INDEX

NAME Suriya Prakash S.R. SUBJECT CN - Observation

STD. CSE DIV. 'O' ROLL NO. 352 SCHOOL REC

SR. NO.	PAGE NO.	TITLE	DATE	TEACHER'S SIGN / REMARKS
1.		Study of Various Network Commands used in Linux and Windows.	14/7	21/7/25 8
2.		Study of different types of network cables.	21/7	31/7/25 9
3.		Study of Packet tracer installation and user interface overview.	24/7	8/9/25 9
5.		Experiments on Packet Capture Tool: Wireshark.	26/7	
5.		Error correction at Data link layer.	07/8	9/10/25 8
6.		Flow control at Data link layer.	28/8	
7.		Setup and configure a LAN using a switch and Ethernet.	28/8	15/10/25 7
8.		Flow NMAP line hosting discovery	1/9	
9.		Implementation of subnetting in CISCO Packet	18/9	

SR. NO.	PAGE NO.	TITLE	DATE	TEACHER'S SIGN / REMARKS
10a)		Internetworking with router in CISCO packet tracer	21/9	
10b)		Design of configure Internetwork	25/9	
11a) 11b)		Simulate static Routing configuration using CISCO packet	25/9	
12a)		Implementation of echo client server using TCP / UDP sockets	28/9	
12b)		Implementation of chat client server using TCP / UDP sockets	29/9	
13)		Implementation of ping program	6/10	
14)		Implementation of packet sniffing using packet sniffing	6/10	

EXPERIMENT No: 1 DATE: 14/07/25	STUDY of Various Network Commands used In Linux and Windows.

Aim:- To study various networking commands used in linux and windows

Basic Networking Commands:-

1. arp-a:-

ARP is a short form of address resolution protocol. It will show the IP address of your computer along with IP address and MAC address of your router.

OUTPUT:-

Interface: 192.168.29.99 ---- 0x9

Internet address	Physical address	Type.
192.168.29.1	04-ab-08-2a-f3-b1	dynamic
192.163.29.255.	ff-ff-ff-ff-ff-ff	static
244.0.0.22	01-00-5e-00-00:	static
244.0.0.251	01-00-5e-00-00-fb	static
244.0.0.252.	01-00-5e-00-00-fc	static.

2. hostname:-

This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

OUTPUT:-

ks03-81

3. Ipconfig /all: This command displays detailed configuration information about your TCP/IP connection including router. gateway. DNS, DHCP,

OUTPUT:-

Windows IP configuration.

Host Name .....

Primary DNS suffix .....

Node type ..... Mixed.

IP routing Enabled ..... No

WINS Proxy Enabled ..... No.

4. nbtstat -a

This command helps solve problems with NetBIOS name resolution (Nbt stands for NetBIOS over TCP/IP)

OUTPUT:-

NBTSTAT [[-a RemoteName] [-A IP address] [-c] [-n] [-r] [-RR] [-s] [-S] [interval]].

-a (adapter status) lists the remote machine name table given its name

5. netstat (Network Statistics) netstat displays a variety of statistical about a computer active TCP/IP connections. It is a command line for monitoring network connections with incoming and outgoing as idle as viewing routing tables interforce statistics etc...

OUTPUT:-

Proto	Local address	Foreign address	State
TCP	127.00.1 : 49672	KS03-69 : 49673	Establish-ed
TCP	127.00 : 49673	KS03-69 : 49672	Establish-ed
TCP	127.16.75.69.4180	KS02.108.61210	TIME-WAIT

6. nslookup: (nameserver lookup) is a tool used to perform DNS lookups in Linux. It is used to display DNS details, such as the IP address of a particular computer, the MX records for a domain or the NS servers of a domain. nslookup can operate in two modes: interactive and non-interactive.

OUTPUT:-

Default Server : Unknown

Address : 172.16.72.1

7. Pathping: pathping is unique to windows and is basically a combination of the ping and the tracert commands. pathping traces the route to the destination address then

launches a 25 second test of each route along the way gathering statistics on the rate or data loss along each hop.

OUTPUT:

pathping [-g host-list] [-h maximum-hops] [-i address] [-n] [-p period] [-q num-queries] [-w timeout] [-4] [-6] target-name

8. Ping: (Packet Internet Groper): command's the best way to test connectivity between two nodes. Ping use ICMP (Internet control Message protocol) to communicate to other devices

OUTPUT:-

ping [-t] [-a] [-n count] [-l size] [-t] [-i TTL] [-r TOS] [-r count] [-s count] [[-j host-list] | [k-host list]] [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p] [-4] [-6] target-name

9. Route :- Route command is the used to show / manipulate the IP routing table. It is primarily used to setup static routes to specific host or network via an interface.

OUTPUT :-

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

interface list

19 .. 20 88 10 87 9e ad ... (intel (R)

8 .. 4e 82 a9 78 00 55 ... microsoft with

7 ... 42 82 a9 78 00 55 ... Microsoft with.

Linux Networking Commands.

1) Show IP address

$ ip address show

wlp 250

inet 192.16.75.11/2.

inet fe50. 2726 b845 abab a/25/64.

2) Add an IP address.

$ Sudo ip address add 192.168.1.284/24

dev wlp 250 RTNETLINK answers: File fairts

3) Delete an IP

$ sudo ip address del 192.168.1.284/24

dev wlp 230

4) Bring interface up

$ sudo ip link set wlp50 ~~down~~ up

5) Bring interface down:

$ sudo ip link set wlp230 ~~promis on~~ down

6) Enable promiscosous mode.

$ sudo ip link set wlp230 promisc on.

7) Add default route

$ sudo ip address add 192.168.1.254/24 dev wtp250

$ sudo ip route add default via 192.168.1.254 dev wtp250.

8) Add a route to 192.168.1.0/24 via gateway 192.162.1.259.

$ sudo ip route add 192.168.1.0/24 via 192.168.1.254.

9) Add a route to 192.168.1.0/24 reachable on device wtp250.

$ sudo ip route add 192.168.1.0/24 dev wtp250.

10) Delete route for 192.168.1.0/24 via gateway 192.168.1.254

$ sudo ip route delete 192.168.1.0/24 via 192.168.1.254

11) Display route taken IP 10.10.1.4.

$ ip route get 10.10.1.4

10.10.1.4 via 172.16.72.1 dev wtp250 src 172.16.15.17 uid 1000 cache.

12) ifconfig.

