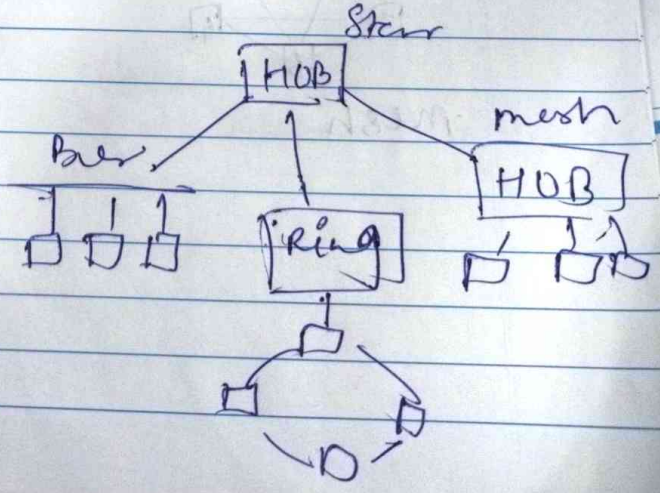




Hybrid Topology
It combines two or more different topology to form a versatile a scalable network.

Advantage
1 flexibility
2 Scalability
3 Resilience

Disadvantage
1. Complexity
2. Cost
3. Maintenance



Shreya Malwade

Shreya Malwade
SE-IT-S13-55

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Assignment NO.-2

Q5] Explain Vlan. and Vpn.

Ans (A) VLAN is a Logical Grouping of Devices on a network that allows them to communicate on a same physical Network, Regardless of the actual physical location. VLAN is used to segment network for various reasons, improving performance, security & better management of the network resources.

Key Concepts :-

1. Logical segmentation.

The VLANs allows the network administrators to have seperate segments ~~of~~ Broadcast Domain VLANs in the single physical Network. i.e different VLANs cannot communicate directly without a router or Layer 3 device.

2. Broadcast Control :-

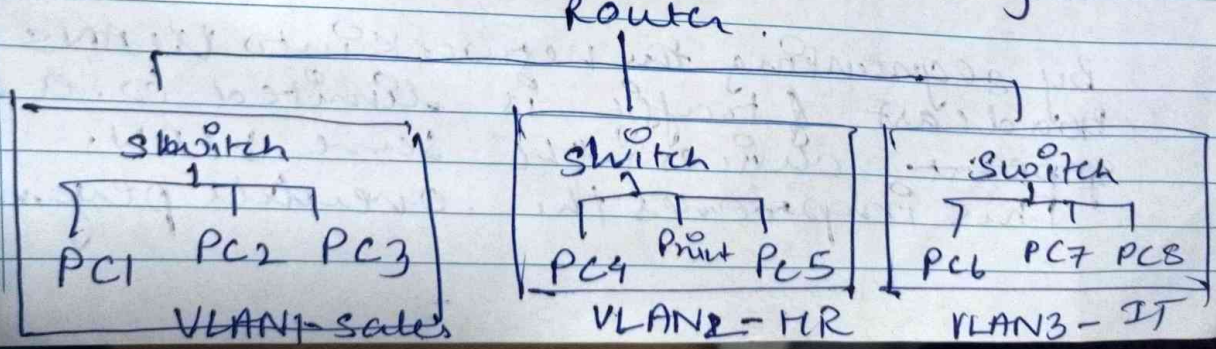
By segmenting the network into VLANs, Broadcast traffic is limited to a devices with same VLAN. This improves the overall peformance

3. Security:-
VLAN seperates / isolates the sensitive data & devices.

4. flexibility & scalability
Devices can move between VLAN without physically reconfiguring the network. This is easier to scale a growing organizations.

Types of VLAN:-
Data Vlan, Voice VLAN, Management VLAN & Native VLAN.

Example:- Considering company have 3 departments i.e IT, HR & Sales. Instead of having them placed in a separate physical Network, the computers are in VLAN placed such that they cannot access the information each other directly.

