Atharva Yadav Roll No :127

Batch: S23 Network Lab

Assignment No. 1 Basic Networking Commands for Windows and Linux OS.

Aim : All 17 networking commands with description and appropriate options.

1.IPCONFIG

Syntax : ipconfig

Description :

IPCONFIG stands for INTERNET PROTOCOL CONFIGURATION .

ipconfig provides information about a computer's IP address, subnet mask, default gateway, DNS servers, MAC address, and connection-specific DNS suffix. It is a command-line utility in Windows, offering details on network configuration.

```
C:\Users\hp>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

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

Ethernet adapter Ethernet 2:

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

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::fe35:7ced:3cda:6b66%21
   IPv4 Address. . . . . . . . . . . : 192.168.0.103
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Ethernet adapter Bluetooth Network Connection:

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

The ipconfig command in Windows has several options that you can use to customize its output and gather specific information. Some options include:

A.ipconfig/all

Description : Displays detailed configuration information for all network interfaces, including DNS settings, DHCP information, and more.

```
C:\Users\hp>ipconfig/all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : LAPTOP-CD6EFDOA
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : SVV.local
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : BC-E9-2F-BF-62-BB
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : ExpressVPN TAP Adapter
   Physical Address. . . . . . . . . : 00-FF-6F-1D-B7-DA
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   Physical Address. . . . . . . . . : F8-AC-65-03-B9-35
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
   Physical Address. . . . . . . . . : FA-AC-65-03-B9-34
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
   Physical Address. . . . . . . . . : F8-AC-65-03-B9-34
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::fe35:7ced:3cda:6b66%21(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.0.103(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 03 February 2024 12:03:26
   Lease Expires . . . . . . . . . . : 03 February 2024 15:26:33
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 335064165
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-26-80-BB-18-BC-E9-2F-BF-62-BB
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Bluetooth Device (Personal Area Network)
   Physical Address. . . . . . . . . : F8-AC-65-03-B9-38
   DHCP Enabled. . . . . . . . . . . : Yes
```

b.ipconfig/renew.

Description :Renews the IP address for all network interfaces.

```
C:\Users\hp>ipconfig/renew

Windows IP Configuration

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

Ethernet adapter Ethernet:

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

Ethernet adapter Ethernet 2:

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

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::fe35:7ced:3cda:6b66%21
   IPv4 Address. . . . . . . . . . . : 192.168.0.103
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Ethernet adapter Bluetooth Network Connection:

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

c.ipconfig/release

Description : Releases the currently assigned IP address for all network interfaces.

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

Windows IP Configuration

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

Ethernet adapter Ethernet:

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

Ethernet adapter Ethernet 2:

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

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::fe35:7ced:3cda:6b66%21
   Default Gateway . . . . . . . . . :

Ethernet adapter Bluetooth Network Connection:

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

d.ipcongig/release6

Description : releases the IPV6 address

```
C:\Users\hp>ipconfig/release6

Windows IP Configuration

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

Ethernet adapter Ethernet:

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

Ethernet adapter Ethernet 2:

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

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::fe35:7ced:3cda:6b66%21
   IPv4 Address. . . . . . . . . . . : 192.168.0.103
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Ethernet adapter Bluetooth Network Connection:

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

f.ipconfig/dispalydns

Description : Shows the contents of the DNS client resolver cache.

```
C:\Users\hp>ipconfig/displaydns

Windows IP Configuration

    ssl.gstatic.com
    ----------------------------------------
    Record Name . . . . . : ssl.gstatic.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 118
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.183.99


    mtalk.google.com
    ----------------------------------------
    Record Name . . . . . : mtalk.google.com
    Record Type . . . . . : 5
    Time To Live  . . . . : 108
    Data Length . . . . . : 8
    Section . . . . . . . : Answer
    CNAME Record  . . . . : mobile-gtalk.l.google.com


    Record Name . . . . . : mobile-gtalk.l.google.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 108
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 74.125.200.188
```

2.ifconfig

Description : The command ifconfig stands for interface configurator. This command enables us to initialize an interface, assign IP address, enable or disable an interface. It display route and network interface.You can view IP address, MAC address and MTU (Maximum Transmission Unit) with ifconfig command.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ ifconfig
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.141  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::4ed1:4a9b:4a19:c19a  prefixlen 64  scopeid 0x20<link>
        ether f4:39:09:49:6c:fc  txqueuelen 1000  (Ethernet)
        RX packets 118  bytes 13265 (13.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 91  bytes 13130 (13.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 200  bytes 18896 (18.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 200  bytes 18896 (18.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lab1003@lab1003-HP-280-G4-MT-Business-PC:~$
```

To find IP address of all three differently, use command ifconfig eth0 ifconfig lo ifconfig wlan0

3.nslookup

Description: nslookup is a command-line tool for querying DNS servers, retrieving information such as IP addresses or mail server details for a given domain. It is commonly used for troubleshooting DNS issues, verifying proper DNS configuration, and conducting reverse DNS lookups. Users can test connectivity and diagnose network problems by querying DNS information with nslookup in the command prompt or terminal.

A.nslookup <url>

```
C:\Users\hp>nslookup
Default Server:  UnKnown
Address:  192.168.0.1

> www.tsec.org
Server:  UnKnown
Address:  192.168.0.1

Non-authoritative answer:
Name:    tsec.org
Addresses:  3.33.130.190
          15.197.148.33
Aliases:  www.tsec.org

> www.google.com
Server:  UnKnown
Address:  192.168.0.1

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4009:829::2004
          142.250.183.196
```

B.nslookup<IP_adress>

Performs revers lookup of the ip address and returns the corresponding domain name(if available)

```
C:\Users\hp>nslookup 172.217.174.68
Server:  UnKnown
Address:  192.168.0.1

Name:    bom07s25-in-f4.1e100.net
Address:  172.217.174.68
```

4.ip

Description :Linux IP command is the newer version of the ifconfig command. It is a handy tool for configuring the network interfaces for Linux administrators. It can be used to assign and remove addresses, take the interfaces up or down, and much more useful tasks.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
                   tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm |
                   netns | l2tp | fou | macsec | tcp_metrics | token | netconf | ila |
                   vrf | sr }
       OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                    -h[uman-readable] | -iec |
                    -f[amily] { inet | inet6 | ipx | dnet | mpls | bridge | link } |
                    -4 | -6 | -I | -D | -B | -0 |
                    -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                    -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                    -rc[vbuf] [size] | -n[etns] name | -a[ll] | -c[olor]}
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$
```

5.ping

This command sends four experimental packets to the destination host to check whether it receives them successfully, if so, then, we can communicate with the destination host. But in case the packets have not been received, that means, no communication can be established with the destination host.

```
C:\Users\hp>ping www.tsec.org

Pinging tsec.org [3.33.130.190] with 32 bytes of data:
Reply from 3.33.130.190: bytes=32 time=3ms TTL=246
Reply from 3.33.130.190: bytes=32 time=11ms TTL=246
Reply from 3.33.130.190: bytes=32 time=2ms TTL=246
Reply from 3.33.130.190: bytes=32 time=2ms TTL=246

Ping statistics for 3.33.130.190:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 11ms, Average = 4ms

C:\Users\hp>ping www.google.com

Pinging www.google.com [172.217.174.68] with 32 bytes of data:
Reply from 172.217.174.68: bytes=32 time=3ms TTL=118
Reply from 172.217.174.68: bytes=32 time=3ms TTL=118
Reply from 172.217.174.68: bytes=32 time=3ms TTL=118
Reply from 172.217.174.68: bytes=32 time=4ms TTL=118

Ping statistics for 172.217.174.68:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 4ms, Average = 3ms
```

6.tracepath

It is similar to traceroute command, but it doesn't require root privileges. By default, it is installed in Ubuntu but you may have to download traceroute on Ubuntu. It traces the network path of the specified destination and reports each hop along the path. If you have a slow network then tracepath will show you where your network is weak.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ tracepath www.google.com
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              0.711ms
 1:  _gateway                                              0.612ms
 2:  no reply
 3:  no reply
 4:  no reply
 5:  no reply
 6:  no reply
 7:  no reply
 8:  no reply
 9:  no reply
10:  no reply
11:  no reply
12:  no reply
13:  no reply
14:  no reply
15:  no reply
16:  no reply
17:  no reply
18:  no reply
19:  no reply
20:  no reply
21:  no reply
22:  no reply
23:  no reply
24:  no reply
25:  no reply
26:  no reply
27:  no reply
28:  no reply
29:  no reply
30:  no reply
     Too many hops: pmtu 1500
     Resume: pmtu 1500
```

7.tracert

tracert, short for "traceroute," is a command-line utility used to trace the route that packets take to reach a destination on a computer network. It shows the sequence of routers or hops that data packets traverse from the source to the specified destination, providing information on the time it takes for each hop. tracert is valuable for diagnosing network connectivity issues and identifying bottlenecks by revealing the path and potential delays between the source and destination. To use it, enter "tracert" followed by the destination address or domain name in the Command Prompt or terminal

```
C:\Users\hp>tracert google.com

Tracing route to google.com [142.250.192.142]
over a maximum of 30 hops:

  1     1 ms     1 ms    10 ms  192.168.0.1
  2   208 ms     2 ms     3 ms  172.25.4.7
  3     8 ms     *        *     172.25.4.1
  4     *        8 ms     *     172.16.2.202
  5    11 ms     7 ms     3 ms  175.100.188.22
  6    16 ms     8 ms     7 ms  172.253.69.227
  7    12 ms     *      244 ms  142.250.238.81
  8     5 ms     4 ms     4 ms  bom12s18-in-f14.1e100.net [142.250.192.142]

Trace complete.
```

8.netstart

The netstat command is a command-line utility used to display information about network connections, routing tables, interface statistics, masquerade connections, and more on a computer. It provides details about open ports, active network connections, and listening sockets. netstat is valuable for diagnosing network issues, identifying active connections, and monitoring network activity. You can use parameters such as "-a" to display all connections and listening ports or "-n" to show numerical addresses.

```
C:\Users\hp>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49684        LAPTOP-CD6EFDOA:49685  ESTABLISHED
  TCP    127.0.0.1:49685        LAPTOP-CD6EFDOA:49684  ESTABLISHED
  TCP    127.0.0.1:49686        LAPTOP-CD6EFDOA:49687  ESTABLISHED
  TCP    127.0.0.1:49687        LAPTOP-CD6EFDOA:49686  ESTABLISHED
  TCP    127.0.0.1:49719        LAPTOP-CD6EFDOA:49720  ESTABLISHED
  TCP    127.0.0.1:49720        LAPTOP-CD6EFDOA:49719  ESTABLISHED
  TCP    127.0.0.1:49721        LAPTOP-CD6EFDOA:49722  ESTABLISHED
  TCP    127.0.0.1:49722        LAPTOP-CD6EFDOA:49721  ESTABLISHED
  TCP    127.0.0.1:49723        LAPTOP-CD6EFDOA:49724  ESTABLISHED
  TCP    127.0.0.1:49724        LAPTOP-CD6EFDOA:49723  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-CD6EFDOA:49726  ESTABLISHED
  TCP    127.0.0.1:49726        LAPTOP-CD6EFDOA:49725  ESTABLISHED
  TCP    192.168.0.103:51815    li695-222:https        ESTABLISHED
  TCP    192.168.0.103:51824    li781-4:https          ESTABLISHED
  TCP    192.168.0.103:52405    20.212.88.117:https    ESTABLISHED
  TCP    192.168.0.103:63850    52.123.168.210:https   ESTABLISHED
  TCP    192.168.0.103:64649    52.114.44.79:https     ESTABLISHED
  TCP    192.168.0.103:64655    20.198.119.143:https   ESTABLISHED
  TCP    192.168.0.103:64785    whatsapp-cdn-shv-01-bom2:https  ESTABLISHED
  TCP    192.168.0.103:64786    whatsapp-cdn-shv-01-bom1:https  ESTABLISHED
  TCP    192.168.0.103:64787    whatsapp-cdn-shv-01-bom1:https  ESTABLISHED
  TCP    192.168.0.103:64788    whatsapp-cdn-shv-01-bom2:https  ESTABLISHED
  TCP    192.168.0.103:64789    whatsapp-cdn-shv-01-maa2:https  ESTABLISHED
  TCP    192.168.0.103:64790    whatsapp-cdn-shv-02-maa2:https  ESTABLISHED
  TCP    192.168.0.103:64818    bom07s31-in-f10:https  ESTABLISHED
  TCP    192.168.0.103:64832    sl-in-f188:5228        ESTABLISHED
  TCP    192.168.0.103:64833    bom12s09-in-f10:https  ESTABLISHED
  TCP    192.168.0.103:64834    bom12s09-in-f10:https  ESTABLISHED
  TCP    192.168.0.103:64835    162.247.243.29:https   ESTABLISHED
  TCP    192.168.0.103:64841    whatsapp-chatd-edge-shv-02-bom2:https  FIN_WAIT_2
  TCP    192.168.0.103:64842    103.226.191.225:https  ESTABLISHED
  TCP    192.168.0.103:64843    bom12s18-in-f5:https   ESTABLISHED
```