& ifconfig

wlp250: flags: 4163 < up, BROADCAST, RUNNING, MULTICAST > mtu 1501 inet 172.16.75.17 netmask 255.255.240.0 broadcast 172.16.79.255 ether 4c:82:67:17:[illegible] txqueuelen 1000 (Ethernet) RX packets 126870 bytes 70.4 MiB dropped 622 TX periods 24436 bytes 4.3 MiB

13) mtx google.com

Host	Packets Loost	Snt	Last	Avg	Pings best	worst	St. dev
1. IN 115.245.95.249	0.0-1	302	2.5	12.3	2.6	8 12.2	50
2. 45 72.14.217.252	0.01	321	6.5	15.5	5.9	365.8	48

14) mtr - b google.com

	Packets loss. 1.	snt	Last	Avg	Pings Best	worst	St.
1) 172 16.12.122 (172.16.12.122)	0.0 4.83	62	122.2	5.8	201.2	[illegible]	
172.16.12.122 (172.16.12.122)	0.0 1.39	64	82.0	5.5	263.9	5[illegible]	

15) mtr. - c 10 google.com

	Packets loss.t	snt	Last	Avg.	Pings Best	worst	Stdev
1) 115.245 95 2490	0.0.1	8	281	57.6	2.9	261.0	117.8
172.16.12.122	0.0.1	8	5.81	65.9	5.4	64.5	19.1

16) Capture traffic on your wifi interface wlp250.

$ sudo tcpdump wlp250

Listening on wlp250.

01:47:07 IP 172.16.75.142 > 239.0.0.1 UDP, length 50

01:47:07 &p fedora > gateway PTR query.

34 packets captured

722 packets received by filter.

612 packets dropped by kernel

17) Capture only 10 packets.

→ ping.

It is used to troubleshoot connectivity. reachability and name resolution.

ping google.com

Output:

PING google.com (142.258.221.266) 56 (84) bytes of data:

From fedora (192.168.1.294) icmp seq = 1 Destination Host unreachable

From fedora (192.168.1.294) icmp - seq = 2 Destination Host unreachable.

Student Observation :-

1) Which command is used to find the reachability of a host machine from your device.

→ ping <hostname or IP> ping google.com

2) Which command will be given the details of hops taken by a packet to reach its destination?

→ mtr <hostname>

mtr google.com

3) Which command displays the ip configuration of your machine?

→ on linux: ip address show

on windows: ip config /all

4) Which command displays the TCP port status?

→ netstat

netstat -r

5) Which the command to modify the IP configuration in a linux machine?

→ to add: ip address add 192.168.1.254/24 dev enps031ff6.

→ to del: ip address del 192.168.1.254/24 dev enps031ff6.

RESULT :- The networking commands in Windows and Linux have been executed successfully.

EXPERIMENT No: 2.
DATE : 21 / 7 / 25.

STUDY OF NETWORK CABLES

Aim:- Study of different types of network cable

a) Understand different types of Network Cable

Different type of cables used in networking are:-

1) Unshielded Twisted Pair (UTP) Cable.
2) Shielded twisted Pair (STP) cable.
3) Coaxial Cable.
4) Fibre optic cable.

Cable Type	Category	Maximum Data Transmission	Advantages/ Disadvantages	Application / Use.
UTP	Category 3	10 bps	Advantages • Cheaper in cost • Easy to install as they have a smaller overall diameter. Disadvantage • More prone to (EMI) Electromagnetic Interference and noice.	10 Base T Ethernet Fast Ethernet, Gigabit Ethernet
	Category 5	Upto 100 mbps		
	Category 5e	1 Gbps.		
STP	Category 6, 6a.	10Gbps	Advantages • Shielded • Faster than UTP • Less susceptible.	Gigabit Ethernet, 10G Ethernet (55 m)

SSTP	Catagory 7	10Gbps	Disadvantages • Expensive • Greater installation effort	Widely used in data centres Gigabit Ethernet 10G Ethernet (100m)
Coaxial Cable	RG-6 RG-59 RG-11	10-100 Mbps.	• High bandwidth • Immune to interface. • Low loss bandwidth • Versatile Disadvantages. • Limited distance • Cost • Size in bulky	Speed of signal in 500m Television network High speed Internet connections.
Fibre optics Cable	Single mode Multi mode	100 Gbps	Advantages • High speed • High bandwidth • High security • Long distance Disadvantages. • Expensive • Requires skilled installers.	Maximum distance of fibre optics cable is around 100 metres.

Student Observation :-

1) Difference between cross cable and straight cable.

→ A straight cable connects different devices, while cross value cable connects the similar devices by swapping transmit and receive wires.

2) Which type of cable is used by connect two PC? → cross cable is used to connect two PC's.

3) Which cable is used to connect a route to PC?

→ Straight cable is used to connect a router to a PC.

4) Find the category of twisted pair cables. used to connect the PC to network socket?

→ Category 5e and category 6, 6a are used to connect the PC to network socket.

5) Write down understanding, challenging faced and output received?

→ Learnt the colour codes and wiring standards.

→ Difficulty in arranging wires in correct order.

→ Successfully made working cables that connected devices and enable proper network communication.

RESULT:-

Hence, the different types of network cables have been successfully studied.

31/8/25

EXPERIMENT No: 3	Experiments on CISCO PACKET TRACER (Simulation Tool)
DATE : 24/07/25.	

AIM:- Analyse the behaviour of network devices using CISCO PACKET TRACER simulator

1) From the network component box, click and drag and drop the below components

a) 4 Generic PC's and One HUB.

b) 4 Generic PC's and One Switch.

2) Click on connections:

a) Click on Copper-straight-Through cable

b) Select one of the PC and connect it to HUB using the cable. The link LED should glow in green, indicating that the link is up. Similarly connect remaining 3 PC's to the HUB.

c) Similarly connect 4 PC's to the switch using copper straight-through cable.

3) Click on the PCs connected to hub, go to the Desktop tab, click on IP configuration, and enter an IP address and subnet mask. Here, the default gateway and DNS server information is not needed as there are only.