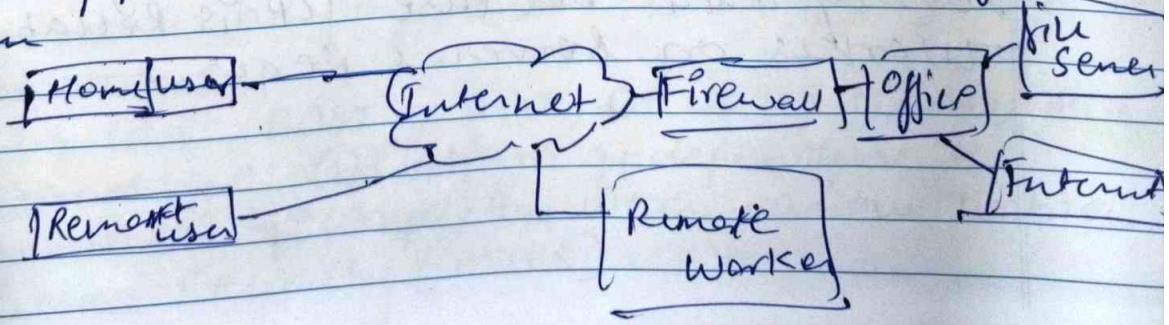


(B) VPN (Virtual Private Network) is technology that creates a secure & encrypted connection over a less secure network, such as internet. VPNs are used to protect private web traffic from snooping, interference & censorship. They are commonly employed by businesses to allow remote employees to securely access the company's internal network.

VPNs use protocols like IPsec, Open VPN & SSL/TLS to protect data from hackers & surveillance.
VPN uses a "tunnel" to transfer data securely.

Example :- An employee working from home uses VPN to access the internal Network of the company to securely. This creates a secure, encrepted tunnel to access the internal files & applications as if they were in office.

Diagram



Q4) HTTP :-

It is defined as How the client and server Internet with each other to get the web pages from the web.

HTTP Client send a Request ; an HTTP Server Returns a Reponse.

Server Port No. is 80, and client uses temenary port No.

HTTP uses services of TCP ~~on the~~ i.e connection oriented and Reliable. which means before transaction between the client and the server can take place, a connection needs to be established between them.

After a transaction, the connection between them is terminated.

~~TCP~~ The client & server, do need to worry about the error messages and loss of data because TCP is Reliable.

Works on Several Request and Respon

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

* Two ways of connection:-

1. Persistent connection	Non Persistent connection
TCP connection opened to server.	TCP connection is opened.
Multiple objects can be sent over single TCP connection between client & the server.	At most a single object is sent over TCP connection.
The connection is closed	The TCP connection is closed.

features.

1] stateless:- each Request is independent
2] Text-based: They readable Request/Respond format.
4] Mode:- client-server:- The client sends a ~~Request~~ HTTP Request to a server, which then process the Request and sends back to the client response.
5] Request methods:- indicate the desired action to be performed on the specified Resource.
Eg GET, POST, PUT, DELET, HEAD, OPTION
Advantage:- ① Platform Independent & compatable & compatible with various Webter
② Ensures Security (HTTPS).

(b) FTP (file Transfer Protocal) :- is a standard network Protocal to transfer files between client and server over a TCP/IP network. It is used to upload and downloading files to and from web servers, as well as sharing files between computers. FTP transfers text, binary or images files unlike HTTP which deals with web resources.

Key features :-

(1) Client - server model :- Client establish a connection to the server to request files transfer.

(2) Two modes of Operation

- Active mode :- The client opens a Random port and inform a server to connect back to that port for data transfer. - Problem can't work with firewall
- Passive mode :- The server opens the Random Port and inform the client to connect back to the port for data transfer. more friendly to fire wall.

(3) Two type of connection

(a) Control connection (port 21) used for sending commands & receiving response

(b) Data connection (port 20) used for send files / data

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

HTTP

Disadvantage:- ① vulnerable to security threats like man in - middle.
② Requires extra mechanisms for session management as it is stateless.

Uses:- ① Browsing websites (GET Requs)
② Submitting forms (POST Requs)
③ REST API communication.

FTP:-

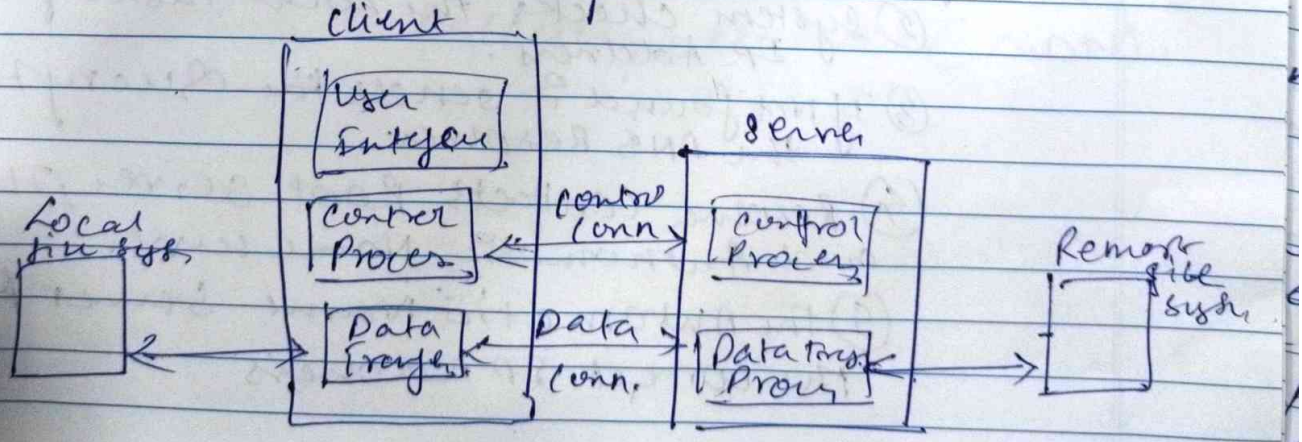
Commands: are USER, PASS, DELE, MKD, RMD, RETR.

