THADOMAL SHAHANI ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR: 2021-2022

Subject: Network Lab **Class: SE- IT** **Semester: IV**

Lab Outcomes

LO 1: -To get familiar with the basic network administration commands.

LO 2: -To install and configure network simulator and learn basics of TCL scripting.

LO 3: -To understand the network simulator environment and visualize a network topology and observe its performance.

LO 4: -To implement client-server socket programs.

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

LO 6: -To design and configure a network for an organization.

Experiment List

Sr. No.	Name of Experiment	LO Mapped
1	Understanding Basic networking Commands.	LO1
2	Installation and configuration of NS2 and implementation of TCL Hello Programming	LO2
3	Implementation of Specific Network topology with respect to TCP	LO3, LO5
4	Implementation of Specific Network topology with respect to UDP	LO3, LO5
5	Simulation of Network with specific routing protocols (Distance Vector, Link State)	LO3, LO5
6	Installation of Wire shark and Analysis of Packet headers – TCP, IP, UDP using Wireshark	LO5
7	Analysis of Packet headers – TCP, IP, UDP using TCPDUMP	LO5
8	Socket Programming with C/Java - TCP Client, TCP Server	LO4
9	Socket Programming with C/Java - UDP Client, UDP Server	LO4
10	A case study to design and configure any organization network.	LO6

Ms. Kumkum Saxena & Ms. Vijal Jain

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
(1)	Understanding Basic network Commands	18/01/22	1-2	
(2)	Installation and configuration of NS2 and implementation of TCL Hello Programing	25/01/22	3	
(3)	Implementation of specific network topology with respect to TCP.	7/02/22	4-5	
(4)	Implementation of specific network topology with respect to UDP	13/02/22	6	8/3/22
(5)	Simulation of Network with specific routing protocols (Distance Vector, Link State).	22/02/22	7-8	
(6)	Installation of wireshark and analysis of packet headers - TCP, IP, UDP using wireshark	08/05/22	9	
(7)	Socket Programming with Java UDP clients, UDP servers	15/03/22	10-11	
(8)	Socket Programming with Java TCP client, TCP servers	22/03/22	12-13	29/3/22
(9)	Analysis of Packet Headers - TCP, IP, UDP using TCP Dump.	29/03/22	14	
(10)	A case study to design and configure any organisation network.	29/03/22	15	
(11)	Report for lab 3-1 : UDP	08/03/22	12	
(12)	Report for lab 3-2 : TCP	08/03/22	17-18	
(13)	Written Assignment - 1	29/03/22	24-25	
(14)	Written Assignment - 2	29/03/22	26-27	5/4/22

Krishna Khadke
S13-38

8/01/22

Assignment - 1

Aim : Understanding Basic networking commands.

Theory :-

(1) **ping :-**

PING (Packet Internet Groper) command is to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address with the message "PING" and get a response from the server/host this time is recorded which is called latency.

(a) ping -4 or ping -6
- To request IPv4 or IPv6.

(b) ping -i TTL
- The default interval between each ping request is set to one second. You can increase or decrease that time using the -i switch. To decrease the ping interval, use values lower than 1.

eg. ping -i 0.5 google.com.

(c) ping -s.
- In some scenarios, you may want to use -s to increase the packet size from the default value of 56(84) bytes. The number in parenthesis represents the ping bytes sent including 28 bytes of the header packet.

eg. ping -s 1000 google.com.

(d) ping -c
- To make the ping command automatically stop after it sends a certain number of packets, use -c and a number. This sets the desired amount of ping requests.

(e) ping -a

- When you use the -a switch, the system plays a sound when there is a response from a host.

(2) ifconfig:-

ifconfig command is used to configure the kernel-resident network interfaces. It is used at the boot time to setup the interfaces are necessary. After that, it is usually used when needed during debugging or when you need system tuning.

(a) ifconfig -a

- This option is used to display all the interfaces available, even if they are down.

(b) ifconfig -s

- Display a short list, instead of details.

(c) ifconfig -v

- Run the command in verbose mode - log more details about

(c) netstat -au
- To list all udp ports.
(d) netstat -l.
- To list only the listening ports.
(e) netstat -lt
- To list only the listening tcp ports.

(4) Nslookup.
Nslookup (stands for "Name Server Lookup") is a useful command for getting information from the DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record.

(a) nslookup google.com
- nslookup followed by the domain name will display the "A Record" (IP Address) of the damain. Use this command to find the address record for a domain. It queries to domain name servers and gets the details.

(b) nslookup -type=any google.com
- Lookup for any record. We can also view all the available DNS records using the -type=any option.

(c) nslookup -type=soa google.com.
- Lookup for an soa record. SOA record (start of authority), provides the authorative information about the domain, the e-mail address of the domain admin, the domain serial number, etc.

(d) nslookup -type=ns google.com
- Lookup of ns record. NS (NameServer) record maps down name to a list of DNS servers authoritative for that domain. It will output the name servers which are associated with the given domain.

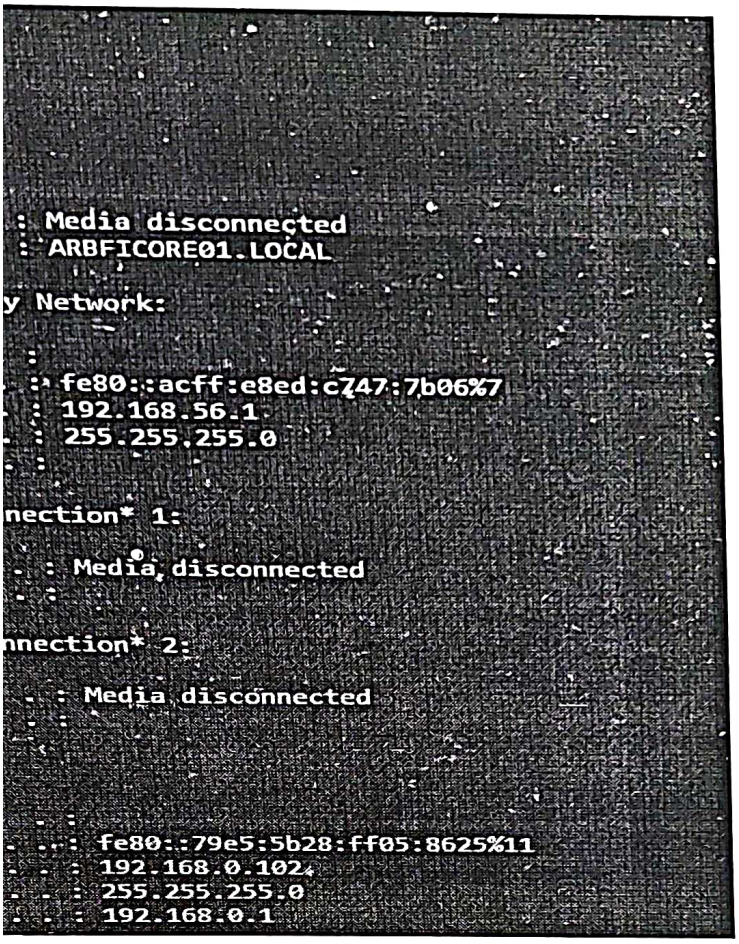
(c) nslookup -type=a google.com
- Lookup for an a record. We can also view all the available DNS records for a particular record using the -type=a option.

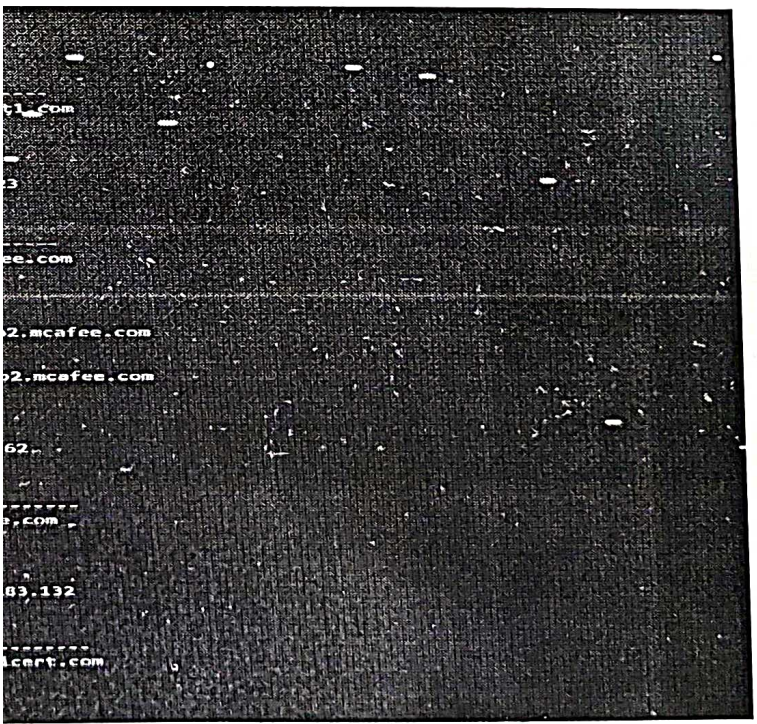
(5) traceroute
- traceroute command prints the route that a packet takes to reach the host. This command is useful when you ~~want~~ want to know about the route and about all the hops that a packet takes.

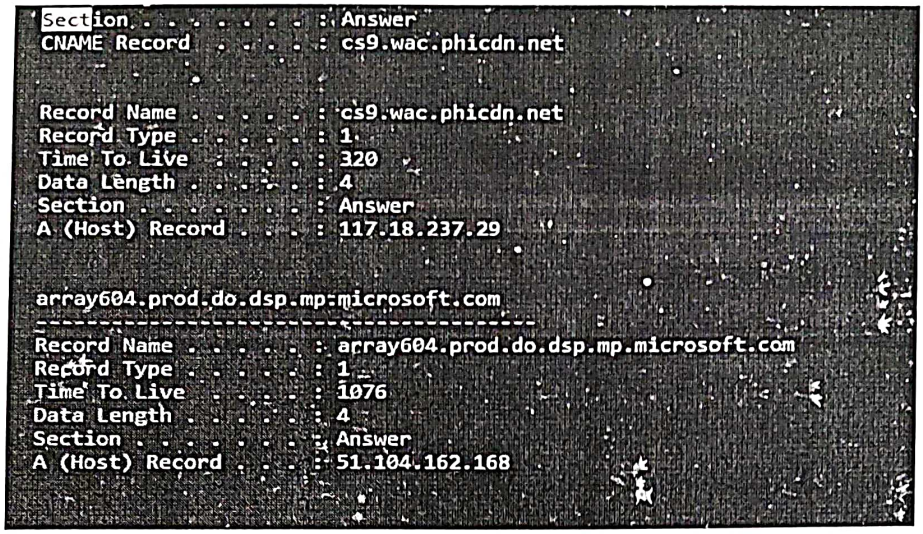
(a) traceroute -4
- Use ip version 4 i.e us IPv4

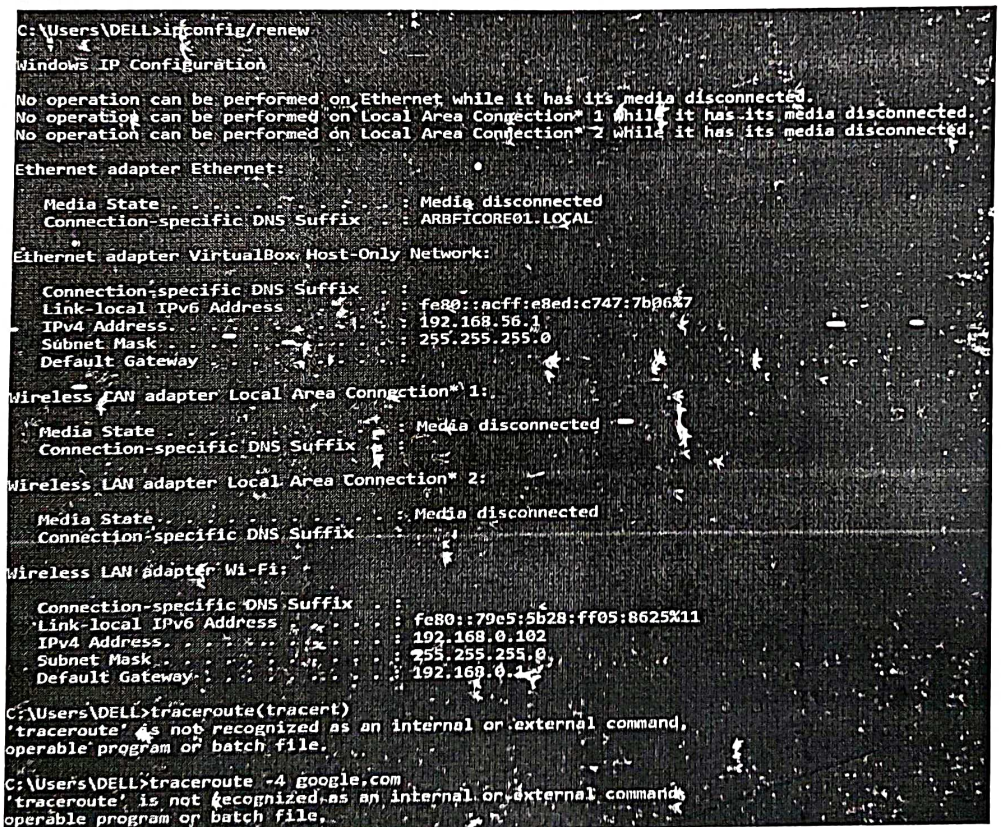
(b) traceroute -6
- Use ip version 6 i.e use IPv6

(c) ~~traceout -F~~ traceroute -F
- Do not fragment packet.