two end devices in the network

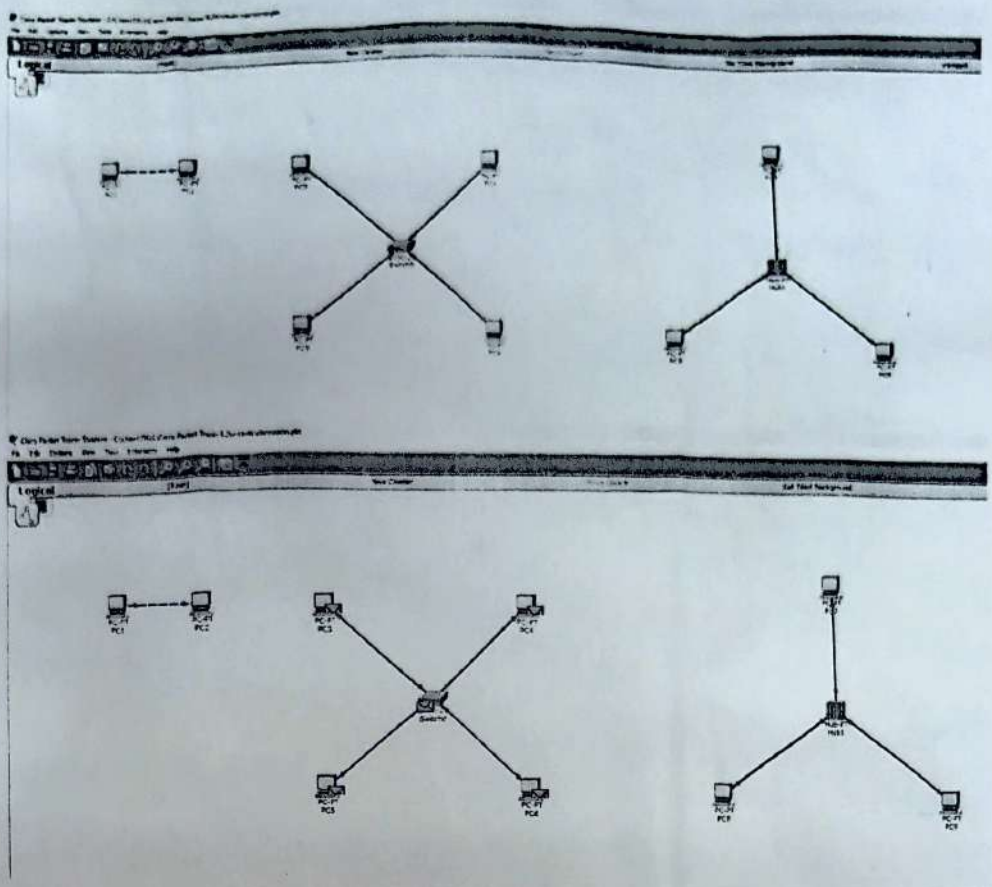
click on the PDU form the common tool bar

a) Drag and drop it ~~from the common tool bar~~ on one of PC and then drop it on another PC connected to the HUB.

4) Observe the flow of PDU from Source PC to destination PC by selecting the Realtime mode of simulation.

5) Repeat Step 3 to step 5 for the PC's connected to the switch

6) Observe how HUB and switch are forwarding the PDU and about the behaviour of switch and HUB.



Student Observation :-

a) From your observation write down the behaviour of switch and HUB in terms of forwarding the Packet received by them.

→ A switch forwards packets only to the specific device (port) based on MAC address, while a hub broadcasts packets to all connected devices.

b) Find out the network topology implemented in your college and draw and label that topology in your observation book.

→ The network topology commonly used in colleges is star topology where all devices are connected a central switch or hub.

RESULT:-

The packet tracer tool installation and user interface overview is studied.

3/9/25

EXPERIMENT No: 5
DATE: 26/7/25.

Experiments on Packet Capture Tool: Wireshark

AIM:

Experiments on packet capture tool: wireshark.

Packet sniffer

- Sniffs messages being sent / received from by your computer
- Store and display the contents of the various protocol fields in messages.
- Passive program.
  - never sends packets itself
  - no packets addressed to it
  - receives a copy of all packets. (sent / received)

Packet sniffer diagnostic tools.

- Tcpdump
- Wireshark.

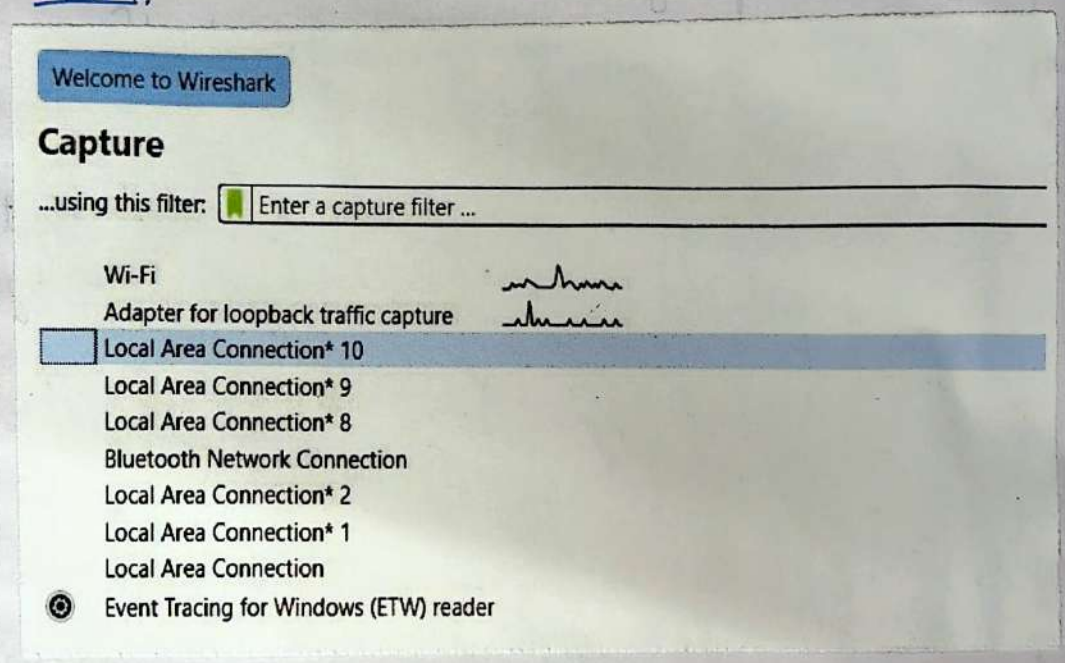
Capturing Packets

After downloading and installing wireshark, launch it and double-click the name of a network interface under capture to start capturing packets on that surface.

Procedure :-

- Select local area connection in wireshark
- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click start capture
- Save the packets.

Output :-



1. Create a filter to display only TCP/ UDP packets, inspect the packets and provide the flow graph.

Filtering packets.

If you're trying to inspect something specific, such as the traffic a program send when phoning home, it helps to close down all other applications using network so you can narrow down the traffic. still, you'll likely have a large amount of packet to stiff through. That's where wireshark filters come in.

tcp

No.	Time	Source	Destination	Protocol	Length	Info
1875	3.692924	192.168.0.111	109.61.45.167	TCP	54	63498 → 443 [ACK]
1877	3.856422	109.61.45.167	192.168.0.111	TLSv1.2	95	Application Data
1878	3.908955	192.168.0.111	109.61.45.167	TCP	54	63498 → 443 [ACK]
1880	4.561445	18.67.161.50	192.168.0.111	TLSv1.2	251	Application Data
1881	4.611109	192.168.0.111	18.67.161.50	TCP	54	63723 → 443 [ACK]
1883	5.175807	89.43.104.224	192.168.0.111	TLSv1.2	78	Application Data
1884	5.176992	109.61.45.167	192.168.0.111	TLSv1.2	97	Application Data
1885	5.177230	192.168.0.111	89.43.104.224	TLSv1.2	82	Application Data
1886	5.228271	192.168.0.111	109.61.45.167	TCP	54	63498 → 443 [ACK]
1887	5.380624	18.67.161.50	192.168.0.111	TLSv1.2	253	Application Data
1888	5.403001	89.43.104.224	192.168.0.111	TCP	54	443 → 63566 [ACK]
1889	5.428387	192.168.0.111	18.67.161.50	TCP	54	63723 → 443 [ACK]
1890	5.691231	109.61.45.167	192.168.0.111	TLSv1.2	92	Application Data
1891	5.750815	192.168.0.111	109.61.45.167	TCP	54	63498 → 443 [ACK]
1892	5.763167	109.61.45.167	192.168.0.111	TLSv1.2	165	Application Data
1893	5.813240	192.168.0.111	109.61.45.167	TCP	54	63498 → 443 [ACK]
1895	6.716279	109.61.45.167	192.168.0.111	TLSv1.2	97	Application Data