9.wget

wget is a command-line utility for non-interactive downloading of files from the web. It is widely used on Unix-like operating systems, including Linux. With wget, you can retrieve files using various protocols such as HTTP, HTTPS, FTP, and FTPS. Some common use cases include downloading files, mirroring entire websites, and fetching content for automated tasks or scripts. To use wget, you typically enter a command like wget [URL] in the terminal, where [URL] represents the web address of the file you want to download

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ wget www.geeksforgeeks.com
--2024-02-02 15:57:11--  http://www.geeksforgeeks.com/
Resolving www.geeksforgeeks.com (www.geeksforgeeks.com)... 199.59.243.225
Connecting to www.geeksforgeeks.com (www.geeksforgeeks.com)|199.59.243.225|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1066 (1.0K) [text/html]
Saving to: 'index.html.1'

index.html.1                      100%[=================================================================>]   1.04K  --.-KB/s    in 0s

2024-02-02 15:57:12 (40.6 MB/s) - 'index.html.1' saved [1066/1066]
```

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ wget www.google.com
--2024-02-02 15:52:48--  http://www.google.com/
Resolving www.google.com (www.google.com)... 172.217.27.196, 2404:6800:4009:800::2004
Connecting to www.google.com (www.google.com)|172.217.27.196|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html                          [ <=>                                                ]  19.98K  --.-KB/s    in 0s

2024-02-02 15:52:48 (77.1 MB/s) - 'index.html' saved [20464]
```

10.dig

dig, which stands for Domain Information Groper, is a command-line utility for querying Domain Name System (DNS) servers. It is commonly used on Unixlike operating systems, including Linux. dig provides detailed information about DNS queries and can be used to retrieve various types of DNS records such as A (IPv4 address), AAAA (IPv6 address), MX (mail exchange), and others. It's a versatile tool for troubleshooting DNS-related issues, checking DNS configurations, and obtaining DNS information for domain names

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ dig

; <<>> DiG 9.11.3-1ubuntu1.18-Ubuntu <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 31634
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;.                              IN      NS

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Fri Feb 02 15:45:01 IST 2024
;; MSG SIZE  rcvd: 28
```

11.hostname

The hostname command is a command-line utility that provides the hostname of the current system. On Unix-like operating systems (including Linux and macOS) and Windows, using the hostname command without any options typically displays the host or computer name assigned to that system.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ hostname
lab1003-HP-280-G4-MT-Business-PC
```

```
C:\Users\hp>hostname
LAPTOP-CD6EFDOA
```

12.arp

The arp command is a network utility available on various operating systems, including Windows and Unix-like systems. It stands for Address Resolution Protocol and is used to display and manipulate the ARP cache, which is a table that maps IP addresses to MAC addresses on a local network.

The ARP command is useful for troubleshooting and verifying connectivity at the link layer of the OSI model. It helps in identifying and resolving issues related to MAC address resolution on a local network.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   10:27:f5:a9:23:47   C                     enp4s0
```

```
C:\Users\hp>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 Physical
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each ARP
                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.

C:\Users\hp>
```

13.ss

The ss command is a utility for investigating sockets in Unix-like operating systems, providing information about network connections, listening ports, and socket statistics. It is often used as an alternative to the older netstat command.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ ss
Netid      State       Recv-Q      Send-Q                                Local Address:Port                  Peer Address:Port
u_str      ESTAB       0           0                                                 * 47439                            * 43957
u_str      ESTAB       0           0                                                 * 50450                            * 49490
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 31278                            * 34918
u_str      ESTAB       0           0                                                 * 32896                            * 31949
u_str      ESTAB       0           0                                                 * 29270                            * 26521
u_str      ESTAB       0           0                            @/tmp/.X11-unix/X0 38531                            * 41184
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 32075                            * 34955
u_str      ESTAB       0           0                                                 * 31096                            * 31809
u_str      ESTAB       0           0                  /var/run/dbus/system_bus_socket 34035                            * 32105
u_str      ESTAB       0           0                                                 * 30566                            * 30567
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 32888                            * 33944
u_str      ESTAB       0           0                                                 * 29474                            * 31898
u_str      ESTAB       0           0                                                 * 30099                            * 30100
u_str      ESTAB       0           0                  /var/run/dbus/system_bus_socket 26319                            * 28027
u_str      ESTAB       0           0                  /var/run/dbus/system_bus_socket 26803                            * 19433
u_str      ESTAB       0           0                                                 * 22277                            * 20461
u_seq      ESTAB       0           0                                                 * 39411                            * 39410
u_str      ESTAB       0           0                            /run/user/1000/bus 37199                            * 33264
u_str      ESTAB       0           0                  /var/run/dbus/system_bus_socket 32897                            * 34863
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 25434                            * 27637
u_str      ESTAB       0           0                                                 * 27275                            * 28503
u_str      ESTAB       0           0                            /run/user/121/bus 26371                            * 24993
u_str      ESTAB       0           0                                                 * 24924                            * 27934
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 23672                            * 19089
u_str      ESTAB       0           0                  /var/run/dbus/system_bus_socket 43960                            * 47441
u_seq      ESTAB       0           0                                                 * 34648                            * 34647
u_str      ESTAB       0           0                                                 * 34921                            * 32005
u_str      ESTAB       0           0                                                 * 29264                            * 25379
u_str      ESTAB       0           0                                                 * 25355                            * 28464
u_str      ESTAB       0           0                                                 * 41163                            * 41162
u_str      ESTAB       0           0                  /var/run/dbus/system_bus_socket 34026                            * 34968
u_str      ESTAB       0           0                                                 * 31092                            * 31091
u_str      ESTAB       0           0                                                 * 48311                            * 50483
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 34872                            * 33968
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 31903                            * 33854
u_str      ESTAB       0           0                                                 * 29471                            * 32811
u_str      ESTAB       0           0                                                 * 30102                            * 25393
u_str      ESTAB       0           0                            /run/user/121/bus 28794                            * 27816
u_str      ESTAB       0           0                     /run/systemd/journal/stdout 34265                            * 35436
```

14.route

The route command is a network utility used to display or manipulate the IP routing table on Unix-like operating systems, including Linux. The routing table is a key component of a computer's network configuration, specifying how network packets should be forwarded to their destination.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp4s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp4s0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 enp4s0
```

```
C:\Users\hp>route

Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
                  [MASK netmask]  [gateway] [METRIC metric]  [IF interface]

  -f            Clears the routing tables of all gateway entries.  If this is
                used in conjunction with one of the commands, the tables are
                cleared prior to running the command.

  -p            When used with the ADD command, makes a route persistent across
                boots of the system. By default, routes are not preserved
                when the system is restarted. Ignored for all other commands,
                which always affect the appropriate persistent routes.

  -4            Force using IPv4.

  -6            Force using IPv6.

  command       One of these:
                  PRINT     Prints  a route
                  ADD       Adds    a route
                  DELETE    Deletes a route
                  CHANGE    Modifies an existing route
  destination   Specifies the host.
  MASK          Specifies that the next parameter is the 'netmask' value.
  netmask       Specifies a subnet mask value for this route entry.
                If not specified, it defaults to 255.255.255.255.
  gateway       Specifies gateway.
  interface     the interface number for the specified route.
  METRIC        specifies the metric, ie. cost for the destination.

All symbolic names used for destination are looked up in the network database
file NETWORKS. The symbolic names for gateway are looked up in the host name
database file HOSTS.

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.

If Dest contains a * or ?, it is treated as a shell pattern, and only
matching destination routes are printed. The '*' matches any string,
and '?' matches any one char. Examples: 157.*.1, 157.*, 127.*, *224*.

Pattern match is only allowed in PRINT command.
Diagnostic Notes:
    Invalid MASK generates an error, that is when (DEST & MASK) != DEST.
    Example> route ADD 157.0.0.0 MASK 155.0.0.0 157.55.80.1 IF 1
             The route addition failed: The specified mask parameter is invalid. (Destination & Mask) != Destination.

Examples:

    > route PRINT
    > route PRINT -4
    > route PRINT -6
    > route PRINT 157*            .... Only prints those matching 157*

    > route ADD 157.0.0.0 MASK 255.0.0.0  157.55.80.1 METRIC 3 IF 2
              destination^      ^mask      ^gateway     metric^    ^
                                                        Interface^
      If IF is not given, it tries to find the best interface for a given
      gateway.
    > route ADD 3ffe::/32 3ffe::1

    > route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2 IF 2

      CHANGE is used to modify gateway and/or metric only.

    > route DELETE 157.0.0.0
    > route DELETE 3ffe::/32

C:\Users\hp>
```

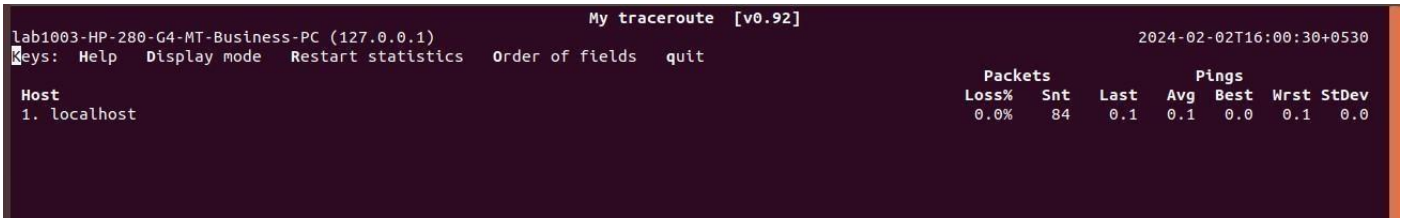
15.host

The host command is a utility used to perform Domain Name System (DNS) lookups and retrieve information about domain names or IP addresses. It is available on Unix-like operating systems, including Linux.

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ host
Usage: host [-aCdilrTvVw] [-c class] [-N ndots] [-t type] [-W time]
            [-R number] [-m flag] hostname [server]
       -a is equivalent to -v -t ANY
       -c specifies query class for non-IN data
       -C compares SOA records on authoritative nameservers
       -d is equivalent to -v
       -i IP6.INT reverse lookups
       -l lists all hosts in a domain, using AXFR
       -m set memory debugging flag (trace|record|usage)
       -N changes the number of dots allowed before root lookup is done
       -r disables recursive processing
       -R specifies number of retries for UDP packets
       -s a SERVFAIL response should stop query
       -t specifies the query type
       -T enables TCP/IP mode
       -v enables verbose output
       -V print version number and exit
       -w specifies to wait forever for a reply
       -W specifies how long to wait for a reply
       -4 use IPv4 query transport only
       -6 use IPv6 query transport only
```

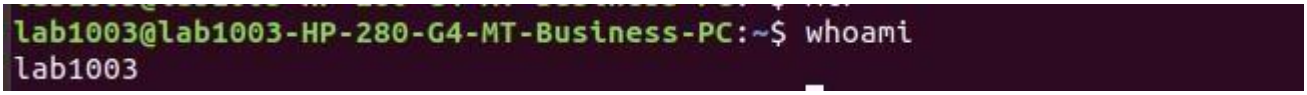
16.mtr

The mtr command, which stands for "My Traceroute," is a network diagnostic tool that combines the functionalities of traceroute and ping. It provides a continuous traceroute by sending packets to each hop on the route to a destination and measuring the response times. mtr is available on Unix-like operating systems, including Linux.



17.whoami

The whoami command is a simple command-line utility that prints the username associated with the current user who is executing the command. When you run whoami in a terminal or command prompt, it returns the username of the user logged in or executing the session.



CONCLUSION : problems and ensure smooth communication within a network infrastructure. configure network settings. With this newfound knowledge, we can effectively diagnose basic network ipconfig (or ifconfig on macOS/Linux), allowing you to verify connectivity, identify network paths, and navigating and troubleshooting network issues. We explored essential commands like ping, traceroute, and The network assignment on basic networking commands equips you with a foundational skillset for

BASED ON LO1 : To get familiar with the basic network administration commands

Atharva Yadav

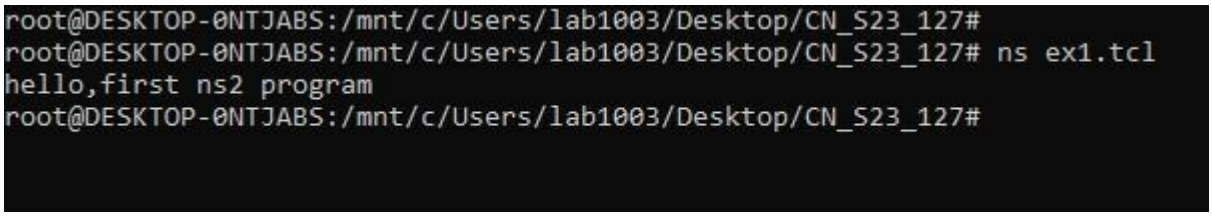
Roll No : 127 Batch : S23

Network Lab

Assignment No. 2 Install and configure the NS2.