: Media disconnected
: ARBFICORE01.LOCAL
y Network:
fe80::acff:e8ed:c747:7b06%7
192.168.56.1
255.255.255.0
nection* 1:
: Media disconnected
nnection* 2:
: Media disconnected
fe80::79e5:5b28:ff05:8625%11
192.168.0.102
255.255.255.0
192.168.0.1


2.mcafee.com
2.mcafee.com
83.132
icert.com


Section . . . . . : Answer
CNAME Record . . . . . : cs9.wac.phicdn.net
Record Name . . . . . : cs9.wac.phicdn.net
Record Type . . . . . : 1
Time To Live . . . . . : 320
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . : 117.18.237.29
array604.prod.do.dsp.mp.microsoft.com
Record Name . . . . . : array604.prod.do.dsp.mp.microsoft.com
Record Type . . . . . : 1
Time To Live . . . . . : 1076
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . : 51.104.162.168


C:\Users\DELL>ipconfig/renew
Windows IP Configuration
No operation can be performed on Ethernet 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.
Ethernet adapter Ethernet:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : ARBFICORE01.LOCAL
Ethernet adapter VirtualBox Host-Only Network:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::acff:e8ed:c747:7b06%7
IPv4 Address. . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
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::79e5:5b28:ff05:8625%11
IPv4 Address. . . . . : 192.168.0.102
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1
C:\Users\DELL>traceroute(tracert)
'traceroute' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\DELL>traceroute -4 google.com
'traceroute' is not recognized as an internal or external command,
operable program or batch file.

Traceroute

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

Tracing route to google.com [172.217.160.206]
over a maximum of 30 hops:

  1   467 ms     3 ms     3 ms  192.168.0.1
  2     7 ms     4 ms     5 ms  reliance.reliance [192.168.29.1]
  3    10 ms     6 ms     9 ms  10.37.248.1
  4    16 ms     7 ms    12 ms  172.31.0.238
  5    11 ms     7 ms     8 ms  192.168.53.188
  6    12 ms     7 ms     7 ms  172.26.76.213
  7     9 ms     7 ms    28 ms  172.26.76.195
  8     7 ms     9 ms    10 ms  192.168.53.178
```

```
:\Users\DELL>tracert -g google.com
g is not a valid command option.

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.
```

```
:\Users\DELL>tracert -d google.com

racing route to google.com [172.217.160.206]
ver a maximum of 30 hops:

 1     6 ms     5 ms     3 ms  192.168.0.1
 2    14 ms     7 ms     4 ms  192.168.29.1
 3    10 ms     6 ms     4 ms  10.37.248.1
 4    10 ms    16 ms    10 ms  172.31.0.208
 5    13 ms     9 ms    10 ms  192.168.53.188
 6    10 ms     9 ms     8 ms  172.26.76.213
 7    12 ms     9 ms     7 ms  172.26.76.195
 8    42 ms    13 ms    13 ms  192.168.53.176
 9    12 ms    13 ms    10 ms  192.168.53.177
10    12 ms    11 ms     9 ms  172.31.2.71
11    12 ms    15 ms    10 ms  74.125.51.166
12    10 ms     7 ms     8 ms  142.251.76.31
13    14 ms    13 ms    11 ms  216.239.47.149
14    11 ms    10 ms    10 ms  172.217.160.206

race complete.
```

-Ping

```
C:\Users\DELL>ping google.com

Pinging google.com [142.251.42.14] with 32 bytes of data:
Reply from 142.251.42.14: bytes=32 time=21ms TTL=52
Reply from 142.251.42.14: bytes=32 time=11ms TTL=52
Reply from 142.251.42.14: bytes=32 time=8ms TTL=52
Reply from 142.251.42.14: bytes=32 time=12ms TTL=52

Ping statistics for 142.251.42.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 21ms, Average = 13ms
```

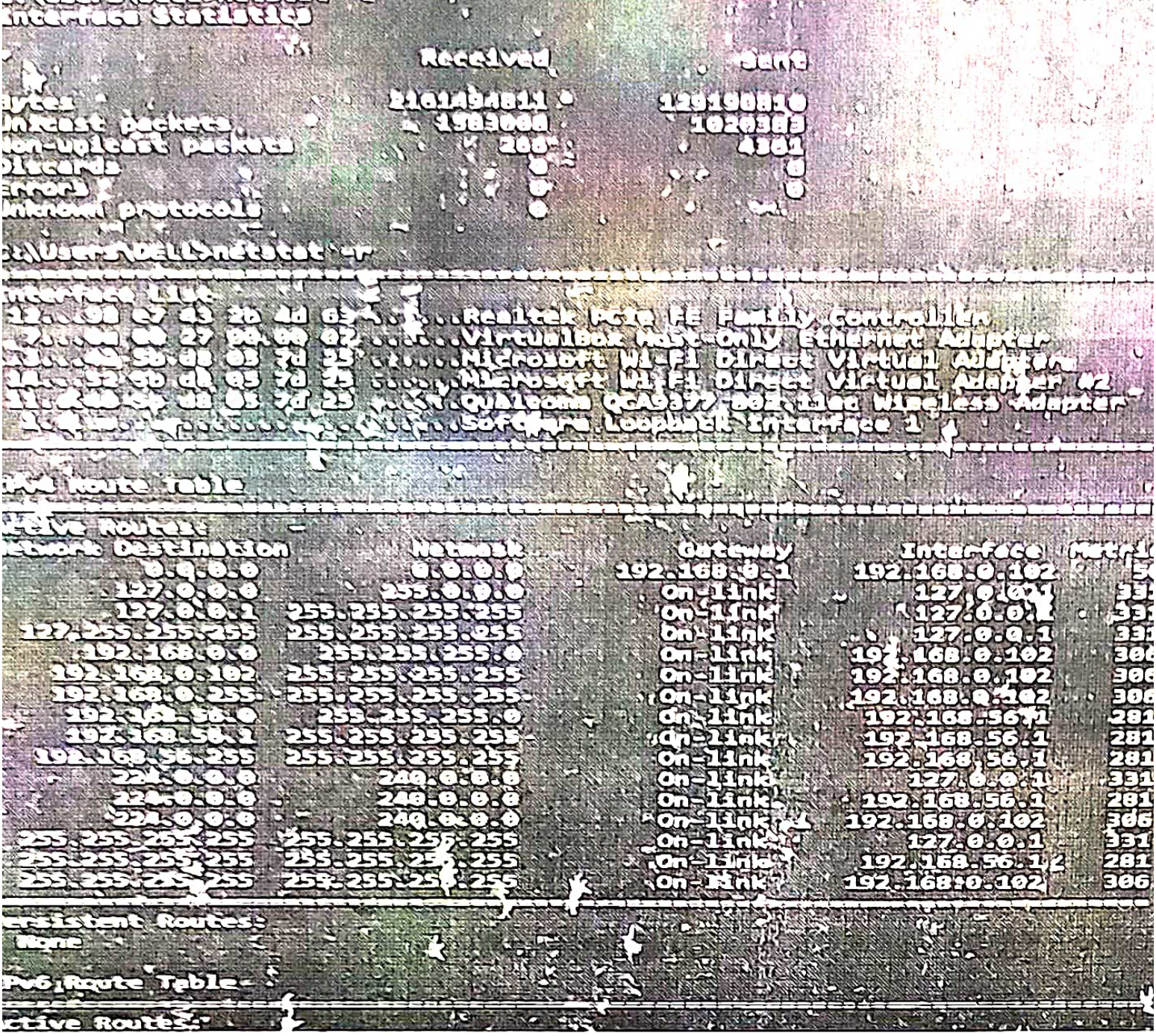
```
C:\Users\DELL>ping -n 10 google.com

Pinging google.com [142.251.42.14] with 32 bytes of data:
Reply from 142.251.42.14: bytes=32 time=13ms TTL=52
Reply from 142.251.42.14: bytes=32 time=17ms TTL=52
Reply from 142.251.42.14: bytes=32 time=30ms TTL=52
Reply from 142.251.42.14: bytes=32 time=11ms TTL=52
Reply from 142.251.42.14: bytes=32 time=13ms TTL=52
Reply from 142.251.42.14: bytes=32 time=16ms TTL=52
Reply from 142.251.42.14: bytes=32 time=12ms TTL=52
Reply from 142.251.42.14: bytes=32 time=15ms TTL=52
Reply from 142.251.42.14: bytes=32 time=24ms TTL=52
Reply from 142.251.42.14: bytes=32 time=20ms TTL=52

Ping statistics for 142.251.42.14:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 30ms, Average = 17ms
```

```
C:\Users\DELL>ping -i 1 google.com

Pinging google.com [142.251.42.14] with 32 bytes of data:
Reply from 192.168.0.1: TTL expired in transit.
Reply from 192.168.0.1: TTL expired in transit.
Reply from 192.168.0.1: TTL expired in transit.
Reply from 192.168.0.1: TTL expired in transit.