You can also click Analyze > Display filters to choose a filter from among the default filters included in wireshark close the window and you'll find a filter has been applied automatically wireshark is showing you the packets that make up the conversation

No.	Time	Source	Destination	Protocol	Length	Info
163	6.932229	109.61.45.167	192.168.0.111	TLSv1.2	127	Application Data
164	6.980095	192.168.0.111	109.61.45.167	TCP	54	61498 → 443 [ACK] Seq=1 Ack=961 Win=509 Len=0
165	7.097540	192.168.0.111	142.251.223.229	TLSv1.2	79	Application Data
166	7.097540	192.168.0.111	142.250.183.174	TLSv1.2	79	Application Data
167	7.097631	192.168.0.111	142.251.223.229	TCP	54	63747 → 443 [FIN, ACK] Seq=25 Ack=1 Win=512 Len=0
168	7.097713	192.168.0.111	142.250.183.174	TCP	54	63748 → 443 [FIN, ACK] Seq=25 Ack=1 Win=512 Len=0
169	7.106812	142.251.223.229	192.168.0.111	TCP	54	443 → 63747 [ACK] Seq=1 Ack=25 Win=639 Len=0
170	7.106812	142.250.183.174	192.168.0.111	TCP	54	443 → 63748 [FIN, ACK] Seq=1 Ack=25 Win=1047 Len=0
171	7.106812	142.250.183.174	192.168.0.111	TCP	54	443 → 63748 [ACK] Seq=2 Ack=26 Win=1047 Len=0
172	7.106812	142.251.223.229	192.168.0.111	TCP	54	443 → 63747 [FIN, ACK] Seq=1 Ack=26 Win=639 Len=0
173	7.107503	192.168.0.111	142.250.183.174	TCP	54	63748 → 443 [ACK] Seq=26 Ack=2 Win=512 Len=0
174	7.107506	192.168.0.111	142.251.223.229	TCP	54	63747 → 443 [ACK] Seq=26 Ack=2 Win=512 Len=0
175	7.444167	89.43.104.224	192.168.0.111	TLSv1.2	78	Application Data
176	7.445471	192.168.0.111	89.43.104.224	TLSv1.2	82	Application Data
177	7.648447	18.67.161.50	192.168.0.111	TLSv1.2	253	Application Data
178	7.672759	89.43.104.224	192.168.0.111	TCP	54	443 → 63566 [ACK] Seq=25 Ack=29 Win=83 Len=0
179	7.704072	192.168.0.111	18.67.161.50	TCP	54	63723 → 443 [ACK] Seq=1 Ack=1193 Win=515 Len=0

> Frame 1: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{3922E75B-0

Inspecting Packets

Click a packet to select it and you can dig down to view its details.

```
> Frame 1: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{3922E75B-0
> Ethernet II, Src: CloudNetwork_38:ee:e7 (90:0f:0c:38:ee:e7), Dst: TpLinkTechno_26:23:6e (3c:84:6a:26:23:
> Internet Protocol Version 4, Src: 192.168.0.111, Dst: 142.250.77.138
> User Datagram Protocol, Src Port: 58877, Dst Port: 443
> Data (29 bytes)
```

You can also create filters from here. just right-click one of the details and use the Apply as filter submenu to create a filter based on it.

Flow Graph:- Gives a better understanding

Capturing and analyzing packets using Wireshark tool.

To filter, capture, view, packets in wireshark tool, capture 100 packets from the Ethernet : IEEE 802.3 LAN Interface and save it

Procedure:

- Select Local Area connection in Wireshark
- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click start capture
- Save the packets.

1. Create a filter to display only TCP/UDP packets, inspect the packets and provide the flow graph

2. Create a filter to display only ARP packets and inspect the packets.

3. Create a filter to display only DNS packets and provide the flow graph

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.111	148.113.9.94	TCP	55	62659 → 443 [ACK] Seq=1 Ack=1
2	0.047691	104.238.216.204	192.168.0.111	TCP	54	443 → 63807 [ACK] Seq=1 Ack=1
3	0.047939	192.168.0.111	104.238.216.204	TCP	680	63807 → 443 [PSH, ACK] Seq=1
4	0.051078	104.238.216.204	192.168.0.111	TCP	66	443 → 63786 [ACK] Seq=1 Ack=1
5	0.061612	192.168.0.111	54.188.199.43	TCP	66	63810 → 443 [SYN] Seq=0 Win=6
6	0.086960	148.113.9.94	192.168.0.111	TCP	66	443 → 62659 [ACK] Seq=1 Ack=2
7	0.091888	192.168.0.111	40.104.107.178	TCP	66	63811 → 443 [SYN] Seq=0 Win=6
8	0.102181	192.168.0.111	40.104.107.178	TLSv1.2	583	Client Hello (SNI=substrate.o
9	0.111174	192.168.0.111	142.251.223.229	TCP	55	63763 → 443 [ACK] Seq=1 Ack=1
10	0.346613	40.104.107.178	192.168.0.111	TCP	66	443 → 63811 [SYN, ACK] Seq=0
11	0.346946	192.168.0.111	40.104.107.178	TCP	54	63811 → 443 [ACK] Seq=1 Ack=1
12	0.347463	192.168.0.111	40.104.107.178	TLSv1.2	583	Client Hello (SNI=substrate.o
13	0.668789	192.168.0.111	18.67.161.50	TLSv1.2	82	Application Data
14	0.963225	192.168.0.111	18.67.161.50	TCP	82	[TCP Retransmission] 63789 →
15	1.127874	192.168.0.111	40.104.107.178	TCP	583	[TCP Retransmission] 63809 →
16	1.127896	192.168.0.111	104.238.216.204	TCP	680	[TCP Retransmission] 63807 →
17	1.231960	192.168.0.111	192.168.0.1	DNS	86	Standard query 0xa917 HTTPS 1
18	1.232206	192.168.0.111	192.168.0.1	DNS	86	Standard query 0xbb3e A lives