SFTP is secure FTP. for sensitive Data Transfer.

Adv:- ① Fast and efficient
② Remote file management

Disadv:- ① Lack of security: password transmitted in plain texts.
② file size limitation of 2 GB

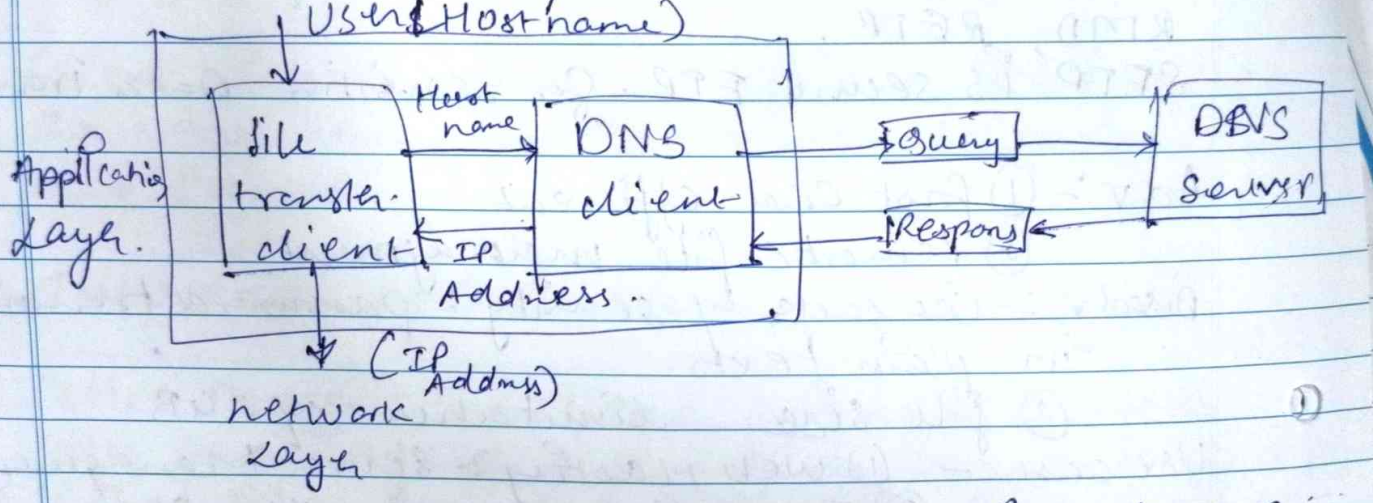
Use case:- ① Web Hosting & server Management
② Enterprises. Internal file sharing & Backup.



THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

DNS (Domain Name System).

It is a hierarchal Decentralized naming system used to translate human Readable domain name (Eg. www.google.com) into IP address (192.266.2.1) This system is essential for the functionality of the internet as it allows users to access websites & services using easy to Remember names instead of numerical IP address.



Working ① User types a website URL in a browser
② System checks the local cache for the IP Address.
③ If Not found it sends the Query to the DNS Resolver
④ Resolver contacts Root server, TLD server and Authomatic Name server
⑤ The Automatic Name Server returns the correct IP address

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

(1) Brower connects to the website server & loads the page.

DNS cashe speeds up internet acces by storing resolved IP address locally Reducing the need for repeated DNS lookeyps.