Ping statistics for 142.251.42.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\DELL>ping -s 1000 google.com
Bad value for option -s, valid range is from 1 to 4.
```

```
C:\Users\DELL>ping -a google.com

Pinging google.com [142.251.42.14] with 32 bytes of data:
Reply from 142.251.42.14: bytes=32 time=13ms TTL=52
Reply from 142.251.42.14: bytes=32 time=13ms TTL=52
Reply from 142.251.42.14: bytes=32 time=26ms TTL=52
Reply from 142.251.42.14: bytes=32 time=12ms TTL=52

Ping statistics for 142.251.42.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 26ms, Average = 16ms
```

netstat

```
C:\Users\DELL>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49671        DESKTOP-EJAR4D4:49672  ESTABLISHED
  TCP    127.0.0.1:49672        DESKTOP-EJAR4D4:49671  ESTABLISHED
  TCP    127.0.0.1:49673        DESKTOP-EJAR4D4:49674  ESTABLISHED
  TCP    127.0.0.1:49674        DESKTOP-EJAR4D4:49673  ESTABLISHED
  TCP    127.0.0.1:56162        DESKTOP-EJAR4D4:56163  ESTABLISHED
  TCP    127.0.0.1:56163        DESKTOP-EJAR4D4:56162  ESTABLISHED
  TCP    127.0.0.1:56164        DESKTOP-EJAR4D4:56165  ESTABLISHED
  TCP    127.0.0.1:56165        DESKTOP-EJAR4D4:56164  ESTABLISHED
  TCP    127.0.0.1:56166        DESKTOP-EJAR4D4:56167  ESTABLISHED
  TCP    127.0.0.1:56167        DESKTOP-EJAR4D4:56166  ESTABLISHED
  TCP    127.0.0.1:56169        DESKTOP-EJAR4D4:56170  ESTABLISHED
  TCP    127.0.0.1:56170        DESKTOP-EJAR4D4:56169  ESTABLISHED
  TCP    127.0.0.1:56659        DESKTOP-EJAR4D4:27300  SYN_SENT
  TCP    192.168.0.102:55934    20.198.162.76:https    ESTABLISHED
  TCP    192.168.0.102:55935    23.98.104.194:https    ESTABLISHED
  TCP    192.168.0.102:56119    52.114.44.78:https     ESTABLISHED
  TCP    192.168.0.102:56149    sd-in-f188:5228        ESTABLISHED
```



```
Interface Statistics

                           Received            Sent

Bytes                    21014948811      1291908810
Unicast packets             1983008         1020363
Non-unicast packets             200            4301
Discards                          0               0
Errors                            0               0
Unknown protocols                 0

C:\Users\DELL>netstat -r

Interface List
 11...08 e7 43 2b 4d 63 ......Realtek PCIe FE Family Controller
  7...0a 00 27 00 00 07 ......VirtualBox Host-Only Ethernet Adapter
 13...50 0b 91 7d 25 ......Microsoft Wi-Fi Direct Virtual Adapter
 14...52 50 d1 95 7d 25 ......Microsoft Wi-Fi Direct Virtual Adapter #2
 11...c8 3a 35 91 7d 25 ......Qualcomm QCA9377 802.11ac Wireless Adapter
  1...........................Software Loopback Interface 1

IPv4 Route Table

Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.0.1    192.168.0.102     50
        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     192.168.0.102    306
    192.168.0.102  255.255.255.255         On-link     192.168.0.102    306
    192.168.0.255  255.255.255.255         On-link     192.168.0.102    306
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
        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      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link     192.168.0.102    306
  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      192.168.56.1    281
  255.255.255.255  255.255.255.255         On-link     192.168.0.102    306
Persistent Routes:
  None

IPv6 Route Table

Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
  7    281 fe80::/64                On-link
 11    306 fe80::/64                On-link
 11    306 fe80::79e5:5b28:ff05:8625/128
                                    On-link
  7    281 fe80::acff:e8ed:c747:7b06/128
                                    On-link
  1    331 ff00::/8                 On-link
  7    281 ff00::/8                 On-link
 11    306 ff00::/8                 On-link

Persistent Routes:
  None
```

-nslookup

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

25/01/22

Assignment-2

Aim: Installation and configuration of NS2 and implementation of TCL Hello programming.

Theory:

NS 2:-

NS2 stands for Network Simulator Version 2. It is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

To install NS2 in Linux we have use the command "sudo apt install ns2".

TCL Hello programming:-

set ns [new Simulator]

→ Creates a new simulator by using the above command.

$ns at 1 "puts \"hello world\""

→ To print text on screen we use puts command here we schedule an event to write at the time 1 - By this script simulator instance "ns" writes "hello world".

$ns at 1.5 "exit"

→ Now we have to exit the simulator object.

$ns run

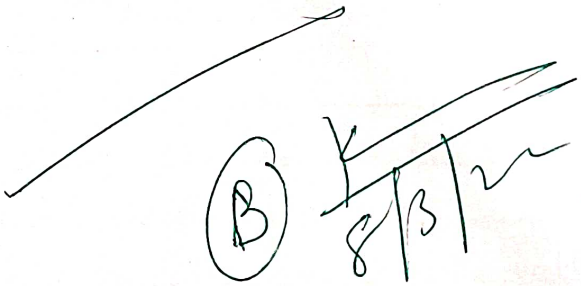
→ To run the simulator object we to use the above command.

Conclusion :- Hence, we have learned about NS2 and implemented the first TCL hello program.

Open HelloWorld.tcl ~/Documents/CNND/Programs Save

```
1 set ns [new Simulator]
2 $ns at 1 "puts \"hello world\""
3 $ns at 1.5 "exit"
4 $ns run
```

```
krishna@krishna-VirtualBox:~/Documents/CNND/Programs$ ns HelloWorld.tcl
hello world
```



7/02/22

Assignment-3

Aim: Implementation of specific network topology with respect to TCP.

Theory:

- Set ns [new Simulator]:

→ Creates a new simulator, and assigns it to variable ns

- $ns color 1 blue

→ This command is used to set color of the packets for a flow specified by the flow id ($1 (1).

- set nf [open out.nam w]

→ Opening out.nam file in write mode and assign it to variable nf.

- ~~$ns nam-trace~~

- $ns namtrace-all $nf.

→ The member function tells the simulator to record simulation traces in NAM input format.

Similarly we will also open trace file.

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

→ Make ns nf np global

flush-trace command flushes the trace buffer and typicalled called before simulation run ends.

Close the Nam trace file, and executed nam on the trace file within the program then exit the finish procedure.

• Set n0 [$ns node]
→ The member function node creates a node. A node in NS is compound object made of address and port classifiers.

• $ns duplex-link $n0 $n1 2Mb 10ms Droptail
→ It creates two simplex links of specified bandwidth and delay, and connects the two specified nodes. Here, n_0, n_1 are the nodes, 2Mb is bandwidth, 10ms is delay and Droptail is queue type. So, we are basically creating the links between the nodes by using the above command.

• $ns queue-limit $n0 $n1 5
→ This line sets the queue limit of the two simplex links that connect node1 and node2 to the number specified.

$ns duplex-link-op $n0 $n1 queuePos 0.5
→ This line is used to position the nodes for nam-display.

Set tcp [new Agent/TCP]
→ This line shows how to create TCP agent

$ns attach-agent $n0 $tcp
→ The attach-agent member function attaches an agent

object created to a node object.

set sink [new Agent/TCPSink]
→ This line shows how to create TCPSink agents.

$ns connect $tcp $sink.
→ After two agents that will communicate with each-other are created, the next thing is to establish a logical network connection between them.

set ftp [new Application/FTP]
→ Create an FTP source "application".

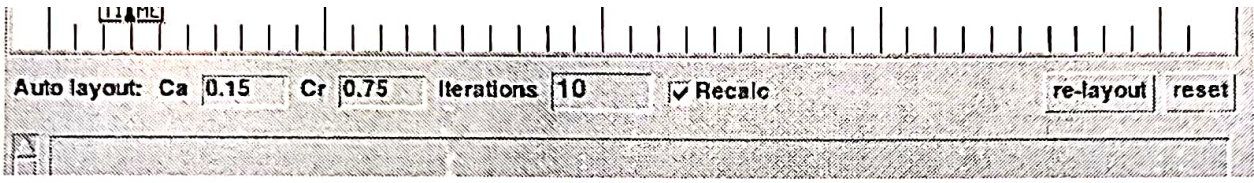
$ftp attach-agent $tcp
→ Setsup a FTP over TCP connection.

$ns at 0.1 "$ftp start"
→ This member function of a simulator object makes the scheduler to schedule the execution of the specified string at given simulation time.

$ns at 5.0 "finish"
→ Calls the finish procedure after 5 seconds of simulation time.

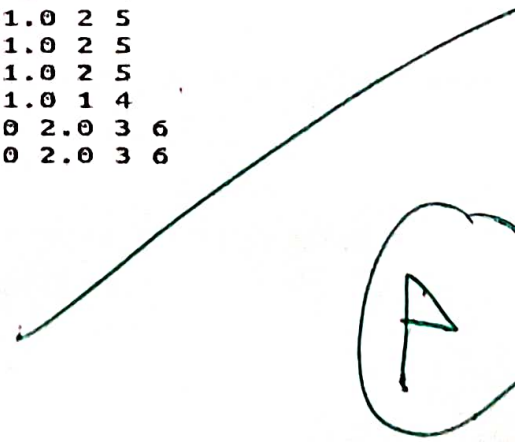
$ns run.
→ Above command is used to run the simulation.

Conclusion:- Hence, we learned how tcp connections are made by implementing in ns (network simulator), NAM (Network animation) is used for node to node connections.



Open out.tr ~/Documents/CNND/Programs/Tcp

```
 1 + 0.1 1 0 tcp 40 ------- 1 1.0 2.0 0 0
 2 - 0.1 1 0 tcp 40 ------- 1 1.0 2.0 0 0
 3 r 0.11016 1 0 tcp 40 ------- 1 1.0 2.0 0 0
 4 + 0.11016 0 2 tcp 40 ------- 1 1.0 2.0 0 0
 5 - 0.11016 0 2 tcp 40 ------- 1 1.0 2.0 0 0
 6 r 0.11532 0 2 tcp 40 ------- 1 1.0 2.0 0 0
 7 + 0.11532 2 0 ack 40 ------- 1 2.0 1.0 0 1
 8 - 0.11532 2 0 ack 40 ------- 1 2.0 1.0 0 1
 9 r 0.12048 2 0 ack 40 ------- 1 2.0 1.0 0 1
10 + 0.12048 0 1 ack 40 ------- 1 2.0 1.0 0 1
11 - 0.12048 0 1 ack 40 ------- 1 2.0 1.0 0 1
12 r 0.13064 0 1 ack 40 ------- 1 2.0 1.0 0 1
13 + 0.13064 1 0 tcp 1040 ------- 1 1.0 2.0 1 2
14 - 0.13064 1 0 tcp 1040 ------- 1 1.0 2.0 1 2
15 + 0.13064 1 0 tcp 1040 ------- 1 1.0 2.0 2 3
16 - 0.1348 1 0 tcp 1040 ------- 1 1.0 2.0 2 3
17 r 0.1448 1 0 tcp 1040 ------- 1 1.0 2.0 1 2
18 + 0.1448 0 2 tcp 1040 ------- 1 1.0 2.0 1 2
19 - 0.1448 0 2 tcp 1040 ------- 1 1.0 2.0 1 2
20 r 0.14896 1 0 tcp 1040 ------- 1 1.0 2.0 2 3
21 + 0.14896 0 2 tcp 1040 ------- 1 1.0 2.0 2 3
22 - 0.14896 0 2 tcp 1040 ------- 1 1.0 2.0 2 3
23 r 0.15396 0 2 tcp 1040 ------- 1 1.0 2.0 1 2
24 + 0.15396 2 0 ack 40 ------- 1 2.0 1.0 1 4
25 - 0.15396 2 0 ack 40 ------- 1 2.0 1.0 1 4
26 r 0.15812 0 2 tcp 1040 ------- 1 1.0 2.0 2 3
27 + 0.15812 2 0 ack 40 ------- 1 2.0 1.0 2 5
28 - 0.15812 2 0 ack 40 ------- 1 2.0 1.0 2 5
29 r 0.15912 2 0 ack 40 ------- 1 2.0 1.0 1 4
30 + 0.15912 0 1 ack 40 ------- 1 2.0 1.0 1 4
31 - 0.15912 0 1 ack 40 ------- 1 2.0 1.0 1 4
32 r 0.16328 2 0 ack 40 ------- 1 2.0 1.0 2 5
33 + 0.16328 0 1 ack 40 ------- 1 2.0 1.0 2 5
34 - 0.16328 0 1 ack 40 ------- 1 2.0 1.0 2 5
35 r 0.16928 0 1 ack 40 ------- 1 2.0 1.0 1 4
36 + 0.16928 1 0 tcp 1040 ------- 1 1.0 2.0 3 6
37 - 0.16928 1 0 tcp 1040 ------- 1 1.0 2.0 3 6
```



Assignment-3

Code:

```
#Create a simulator object
set ns [new Simulator]
$ns color 1 blue

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

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

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

#Create two nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
#Create links between the nodes
$ns duplex-link $n0 $n1 2Mb 10ms DropTail
$ns duplex-link $n0 $n2 2Mb 5ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n0 $n1 5
$ns queue-limit $n0 $n2 2
#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n0 $n1 queuePos 0.5
$ns duplex-link-op $n0 $n2 queuePos 1.0

#Setup a TCP connection
set tcp [new Agent/TCP]

$ns attach-agent $n0 $tcp
$ns attach-agent $n1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns attach-agent $n2 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