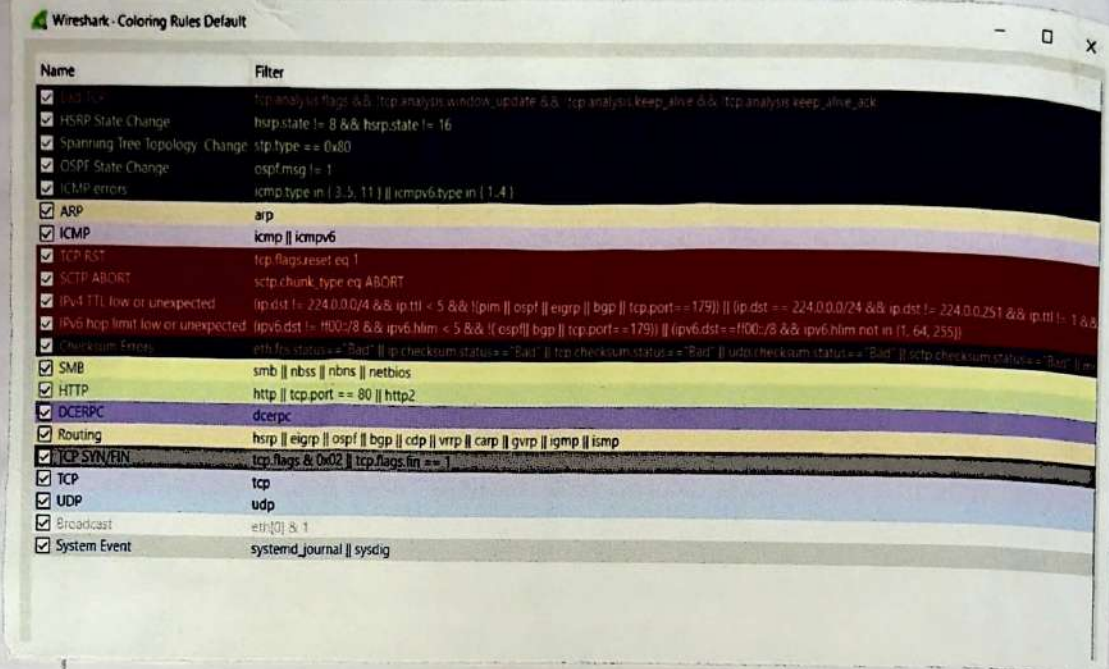
```
Frame 6: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{3922E75B-0
Ethernet II, Src: TpLinkTechno_26:23:6e (3c:84:6a:26:23:6e), Dst: CloudNetwork_38:ee:e7 (90:0f:0c:38:ee:
Internet Protocol Version 4, Src: 148.113.9.94, Dst: 192.168.0.111
Transmission Control Protocol, Src Port: 443, Dst Port: 62659, Seq: 1, Ack: 2, Len: 0
```

4. Create a filter to display only HTTP packets and inspect the packets.

Procedure:

- Select a local Area connection in wireshark
- Go to capture → option.

- Select stop capture automatically after 100 packets.
- Then click start capture.
- Search HTTP packets and save packets



5. Create a filter to display only IP / ICMP packets & inspect the packets

- Select Local Area Connection in 'wireshark'.
- Go to capture → option.
- Select stop capture automatically after 100 packets.
- Then click start capture.

- Search ICMP / IP packets in search bar
- Save the packets.

6. Create a filter to display only DHCP packets and inspect the packets.

Student observation.

1. What is promiscuous mode?

→ Promiscuous mode is NIC [network interface card] setting, where NIC captures all packets not just addressed ones.

2. Does ARP packets have transport layer header? Explain.

→ No, ARP works at Data link, no transport header.

3. Which transport layer protocol is used by DNS?

→ UDP on port 53 (queries).
TCP on port 53 (large data)

4. What is the port numbers used by HTTP protocol?

→ 80 (HTTP)
443 (HTTPS).

5. What is broadcast IP address?

→ A broadcast IP address is an address used to send data to all hosts in a network simultaneously. In IPV4, its highest address in a subnet.

Eg → For network 192.168.1.0/24, the broadcast address is 192.168.1.255.

RESULT :-

Experiments on Packet Capture tool - Wireshark was successfully studied.

Experiment No: 6 | Error Correction at Data link Layer

Date: 07/08/25

Aim:-

Write a Program to implement error detection and correction using Hamming Code Concept.

Sender Program:-

Apply hamming Code Concept on the binary data and add redundant bits to it.

```
def hamming-code (data):
    def insert_bits (data):
        m = len (data)
        r = 0
        while (2**r) < (m+r+1):
            r+=1
        n = m+r
        result = ['0']*n
        j = 0
        for i in range (1,n+1):
            if i as (i-1) == 0
                continue
            result [-i] = data [-(j+1)]
            j+ = 1
            if j == m:
                break
        return result, n, r
    def calc-Parity (Pdata, r):
        n= len (Pdata)
```

```
result = Pdata [:]
for i in range (r)
    Parity - Val = 0
    Parity - Pos = (2**i)
for K in range (1, n+1)
    if K in Parity Pos
        Parity - Val = int ( result [- K])
result [Parity - POS] = Str (Parity - Val)
return result
Pbits, n, r = insert bits (data)
Code = Calc - Parity (Pbits, r)
return join (Code)
input = input ('Enter binary data :)
Print ('Hamming Code = , hamming - Code (input))
```

Reciever Program:

⇒ Apply hamming Code on the binary data to Check for errors.

⇒ If there is any error, display the position of the error.

```
def hamming - Check (hamming - code):
    n = len( hamming - Code)
    r = 0
    while (2 ** r) < n+1 :
        r+ = 1
    syn = 0
    Parity = [ ]
```

```
for i in range (x):
    Parity_Pos = 2** i
    Parity_Val = 0
for k in range (1, n+1):
if k as Parity_pos
    Parity_Val = int (hamming_Code [-k])
Parity.append (Parity_Val)
    Syn |= (Parity_Val << i)
Synbits = ''.join (Str (x) for x in reversed (Parity))
return Synbits, Syn
Code = input ("Enter recieved hamming Code.")
res, error = hamming_check (Code)
Print ('Error bits:', res)
    if error == 0
        Print ('No Error detected')
    else:
        Print ('Error detected at bit position:
        , error)
```

Output :-

Enter binary data: 1001101

Hamming Code: 1001110101

Enter recieved Hamming Code: 1001010101

Error Syndrome bits: 0111

Error detected at bit Position: 7

Result :-

Sender and Reciever Program for hamming Code concipt was executed and go the output:

9/10/19

Ex. no : 7
28/08/25

Flow Control at Data Link Layers

Aim :-

Write a program to implement flow control at data link layer using sliding window protocol, Simulate the flow of frames from one node to another.

Features:-

⇒ Inputs window size & message

⇒ Sends window size frames at a time.

⇒ writes frames to sender buffer.

⇒ Reciever reads frame, sends ACK or NACK to reciever - Buffer.

⇒ Sender reads ACK/NACK and continues or resends frames.

⇒ You Can manually edit the files to simulate errors.

