

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Networks

Submitted by

Avyukth Inna (1BM22CS060)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Academic Year 2024-25 (odd)

B.M.S. College of Engineering

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “ Computer Network (23CS5PCCON)” carried out by **Avyukth Inna (1BM22CS060)**, who is a bonafide student of **B.M.S. College of Engineering**. It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Dr. Nandhini Vineeth Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
---	--

Index

Cycle - I

Sl.No.	Date	Experiment Title	Page No.
1	25/09/2024	Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.	1
2	09/10/2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	4
3	16/10/2024	Configure default route, static route to the router	7
4	23/10/2024	Configure DHCP within a LAN and outside LAN.	12
5	13/11/2024	Configure RIP routing Protocol in Routers.	15
6	20/11/2024	Configure OSPF Routing Protocol	19
7	27/11/2024	Demonstrate the TTL/ Life of a Packet	25
8	18/12/2024	Configure Web Server & DNS	27
9	18/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	30
10	18/12/2024	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	34

11	18/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	37
12	18/12/2024	To construct a WLAN and make the nodes communicate wirelessly	40

Cycle-II

Sl.No.	Date	Experiment Title	Page No.
1	1/1/2025	Write a program for error detecting code using CRC-CCITT (16-bits).	44
2	1/1/2025	Write a program for congestion control using Leaky bucket algorithm	47
3	1/1/2025	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	49
4	1/1/2025	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	52
5	1/1/2025	Tool Exploration - Wireshark	55

Github Link:

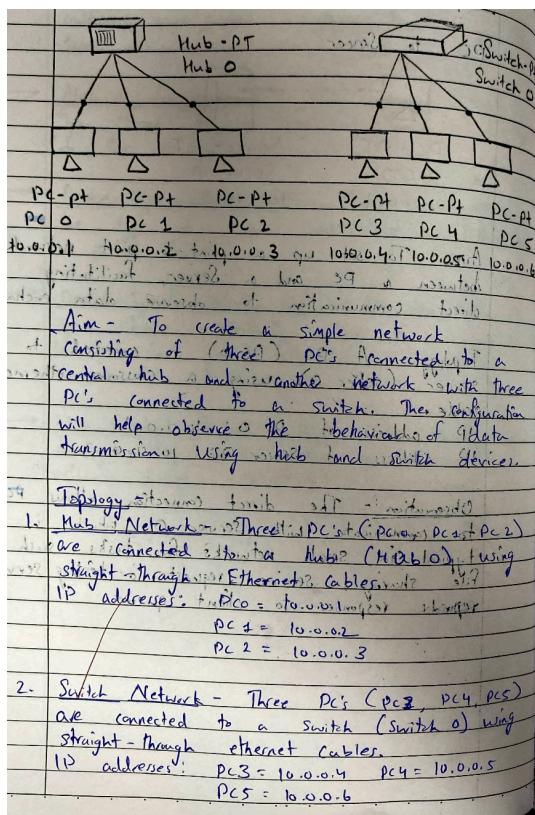
https://github.com/avyukthinna/CN_Lab

Program 1

Aim of the program:

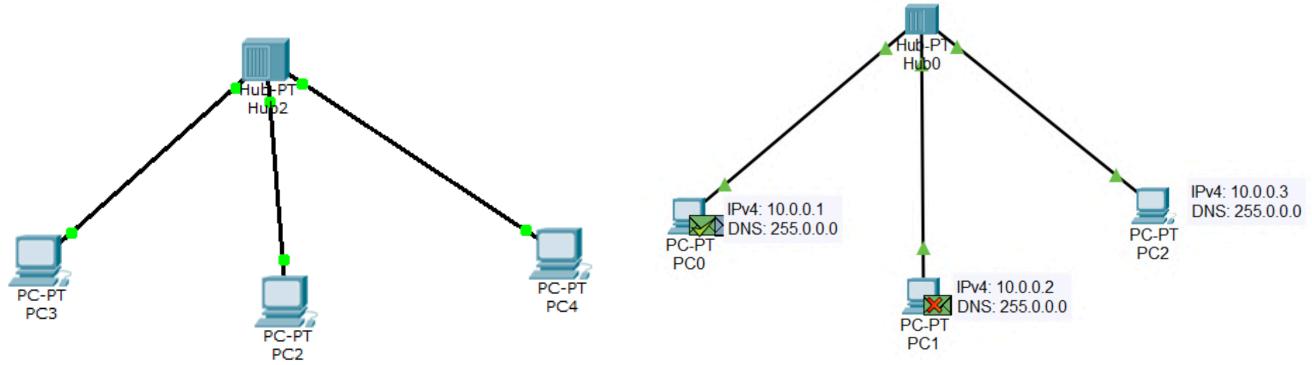
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