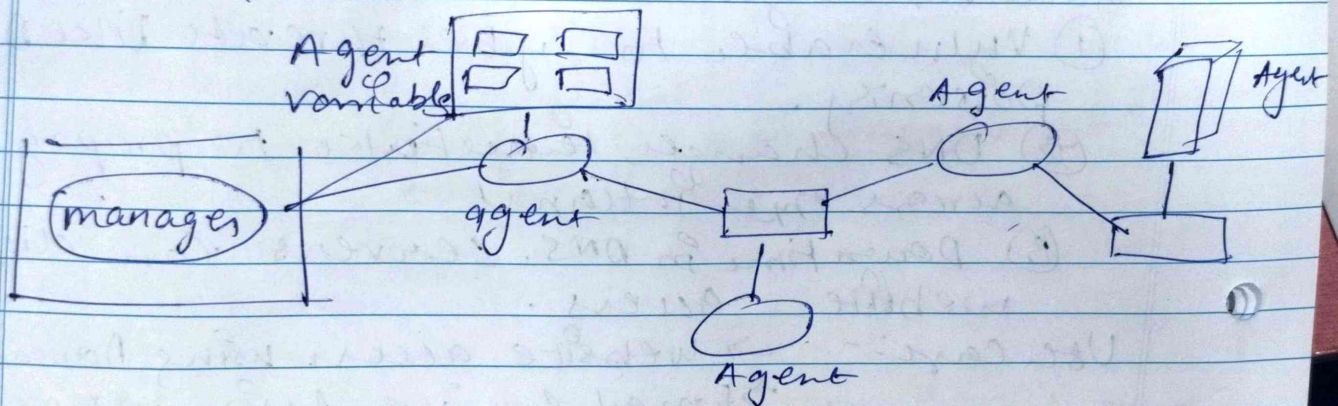
Advantages :-
→ Making Browsing easier.
→ DNSSEC is used for security to prevent Spoofing attacks.

Disadvantages :-
(1) Vulnerable to cyber threats like DNS poisoning.
(2) DNS changes take time to propogate across the internet.
(3) Downtime in DNS. servers can disrupt website acces.

Use Case :- → website access using Domain Name
→ Email Service using MX records
→ Loading balance & content Delivery network (CDN).

(d) SNMP (Simple Network management Protocol) used to manage and monitor network devices and systems. It enables network administrators to collect information about network performance, detect faults, and configure devices remotely. SNMP is essential tool for network management, providing a standardized framework for communication between devices and management systems.

Architecture



(i) Managed Devices :- The Network Devices are managed and monitored using SNMP Eg Printer, server.

(2). SNMP Agent :- A software component which ~~is~~ Running on managed devices that collects and stores information about the device status & performance

(3). SNMP Manager :- A centralized network management system that communicates

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

with SNMP agents to collect Data, send commands, & monitor the network

Messages:- (1) GET Request: Retrieves data from an agent

(2) SET Request: Set a value on an agent

(3) Trap:- Sent by the agent when fault occurs.

(4) Response: Confirms request data or changes.

Advantages:- 1) Real time network monitoring
2) Configurable Remote devices.
3) Detects network failures.

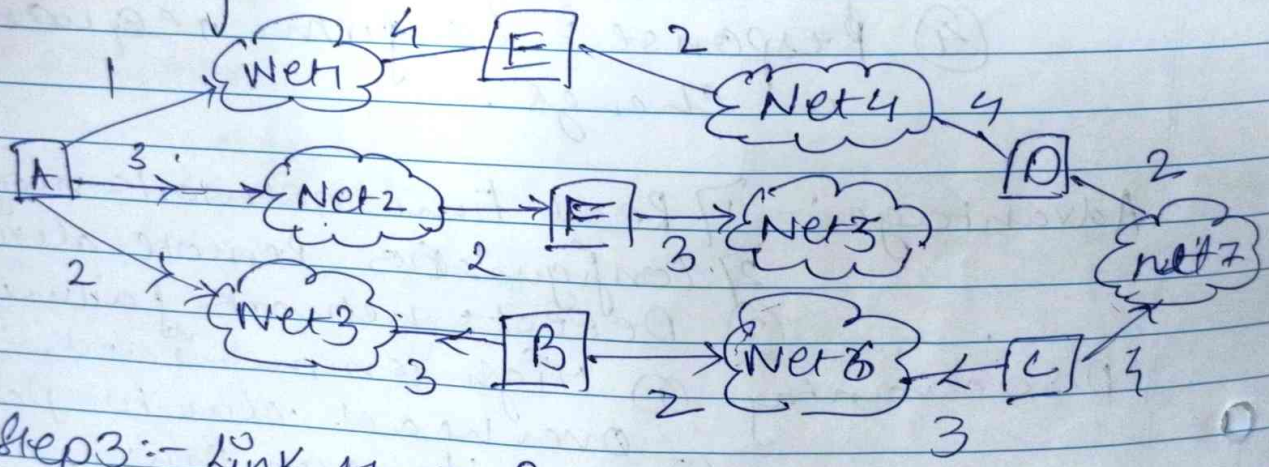
Disadvantage (1) High communication overhead due to polling
(2) Security concerns.
(3) Limited scalability.