Code :-

```
import time
import random
class Sender:
    def_ init_ (self, total_frames, window_size):
        self. total_frames = total_frames
        self. window_size = window_size
        self_base = 0
        self. next_seq = 0
    def send_frames (self):
```

```
Print (f"\n [Sender] Total frames to send :
        {Self.total_frames}")
while self.base < Self.total_frame :
    while self.next_seq < seq.total_frame
Print f(f [Sender] Sending frame {Self.next_seq}
Self.next_seq += 1
    time.sleep (1)
def ack_recieved (Self, ack) :
    Print (f" [Sender] Acknowledgment recieved for
        Frame {ack}")
    if ack >= Self.base :
        Self.base = ack + 1
class Reciever :
    def reciever.frame (self, frame_no, Sender) :
    if random.choice ([True, false]) :
        Print (f" [Reciever] Recieved Frame
            {frame_no}" )
    else :
        Print (f" [Reciever] frame {frame_No}
            lost (No Ack sent)" }

if __name__ == "__main__" :
        total_frames = 5
        window_size = 3
```

```
sender = Sender(total_frames, window_size)
reciever = reciever()
sender.Send_frames(reciever)
```

Output :-

```
Enter total number of frames : 5
Enter window size : 3
[Sender] Total Frames to send : 5
[Sender] Sending frame 0
[Sender] Sending frame 1
[Sender] Sending frame 2
[Reciever] Successfully recieved frames 0 to 2
[Sender] Acknowledgment recieved for frame 2
[Sender] Sending frame 3
[Sender] Sending frame 4
[Reciever] Frame 4 lost or Corrupted
[Sender] Timeout Resending for frame 3
[Sender] Sending frame 3
[Sender] Sending frame 4
[Reciever] Successfully recieved frames 3 to 4
[Sender] Acknowledgment recieved for frame 4
Transmission Completed.
```

Result:-

Sliding window Protocol is executed sucessfully.

EXPERIMENT No. 4.

DATE: 28/8/25

Setup and Configure LAN

AIM:- Setup and configure a LAN using a switch and Ethernet cable in your lab

What is LAN?

A LAN refers to a network that connects devices to a limited area such as office building, school to a limited area or home. It enable users to share resources including data, printer and int access. LAN connects devices to promote collaboration and transfer information between users such as computers, printers, servers and switches

How to SETUP LAN?

Step 1: Plan and design an appropriate network topology taking into account network requirements and equipment location

Step 2: You can take 4 computers a switch with 8, 16, or 24 ports which is sufficient for networks of these

sizes and 4 Ethernet cables

Step 3: Connect your computers to network switch via an ethernet cable, which is as simple as plugging one end of the Ethernet cable into your computer and the other end into your network switch

Step 4: Assign IP address to your PC's

1. Log on to the client computer as Administrator or as Owner
2. Click Network and Internet Connections
3. Right click local Area Connection Ethernet → Go to properties → Select Internet Protocol (TCP/IPv4) → Click on properties → Select use the following IP address option and assign ip address.

Similarly assign IP address to all the PC's connected to switch.

PC1 - IP address : 10.1.1.1, subnet mask : 255.0.0.0

PC2 - IP address : 10.1.1.2, subnet mask 255.0.0.0

PC3 - IP address : 10.1.1.3 subnet mask 255.0.0.0

PC4 - IP address : 10.1.1.4 subnet mask 255.0.0.0

Step 5: Configure a network switch

1) Connect your computer to the switch. To access the switch's web interface you will need to connect your computer to the switch using an Ethernet cable

2) log in to the web interface open a browser and enter the IP address of the switch in the address. This should bring up the login page for the switch web interface. Enter the username & password to log in.

3) Configure basic settings. Once you're logged in you will be able to configure basic settings for the switch

4) Assign IP address as 10.1.1.5, subnet mask 255.0.0.0

Step 6: Check connectivity between switch and other machine by using ping command in the command prompt of the device

Step 7: Select a folder → go to properties → click sharing tabs → share it with everyone on the same LAN

Step 8: Try to access the shared folder from other computers of the network

Result:

Using Switch & Ethernet cables a LAN has been setup and configured.

15/10/

EXPERIMENT No: 8 Experiment on outliving
DATE: 1/9/25 the processes in NMAP
before port scanning to
find online system.

Aim:-
To attempt to port scan offline systems and recognize the waste time and the created connected network (because it is active recon)

PROCEDURE:-

The ARP scan:

This scan uses ARP requests to discover live hosts 3.

ICMP scan:

This scan users ICMP requests to identify live host 3

TCP/UDP ping scan:

This scan users packets to TCP ports and UDP ports to determine live hosts

There will be 2 scanners introduced

1) arp-scan

2) mass-scan

NMAP (network mapper). It is a well known tool for mapping networks, locating lines and detecting running services. NMAP's scripting engine can be used to extend its capabilities such as fingerprinting services. The scans follow the step represented in image below, but several are optional and are conditional on "command line".

Step 1: enumerate the targets

Step 2: Discover live host

Step 3: Reverse DNS lookup

Step 4: Scan ports

Step 5: Detect versions

Step 6: Detect OS

Step 7: Traces route

Step 8: scripts

Step 9: write output

RESULT :- Hence the experiment on outlining the process in NMAP port scanning is successfully.

EXPERIMENT 8b	CONFIGURATION OF WIRELESS
DATE: 1/9/25	LAN using CISCO Packet tracer

AIM:
To configure the wireless LAN using cisco packet tracer.

PROCEDURE:
Design a topology with three PC's connected from Linksys wireless router.
Perform the following configuration:

- Configure static IP on PC and wireless Router
- Set SSID on mother Network
- Set IP address of router to 192.168.0.1, PC0 to 192.168.0.2 & PC2
- Connect PC by using WAP key.

Click on wireless router.

→ Select Administrator tab from top menu, set username and password to admin & click on same setting.

→ Next click on wireless tab and set default SSID to mother Network.

→ Now select wireless security & change security mode to WEP.

→ Set Key 1 to 01234567 8.

→ click on same sitting.

→ configure the static IP on all three PC's

→ Double click on PC select Desktop tab click on IP configuration select static IP and set IP as given

→ Repeat same process PC1 and PC2

Student Observation

a) What is SSID of a wireless router?

→ It is a unique name of a wireless network used to identify and connect to it

b) Security key, what is meant by this?

→ It is the password or pairphrase used to protect a WIFI - network and ensures user authorization

c) Configure a simple wireless LAN

→ Set SSID, enable WPA2 / WPA3 security with a strong key, assign IPS & connect devices

RESULT:

Hence the configuration of wireless LAN using CISCO tracer packet is successfully done.

EXPERIMENT NO: 9 | Subnetting in CISCO
DATE: 18/9/25. | PACKET TRACER Simulator

AIM:

Implementation of subnetting in CISCO packet Tracer Simulator

Class IP subnetting is a technique that allows more efficient use of IP address by allowing for subnet masks that are not just the default masks, that are not just the default masks. This means that we can divide our IP address space into smaller subnets

PROCEDURE:

Step 1: Creating a network topology

Step 2: adding the devices

Step 3: Subnett the network address to provide enough space for atleast 5 address for end devices

Step 4: Configure the devices that were added. Right click on the router and select "CLI". This

Step 5: To configure the 6 gigabit ethernet interface on router, you can follow these steps.