Procedure along with the topology:



- Procedure
1. Add 1 hub, 1 switch and 16 PCs!
PC0, PC1, PC2 for the hub.
PC3, PC4, PC5 for the switch to the Cisco packet tracer workspace.
 2. Use copper straight-through cables to connect PC0, PC1, PC2 to hub 0, connect PC3, PC4, PC5 to switch 0 using same type of cables.
 3. Assign IP address to each PC and obtain subnet mask.
 4. Set switch simulation mode to observe data traffic behaviour when packets are sent between the 2 devices.
 5. In the hub network, notice when the hub broadcasts packets to all devices causing potential traffic overload.
 6. In the switch network, observe how the switch forwards packets only to the intended recipient reducing unnecessary traffic.
 7. The hub broadcasts data to all connected devices leading to more network congestion, while the switch efficiently sends data only to the correct device, optimizing performance.

Screen shots/ output:



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit (edit)	Delete
	Successful	PC0	PC2	ICMP		0.000	N	0	Edit (edit)	Delete

```

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

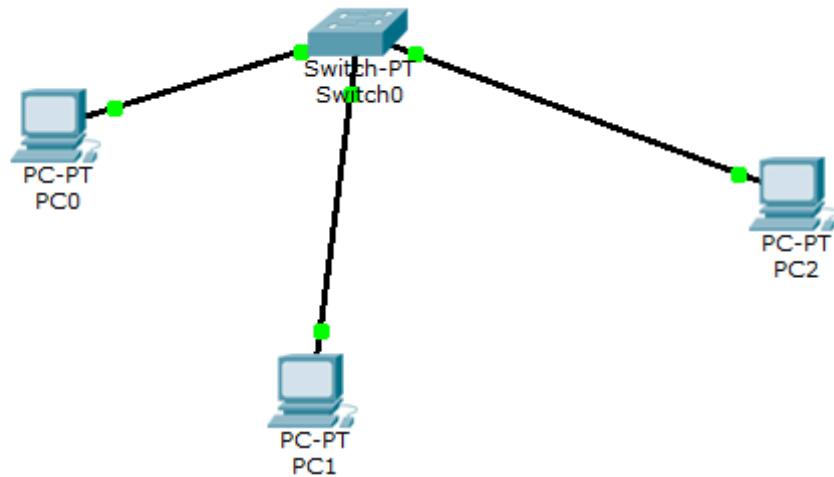
Ping statistics for 20.0.0.1:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.1: bytes=32 time=12ms TTL=125
Reply from 20.0.0.1: bytes=32 time=15ms TTL=125
Reply from 20.0.0.1: bytes=32 time=15ms TTL=125

Ping statistics for 20.0.0.1:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 12ms, Maximum = 15ms, Average = 14ms
PC>

```



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

```
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 9ms, Average = 2ms
```

Observation:

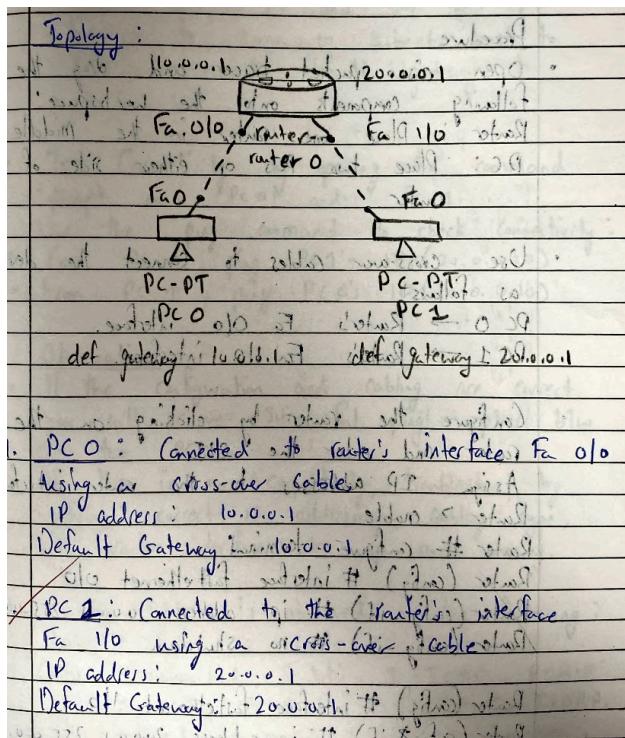
	<u>Observation</u>
1.	The hub broadcasts packets to all devices which may cause unnecessary traffic.
2.	The switching forwards packets only to the appropriate device by learning MAC address, making it more efficient by reducing traffic sent to other hosts.

Program 2

Aim of the program:

Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

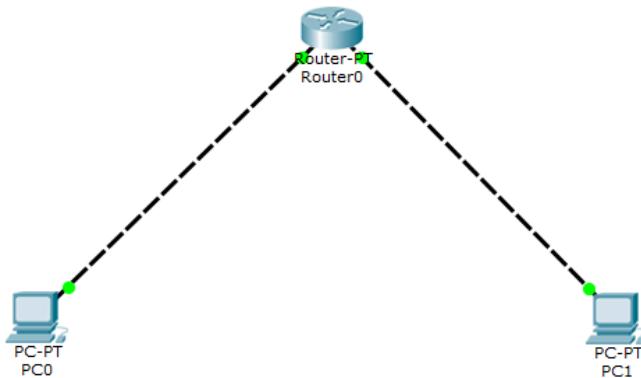
Procedure along with the topology:



3.	Router :
	• Interface Fa 0/0 connected to PC 0
	• Interface Fa 1/0 connected to PC 1
	• IP address of Fa 0/0 : 10.0.0.1
	• IP address of Fa 1/0 : 20.0.0.1
	Router (config-if) # ip address 10.0.0.1 255.0.0.0
	Router (config-if) # no shutdown
	Router (config) #
	Procedure
	• Open Cisco packet tracer and drag the following components onto the workspace: Router : Place one router in the middle. PC : Place two PCs on either side of the router
	• Use cross-over cables to connect the devices as follows: PC 0 → Router's Fa 0/0 interface. PC 1 → Router's Fa 1/0 interface.
	• Configure the router by clicking on the Router and entering the (CLI) : Router# Assign IP addresses to the router's interfaces: Router > enable Router # configure terminal Router (config) # interface fastethernet 0/0 Router (config-if) # ip address 10.0.0.1 255.0.0.0 Router (config-if) # no shutdown
	Router (config) # interface fastethernet 1/0 Router (config-if) # ip address 20.0.0.1 255.0.0.0 Router (config-if) # no shutdown

• Configure the PCs :-
For PC 0 :-
• Click on PC 0 and set the IP address to 10.0.0.10, subnet mask to 255.0.0.0 and default gateway to 10.0.0.1
For PC 1 :-
• Click on PC 1 and set the IP address to 20.0.0.10, subnet mask to 255.0.0.0 and default gateway to 20.0.0.1
• Test Connectivity by opening the command prompt on PC 0 and PC 1.
• Use the ping command to check connectivity :-
→ From PC 0 ; ping PC 1's IP (20.0.0.10)
→ From PC 1 ; ping PC 0's IP (10.0.0.10)

Screen shots/ output:



Router0

Physical	Config	CLI
IOS Command Line Interface		
<pre> Router>enable Router>config terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.2 Router(config)# Router(config)#interface Serial2/0 Router(config-if)#exit Router(config)#exit Router# \$SYS-5-CONFIG_I: Configured from console by console Router#show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area * - candidate default, U - per-user static route, o - ODR P - periodic downloaded static route Gateway of last resort is not set C 10.0.0.0/8 is directly connected, FastEthernet0/0 S 20.0.0.0/8 [1/0] via 30.0.0.2 C 30.0.0.0/8 is directly connected, Serial2/0 Router# Router#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)# Router(config)#interface FastEthernet0/0 </pre>		
<input type="button" value="Copy"/> <input type="button" value="Paste"/>		

PC0

Physical	Config	Desktop	Custom Interface
<p>Command Prompt</p> <pre> PC>ping 30.0.0.2 Pinging 30.0.0.2 with 32 bytes of data: Reply from 30.0.0.2: bytes=32 time=13ms TTL=254 Reply from 30.0.0.2: bytes=32 time=5ms TTL=254 Reply from 30.0.0.2: bytes=32 time=10ms TTL=254 Reply from 30.0.0.2: bytes=32 time=7ms TTL=254 Ping statistics for 30.0.0.2: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 5ms, Maximum = 13ms, Average = 8ms PC>ping 20.0.0.1 Pinging 20.0.0.1 with 32 bytes of data: Reply from 20.0.0.1: bytes=32 time=1ms TTL=126 Reply from 20.0.0.1: bytes=32 time=7ms TTL=126 Reply from 20.0.0.1: bytes=32 time=14ms TTL=126 Reply from 20.0.0.1: bytes=32 time=7ms TTL=126 Ping statistics for 20.0.0.1: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 1ms, Maximum = 14ms, Average = 7ms </pre>			

Observation:

S1	<p><u>Observation - unit 58 - lab 01.0005 NTP</u></p> <ul style="list-style-type: none"> If the configuration and cabling are correct, you will receive successful ping or replies below the terminal PC screen. If there is no connectivity, troubleshoot by verifying: correct IP addressing, cabling type, both router interfaces are up and running. <p>Routing table is observed as the following: Router > show ip route Codes: C-connected, S-static, I-IGRP, R-RIP M-Mobile, B-BGP, D-EIGRP, Ex-EIGRP, external, O-OSPF, IA-OSPF inter area, N1-OSPF NSSA external type 1, N2-OSPF NSSA external type 2, E1-OSPF external type 1, E2-OSPF external type 2, E-EGP</p>
----	--