Aim: Install and configure the NS2 and write a TCL script to display welcome message

Output:



Conclusions: The network assignment on NS-2 equips us with a powerful tool for simulating network behavior.We likely grasped optimizing network protocols and configurations without the risks associated with real-world deployments. scenarios in a controlled environment. This newfound skill is invaluable for designing, troubleshooting, and the fundamentals of working with NS-2, a network simulator that allows you to model and test various network

LO: BASED ON LO1 & LO2

Atharva Yadav
Roll No. 127
Batch S23

NETWORK LAB ASSIGNMENT NO.3

Aim: Implementation of specific network topology which is supported in TCP.

Theory:

In wireless networks nodes communicate Using Communication model that consist of TCP agent, TCP sink agent and FTP Application .

The sender node is attach to TCP agent The receiver node is attach to TCP sink agent The connection between TCP agent and TCP Sink agent is establish using keyword "connect".

In Transport Layer,TCP agent and the FTP Application are connected using keyword "attach-agent".

On receiving packet TCP sink agent sent the acknowledgment to the TCP agent that in turn Process the acknowledgment and adjust the data- transmission rate, The loss of packet are interpreted as sign of congestion.

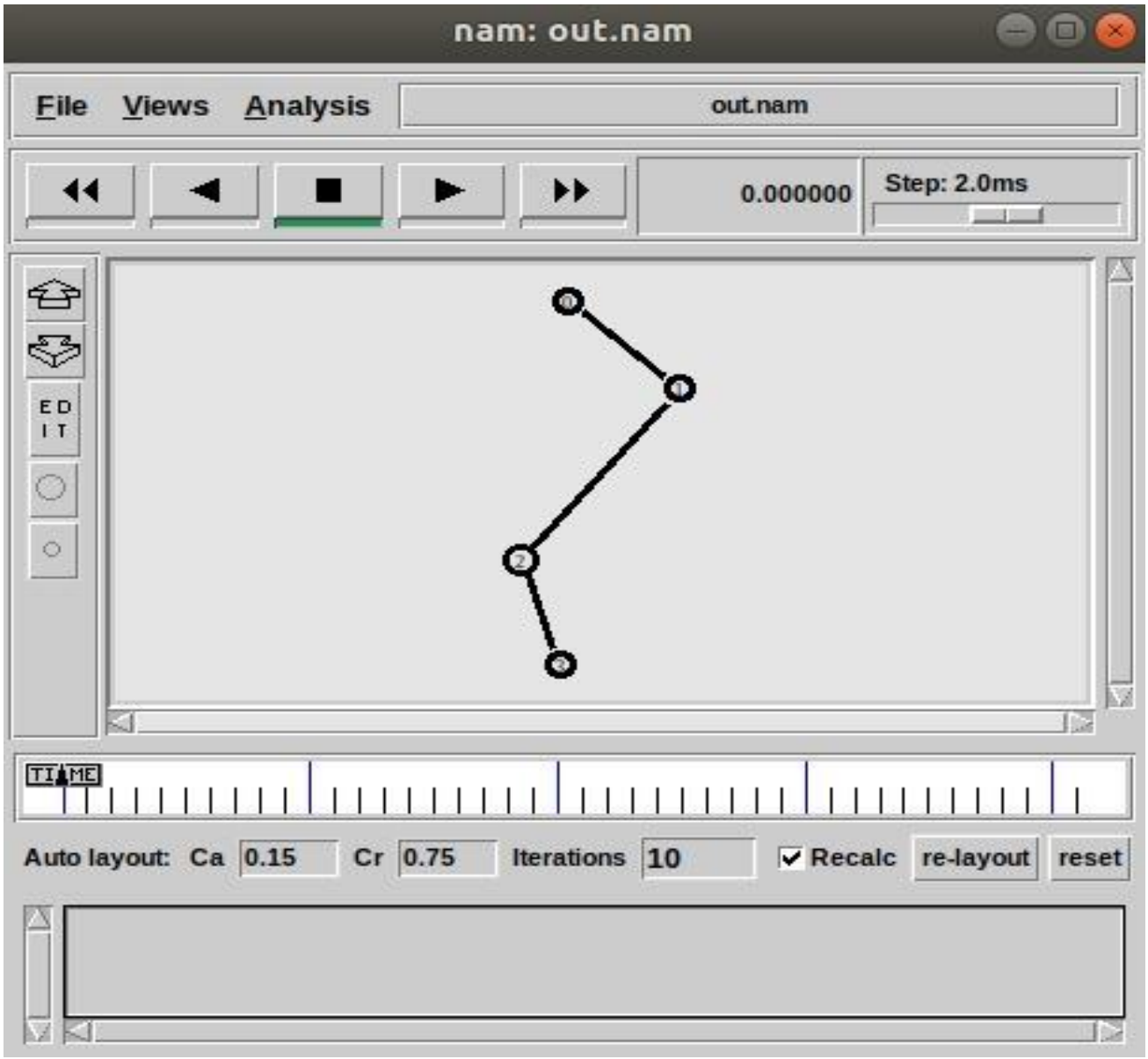
CODE :

```
#Create a Simulator Object set ns
[new Simulator]

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

#define finish procedure proc
finish {} { global ns nf np
$nsflush-trace #Close NAM
Trace 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
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 10ms DropTail #set Queue Size
```

```
$ns queue-limit $n0 $n1 5
$ns queue-limit $n1 $n2 5
$ns queue-limit $n2 $n3 5

#Monitor The queue for link (n0-n1)
$ns duplex-link-op $n0 $n1 queuePos 0.5
$ns duplex-link-op $n1 $n2 queuePos 0.5
$ns duplex-link-op $n2 $n3 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 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 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 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 up a
TCP connection set tcp [new
Agent/TCP] $ns attach-agent $n0
$tcp set sink [new
Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
#Set Packet Colour
$tcp set fid_ 4


#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
```

OUTPUT :



CONCLUSION : Network assignment on implementing a specific network topology with TCP This equips you with a strong understanding or mesh) using network devices and software. Additionally, we delved into the implementation of TCP (Transmission of how network structure and protocols interact.We explored how to configure a chosen network topology (like star, bus, Control Protocol) within this topology, ensuring reliable data transfer between devices. By successfully combining these concepts, we gained valuable practical experience in designing and implementing network communication.

LO: BASED ON LO3 : To understand the network simulator environment and visualize a network topology and observe its performance

Atharva Yadav

Roll No. 127

Batch : s23

Network Lab Assignment 4

AIM: Network topology using UDP protocol.

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection before data transfer. The UDP helps to establish low-latency and loss-tolerating connections over the network. The UDP enables process-to-process communication.

Advantages of UDP

Speed: UDP is faster than TCP because it does not have the overhead of establishing a connection and ensuring reliable data delivery.

Lower latency: Since there is no connection establishment, there is lower latency and faster response time.

Simplicity: UDP has a simpler protocol design than TCP

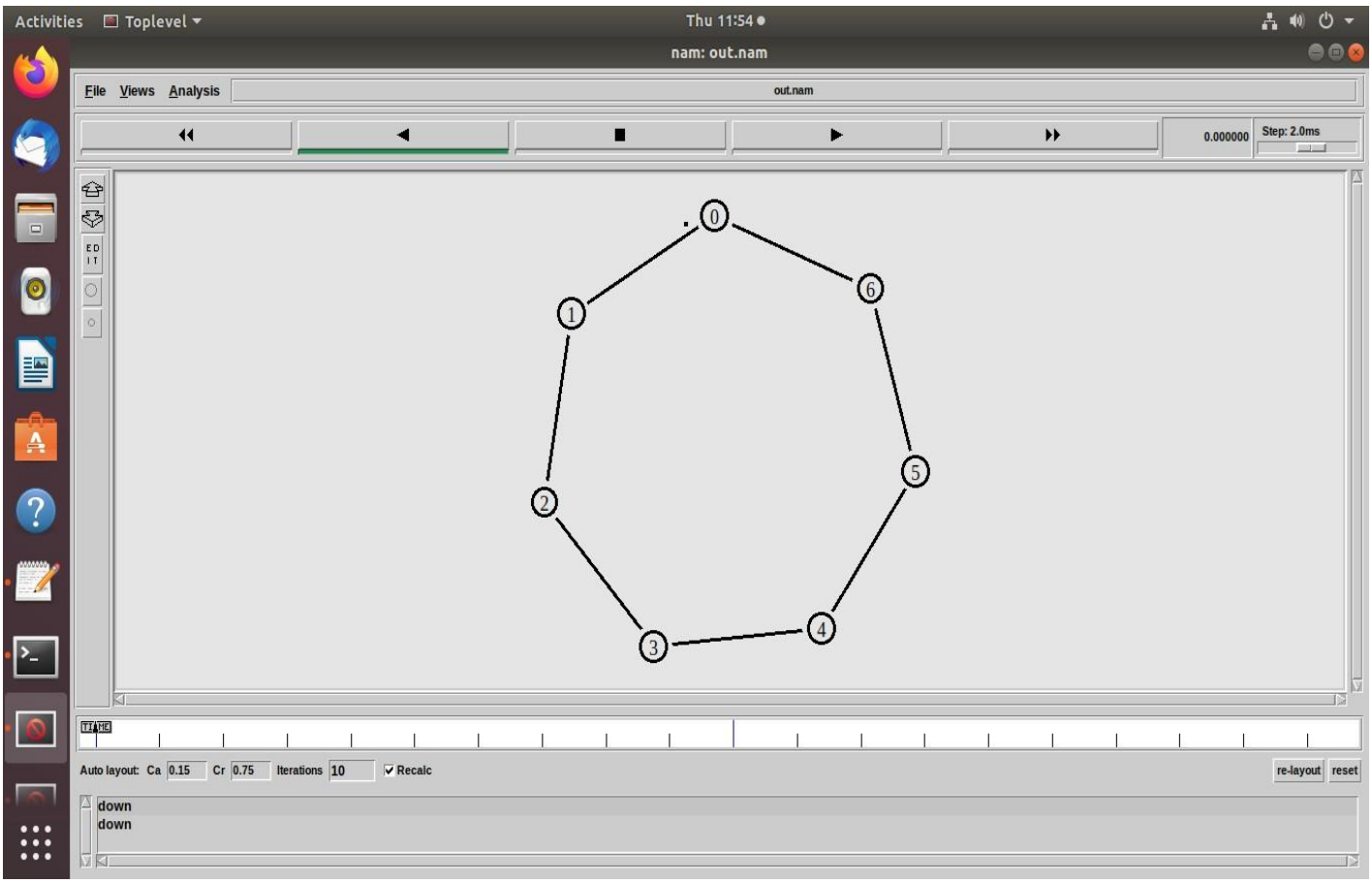
Broadcast support: UDP supports broadcasting to multiple recipients
Smaller packet size: UDP uses smaller packet sizes than TCP, which can reduce

Ring topology is a network architecture in which devices are connected in a ring structure and send information to each other based on their ring node's neighbouring node. As compared to the bus topology, a ring topology is highly efficient and can handle heavier loads. Because packets may only travel in one direction, most Ring Topologies are referred to as one-way unidirectional ring networks. Generally, Bidirectional and Unidirectional are the two types of ring topology. On the basis of devices that are being linked together to form a network, several kinds of ring topology setups work differently.

```
 CODE: set ns [new Simulator]
$ns rtproto DV
 set nf [open out.nam w]
$ns namtrace-all $nf
 proc finish {} {         global ns nf
$ns flush-trace          close $nf
exec nam out.nam          exit 0
        }

#Creating Nodes for {set i 0} {$i<7} {incr i}
{ set n($i) [$ns node]
}

#Creating Links for {set i 0} {$i<7} {incr i}
{
$ns duplex-link $n($i) $n([expr ($i+1)%7]) 512Kb 5ms DropTail
}


$ns duplex-link-op $n(0) $n(1) queuePos 1
$ns duplex-link-op $n(0) $n(6) queuePos 1

#Creating UDP agent and attching to node 0 set udp0 [new
Agent/UDP] $ns attach-agent $n(0) $udp0

$ns attach-agent $n(0) $udp0

#Creating Null agent and attaching to node 3 set null0 [new
Agent/Null] $ns attach-agent $n(3) $null0

$ns connect $udp0 $null0
```

```
#Creating a CBR agent and attaching it to udp0 set cbr0 [new
Application/Traffic/CBR]
$cbr0 set packetSize_ 1024
$cbr0 set interval_ 0.01
$cbr0 attach-agent $udp0

$ns rtmodel-at 0.4 down $n(2) $n(3)
$ns rtmodel-at 1.0 up $n(2) $n(3)

$ns at 0.01 "$cbr0 start"
$ns at 1.5 "$cbr0 stop"

$ns at 2.0 "finish"
$ns run
```

OUTPUT:



Conclusions: efficiency. Keep practicing to explore different network topologies and how they interact with UDP. This will help you network designs. choose the right protocol and network structure for various communication needs, ensuring optimal performance in your implementing UDP in a specific topology, We gained valuable experience in designing networks that prioritize speed and ideal for applications where real-time data delivery is crucial, even if some packets might be lost. By successfully bus, or mesh) and implemented UDP (User Datagram Protocol) within it. UDP prioritizes speed over reliability, making it trade-offs between different protocols in network design.WE explored configuring a chosen network topology .