1. Right - click on the router
2. Enter the commands

Step 6: Testing the network

When the network topology is configured, then test the network. Open a command prompt on each PC and try to ping the other PC.

Student Observation :-

a) Write down what is subnetting.

→ It is the process of dividing a large network into smaller networks to improve management and efficiency

b) Advantage of subnetting

It reduces network congestion improves security and allows efficient use of IP address

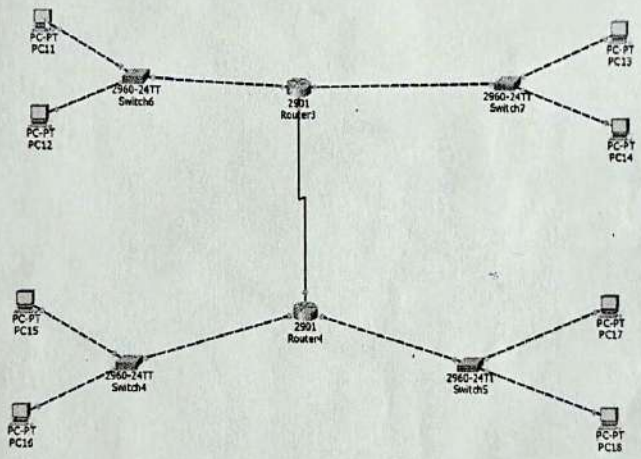
c) Subnetting in college. Explain

Colleges use subnetting to seperate labs, admin and WiFi network

(eg) 192.168.1.0.0/24 for Lab

192.168.3.0/24 for admin

9)



RESULT:-

Hence the experiment of subnetting in cisco packet tracer simulator

EXPERIMENT NO: 10 a
DATE: 21/9/25

Internetworking with Rou
in CISCO Packet trace

AIM:-

a). Internetworking routers in CISCO packet tracer simulator

In this network, a router and 2 PC's are used - Computers are connected with routers using a copper straight - through cable. After forming the network, to check network connectivity a simple PDV is transformed from PC0 to PC1

PROCEDURE:-

Step 1: (Configure Router 1)

→ Select the router and open CLI.

→ Please enter to start configuration Router 1

Step 2: (configuring PCs):

→ Assign IP address to energy PC in the network.

→ Select the PC, go to the desktop and assign an IP address

→ Assign the default gateway of PC0 as 192.168.10.1

→ Assign the default gateway of PC1 as 192.168.20.1

Step 3:- (conflicting PC's with Routers)

→ Connect Fast ethernet 0 port of PC0 with Fast ethernet 0/0 port of Router 1 using a copper straight through cable.

→ Connect Fast ethernet 0 of PC1 with Fast ethernet using a copper straight through cable.

PC configuration table :-

Device Name	IP address	Subnet network	Gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.0
PC1	192.168.20.2	255.255.255.0	192.168.20.0

RESULT :-

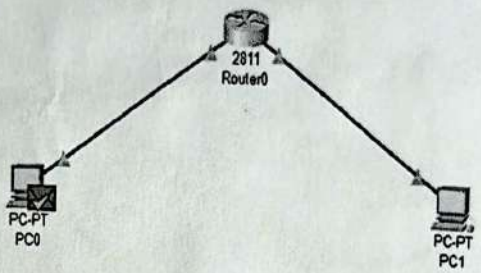
Hence the experiment on internet working with router has been executed successfully.

EXPERIMENT No: 10b
DATE: 21/9/25

Design and Configure an internetworking using wireless Router.

AIM: Design and configure an inter network using wireless router. DHCP server and internet cloud.

10)a)



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)
	Successful	PC0	PC1	ICMP		0.000	N	1	(edit)

Addr

Device	IP addr	Subnet	p'addr	Sub
	fast ethernet	network		network
Router 1	192.168.10.1	285.255.285.0	256.285.20	

Device name	IP addr	Subnet network	gateway
PC0	192.168.10.2	255.255.255.0	192.168.1
PC1	192.168.202	255.255.258.0	

Objectives

Part 1: Build a simple network

Devices

Part 2: Test connectivity between Network Devices.

Part 3: Test connectivity networks

Part 4: Save the file and close packet tracer.

PROCEDURE:-

Step 1: Launch packet tracer

Step 2: Build the topology

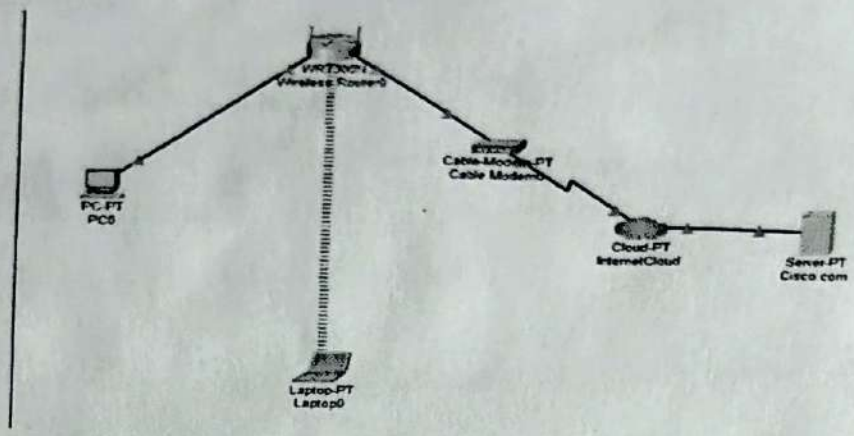
a) Add network devices to the workspace using the device to the workspace

To place a chance into the workspace device - Type selection box

Device - Specific box is finally clicked on a location. If you want a cancel click and drag a device from the device specific solution box onto the workspace.

b) Change display names of the network devices. click verify and check the connectivity.

10)b)



RESULT:-

Hence the experiment on designing and configuring internetwork using wireless.

EXPERIMENT No: 11 | Simulate Static Routing Configuration.
DATE: 25/9/25

AIM: To simulate the RIP using CISCO packet tracer

Assign IP address to PC's - double click PC's and click desktop server items and IP config. Assign IP address refferring the table

Double click Router and click CLI and press enter key to access the command prompt for router

router > enable
router # config terminal
enter config commands, one per live.
end with CNT (Z/2)
router (config) #
interface past ethernet 0/0 command is used to enter in interface mode
IP address 10.0.0.1 255.0.0.0 command will assign IP address to interface.
no shutdown command will bring the interface up
exit command is used to return to global configuration mode.