Q1] Link State Routing:- A protocol used by the computer network to know the best path for the packets to travel from source to destination. Unlike, the DVR, that relies on the neighbours Router, LSR maintains its complete view of the topology. This helps make informed decisions about the best path can be taken.

Working Principle

Step1:- Information Sharing
Discover its neighboors and learn the address. This is done by Router sending the information to every other router.

Step2:- Measuring Packet Cost.
Measuring the delay or cost to each of its neighboors



Step3:- Link State Packet:-

A small packet containing the information sent by all the Routers is Required as Link State Packet (LSP)

The Packet telling all it has just learned

Step4:

Advertise	Network	cost	Neighbor
A	1	1	E
A	2	3	F
A	3	2	B

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Step 4: Flooding of LSP.

After creating LSP, every router forwards it to each and every other router in the Internet.

Step 5:- Compute the shortest path tree

Using the LS Database calculate the shortest path.

Advertiser	Network	Cost	Neighbor
A	1	1	E
A	2	3	F
A	3	2	B
B	3	3	A
B	6	2	C

Advantages:-
① Fast convergence. ② Loop free Routing.
③ Efficient for Large networks.

Disadvantage
Requires more memory, CPU & Bandwidth for processing.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Distance vector Routing.

It is based on principle of sharing information about the distance to. Reach various network destination amongst neighbouring routers.

working

→ Step1:- Each Router prepare its own routing table. By this. Local knowledge each Routing table know,
(1) All the Routers present in the network
(2). Distance to its ~~adjacent~~ neighbouring Routers.

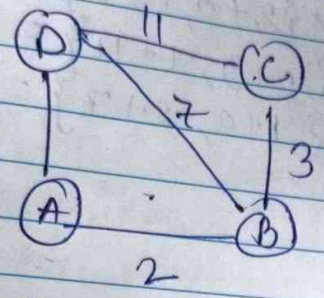
Step2:- Each router ~~sha~~ exchange its Distance vector with its neighbouring routers.
Each Router prepare a new routing table using the distance vectors it has obtained.

~~If~~ This step Repeats for (n-2) times.
~~Ha~~ n is the No. of Routers.
After this, Routing table become stable.
Algorithm used is Bellman-Ford. Algorithm
Used protocol is RIP.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Example

A Router connected to a Network



Step 1:- Prepare a Local Routing table.

At Router A:-

Dest	Dist	NextHop
A	0	A
B	2	B
C	∞	—
D	1	D

At Router B:-

Dest	Dist	Next hop
A	2	A
B	0	B
C	3	C
D	7	A

At Router C:-

Dest	Dis	Next Hop
A	∞	—
B	3	B
C	0	C
D	11	D

At Router D:-

Dest	Dist	next hop
A	1	A
B	7	B
C	11	C
D	0	D

Step 2:- Exchange the Dist vector ~~table~~ obtained in step-01 with its neighbors. New Routing table is formed.

Router A

Way Reaching B from A {2+0, 1+7} = 2 via B
C from A {2+3, 1+11} = 5 via B
D from A {1+0, 2+7} = 1 via D

Dest	Dist	NextHop
A	0	A
B	2	B
C	5	B
D	1	D

ical for all other

vantages.
easy to implement
ow process overhead.
ood for small Network

Disadvantages.
→ Slow convergence
→ Suffer from infinite count problem.
→ Use UDP at Transport Layer

S-12 19 Mar

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Q17 Explain NAT mechanism:-

Ans Def:- Network Address Translation (NAT) is a technique that allows multiple devices in a private network to be used a single public IP address to access the Internet. It helps prevent IPv4 address exhaustion by translating private IP addresses to public ones.

Working:- NAT operates on a Router that connects a local (inside) network to the global (outside) network. It replaces private IP address with public one when sending data outside and does the reverse when receiving response. If the NAT runs out of public IP, packets are dropped.

→ Type of NAT

1) Static NAT:-
One-to-one mapping btwen a private and public IP address

2) Dynamic NAT
Assign private IPs to available public IPs from a pools.

3) Port Address Translation (PAT):-

Advantages of NAT
1) Conserves public IP address
2) Provides security by hiding Internal Ip address

3] Simplify network restructuring by avoiding IP renumbering

→ Disadvantages.

1] Increase processing time, causing slight delay
2] Some applications may not work properly
3] Can't interface with security protocols like IP Sec

Diagram

Devices.
(PC / Tablet / Smart phone)