LO: Based on LO3 : To understand the network simulator environment and visualize a network topology and observe its performance

Atharva Yadav
S23-127
Network Lab

NETWORK LAB ASSIGNMENT NO.5 A

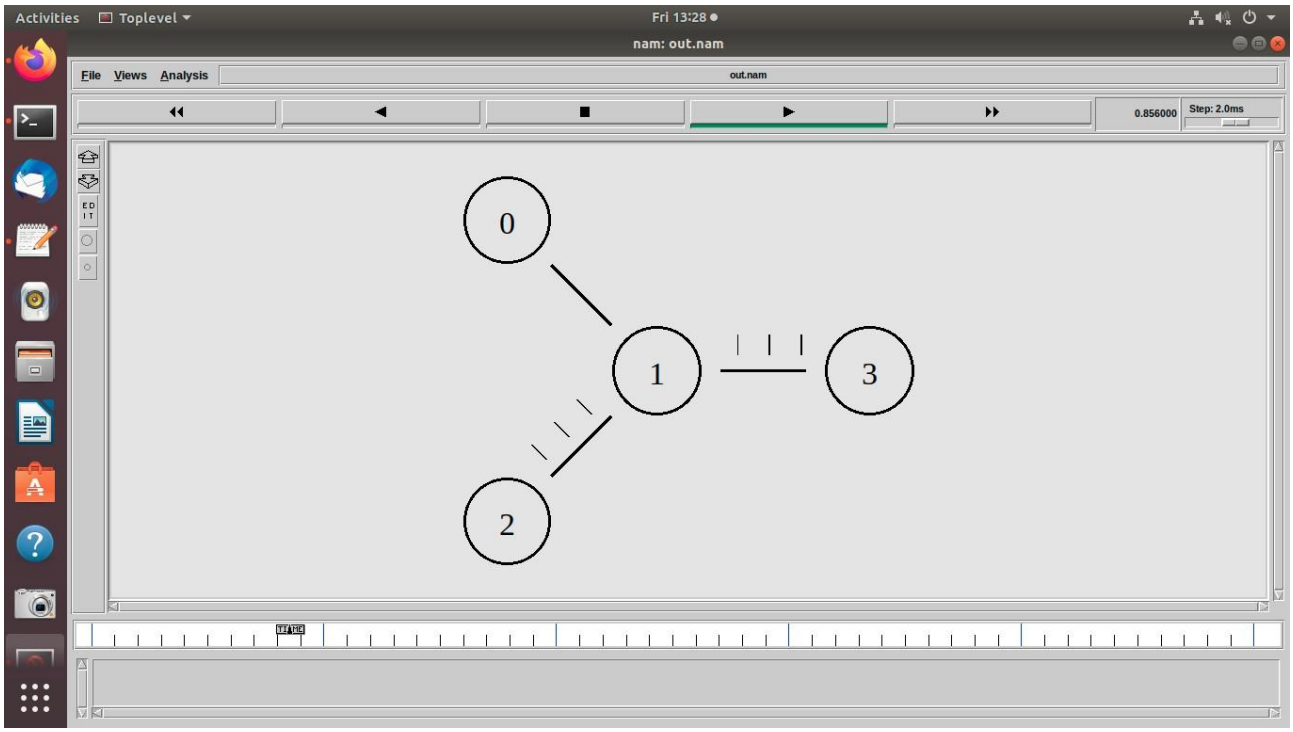
AIM : SIMULATION OF NETWORK WITH SPECIFIC ROUTING : DISTANCE ROUTING

Distant vector routing protocol also called as Bellman-Ford algorithm or Ford Fulkerson algorithm used to calculate a path. A distance-vector protocol calculates the distance and direction of the vector of the next hop from the information obtained by the neighboring router

CODE:

```
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
$tr exec nam out.nam
& exit 0 }
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$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-op $n0 $n1 orient right-down
$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n2 $n1 orient right-up
set tcp [new Agent/TCP] $ns attach-agent
$n0 $tcp set ftp [new Application/FTP]
$ftp attach-agent $tcp set sink [new
Agent/TCPSink] $ns attach-agent $n3
$sink set udp [new Agent/UDP] $ns
attach-agent $n2 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp set
null [new Agent/Null] $ns
attach-agent $n3 $null
```

$ns connect $tcp $sink
$ns connect $udp $null
$ns rtmodel-at 1.0 down $n1 $n3
$ns rtmodel-at 2.0 up $n1 $n3
$ns rtproto DV
$ns at 0.0 "$ftp start"
$ns at 0.0 "$cbr start"
$ns at 5.0 "finish"
$ns run
OUTPUT:



Conclusions: Distance vector protocols like RIP are simple to implement and suitable for small to medium-sized networks with low traffic. However, their slow convergence and susceptibility to routing loops make them less suitable for large, dynamic networks. Consider network size, traffic patterns, and reliability requirements when selecting a routing protocol.

BASED ON LO4 : To implement client-server socket programs.

Atharva Yadav
S23-127

Network Lab

NETWORK LAB ASSIGNMENT NO.5B

Aim:
To simulate and study the link state routing algorithm using simulation using NS2.

Link state routing algorithm is a type of routing algorithm used in computer networks to determine the shortest path from a source node to all other nodes in the network.

Each node in the network maintains a map of the network topology, which includes information about all links and their states (i.e., whether they are up or down).

Each node floods its link state information to all other nodes in the network.

Upon receiving this information, each node constructs a complete map of the network topology.

Using this topology map, each node computes the shortest path to all other nodes using algorithms like Dijkstra's algorithm.

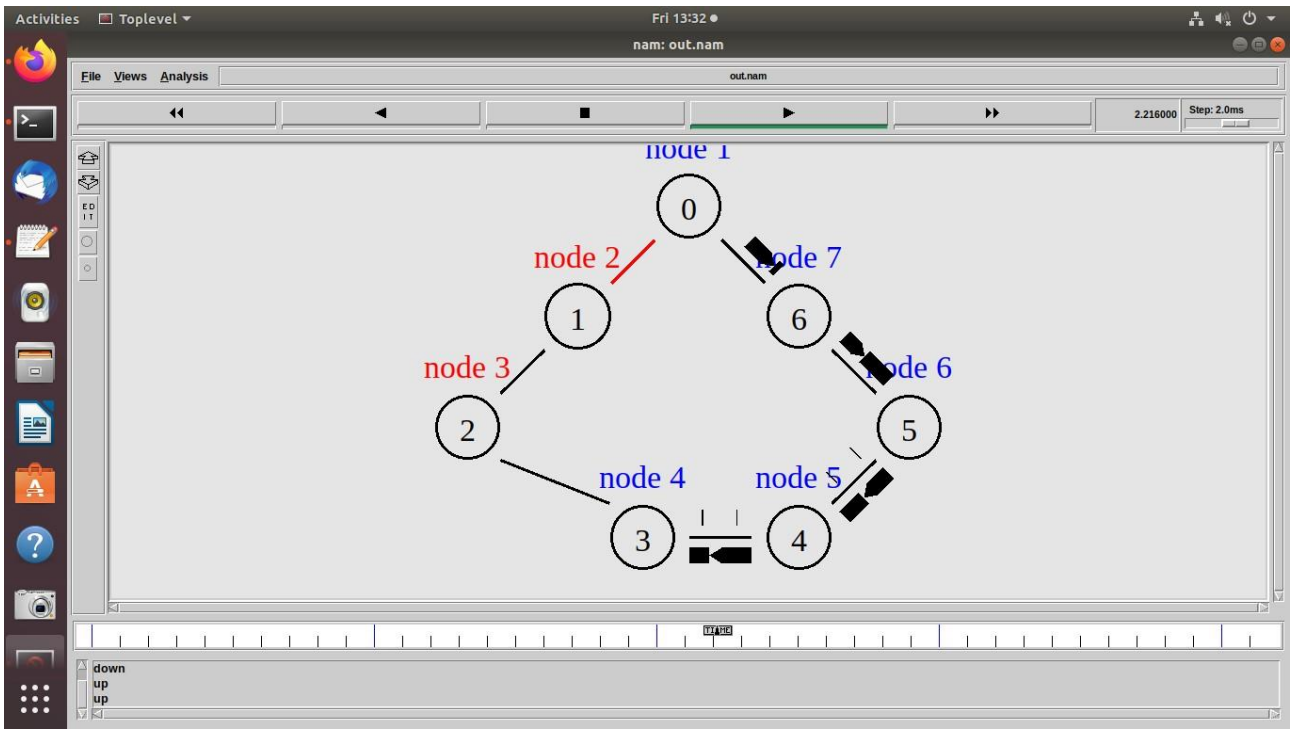
Finally, each node uses this shortest path information to forward packets towards their destination.

CODE:

```
set tf [open out.tr w]
$ns trace-all $tf set nf
[open out.nam w] $ns
namtrace-all $nf
$node1 label "node 1"
$node2 label "node 2"
$node3 label "node 3"
$node4 label "node 4"
$node5 label "node 5"
$node6 label "node 6"
$node7 label "node 7"
$node1 label-color blue
$node2 label-color red

$node3 label-color red
$node4 label-color blue
$node5 label-color blue
$node6 label-color blue
$node7 label-color blue
$ns duplex-link $node1 $node2 1.5Mb 10ms DropTail
$ns duplex-link $node2 $node3 1.5Mb 10ms DropTail
```

```
$ns duplex-link $node3 $node4 1.5Mb 10ms DropTail
$ns duplex-link $node4 $node5 1.5Mb 10ms DropTail
$ns duplex-link $node5 $node6 1.5Mb 10ms DropTail
$ns duplex-link $node6 $node7 1.5Mb 10ms DropTail
$ns duplex-link $node7 $node1 1.5Mb 10ms DropTail
$ns duplex-link-op $node1 $node2 orient left-down
$ns duplex-link-op $node2 $node3 orient left-down
$ns duplex-link-op $node3 $node4 orient right-down
$ns duplex-link-op $node4 $node5 orient right
$ns duplex-link-op $node5 $node6 orient right-up
$ns duplex-link-op $node6 $node7 orient left-up
$ns duplex-link-op $node7 $node1 orient left-up

set tcp2 [new Agent/TCP] $ns attach-
agent $node1 $tcp2 set sink2 [new
Agent/TCPSink] $ns attach-agent
$node4 $sink2 $ns connect $tcp2
$sink2 set traffic_ftp2 [new
Application/FTP] $traffic_ftp2 attach-
agent $tcp2 proc finish {} { global ns
nf $ns flush-trace close $nf exec nam
out.nam & exit 0
}
$ns at 0.5 "$traffic_ftp2 start"
$ns rtmodel-at 1.0 down $node1 $node2
$ns rtmodel-at 2.0 up $node2 $node3
$ns at 3.0 "$traffic_ftp2 start"
$ns at 4.0 "$traffic_ftp2 stop"
$ns at 5.0 "finish"
$ns run
```

OUTPUT:



Conclusions: Link-state protocols like OSPF are suitable for large, dynamic networks with high traffic. Their fast convergence and scalability make them ideal for complex network topologies. Consider network size, traffic patterns, and reliability requirements when selecting a routing protocol for simulation.

LO: Based on LO4 : To implement client-server socket programs.