Serial interface needs two additional parameters clock rate and bandwidth every serial cable has two ends DTE and DCE.

router > enable

router # configure terminal

enter configuration command one per time and with CNTL/Z

router (config) # interface serial 0/0

router (config) # interface ip addr

192.168.1.250 255.255.255.0

router (config) # no shutdown

router (config-if) # clock rate 64000

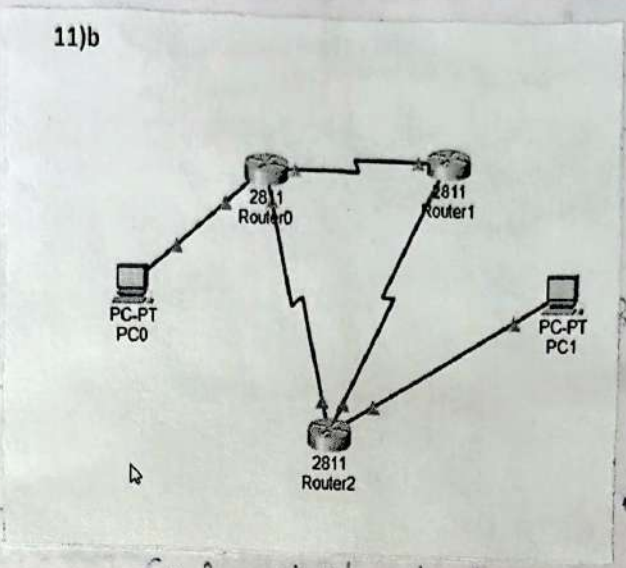
router (config-if) # bandwidth

router (config-if) # no shutdown

router (config-if) # exit

By default RIP will use the route that has low hops count between source and destination. In our network route 1 has low hops counts so it will be selected. We

can. ever trace command to verify it.



RESULT:

Hence the experiment on Simulating RIP config Cisco packet traces has been executed successfully.

EXPERIMENT No: 12a
DATE: 25/9/25

Implementation of echo client server using TCP/UDP sockets

AIM:

To implement an echo client server by using TCP/UDP sockets.

Server-side Algorithm:-

```
import socket
server socket = socket.socket(socket.
                AF_INET, socket.SOCK_STREAM
server_socket.bind((localhost, 12345))
server_socket.listen(1)
printf("connected to {addr}")
while true:
  data = conn.recv(1024).decode()
  if = conn.recv(1024).decode()
  if no/date or data.lower() ==
                'bye'.
  break
  print("Received from client: {data}
  conn.send(data.encode());
  conn.close()
```

Client - side Algorithm :-

```
import socket
client_socket = socket.socket (socket.
                AF_INF, socket.SOCK_STREAM)
client_socket.connect (('local host';12345))
while trace:
message = input ("enter message: ")
client_socket.send (message.encode())
if message.lower () = 'bye'.
break
```

client side :-

enter message : Hello server

echo from server : Hello server

enter message : How are you?

echo from server : How are you?

enter message : Bye.

server side :

server is waiting for connection

connected to ('127.0.0.1', 58944)

Received from client : Hello server

Received from client : How are you?

connection closed.

12 . server.py

```
UDP Server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 63852): Hello
```

client.py

```
Enter message to send: Hello
Echo from server: Hello
```

RESULT:-

The task has been successfully executed.

EXPERIMENT NO: 12 b | Implementation of Chat Client Server.

DATE: 28/9/25

AIM:-

To implement chat client server using the TCP and UDP sockets.

Server Client Algorithm

```
import socket
server_socket: socket (socket. AF_INET)
        socket. SOCK_STREAM)
server. find (('local host', 123456))
server. listen (1)
print ("server is waiting for connection")
conn, addr = server. accept ()
printf (f" connected to {addr}")

while true :
        msg = conn. recv (1024). decode ()
        if msg. lower () = = bye
        print (" Client disconnected")
        break
        printf (' client: {msg}")
                reply = input / you).
```

```
conn.send(reply.encode())
    if reply.lower() == bye.
    break
conn.close()
```

Client side algorithm :-

```
import socket
client = socket.socket(socket.AF_INET
        socket.SOCK_STREAM)
client connect (('local host', 12345))
while true:
    message = input (you:)
    if message.lower() == bye
    break
    reply = client.recv(1024).decode()
        printf (f" server {reply})
    if reply.lower() == bye.
        break
client.close()
```

Sample Input and Output.

Server side:

server waiting for connection

conn to ('127.0.0.1', 570.0)

client : server/

you : Hello client

client : How are you?

you : I'm fine, thanks!

Client : bye

Client side:

you : Hi (server)

server : Hello client.

you : How are you?

server : I'm fine, thanks!

You : bye!

RESULT:

The implementation of chat client server using TCP/UDP has been executed successfully.

EXPERIMENT No: 13 | Implementation of Ping Program.
DATE: 6/10/25

AIM: To implement a ping program by self.

Client Server Algorithm:

```
import socket
import time
def ping-server (host = '127.0.0.1')
        port = 12345)
socket socket /socket AF_INET,
        socket. SOCKET) as s
try:
s. send to (b'Hello', (host, port))
except s. timeout
    print ("Request timed out")
ping. server ()
```

Server side Algorithm:

```
import socket
    def start - server (host
        = '127. 0. 0. 1')
        port = 12345):
```

```
with socket.socket(socket.AF_INET,
    Socket.SOCK_DGRAM) as s.
s.bind((host, port))
print(f"UDP server running on
        {host}:{port}")
while true:
  data, addr = s.recvfrom(1024)
    printf(f"Received message
    from {addr}: {data.decode()}")
        Start_server()
```

Sample Input and Output

Server side:

command: python server.py.

Output:

UDP server running on 127.0.0.1:
12345 received from ('127.0.0.1',
55437): ping.

Client side

Sample input :- enter host: 127.0.0.1

Sample output: reply from 127.0.0.1
ping time = 1.42.

RESULT:

The implementation of ping program has been executed successfully.

15/10/26

EXPERIMENT No: 14 | Implementation of Packet Sniffing using Raw Sockets

DATE: 6/10/25

AIM:

Write a code using Raw sockets to implement packet sniffing.

Algorithm:

```
from scapy.all import sniff
from scapy.layers.inet import
    IP, TCP, UDP, ICMP
def packet = call back (packet)
if IP in packet:
    layer = packet[IP]
    protocol = ip_layer.proto
    src_ip = ip_layer.src
    dst_ip = ip_layer.dst
# Determine the protocol
    protocol_name = ""
    if protocol == 1:
protocol_name = "ICMP"
    elif protocol == 6:
    protocol_name = "TCP"
    elif protocol == 17:
```

```
protocol_name = "UDP"
else:
    protocol_name = "Unknown Protocol"
# print Packet details
print (f"Protocol : {protocol_name}")
print (f"Source IP : {src_ip}")
print (f"Destination IP : {dst_ip}")
print ("-" * 50)
# capture packets on default
network
    sniff (iface=conf.iface, prn=packet
Callback, filter="ip", store=0)
```

<u>Output</u>:

protocol : TCP

Source IP : 192.168.1.400

Destination IP : 142.250.182.14.

<u>RESULT</u>:

Implementation of packet sniffing using RAW sockets is executed successfully.

18/10/26