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


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


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

#Run the simulation
$ns run
```

15/02/2024

Assignment-4

Aim:- Implementation of specific network topology with respect to UDP.

Theory:-

UDP:-

User Datagram Protocol (UDP) is a transport layer protocol (TCL). UDP is a part of the internet protocol suite, reffered to as UDP/IP suite. Unlike TCP, it's is an unreliable and connectionless protocol. So, there is no need to establish a connection prior to data transfer.

Orient:-

Orient short form for Orientation, this is used to decide the position of a node in the output screen. The different types of orientation available in linux are: top, bottom, left, right, left-top

ble bitrate. When referring to codes, constant
ding means that a the rate at which a codec's
ta should be consumed is constant.

nplemented specific network topology with
UDP.

Assignment-4

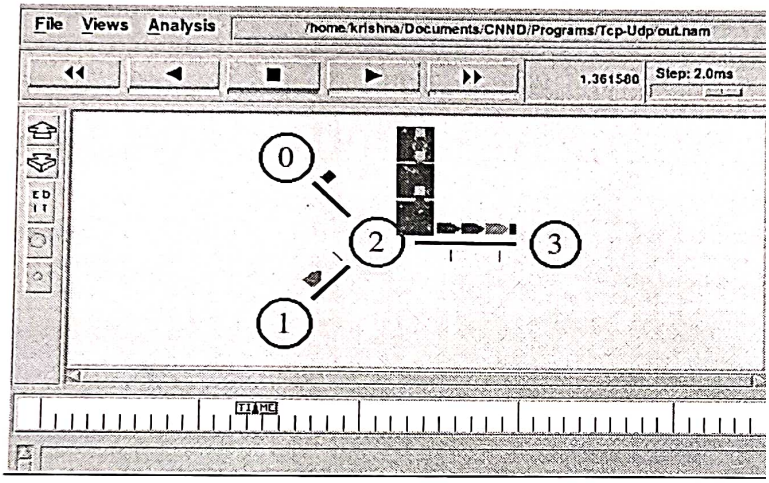
Code:

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

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

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

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

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

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

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

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

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

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

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

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


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


# setting packet size
$cbr set packet_size_ 1000

#setting bit rate
$cbr set rate_ 1mb

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

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

```
$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n

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

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

#Run the simulation
$ns run
```

Output:



Open — out.tr — /Documents/CNND/Programs/Tcp-Udp

```
 1 + 0.1 1 2 cbr 1000 ------- 2 1.0 3.1 0 0
 2 - 0.1 1 2 cbr 1000 ------- 2 1.0 3.1 0 0
 3 + 0.108 1 2 cbr 1000 ------- 2 1.0 3.1 1 1
 4 - 0.108 1 2 cbr 1000 ------- 2 1.0 3.1 1 1
 5 r 0.114 1 2 cbr 1000 ------- 2 1.0 3.1 0 0
 6 + 0.114 2 3 cbr 1000 ------- 2 1.0 3.1 0 0
 7 - 0.114 2 3 cbr 1000 ------- 2 1.0 3.1 0 0
 8 + 0.116 1 2 cbr 1000 ------- 2 1.0 3.1 2 2
 9 - 0.116 1 2 cbr 1000 ------- 2 1.0 3.1 2 2
10 r 0.122 1 2 cbr 1000 ------- 2 1.0 3.1 1 1
11 + 0.122 2 3 cbr 1000 ------- 2 1.0 3.1 1 1
12 - 0.122 2 3 cbr 1000 ------- 2 1.0 3.1 1 1
13 + 0.124 1 2 cbr 1000 ------- 2 1.0 3.1 3 3
14 - 0.124 1 2 cbr 1000 ------- 2 1.0 3.1 3 3
15 r 0.13 1 2 cbr 1000 ------- 2 1.0 3.1 2 2
16 + 0.13 2 3 cbr 1000 ------- 2 1.0 3.1 2 2
17 - 0.13 2 3 cbr 1000 ------- 2 1.0 3.1 2 2
18 + 0.132 1 2 cbr 1000 ------- 2 1.0 3.1 4 4
19 - 0.132 1 2 cbr 1000 ------- 2 1.0 3.1 4 4
20 r 0.138 1 2 cbr 1000 ------- 2 1.0 3.1 3 3
21 + 0.138 2 3 cbr 1000 ------- 2 1.0 3.1 3 3
22 - 0.138 2 3 cbr 1000 ------- 2 1.0 3.1 3 3
23 r 0.138706 2 3 cbr 1000 ------- 2 1.0 3.1 0 0
24 + 0.14 1 2 cbr 1000 ------- 2 1.0 3.1 5 5
25 - 0.14 1 2 cbr 1000 ------- 2 1.0 3.1 5 5
26 r 0.146 1 2 cbr 1000 ------- 2 1.0 3.1 4 4
27 + 0.146 2 3 cbr 1000 ------- 2 1.0 3.1 4 4
28 - 0.146 2 3 cbr 1000 ------- 2 1.0 3.1 4 4
29 r 0.146706 2 3 cbr 1000 ------- 2 1.0 3.1 1 1
30 + 0.148 1 2 cbr 1000 ------- 2 1.0 3.1 6 6
31 - 0.148 1 2 cbr 1000 ------- 2 1.0 3.1 6 6
32 r 0.154 1 2 cbr 1000 ------- 2 1.0 3.1 5 5
33 + 0.154 2 3 cbr 1000 ------- 2 1.0 3.1 5 5
34 - 0.154 2 3 cbr 1000 ------- 2 1.0 3.1 5 5
35 r 0.154706 2 3 cbr 1000 ------- 2 1.0 3.1 2 2
36 + 0.156 1 2 cbr 1000 ------- 2 1.0 3.1 7 7
37 - 0.156 1 2 cbr 1000 ------- 2 1.0 3.1 7 7
```

22/2/2024

Assignment-5

Aim: Simulation of network with specific routing protocols (Distance vector, link state)

Theory :-

Routing Protocols: Routing Protocols are the set of defined rules used by the routers to communicate between source and ~~distance~~ destination. They help computer networks communicate effectively and efficiently.

Distance Vector:-

Distance vector protocols can measure the distance called hops. It takes data to arrive at its destination within a system or application. The number of hops refers to the specific number of routers the data may run through before reaching its destination. These protocol send information to other nearly devices which require large bandwidth for support.

Link state:

It also finds the best routing path and also share information with nearly routers. However, they calculate the speed and the cast of resources associated with each potential path.

Real time Transfer Protocol (RTP):-

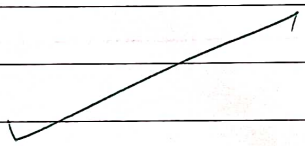
A protocol is designed to handle real time traffic of internet, it is known as RTP. It may be used

with UDP. RTP supports different file formats like HPEG and MIPEG.

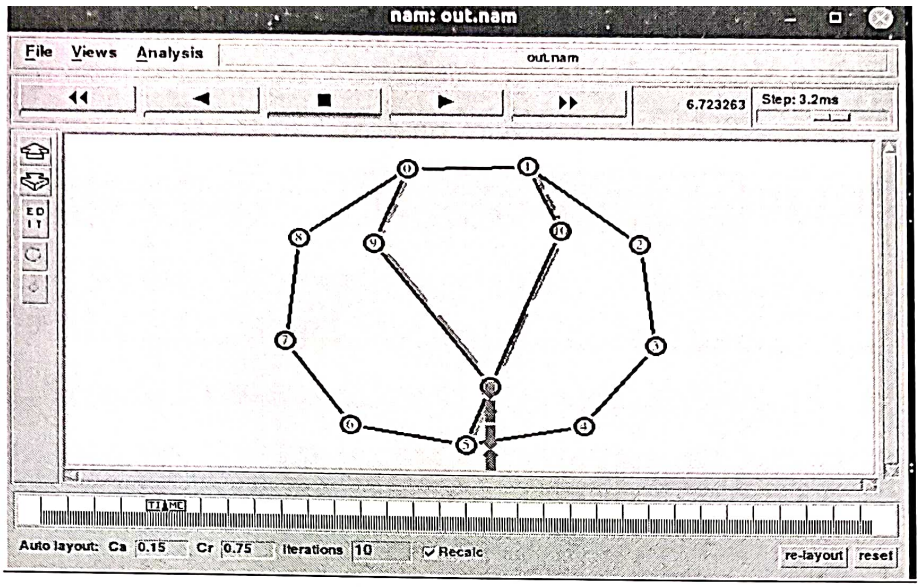
Application of RTP:-

(1) It supports voice Over Internet Protocol (VOIP)
(2) Video Teleconferencing over Internet.
(3) Internet Audio and Video streaming.

Conclusion: Hence, we have created networks on ns (network simulator) with specific routing protocols like Distance Protocol, Link state protocol and RTP.



```
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$ns attach-agent $udp0 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0

$ns rtproto DV

$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)
$udp0 set fid_ 1
$udp1 set fid_ 2

$ns color 1 Red
$ns color 2 Green

$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"

$ns at 45 "finish"
$ns run
```



Open ~/Documents/CNND/Programs/Routing/out.tr Save