ATHARVA YADAV
ROLL NO. 127
BATCH : S23

NETWORK LAB ASSIGNMENT

AIM : Establish a WireShark Connection and analyse the protocols and its headers : WIRESHARK

INSTALATION:

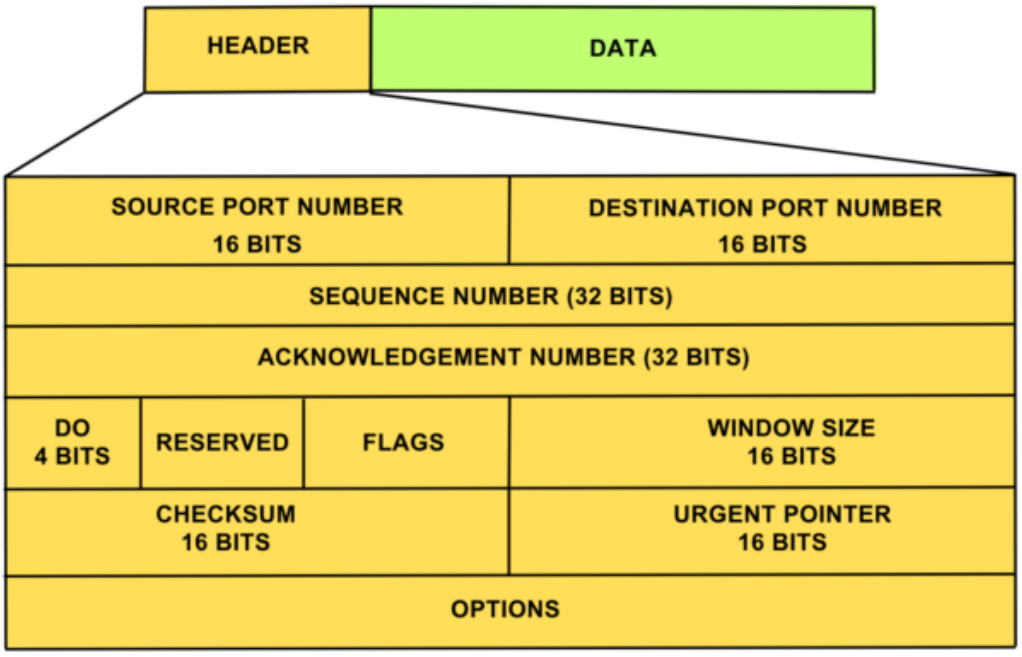
ChatGPT
ChatGPT
Here are the concise steps to install Wireshark on Windows:

1. **Download**:
   - Go to the Wireshark website.
   - Download the installer for Windows.
2. **Run Installer**:
   - Open the downloaded `.exe` file.
3. **Install Wireshark**:
   - Follow the installation prompts.
   - Choose components to install (like WinPcap or npcap).
4. **Add to PATH**:
   - Select the option to add Wireshark to the system PATH.
5. **Complete Installation**:
   - Finish the installation process.
6. **Launch Wireshark**:
   - Open Wireshark from the Start menu or desktop shortcut.
7. **Capture Privileges**:
   - If needed, run Wireshark as Administrator for packet capture.

Wireshark is a network protocol analyzer used for:

- **Packet Analysis**: Capturing and inspecting network packets.
- **Network Troubleshooting**: Diagnosing and resolving network issues.
- **Protocol Analysis**: Understanding and analyzing various network protocols.
- **Security Analysis**: Identifying and investigating security threats in network traffic.
- **Network Performance Monitoring**: Monitoring network performance metrics.
- **Application Protocol Debugging**: Troubleshooting application-level protocols.
- **Educational Tool**: Teaching networking and packet analysis concepts.
- **Development and Testing**: Testing network-enabled applications.

TCP PROTOCOL :

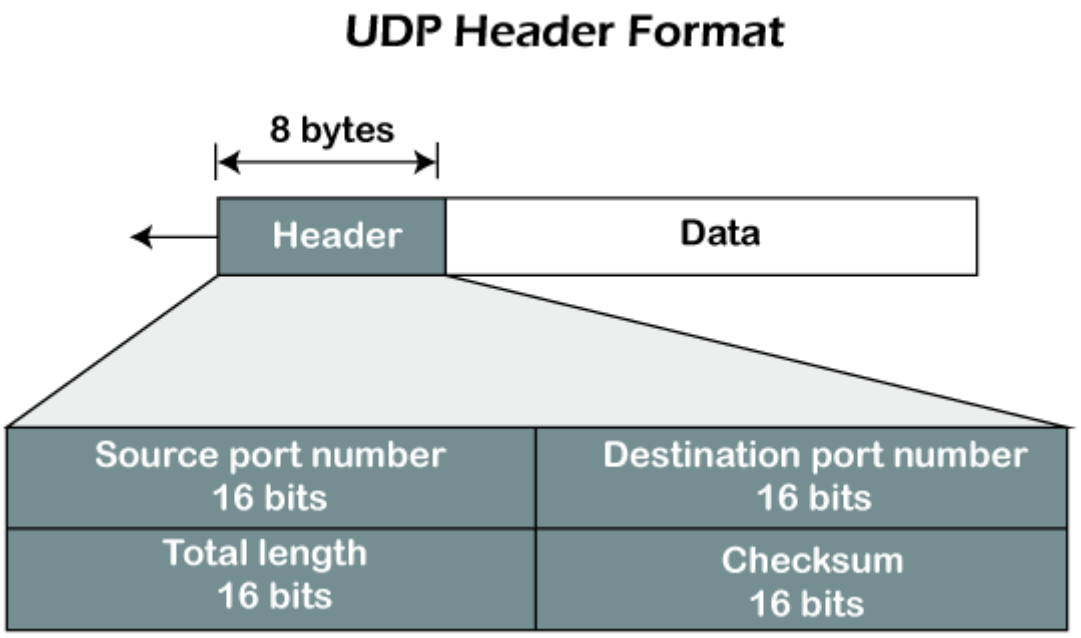


SOURCE PORT: 443	Destination Port: 50874

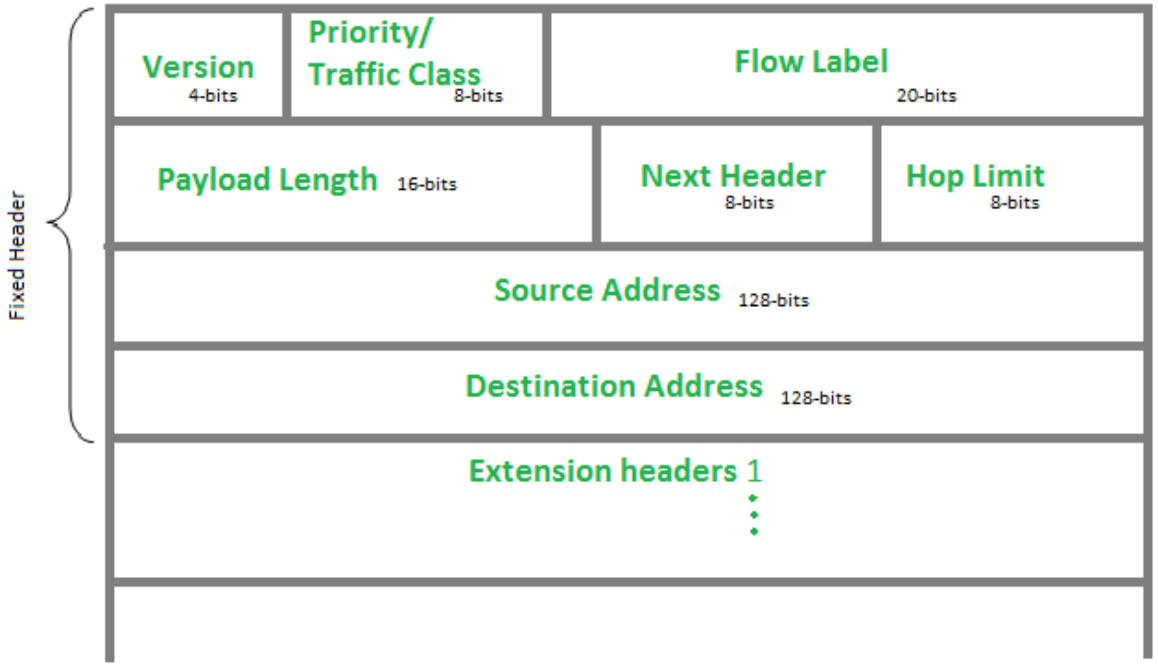
Sequence Number (raw): 1739018096			
Acknowledgment Number: 593 (relative ack number)			
0101 .... = Header Length: 20 bytes (5)	Reserved	Flags: 0x011 (FIN, ACK)	Window: 261
Checksum: 0xa5a7			Urgent Pointer: 0

UDP PROTOCOL :

Source Port: 54799	Destination Port: 443
Length: 37	Checksum: 0x6b51 [unverified]


UDP Header Format
8 bytes
Header
Data
Source port number
16 bits
Destination port number
16 bits
Total length
16 bits
Checksum
16 bits

IP HEADER:


Fixed Header
Version
4-bits
Priority/
Traffic Class
8-bits
Flow Label
20-bits
Payload Length
16-bits
Next Header
8-bits
Hop Limit
8-bits
Source Address
128-bits
Destination Address
128-bits
Extension headers 1

<table>
<tr><td><table><tr><td>Version: 6</td><td>Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)</td></tr></table></td><td>Flow Label: 0x00000</td></tr>
<tr><td>Payload Length: 36</td><td><table><tr><td>Next Header: IPv6 Hop-by-Hop Option (0)</td><td>Hop Limit: 1</td></tr></table></td></tr>
<tr><td colspan="2">Source Address: fe80::ba63:928b:ccaa:d22b</td></tr>
<tr><td colspan="2">Destination Address: ff02::16</td></tr>
</table>

CONCLUSION : The network assignment on installing Wireshark equips you with a valuable tool for network analysis and troubleshooting. Wireshark is a powerful packet sniffer that allows you to capture and inspect network traffic data flowing across your network. By successfully installing Wireshark

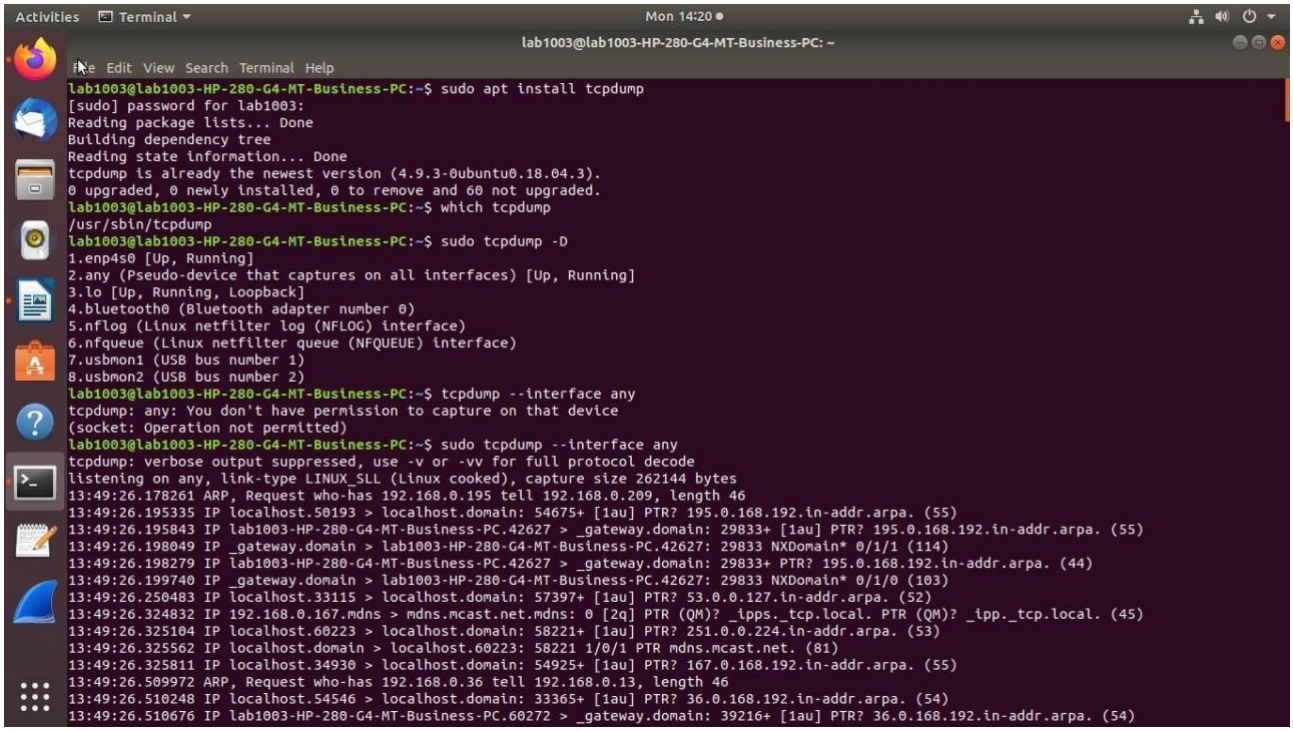
LO 5: To observe and study the traffic flow and the contents of protocol frames.