```
 1 + 0.00017 0 1 rtProtoDV 12 ------- 0 0.2 1.2 -1 0
 2 - 0.00017 0 1 rtProtoDV 12 ------- 0 0.2 1.2 -1 0
 3 + 0.00017 0 8 rtProtoDV 12 ------- 0 0.2 8.1 -1 1
 4 - 0.00017 0 8 rtProtoDV 12 ------- 0 0.2 8.1 -1 1
 5 + 0.00017 0 9 rtProtoDV 12 ------- 0 0.2 9.1 -1 2
 6 - 0.00017 0 9 rtProtoDV 12 ------- 0 0.2 9.1 -1 2
 7 + 0.007102 2 1 rtProtoDV 12 ------- 0 2.1 1.2 -1 3
 8 - 0.007102 2 1 rtProtoDV 12 ------- 0 2.1 1.2 -1 3
 9 + 0.007102 2 3 rtProtoDV 12 ------- 0 2.1 3.1 -1 4
10 - 0.007102 2 3 rtProtoDV 12 ------- 0 2.1 3.1 -1 4
11 r 0.010266 0 1 rtProtoDV 12 ------- 0 0.2 1.2 -1 0
12 + 0.010266 1 0 rtProtoDV 12 ------- 0 1.2 0.2 -1 5
13 - 0.010266 1 0 rtProtoDV 12 ------- 0 1.2 0.2 -1 5
14 + 0.010266 1 2 rtProtoDV 12 ------- 0 1.2 2.1 -1 6
15 - 0.010266 1 2 rtProtoDV 12 ------- 0 1.2 2.1 -1 6
16 + 0.010266 1 10 rtProtoDV 12 ------- 0 1.2 10.1 -1 7
17 - 0.010266 1 10 rtProtoDV 12 ------- 0 1.2 10.1 -1 7
18 r 0.010266 0 8 rtProtoDV 12 ------- 0 0.2 8.1 -1 1
19 + 0.010266 8 0 rtProtoDV 12 ------- 0 8.1 0.2 -1 8
20 - 0.010266 8 0 rtProtoDV 12 ------- 0 8.1 0.2 -1 8
21 + 0.010266 8 7 rtProtoDV 12 ------- 0 8.1 7.1 -1 9
22 - 0.010266 8 7 rtProtoDV 12 ------- 0 8.1 7.1 -1 9
23 r 0.010266 0 9 rtProtoDV 12 ------- 0 0.2 9.1 -1 2
24 + 0.010266 9 0 rtProtoDV 12 ------- 0 9.1 0.2 -1 10
25 - 0.010266 9 0 rtProtoDV 12 ------- 0 9.1 0.2 -1 10
26 + 0.010266 9 11 rtProtoDV 12 ------- 0 9.1 11.1 -1 11
27 - 0.010266 9 11 rtProtoDV 12 ------- 0 9.1 11.1 -1 11
28 r 0.017198 2 1 rtProtoDV 12 ------- 0 2.1 1.2 -1 3
29 + 0.017198 1 0 rtProtoDV 12 ------- 0 1.2 0.2 -1 12
30 - 0.017198 1 0 rtProtoDV 12 ------- 0 1.2 0.2 -1 12
31 + 0.017198 1 2 rtProtoDV 12 ------- 0 1.2 2.1 -1 13
32 - 0.017198 1 2 rtProtoDV 12 ------- 0 1.2 2.1 -1 13
33 + 0.017198 1 10 rtProtoDV 12 ------- 0 1.2 10.1 -1 14
34 - 0.017198 1 10 rtProtoDV 12 ------- 0 1.2 10.1 -1 14
35 r 0.017198 2 3 rtProtoDV 12 ------- 0 2.1 3.1 -1 4
36 + 0.017198 3 2 rtProtoDV 12 ------- 0 3.1 2.1 -1 15
37 - 0.017198 3 2 rtProtoDV 12 ------- 0 3.1 2.1 -1 15
```

A+ 8/5/22

01/03/22

Assingment-6

Aim:- Installation of wireshark and Analysis of packet headers- TCP, IP, UPP using wireshark

Theory:-

Run the following command to install wireshark on your ubuntu machine:

- $ sudo apt install wireshark

Now press y and then press <Enter>

By default, wireshark must be started as root (can also be done with sudo) privileges in order to work. If you want to run Wireshark without root privileges or without sudo, then select <Yes> and press <Enter>

wireshark should be installed

- $ Sudo groupadd wirshark

→ Creates a group with the name wireshark.

- $ sudo usermod -a -G wireshark krishna

→ By using -a we append the group wireshark with the username (krishna) and lets the user access the group.

- $ Sudo chgrp wireshark /usr/bin/dumpcap

→ Using chgrp command we change the path of the group (wireshark) to the specified path. Here dumpcap provide capabilities to copy network traffic.

- $ sudo chmod 750 /usr/bin/dumpcap

→ chmod used to change or allocate different mods to different user.

7 → Owner has all the rwx access
5 → group has only rw access.
0 → Others no read no write access.

• Sudo ~~apt~~ setcap cap_net_admin, cap_net_admin+eip /usr/bin/dumpcap

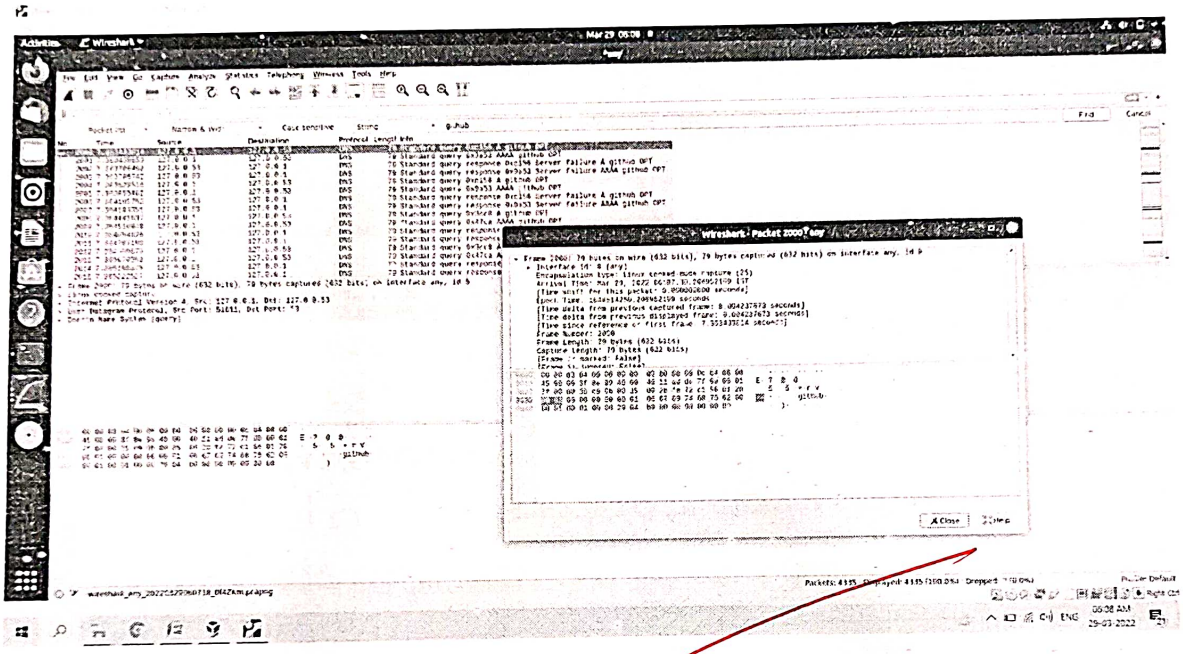
→ setcap :- Setcap sets the capabilities of each file name to the ~~of~~ capabilities to each specified filename.

• Sudo getcap /usr/bin/dumpcap

→ Above command is used to get all the capabilities to the path.

Wireshark :-

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see whats happening on your network at a microscopic level and is the defacto (and often de jure) standard ~~accross~~ across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998.



Page: 10
Date: 15/03/22

Assignment - 7

Aim: Socket Programming with Java - UDP client, UDP server

Theory:-

Socket Programming:

Socket programming is used for communication between the applications running on different JRE. It can be connection oriented or connection-less.

Socket and serversocket classes are used for connection-oriented socket programming. Datagramsocket and Datagrampacket classes are used for connection-less socket programming.

The client in socket programming must know two things
(1) IP Address of server
(2) Port Number.

Datagram Packet:-

In UDP's terms, data transferred is encapsulated in a unit called datagram. It is an independent self-contained message sent over the network whose arrival, time and content are not guranteed. In Java, DatagramPacket represents a datagram.

There are 2 ways to create a DatagramPacket object.
(1) DatagramPacket (byte[] byf, int length)
(2) DatagramPacket (byte[] buf, int length, Inet Address, int port)

Datagramsocket:-

We use Datagramsocket to send and receive datagram packets. This represents a UDP connection between two computers in a network. Basically its the path through which all Datagram Packets travel.

Page: 11
Date:

In Java, we use Datagramsocket for both client and server. There are not seperate TCP sockets for client and server. We create Datagramsocket object to establish a UDP connection for sending and receiving datagram. It can be created using the following constructors:

(1) DatagramSocket()
(2) DatagramSocket(int port)
(3) DatagramSocket(int port, InetAddress ladder)