Atharva Yadav
Roll No. 127
Batch : S23
Network Lab

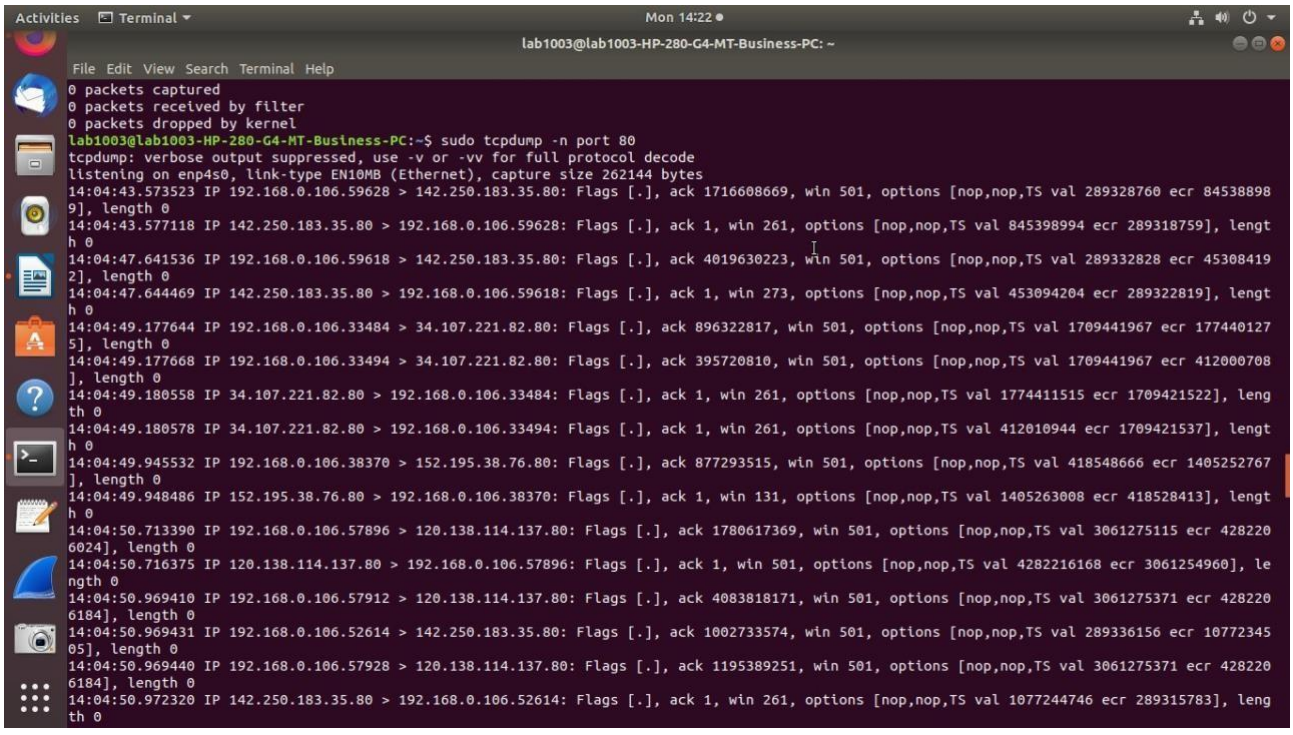
Assignment: TCPDUMP

Theory: 1. TCP (Transmission Control Protocol):

- TCP headers contain essential information for reliable, connection-oriented communication.
- Fields include source and destination ports, sequence and acknowledgment numbers, window size, and flags like SYN, ACK, and FIN.
- Analysis of TCP headers helps in monitoring connection establishment, data transfer, and connection termination. 2. IP (Internet Protocol): • IP headers encapsulate data packets and facilitate routing across network devices.
- Fields include source and destination IP addresses, version, header length, protocol, and checksum.
- Analysis of IP headers provides insights into packet routing, network addressing, and protocol version used for communication. 3. UDP (User Datagram Protocol):
- UDP headers support connectionless, unreliable communication, ideal for real□time applications.
- Fields include source and destination ports, length, and checksum. • Analysis of UDP headers helps in understanding datagram transmission and reception without the overhead of connection establishment and acknowledgment.




```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any -c3 -nn
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
13:53:28.538166 IP 192.168.0.54.138 > 192.168.0.255.138: UDP, length 206
13:53:28.634541 ARP, Request who-has 192.168.0.195 tell 192.168.0.33, length 46
13:53:28.856805 ARP, Request who-has 192.168.0.168 tell 192.168.0.168, length 46
3 packets captured
3 packets received by filter
0 packets dropped by kernel
```



```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any -c4 host www.google.com
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
14:06:26.416139 IP lab1003-HP-280-G4-MT-Business-PC.33642 > bom12s20-in-f4.1e100.net.443: UDP, length 685
14:06:26.419513 IP bom12s20-in-f4.1e100.net.443 > lab1003-HP-280-G4-MT-Business-PC.33642: UDP, length 32
14:06:26.440443 IP lab1003-HP-280-G4-MT-Business-PC.33642 > bom12s20-in-f4.1e100.net.443: UDP, length 34
14:06:26.484209 IP bom12s20-in-f4.1e100.net.443 > lab1003-HP-280-G4-MT-Business-PC.33642: UDP, length 234
4 packets captured
13 packets received by filter
1 packet dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ tcpdump -i any -c6 udp
tcpdump: any: You don't have permission to capture on that device
(socket: Operation not permitted)
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any -c6 udp
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
14:07:15.406282 IP localhost.54028 > localhost.domain: 59068+ [1au] A? metrics.ubuntu.com. (47)
14:07:15.406300 IP localhost.54028 > localhost.domain: 973+ [1au] AAAA? metrics.ubuntu.com. (47)
14:07:15.406679 IP lab1003-HP-280-G4-MT-Business-PC.33033 > _gateway.domain: 45518+ [1au] A? metrics.ubuntu.com. (47)
14:07:15.406803 IP localhost.domain > localhost.54028: 973 0/0/1 (47)
14:07:15.407346 IP localhost.33713 > localhost.domain: 39091+ [1au] PTR? 1.0.168.192.in-addr.arpa. (53)
14:07:15.408907 IP _gateway.domain > lab1003-HP-280-G4-MT-Business-PC.33033: 45518 1/3/1 A 162.213.33.48 (127)
6 packets captured
20 packets received by filter
8 packets dropped by kernel
```

```
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ tcpdump -r downloadedPackets.pcap
reading from file downloadedPackets.pcap, link-type LINUX_SLL (Linux cooked)
14:10:30.558769 ARP, Request who-has 192.168.0.195 tell 192.168.0.238, length 46
14:10:30.621474 IP 192.168.0.145.mdns > mdns.mcast.net.mdns: 0 PTR (QM)? _microsoft_mcc._tcp.local. (43)
14:10:31.417460 ARP, Request who-has 192.168.0.195 tell 192.168.0.238, length 46
14:10:32.417553 ARP, Request who-has 192.168.0.195 tell 192.168.0.238, length 46
14:10:32.985628 ARP, Request who-has _gateway tell lab1003-HP-280-G4-MT-Business-PC, length 28
```

```
1 packet dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any -c6 host 192.168.0.106 and  port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
14:15:22.265622 IP lab1003-HP-280-G4-MT-Business-PC.56192 > bom12s09-in-f3.1e100.net.http: Flags [.], ack 2414983451, win 501, options [nop,no
p,TS val 2081863064 ecr 4122624900], length 0
14:15:22.267944 IP bom12s09-in-f3.1e100.net.http > lab1003-HP-280-G4-MT-Business-PC.56192: Flags [.], ack 1, win 261, options [nop,nop,TS val
4122635140 ecr 2081832620], length 0
14:15:32.505655 IP lab1003-HP-280-G4-MT-Business-PC.56192 > bom12s09-in-f3.1e100.net.http: Flags [.], ack 1, win 501, options [nop,nop,TS val
2081873304 ecr 4122635140], length 0
14:15:32.508534 IP bom12s09-in-f3.1e100.net.http > lab1003-HP-280-G4-MT-Business-PC.56192: Flags [.], ack 1, win 261, options [nop,nop,TS val
4122645380 ecr 2081832620], length 0
14:15:42.745645 IP lab1003-HP-280-G4-MT-Business-PC.56192 > bom12s09-in-f3.1e100.net.http: Flags [.], ack 1, win 501, options [nop,nop,TS val
2081883544 ecr 4122645380], length 0
14:15:42.752235 IP bom12s09-in-f3.1e100.net.http > lab1003-HP-280-G4-MT-Business-PC.56192: Flags [.], ack 1, win 261, options [nop,nop,TS val
4122655620 ecr 2081832620], length 0
6 packets captured
6 packets received by filter
0 packets dropped by kernel
lab1003@lab1003-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -i any -c6 host 192.168.0.106 and  port 443
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
14:17:15.885076 IP lab1003-HP-280-G4-MT-Business-PC.51170 > bom12s20-in-f4.1e100.net.443: UDP, length 1357
14:17:15.894473 IP lab1003-HP-280-G4-MT-Business-PC.51170 > bom12s20-in-f4.1e100.net.443: UDP, length 1357
14:17:15.894489 IP lab1003-HP-280-G4-MT-Business-PC.51170 > bom12s20-in-f4.1e100.net.443: UDP, length 1357
14:17:15.898433 IP bom12s20-in-f4.1e100.net.443 > lab1003-HP-280-G4-MT-Business-PC.51170: UDP, length 1357
14:17:15.899706 IP bom12s20-in-f4.1e100.net.443 > lab1003-HP-280-G4-MT-Business-PC.51170: UDP, length 1357
14:17:15.907283 IP lab1003-HP-280-G4-MT-Business-PC.51170 > bom12s20-in-f4.1e100.net.443: UDP, length 1357
6 packets captured
7 packets received by filter
0 packets dropped by kernel
```

Conclusions:

TCPDUMP enables detailed examination of packet headers, facilitating network troubleshooting, performance monitoring, and security analysis.

- By analyzing TCP headers, network administrators can diagnose connection issues, monitor traffic flow, and detect potential security threats.
- Examination of IP headers aids in understanding packet routing, identifying network congestion points, and ensuring proper addressing.
- Analysis of UDP headers helps in optimizing real-time applications,

diagnosing packet loss, and ensuring efficient data transmission. Overall, TCPDUMP provides valuable insights into network traffic behavior and protocol usage, empowering administrators to maintain and optimize network performance and security.

Atharva Yadav
Roll No. 127
Batch : s23

Network Lab Assignment No.8

AIM: SOCKET PROGRAMMING USING TCP/IP PROTOCOL

Theory: To connect to another machine we need a socket connection. A socket connection means the two machines have information about each other's network location (IP Address) and TCP port. The java.net.Socket class represents a Socket. environments.

Java Program for Client application

CODE:

```
import java.io.*; import java.net.*; public class Client {
public static void main(String[] args) {  try {
 Socket socket = new Socket("localhost", 5000);  System.out.println("Connected to server.");
 BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
 PrintWriter writer = new PrintWriter(socket.getOutputStream(), true);
 BufferedReader consoleReader = new BufferedReader(new
InputStreamReader(System.in));

 String inputLine, outputLine;  while (true) {
 System.out.print("Client: ");  outputLine =
consoleReader.readLine();  writer.println(outputLine);  if
(outputLine.equalsIgnoreCase("bye")) break;  inputLine =
reader.readLine();
 System.out.println("Server: " + inputLine);  if
(inputLine.equalsIgnoreCase("bye")) break;
}  writer.close();  reader.close();
socket.close();  } catch (IOException
e) {
 e.printStackTrace();
 }
 }
}
```

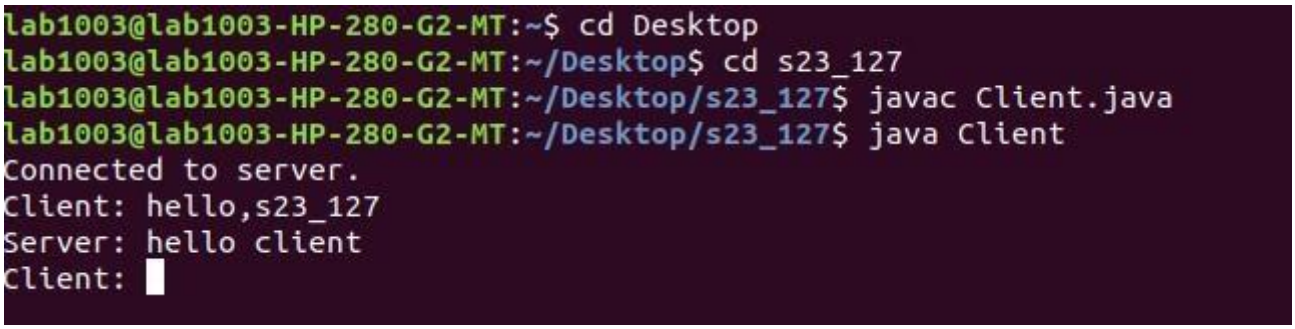
Java Program for Server application

CODE:

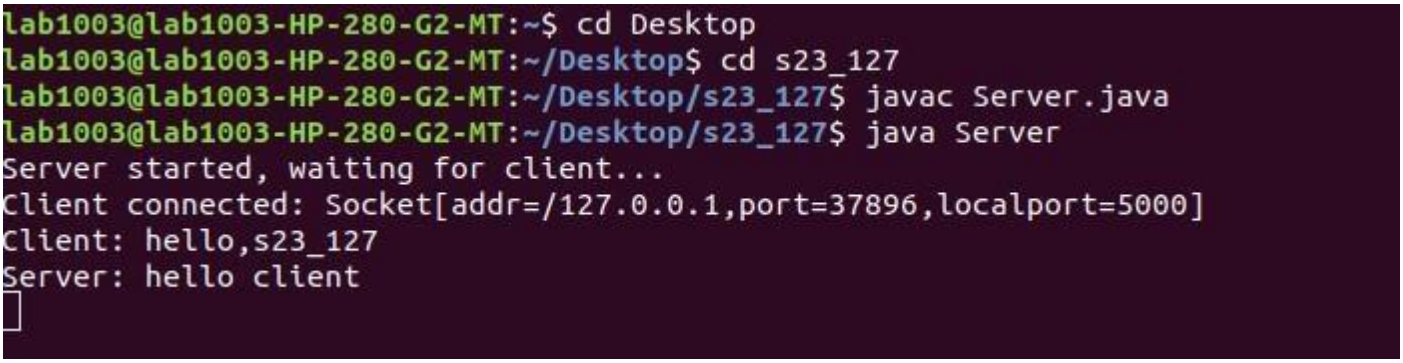
```
import java.io.*; import java.net.*; public class Server {
public static void main(String[] args) {  try {
 ServerSocket = new ServerSocket(5000);  System.out.println("Server started, waiting for client...");
 Socket clientSocket = serverSocket.accept();
 System.out.println("Client connected: " + clientSocket);
 BufferedReader reader = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
 PrintWriter writer = new PrintWriter(clientSocket.getOutputStream(), true);
 BufferedReader consoleReader = new BufferedReader(new
InputStreamReader(System.in));  String inputLine,
outputLine;
 while ((inputLine = reader.readLine()) != null) {
System.out.println("Client: " + inputLine);  if
(inputLine.equalsIgnoreCase("bye")) break; System.out.print("Server:
");  outputLine = consoleReader.readLine();  writer.println(outputLine);
if (outputLine.equalsIgnoreCase("bye")) break;
 }  writer.close();  reader.close();
clientSocket.close();
serverSocket.close();  } catch
(IOException e) {
 e.printStackTrace();
 }
 }
}
```

OUTPUT:

Client:



Server:



Conclusion:TCP protocol, ensuring reliable data transfer between programs on different machines. applications.We likely explored the concepts of sockets, which act as endpoints for communication, and delved into the The network assignment on socket programming using TCP equips us with a foundational skill for building network

Based On IO 6: To design and configure a network for an organization

Atharva Yadav
Roll No.127
Batch: s23

NETWORK LAB ASSG NO. 9

AIM: SOCKET PROGRAMMING USING UDP PROTOCOL

Theory: Datagram Sockets are Java's mechanism for network communication via UDP instead of TCP. Java provides
DatagramSocket to communicate over UDP instead of TCP. It is also built on top of IP. DatagramSockets can be used to both send and receive packets over the Internet. One of the examples where UDP is preferred over TCP is the live coverage of TV channels. In this aspect, we want to transmit as many frames to live audience as possible not worrying about the loss of one or two frames. TCP being a reliable protocol add its own overhead while transmission. Another example where UDP is preferred is online multiplayer gaming. In games like counter-strike or call of duty, it is not necessary to relay all the information but the most important ones. It should also be noted that most of the applications in real life uses careful blend of both UDP and TCP; transmitting the critical data over TCP and rest of the data via UDP.

Java Program for Client application

CODE:

```
// UDP Client import java.io.*; import java.net.*; public
class UDPClient {  public static void main(String[] args) {
DatagramSocket clientSocket = null;  try {
 // Create a DatagramSocket  clientSocket = new
DatagramSocket();
 InetAddress IPAddress = InetAddress.getByName("localhost");  byte[] sendData =
new byte[1024];  byte[] receiveData = new byte[1024];  while (true) {
 // Read input from user
 BufferedReader inFromUser = new BufferedReader(new
InputStreamReader(System.in));
 System.out.print("Enter your message: ");  String message =
inFromUser.readLine();  sendData = message.getBytes();
 // Send packet to server
 DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);
clientSocket.send(sendPacket);  // Receive response from server
 DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
 clientSocket.receive(receivePacket);
 String serverResponse = new String(receivePacket.getData(), 0, receivePacket.getLength());
 System.out.println("Received from server: " + serverResponse);  }
 } catch (IOException e) {
 e.printStackTrace();  } finally {  if
(clientSocket != null) {
clientSocket.close();
 }
 }
 }
}
```

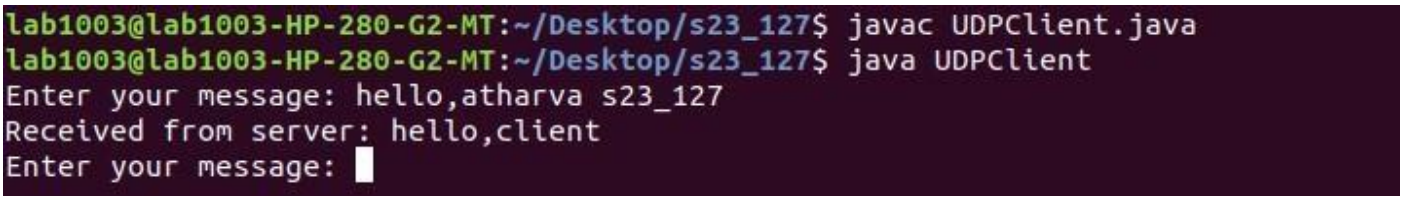
Java Program for Server application

CODE:

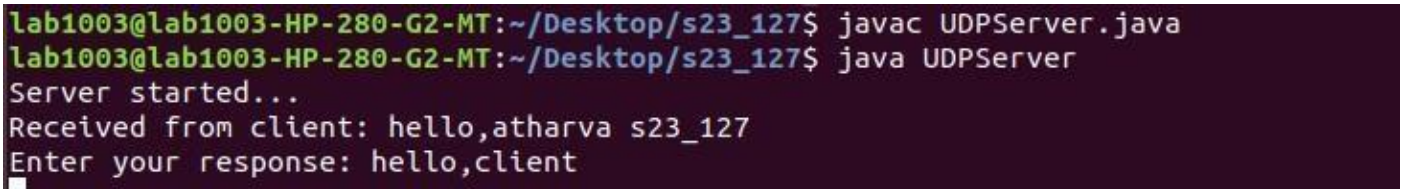
```
// UDP Server import java.io.*; import java.net.*; public

class UDPServer {  public static void main(String[] args) {

DatagramSocket serverSocket = null;  try {

 // Create a DatagramSocket, bind it to port 9876  serverSocket = new

DatagramSocket(9876);  byte[] receiveData = new byte[1024];  byte[]

sendData = new byte[1024];  System.out.println("Server started...");

while (true) {

 // Receive packet from client

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

 String clientMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

 System.out.println("Received from client: " + clientMessage);

 // Get client's address and port

 InetAddress clientAddress = receivePacket.getAddress();  int clientPort =

receivePacket.getPort();


 // Send response back to the client

 BufferedReader inFromUser = new BufferedReader(new
InputStreamReader(System.in));

 System.out.print("Enter your response: ");  String serverResponse =

inFromUser.readLine();  sendData = serverResponse.getBytes();

 DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);

serverSocket.send(sendPacket);

 }

 } catch (IOException e) {
```

```
 e.printStackTrace();  } finally {  if

(serverSocket != null) {

serverSocket.close();

}

 }

 }

}
```

OUTPUT: Client:



Server:



CONCLUSION : ideal for real-time applications where immediate response is crucial, even if sending data packets without error checking or guaranteed delivery, making it into the UDP (User Datagram Protocol) protocol. UDP prioritizes speed by likely explored creating sockets, the endpoints for communication, and delved skills to build applications that prioritize speed over guaranteed delivery. We The network assignment on socket programming using UDP equips us with the some data might be lost

Based On LO 5 : To observe and study the traffic flow and the contents of protocol frames.

Atharva Yadav
S23-127
NL assignment
28/03/2024

AIM : The case study to design and configure any organization

Layout: In this organization we will be using Hybrid topology to connect all the labs and classrooms of 9,10,11 floor

To design and configure a network for the organization with three floors (9th, 10th, and 11th) with specific requirements, we'll opt for a hybrid network topology. To incorporate a hybrid topology with a tree topology for each floor. This design will combine the advantages of both topologies, providing scalability, redundancy, and efficient management.

Network Components:

1. **Switches:** Managed switches will be used for each floor to facilitate connectivity.
2. **Routers:** A router will be placed at each floor to manage inter-floor communication.
3. **Cables:** Ethernet cables (Cat6) will be used for wired connections.
4. **Servers:** A central server room will be established on the 9th floor to manage data and facilitate network operations.
5. **Computers:** Desktop computers will be distributed across labs and classrooms on each floor.
6. **Printers:** Network printers will be strategically placed for shared access.
7. **Wireless Access Points (APs):** Wireless APs will be installed on each floor to provide Wi-Fi connectivity.

Network Design:

9th Floor:

- **Tree Topology:**
- **Root Node (Switch):** Connected to the router and serves as the root of the tree.
- **Branch Nodes (Switches):** Connected to the root switch and serve individual labs and classrooms.
- **Leaf Nodes (Devices):** Computers and printers connected to each branch switch. • **Server Room:** Connected to the root switch for centralized management and data access.

10th Floor:

- **Tree Topology:**
- **Root Node (Switch):** Connected to the router.
- **Branch Nodes (Switches):** Connected to the root switch, serving labs and classrooms.
- **Leaf Nodes (Devices):** Computers and printers connected to each branch switch.

11th Floor:

- **Tree Topology:**
- **Root Node (Switch):** Connected to the router.
- **Branch Nodes (Switches):** Connected to the root switch, serving classrooms.

- **Leaf Nodes (Devices):** Computers connected to each branch switch.

2. IPv4 Addressing and Class C Network:

a. IPv4 Addressing: The Internet Protocol version 4 (IPv4) will be used to assign unique addresses to each device on the network. These addresses will consist of 32 bits, divided into network and host portions.

b. Class C Network: A class C network will be employed, as it allows for a maximum of 254 hosts. The network address will be assigned as follows:

Network Prefix: 192.168.x.0

Network Mask: 255.255.255.0

3. Network Organization:

a. Server Room (Room 902):

i. IP Address: 192.168.x.1 ii.

Subnet Mask: 255.255.255.0 iii.

Default Gateway: 192.168.x.1

iv. DHCP Server: Assign IP addresses to devices on the network

b. Floors 9, 10, and 11:

i. Floor Layout: Tree Layout

ii. Device Assignment: Assign IP addresses to devices on each floor, ensuring they fall within the available host range (2-254) for the class C network.

Hybrid Topology:

Inter-Floor Connectivity:

- Each floor's router will act as a central point connecting to the server room's switch on the 9th floor.
- Inter-floor communication will be facilitated through these routers, creating a hybrid star/bus topology for connectivity between floors.

Tree Topology on Each Floor:

- Each floor will follow a tree topology with switches serving as distribution points.
- Switches will connect to a central router on the floor, which will manage communication within the floor and provide connectivity to other floors.

Conclusions: This document outlines the design and configuration of a network for a multifloor organization using a hybrid topology approach. By incorporating tree topologies within each floor and utilizing routers for inter-floor communication, the network achieves scalability, redundancy, and efficient management. The use of managed switches, routers, cables, servers, computers, printers, and wireless access points ensures comprehensive connectivity and support for various network operations. Additionally, the IPv4 addressing scheme with a Class C network facilitates unique addressing for devices while enabling efficient IP management. Overall, this network design meets the specific requirements of the organization, providing a robust and reliable infrastructure for seamless communication and data management.