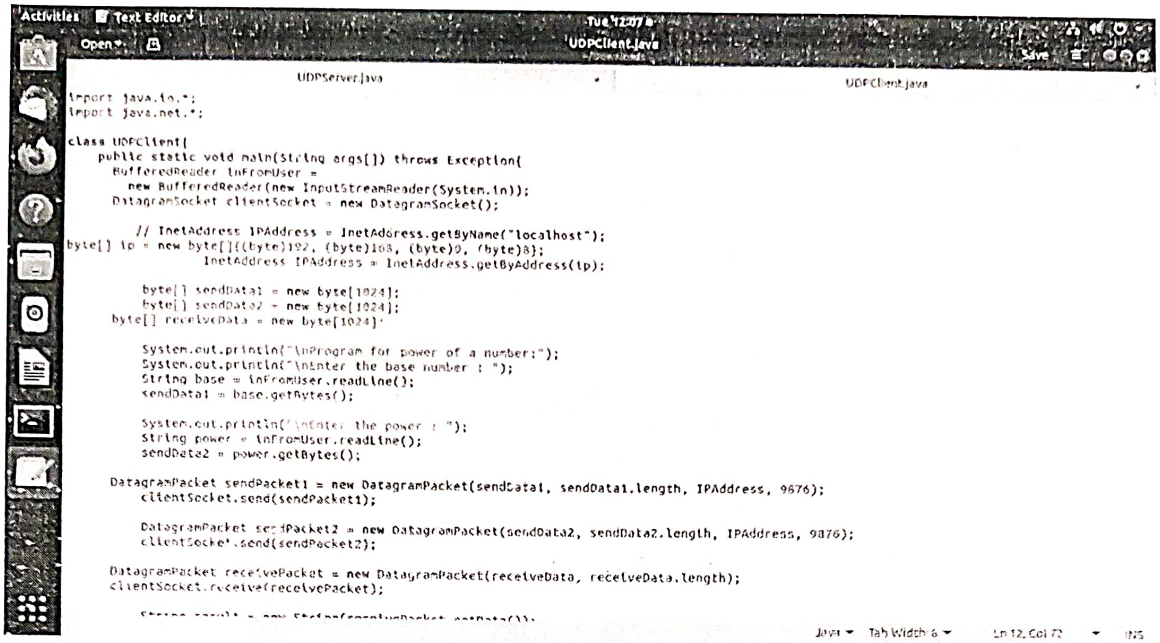
Send():
The send() method of the DatagramPacket class sends a packet from the socket. It contains the data to be sent, its length, IP address and port number of the remote host.

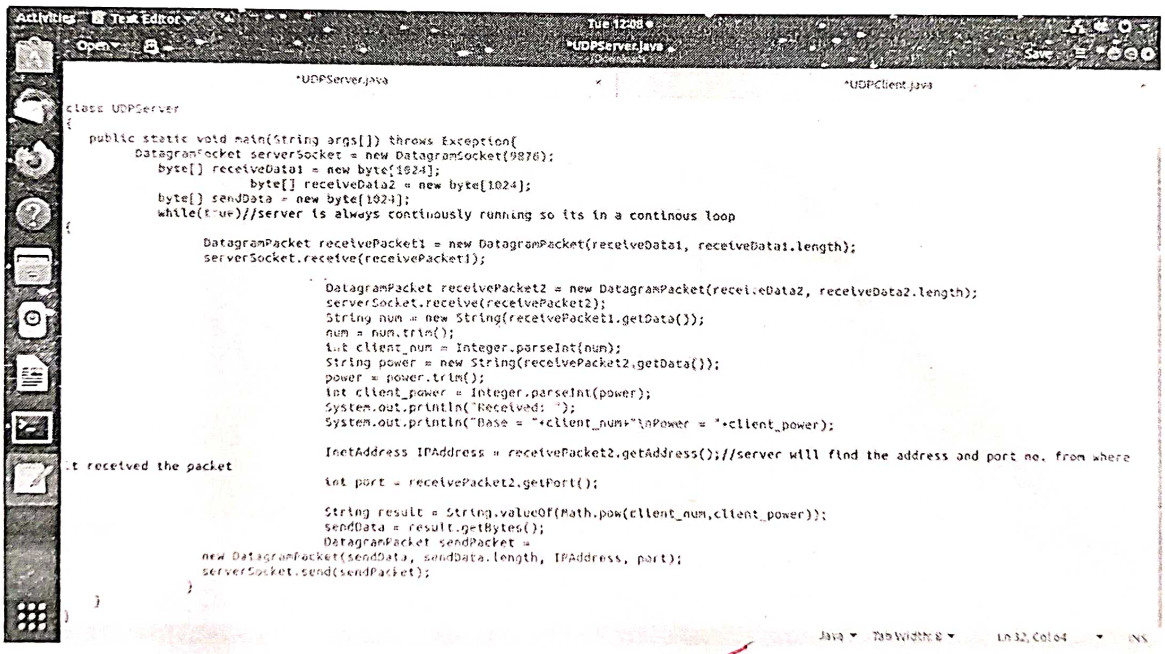
Example:-
DatagramPacket sendPacket1 = new DatagramPacket(sendData1, sendData1.length, IPAddress, 9876);

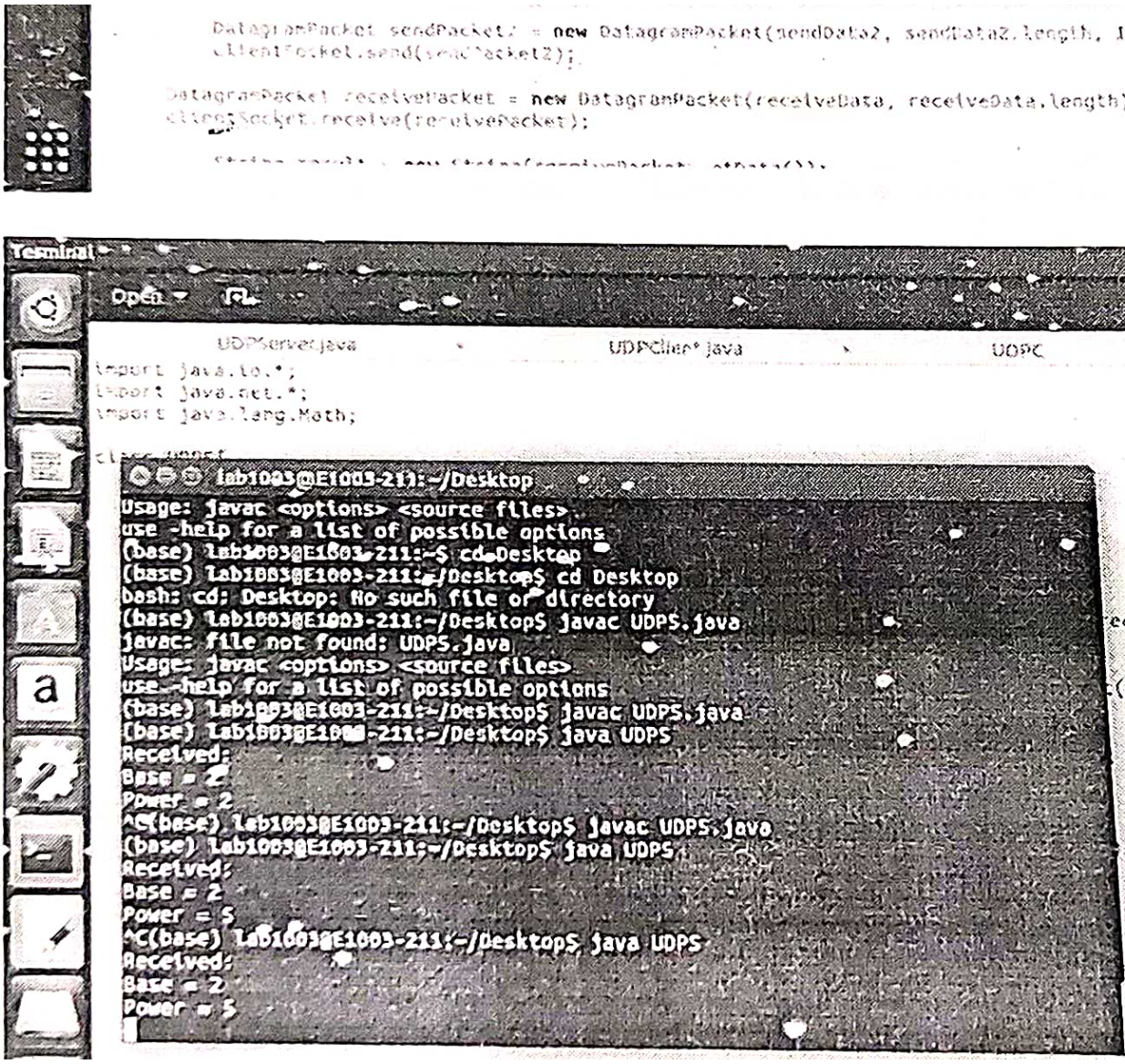
Recieve():
The recieve() method of the datagramPacket class recieves a packet from the socket. It returns the datagram Packet's buffer when it is filled with the data recieved whether the socket is closed or not.

Example:
Datagrampacket recievePacket1 = new DatagramPacket(recieveData1, recieveData.length)

Conclusion:-
We established a connection between ~~TCP client and~~ UDP client and UDP server and also sent and recieved Packets between them.


Activities
Text Editor
Tue 12:07
Open
UDPClient.java
Save
UDPServer.java
UDPClient.java
import java.io.*;
import java.net.*;
class UDPClient{
public static void main(String args[]) throws Exception{
BufferedReader inFromUser =
new BufferedReader(new InputStreamReader(System.in));
DatagramSocket clientSocket = new DatagramSocket();
// InetAddress IPAddress = InetAddress.getByName("localhost");
byte[] ip = new byte[]{(byte)192, (byte)168, (byte)0, (byte)8};
InetAddress IPAddress = InetAddress.getByAddress(ip);
byte[] sendData1 = new byte[1024];
byte[] sendData2 = new byte[1024];
byte[] receiveData = new byte[1024];
System.out.println("\nProgram for power of a number:");
System.out.println("\nEnter the base number : ");
String base = inFromUser.readLine();
sendData1 = base.getBytes();
System.out.println("\nEnter the power : ");
String power = inFromUser.readLine();
sendData2 = power.getBytes();
DatagramPacket sendPacket1 = new DatagramPacket(sendData1, sendData1.length, IPAddress, 9876);
clientSocket.send(sendPacket1);
DatagramPacket sendPacket2 = new DatagramPacket(sendData2, sendData2.length, IPAddress, 9876);
clientSocket.send(sendPacket2);
DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
clientSocket.receive(receivePacket);
Java
Tab Width: 8
Ln 12, Col 72
INS


Activities
Text Editor
Tue 12:08
Open
*UDPServer.java
Save
*UDPServer.java
*UDPClient.java
class UDPServer
{
public static void main(String args[]) throws Exception{
DatagramSocket serverSocket = new DatagramSocket(9876);
byte[] receiveData1 = new byte[1024];
byte[] receiveData2 = new byte[1024];
byte[] sendData = new byte[1024];
while(true)//server is always continously running so its in a continous loop
{
DatagramPacket receivePacket1 = new DatagramPacket(receiveData1, receiveData1.length);
serverSocket.receive(receivePacket1);
DatagramPacket receivePacket2 = new DatagramPacket(receiveData2, receiveData2.length);
serverSocket.receive(receivePacket2);
String num = new String(receivePacket1.getData());
num = num.trim();
int client_num = Integer.parseInt(num);
String power = new String(receivePacket2.getData());
power = power.trim();
int client_power = Integer.parseInt(power);
System.out.println("Received: ");
System.out.println("Base = "+client_num+"\nPower = "+client_power);
InetAddress IPAddress = receivePacket2.getAddress();//server will find the address and port no. from where
t received the packet
int port = receivePacket2.getPort();
String result = String.valueOf(Math.pow(client_num,client_power));
sendData = result.getBytes();
DatagramPacket sendPacket =
new DatagramPacket(sendData, sendData.length, IPAddress, port);
serverSocket.send(sendPacket);
}
}
}
Java
Tab Width: 8
Ln 32, Col 64
INS


DatagramPacket sendPacket2 = new DatagramPacket(sendData2, sendData2.length,
clientSocket.send(sendPacket2);
DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length)
clientSocket.receive(receivePacket);
Terminal
Open
UDPServer.java
UDPClient.java
UDPC
import java.io.*;
import java.net.*;
import java.lang.Math;
lab1003@E1003-211:~/Desktop
Usage: javac <options> <source files>
use -help for a list of possible options
(base) lab1003@E1003-211:~$ cd Desktop
(base) lab1003@E1003-211:~/Desktop$ cd Desktop
bash: cd: Desktop: No such file or directory
(base) lab1003@E1003-211:~/Desktop$ javac UDPS.java
javac: file not found: UDPS.java
Usage: javac <options> <source files>
use -help for a list of possible options
(base) lab1003@E1003-211:~/Desktop$ javac UDPS.java
(base) lab1003@E1003-211:~/Desktop$ java UDPS
Received:
Base = 2
Power = 2
^C(base) lab1003@E1003-211:~/Desktop$ javac UDPS.java
(base) lab1003@E1003-211:~/Desktop$ java UDPS
Received:
Base = 2
Power = 5
^C(base) lab1003@E1003-211:~/Desktop$ java UDPS
Received:
Base = 2
Power = 5

Report for Lab 3-1: UDP

Name: Krishna	Student ID: 38	Date:

1	a. Source port number: 443	b. Destination port number: 46915
	c. Total length of user diagram 1401	d. Length of data 1357
	e. Is the packet from client or server? client	f. Application layer protocol DNS
	g. Is checksum calculated? Yes	
2	Are answer in number 1 are verified by the information in the detail pane lane? Yes	
3	Source and destination IP addresses in the query message: Src: 10.0.2.15 Dst: 172.217.166.36	
	Source and destination IP addresses in the response) message: Src: 172.217.166.36 Dst: 10.0.2.15	
	Relation between IP addresses: It got interchanged	
4	Source and destination port number in the query message: Src: 46915 Dst: 443	
	Source and destination port number in the response message: Src: 46915, Dst: 443	
	Relation between port numbers: Interchanged	
	Which port number is well-known? Destination is well known in query message	
5	The length of the first UDP packet: 84	
	How many bytes of payload are carried by the first UDP packet? —	
6	Number of bytes in the DNS message: —	
	Does the count agree with the answer to question 5?	
7	Is the checksum calculated for the first UDP packet? Yes	
	Value of the checksum: 0x6ddd	

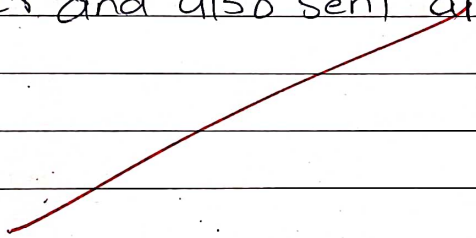
Assignment-8

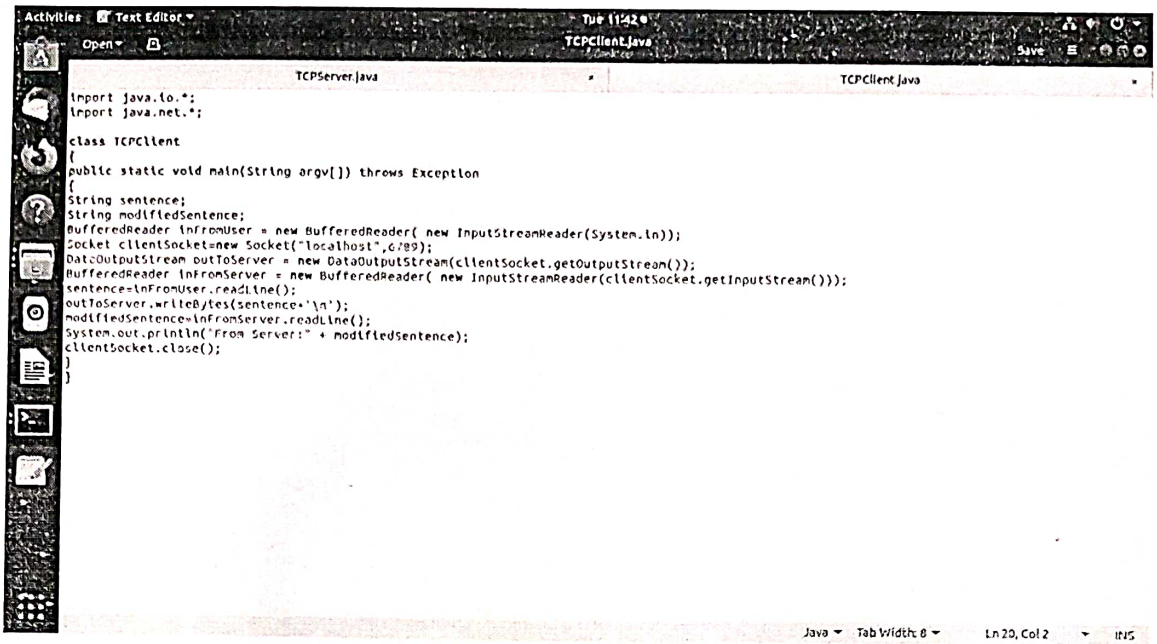
Aim: Socket programming with Ja
server.

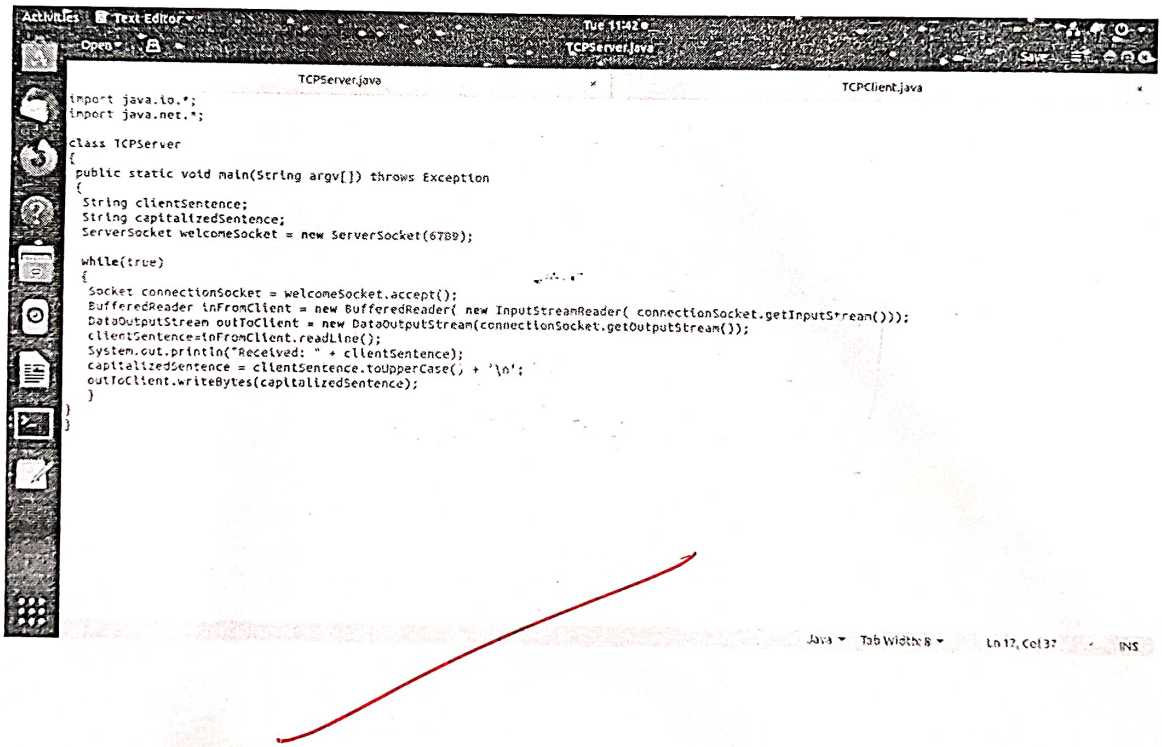
Theory:
TCP is a network protocol that stan
protocol, which allows well-founded c
applications. TCP is consistently use
Protocol; and that is why it is
TCP/IP.
The communication mechanism betu
TCP, can be established using sockets c
~~Cor~~ socket programming.

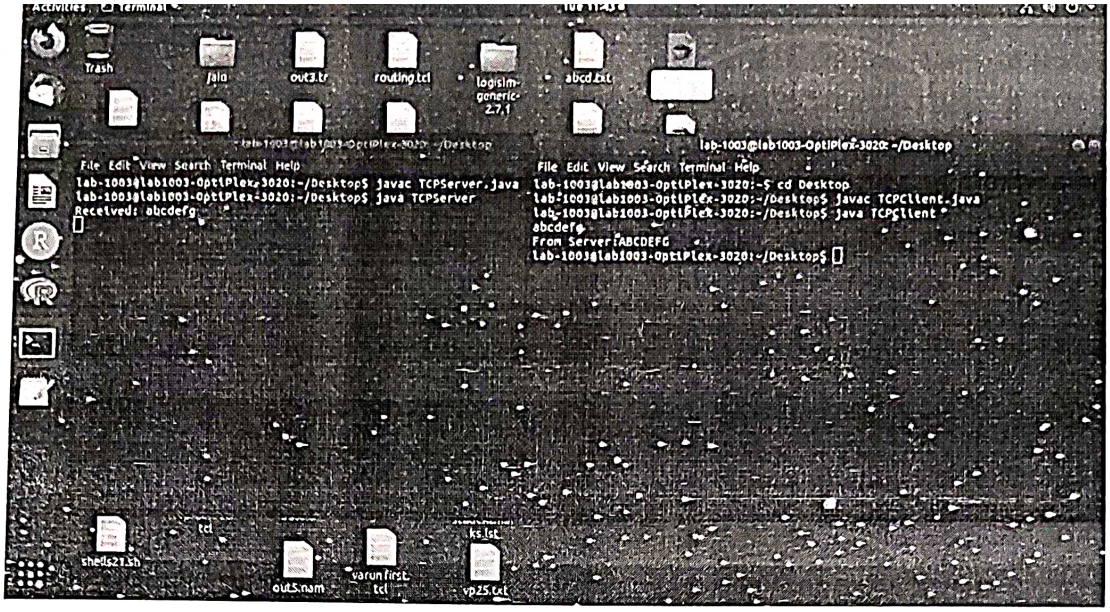
Mechanism for Socket Programming:-
(1) An object of serversocket is ~~for~~ instan
number is specified, on which connectio
place.
(2) The accept method of serversocket is
hold the server in listening mode. This
until a client is connected to the seru
port number.
(3) On the client side, an object of socket
and desired port number and IP addres
the connection.
(4) Since the client is connected to the s
method on the server side resumes, pro
that is capable of communicating to
client

TCP server and also sent and received packets between them.




Activities
Text Editor
Tue 11:42
TCPClient.java
Open
Save
TCPServer.java
TCPClient.java
import java.io.*;
import java.net.*;
class TCPClient
{
public static void main(String argv[]) throws Exception
{
String sentence;
String modifiedSentence;
BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
Socket clientSocket=new Socket("localhost",6789);
DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
sentence=inFromUser.readLine();
outToServer.writeBytes(sentence+'\n');
modifiedSentence=inFromServer.readLine();
System.out.println("From Server:" + modifiedSentence);
clientSocket.close();
}
}
Java
Tab Width: 8
Ln 20, Col 2
INS


Activities
Text Editor
Tue 11:42
TCPServer.java
Open
Save
TCPServer.java
TCPClient.java
import java.io.*;
import java.net.*;
class TCPServer
{
public static void main(String argv[]) throws Exception
{
String clientSentence;
String capitalizedSentence;
ServerSocket welcomeSocket = new ServerSocket(6789);
while(true)
{
Socket connectionSocket = welcomeSocket.accept();
BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));
DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());
clientSentence=inFromClient.readLine();
System.out.println("Received: " + clientSentence);
capitalizedSentence = clientSentence.toUpperCase() + '\n';
outToClient.writeBytes(capitalizedSentence);
}
}
}
Java
Tab Width: 8
Ln 12, Col 37
INS


Activities
Terminal
Trash
jala
out3.tr
routing.tcl
logisim-generic-2.7.1
abcd.txt
File Edit View Search Terminal Help
lab-1003@lab1003-OptiPlex-3020:~/Desktop$ javac TCPServer.java
lab-1003@lab1003-OptiPlex-3020:~/Desktop$ java TCPServer
Received: abcdefg
lab-1003@lab1003-OptiPlex-3020:~/Desktop
File Edit View Search Terminal Help
lab-1003@lab1003-OptiPlex-3020:~$ cd Desktop
lab-1003@lab1003-OptiPlex-3020:~/Desktop$ javac TCPClient.java
lab-1003@lab1003-OptiPlex-3020:~/Desktop$ java TCPClient
abcdefg
From Server:ABCDEFG
lab-1003@lab1003-OptiPlex-3020:~/Desktop$
tcl
shells21.sh
out5.nam
varun first.tcl
ks.tcl
vp25.txt

17

Report for LAB 3-2: TCP

Name: Krishna	Student ID: 38	Date:

	Part I	
1	Socket addresses: (IP)	34.120.237.76 443
2	Set flags:	0x010 (ACK)
3	Sequence number and acknowledgement number:	15210023, 688316129
4	Window size:	65535

	Part II	
1	Set flag in HTTP GET message:	0x4000, Dont Fragment
2	Number of bytes transmitted by the HTTP GET message:	255
3	Acknowledgement frequ...	

Part III		
1	Number of TCP segments exchanged for connection termination: 3	
1	Which end point started the connection termination phase? client	
2	Flags sets in each of the segments used for connection termination: Fin + ACK	

Part IV			
1	a. Source port number: 33968	b. Destination port number: 443	
	c. Sequence number 3166440215	d. Acknowledgement number 0	
	e. Heather length: 40 bytes	f. Set flags: 0x002 (SYN)	
	g. Window size: 64240	h. Urgent pointer: 0	
2	Are answer in the question number 1 verified by the information in the detail pane lane? Yes		
3	Does any of the TCP packet headers carry options? Yes Explain TCP has provision for option header fields identified by an option field.		
4	Size of a TCP packet with no option: 20 bytes Size of a TCP packet with options: 60 bytes		
5	Is window size in any of the TCP packet zero? Yess Explain: If window size in any of the tcp packet is zero then we cannot recieve any further message		

dump to our

ets with ~~te~~

tcp packets.

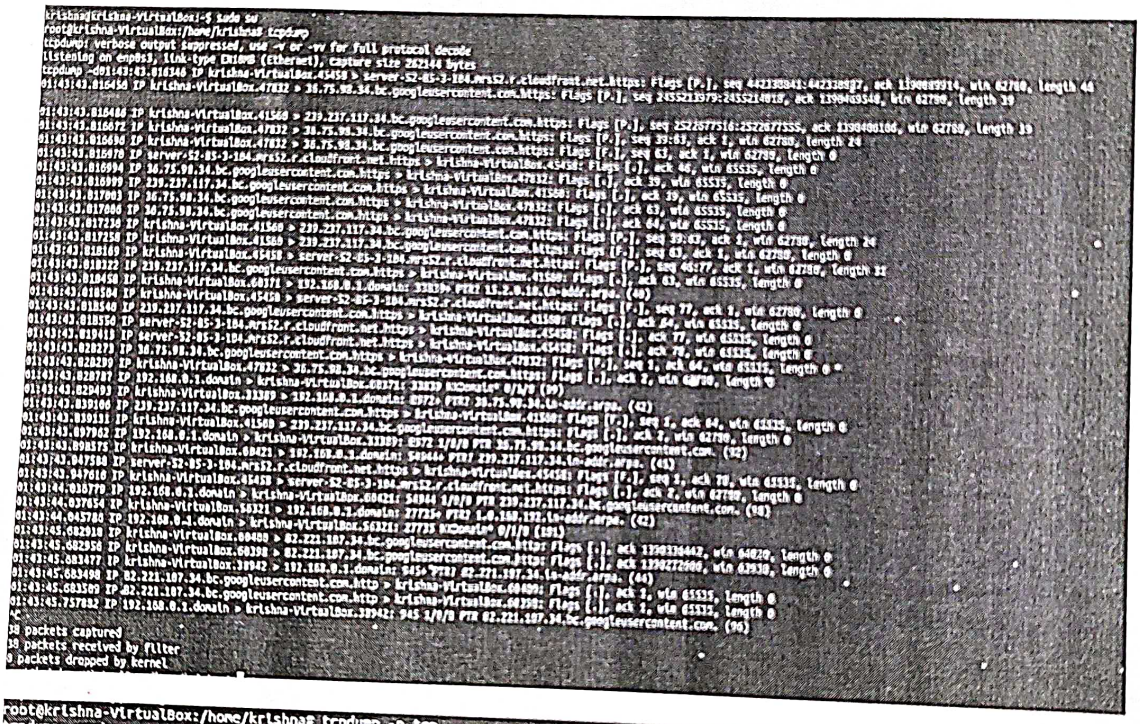
with ver-

- tcpdump -n port 80:
This command is used to capture packets from a specific port.

- tcpdump port 80 -w krishna:
This command is used to capture packets and write them to a file. All tcpdump files are ~~sav~~ saved with the '.pcap' extension and are accessible through wireshark.

Conclusion:
We understood how to capture and analyze packets using TCPDUMP.