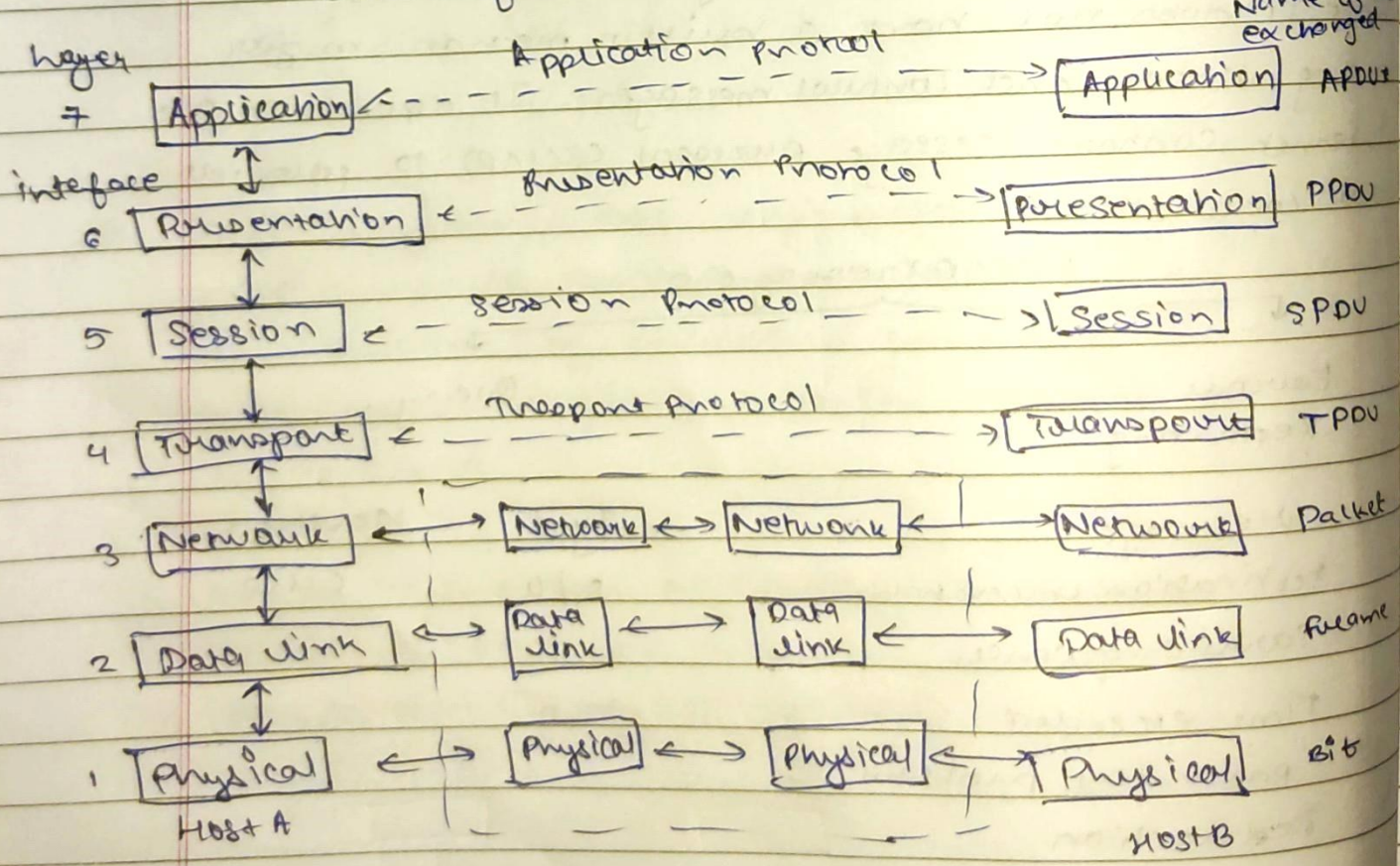
ATHARVA YADAV
S23-127
CN WRITTEN ASSIGNO-1
[ANY THREE]

Page No.
Date

Q1] Explain ISO/OSI Ref Model in details
Q2] write a shortnote on 1. Topology
2. IP Addresses
Q3] Differentiate between LAN, WAN, MAN
Q4] Explain Error control mechanism in DLL
Q5] Explain Ethernet standard

Q1] Ans:-
The international standards organization (ISO) is a multinational body dedicated to worldwide agreement on internation standards. It divides the ~~process~~ communication processes into seven layers

OSI Reference model



i) Physical layer
converts bits into electronic signal for outgoing messages. converts electronic signals into bits for incoming messages.

(ii) Data link layer
The main task of the data link layer is to detected transmission errors. It accomplishes this task by having the sender break up the input data into data frames and transmits the frames sequentially. At the receiving end, this layer packages raw data from the physical layer into data frames for delivery to the network layer.

(iii) Network Layer.
The network layer controls the operation of the subnet the network layer is responsible for the delivery of individual packets from the source host to the destination host. A key design issue is determining how packets are routed from source to destination.

(iv) Transport layer.
Manages the data transmission across a network Manages the flow of data btw parties by segmenting long data streams into smaller data chunks Provides acknowledgements of successful transmission and requests resends for packets which arrives with errors.

(v) Session Layer

The session layer allows users on different machines to establish sessions between them. Various services offer by session layer are: dialog control, token management, Synchronization.

(vi) Presentation layer:

The presentation layer is concerned with the syntax and semantics of the information transmitted.

(vii) Application layer:

Application layer is responsible for providing services to the user. The application layer contains a variety of protocol that are commonly need by user.

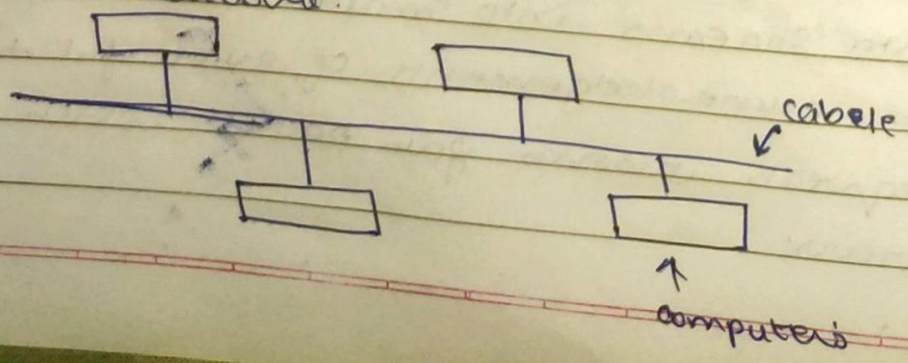
Q2] Ans.

i] Topology.

Topology defines the physical or logical arrangement of links in a network.

① Bus topology:

One long cable act as a backbone to link all the devices in a network.



Page No.
Date

Advantages of Bus Topology.

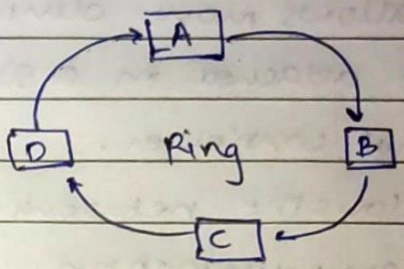
Less Expensive

Suitable for temporary network

Disadvantages:

Not a fault tolerant

limited cable length.



(2) Ring Topology:

A Ring topology is a bus topology in a closed loop.

peer to peer LAN topology

Advantages

~~Com~~ Easy to install and reconfigure.

All nodes with equal access.

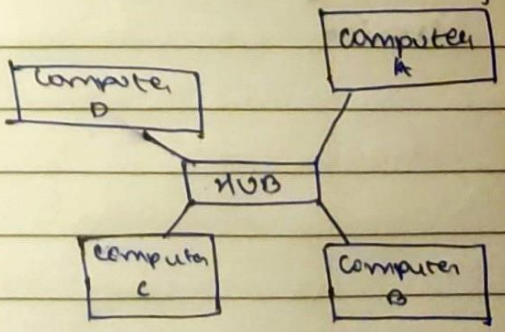
Disadvantages:

Inc in load leads to dec in performance

No security.

(3) Star topology.

Each device has a dedicated point to point link btw only a central controller or "HUB". The devices are not directly linked to some other device



Advantages:

Easy to design and implement

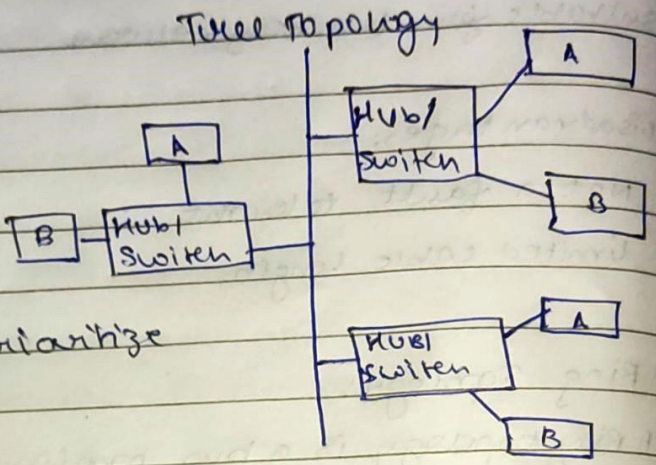
Scalable.

Disadvantages:

Each device must connected to controller.

Bottle neck due to overloaded Switch and Hub.

Tree topology: Tree topology has some variation from star topology. The nodes in the tree are linked to central controller.



Adv:

It allows more devices to be attached in a single central controller.

It allows the network to prioritize the communication.

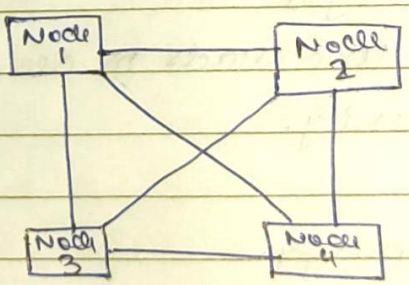
Disadv:

Each device must be linked to controller

It requires more installation processes.

Mesh topology:-

Here every device has a direct point to point link between every other device.



Adv

It eliminates the traffic problem.

It is robustness.

Dis adv

More number of cables to be used.

Every devices must be connected to some other devices

Q3] Ans

① LAN : Local Area Network

LAN's are privately owned networks within a single building or campus of up to few kilometers in size.

LAN's are distinguished based on

i) Their size

ii) Their transmission Technology.

iii) Their topology.

LAN's are restricted in size

LAN's use a transmission technology consisting of a single cable to which all machines are attached like telephone company lines in rural area.

Advantages	Disadvantages.
1) sharing files	1) Reliability
2) sharing of programs	2) capacity
3) communication exchange	3) High cost.

② MAN : Metropolitan Area Network.

Interconnects users with computer resources in a geographic area or region larger than that covered by even a large.

MAN supports upto 150 KM distance

It uses the standard DQDB.

Advantages.	Disadvantages.
1) High Bandwidth	1) Large Space Req.
2) It supports large no. of clients.	2) Slower Data Access
3) Reduce the Errors.	3) High cost.

Page No.
Date

(3) WAN: Wide Area Network

WAN's spans a large geographical area, often a country or continent

It contains collection of machine for running.

Atharva Yadav
Roll No. 127
Batch : s23

1. Explain in detail VLAN

VLAN stands for Virtual Local Area Network. It's a method of logically segmenting a single physical network into multiple distinct virtual networks. These virtual networks operate as if they are physically independent, even though they may share the same physical infrastructure. VLANs offer several benefits, including increased security, improved network performance, and simplified network management.

Purpose:

- VLANs are primarily used to improve network performance, security, and scalability.
- They allow network administrators to logically group devices together based on factors such as department, function, or security requirements, regardless of their physical location.

How VLANs Work:

- VLANs are created by assigning ports on network switches to specific VLANs.
- Switches use a process called VLAN tagging to identify which VLAN a packet belongs to. This tagging adds a small piece of extra information to the Ethernet frame, indicating the VLAN membership of the packet.
- When a switch receives a packet, it examines the VLAN tag and forwards the packet only to the ports assigned to the same VLAN as the source of the packet.

Benefits:

- *Improved Security*: VLANs can enhance network security by isolating sensitive data or critical systems from other parts of the network. For example, finance department computers can be placed on a separate VLAN to prevent access from other departments.
- *Better Performance*: By segregating network traffic, VLANs can reduce broadcast traffic and congestion, leading to improved overall network performance.

2.

a. HTTP (Hypertext Transfer Protocol): HTTP is the foundation of data communication on the World Wide Web. It defines how messages are formatted and transmitted, enabling web browsers to retrieve and display web pages. HTTP operates on a client-server model, facilitating the exchange of hypertext documents.

b. SNTP (Simple Network Time Protocol): SNTP is a simplified version of NTP (Network Time Protocol) used for synchronizing the time of networked devices. It ensures accurate timekeeping by synchronizing devices' clocks to a reference time source over a network, vital for tasks such as logging and security protocols.

c. FTP (File Transfer Protocol): FTP is a standard network protocol for transferring files between a client and a server on a computer network. It offers a straightforward method for uploading, downloading, and managing files across the internet. FTP operates on a client-server architecture with control and data connections.

d. DNS (Domain Name System): DNS translates domain names to IP addresses, facilitating human-readable web addresses. It functions as the internet's address book, mapping domain names like example.com to their corresponding IP addresses. DNS operates in a distributed hierarchical system, translating domain names into IP addresses and vice versa.

e. SNMP (Simple Network Management Protocol): SNMP is an internet standard protocol for collecting and organizing information about managed devices on IP networks. It enables network administrators to monitor network performance, detect and resolve issues, and manage network devices such as routers, switches, and servers remotely.