```
krishna@krishna-VirtualBox:~$ sudo apt-get install tcpdump
[sudo] password for krishna:
Reading package lists... Done
Building dependency tree
Reading state information... Done
tcpdump is already the newest version (4.9.3-4).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2 linux-image-5.13.0-27-generic
  linux-modules-5.13.0-27-generic linux-modules-extra-5.13.0-27-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
krishna@krishna-VirtualBox:~$
```



```
root@krishna-VirtualBox:/home/krishnas tcpdump -n tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
01:44:55.828896 IP 10.0.2.15.35382 > 185.199.109.154.443: Flags [P.], seq 1894705561:1894706600, ack 1371881952, win 65515, length 39
01:44:55.829127 IP 10.0.2.15.35382 > 185.199.109.154.443: Flags [P.], seq 39:63, ack 1, win 65535, length 24
01:44:55.829158 IP 10.0.2.15.35382 > 185.199.109.154.443: Flags [F.], seq 63, ack 1, win 65535, length 0
01:44:55.829373 IP 185.199.109.154.443 > 10.0.2.15.35382: Flags [.], ack 39, win 65535, length 0
01:44:55.829398 IP 185.199.109.154.443 > 10.0.2.15.35382: Flags [.], ack 63, win 65535, length 0
01:44:55.829401 IP 185.199.109.154.443 > 10.0.2.15.35382: Flags [.], ack 64, win 65535, length 0
01:44:55.851764 IP 185.199.109.154.443 > 10.0.2.15.35382: Flags [P.], seq 1:25, ack 64, win 65535, length 24
01:44:55.851885 IP 10.0.2.15.35382 > 185.199.109.154.443: Flags [R], seq 1894706625, win 0, length 0
01:44:55.851979 IP 185.199.109.154.443 > 10.0.2.15.35382: Flags [F.], seq 25, ack 64, win 65535, length 0
01:44:55.851992 IP 10.0.2.15.35392 > 185.199.109.154.443: Flags [R], seq 2981165345, ack 64, win 0, length 0
01:44:55.852124 IP 185.199.109.154.443 > 10.0.2.15.35382: Flags [R.], seq 1894706625, ack 64, win 0, length 0
^C
11 packets captured
11 packets received by filter
0 packets dropped by kernel
```

28

29/03/22

Assignment - 10

Aim: A case study to design and configure any organisation network

Router:

The router is a physical or vitual networking device that is designed to, receive, analyse and forward data packets between compter networks. A router examines a destination IP address of a given data packet, and it uses the headers and forwarding tables to decide the best way to transfer the packets. There are some popular companies that develop routers; such as Cisco, D-link etc.

Cat6 cable:

Category 6 cable (Cat6) is a standardized twisted pair cable for Ethernet and other network physical layers that is backward compatible with the Category 5/5e and category 3 cable standards. Cat6 must meet more stringet specifications for crosstalk and system noise than cat 5 and cat 5e.

Switches :-

Switches are networking devices operating at layer 2 or a data link layer of the OSI model. They conneet devices in a network and use packet switching to send, receive or forward data packets or data frames over the network

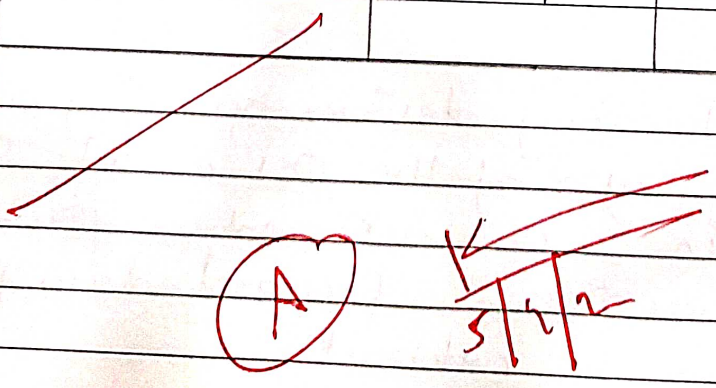
- Firewall

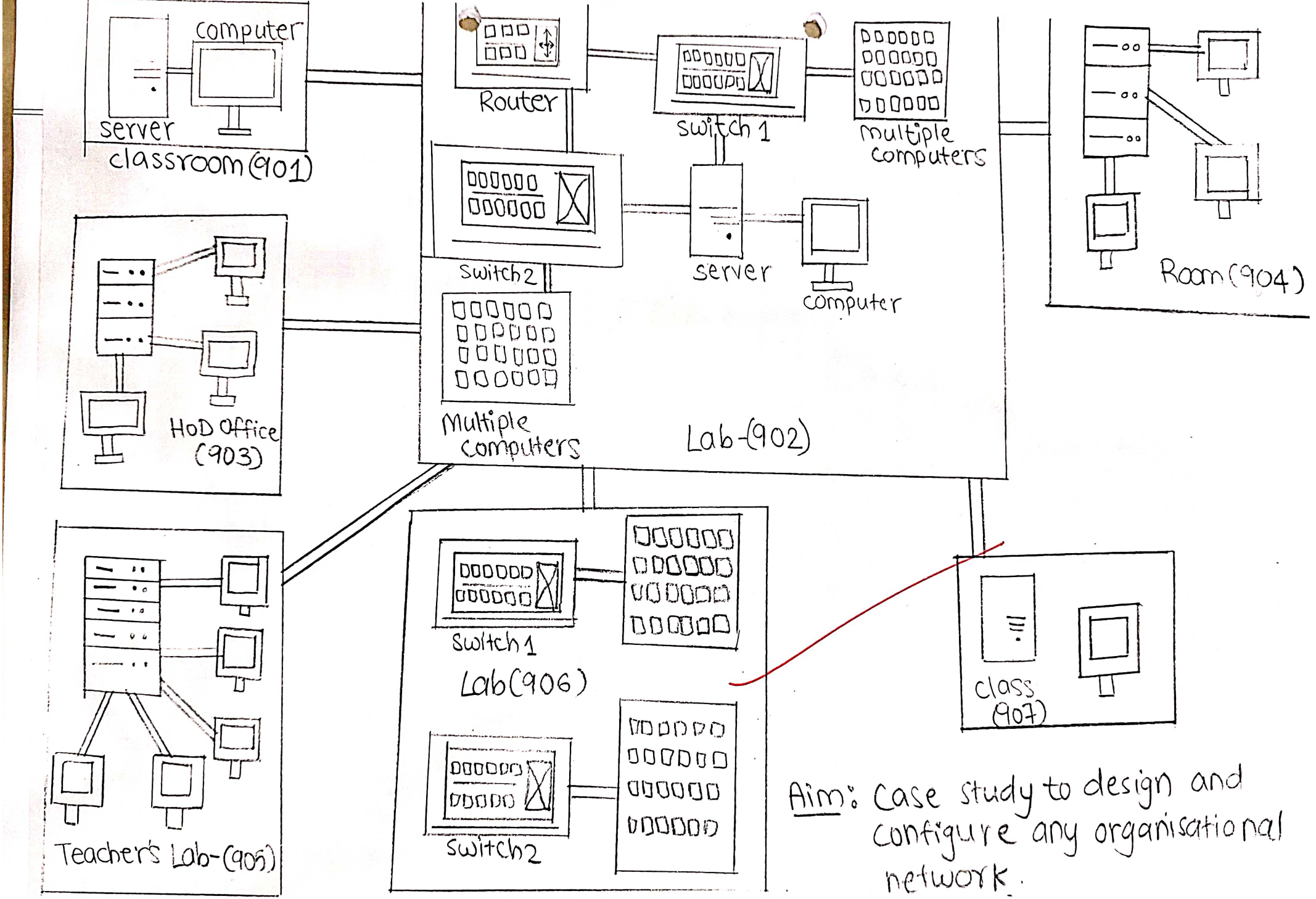
A firewall is a network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies. Firewall is integrated in the router in our model.

- RJ45 connectors.

The eight-pin RJ45 connector is a standardised interface which often connects a computer to a local area network (LAN). This type of connector was originally developed for telephone communications but is now used in a range of applications. The abbreviation RJ45 stands for Registered Jack-45.

		Price Per Piece	No. of pieces	Total price
(1)	RJ45 Connectors	4	100	400
(2)	Cable CAT6	26	1100	28600
(3)	Router	21600	1	21600
(4)	Internet Connection - Jio	18000	1	18000
(5)	Switch	11689	4	46756
				115356/-





Aim: Case study to design and configure any organisational network.

Written Assignment - 1

(Q1) Explain www and HTTP in detail.

→ The world wide web abbreviated as www and commonly known as web. The www was initiated by CERN in 1989.

System Architecture:

From user point of view, the web consists of a vast, world wide connection of documents or web pages. The pages can be retrieved and viewed by using browsers of which chrome, Firefox, etc are the popular ones.

Working of www:

It is based on several different technologies: web browsers, HTTP, HTML

A web browser is used to access web pages. Web browsers can be defined as programs which display text, data, pictures animations and

Architecture:

The communication protocol used to connect to web servers on the Internet or on local internet. The primary function of HTTP is to establish a connection with the server and send HTML pages back to user browser.

The HTTP protocol is a request/response protocol based on the client/server based architecture where web browser, robots and search engines, etc acts like HTTP clients.

Working:

HTTP gives user a way to interact with web resources such as HTML files by transmitting hypertext messages between clients and servers. HTTP clients generally use TCP connections to communicate with server.

Features:

(1) It is the protocol that allows web servers and browsers to exchange data over the web.
(2) It is a request response protocol.
(3) It is stateless, means each request is considered as the new request.

(Q2) Explain Domain name System.

→ DNS is a ~~fast~~ host name to IP address translation service. It is distributed database implemented in a hierarchy of name servers. It is application layer protocol for message ~~ext~~ exchange between clients and servers.

Every host is identified by the IP address but IP address are not ~~stata~~ static therefore a mapping is required to change the domain name to IP address

Domain:

(i) Generic Domain: .com, .edu, .org, .net, are generic domain.

(ii) Country domain: .in, .us, .uk.

(Q3) Explain FTP

→ FTP is an internet protocol tool provided by TCP/IP. It helps to transfer files from 1 PC to another by providing access to directions or folders on remote computers and allows software, data, text file to be transferred between different ~~fi~~ kinds of computers

There are two types of FTP connections.

(i) Active FTP connections:

The client establish the command channel and the server establish the data channel. When the client requests the data over the connection the server initiates the transfer of data to the client.

Written Assignment - 2

(Q1) List the design issues of session layer.

→ The design issues of session layer are as follows :-

(i) Establish session between machines :-

The establishment of session between machines is an important service provided by session layer. This session is responsible for creating a dialogue between connected machines. The session layer provides mechanism for opening, closing and managing a session between end-user application processes.

(ii) Enhanced Services :

Certain services such as checkpoints and management of tokens are the key features of session layer and thus it becomes necessary to keep ~~end~~ enhancing these features during the layer design.

(iii) To ~~hel~~ help in token management and Synchronization

The session layer plays an important role in preventing collision of severed critical operations as well as ensuring ~~b~~ ~~better~~ better data transfer over network by establishing synchronization points at specific intervals.

(Q2) Explain RPC in detail.

→ Remote Procedure Cell is a software communication protocol that are program can use to request a service from a program located in another computer on a network with having to understand the network's details. RPC is used to call other proceeds on the remote system like a local system.

When a RPC is invoked, the calling environment is suspended, the procedure parameters are transferred across the network to the environment where the process is to execute and the procedure is then executed in that environment.

Advantages:-

(1) Can be used in distributed environment.
(2) Supports process oriented and thread-oriented models.
(3) Hides the internal message-passing mechanism from user.

Disadvantages:-

(i) Highly Vulnerable to failure.
(ii) No uniform standard for RPC
(iii) Interaction based.

(Q3) Explain VLAN and VPN in detail along with Example

→ VLAN is a customer network which is created from one or more LAN. It enables a group of devices available in multiple networks to be combined into logical network. The result becomes a virtual LAN that is administered like a physical layer.

Without VLAN, a broadcast sent from a host can easily reach all network devices. Each and every devices will process broadcast received frames. It can increase CPU overhead on each device and reduce the overall network security.

Example:- To separate network management traffic from end-user or server traffic. To isolate sensitive infraistructure, services and host such as corporate user from guest users.

VPN connection establishes a secure connection between you and the internet. Via the VPN, all your data traffic is routed through an encrypted virtual tunnel. This disguises your IP address when you use the internet, making its location invisible to everyone. It is also secure against external attacks

There are five common VPN protocols.

(1) PPTP
(2) L2TP
(3) SSTP
(4) IKEV2
(5) OpenVPN

Example:-
Employees at a branch office could use a VPN to connect to the main office's internal network. Alternatively, a remote worker who may be working from could need to connect to their company's internet.