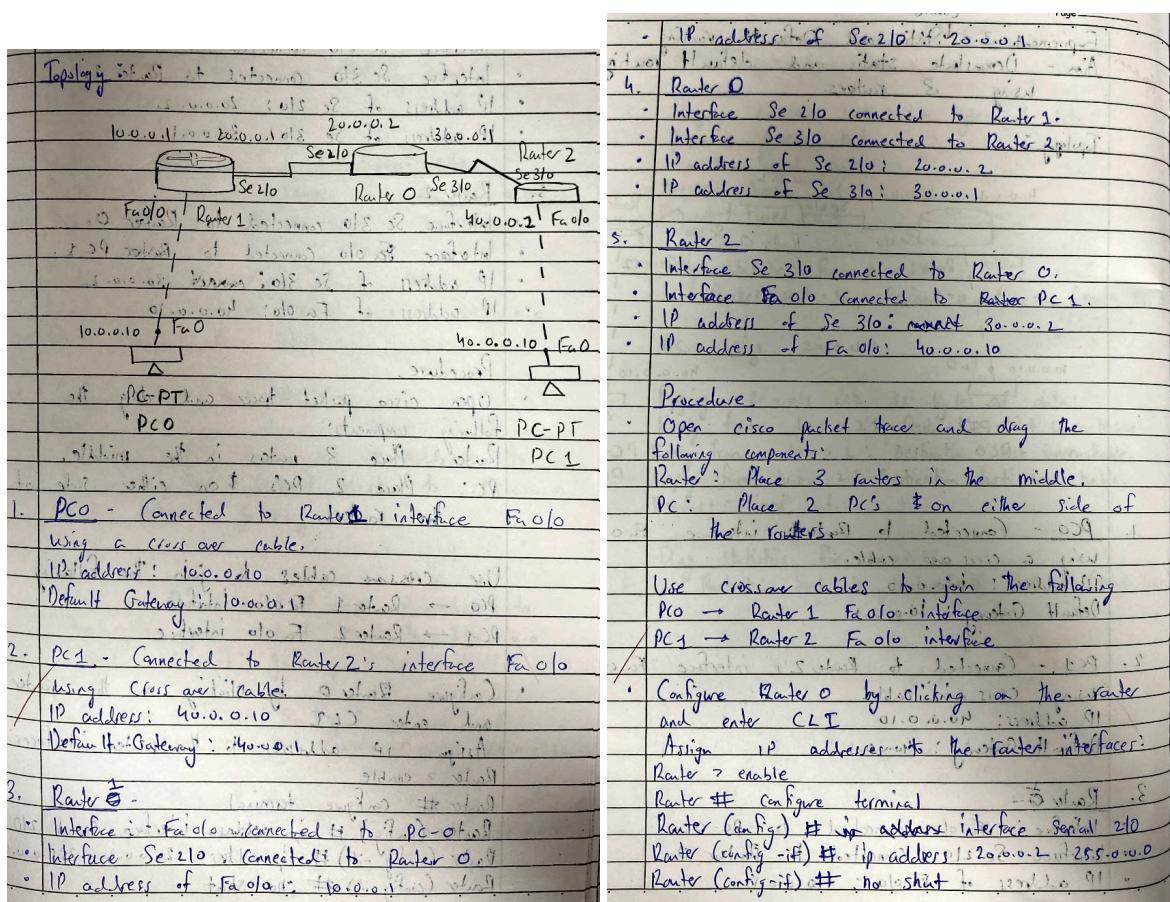
i - IS-IS, L1 - IS-IS level-1 L2 - IS-IS level-2, ia - IS-IS inter area * - Candidate default, o - per-link static route, ! - 0.0.0.0/0; npr - periodically downloaded static route, !# - 0.0.0.0/0, lms
Gateway of last resort is not set.
C1 10.0.0.1/8 is directly connected, Fast Ethernet 0/0 C 20.0.0.0/8 is directly connected, Fast Ethernet 1/0
The ping results are as follows:
<pre> Router> ping 20.0.0.10 Pinging 20.0.0.10 with 32 bytes of data: Reply from 20.0.0.10: bytes=32 time=0ms (0.000000) 0.000000 ms TTL=127 Reply from 20.0.0.10: bytes=32 time=0ms TTL=127 Reply from 20.0.0.10: bytes=32 time=0ms TTL=127 Reply from 20.0.0.10: bytes=32 time=0ms TTL=127 Ping statistics for 20.0.0.10: Packets: Sent = 4, Received = 4, Lost = 0 Approximate round trip times in milliseconds: Minimum = 0ms, Maximum = 0ms, Average = 0ms </pre>

Program 3

Aim of the program:

Configure default route, static route to the Router

Procedure along with the topology:

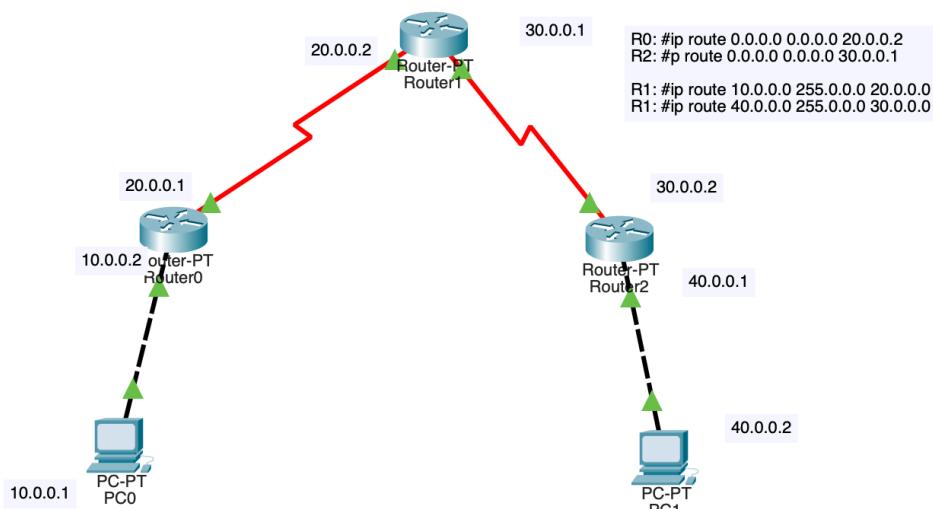


```

Router (config) # interface serial 3/0
Router (config-if) # ip address 130.0.0.1 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # exit
Router (config) # Router 1 is in the following configuration
Router > enable
Router # configure terminal
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shutdown
Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
Router (config-if) # no shutdown
Router (config) # Configure the two PC's -
PC0: 10.0.0.1, IP address is 20.0.0.1
PC1: 10.0.0.2, IP address is 20.0.0.2
Click on PC0 and set up IP address to
10.0.0.10 and subnet mask to 255.0.0.0
and default gateway 10.0.0.1.
PC1: 10.0.0.11, IP address is 20.0.0.2
Click on PC1 and set up IP address to
10.0.0.11 and subnet mask to 255.0.0.0
and default gateway 10.0.0.1.
Router # ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router 2 - Default Routing is fitted
Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1
: this will be learnt by default gateway.

```

Screen shots/ output:



PC1

Physical Config Desktop Custom Interface

Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=9ms TTL=125
Reply from 10.0.0.1: bytes=32 time=13ms TTL=125
Reply from 10.0.0.1: bytes=32 time=10ms TTL=125
Reply from 10.0.0.1: bytes=32 time=15ms TTL=125

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 15ms, Average = 11ms

PC>
```

PC1

Physical Config Desktop Custom Interface

Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=9ms TTL=125
Reply from 10.0.0.1: bytes=32 time=13ms TTL=125
Reply from 10.0.0.1: bytes=32 time=10ms TTL=125
Reply from 10.0.0.1: bytes=32 time=15ms TTL=125

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 15ms, Average = 11ms

PC>
```

PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Pinging 40.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 40.0.0.2:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>show ip route
Invalid Command.

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:
Reply from 40.0.0.2: bytes=32 time=3ms TTL=125
Reply from 40.0.0.2: bytes=32 time=4ms TTL=123
Reply from 40.0.0.2: bytes=32 time=13ms TTL=123
Reply from 40.0.0.2: bytes=32 time=12ms TTL=123

Ping statistics for 40.0.0.2:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 13ms, Average = 8ms
PC>
```

Realtime

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Failed	PC0	PC2	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC0	PC2	ICMP		0.000	N	1	(edit)	(delete)

New Delete

Toggle PDU List Window

Router0

Physical Config CLI

IOS Command Line Interface

```
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
S    20.0.0.0/8 [1/0] via 30.0.0.2
      [1/0] via 40.0.0.2
C    30.0.0.0/8 is directly connected, Serial2/0
S    40.0.0.0/8 [1/0] via 20.0.0.2
Router#enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
S    20.0.0.0/8 [1/0] via 30.0.0.2
      [1/0] via 40.0.0.2
C    30.0.0.0/8 is directly connected, Serial2/0
S    40.0.0.0/8 [1/0] via 20.0.0.2
Router#
```

Copy Paste

Observation:

	<u>Observation</u> : If configuration and cabling are correct, you will receive two successive ping replies from both the two PC's.
	Router # show ip route
	C 10.0.0.0/8 [1/0] via 20.0.0.1
	C 20.0.0.0/8 is directly connected, Serial 2/0
	C 20.0.0.18 is directly connected, serial 3/0
	S* 0.0.0.0/0 [1/0] via 20.0.0.2

.	Static Routing of Router 1
	Router # show ip route
	S 10.0.0.0/8 [1/0] via 20.0.0.1
	C 20.0.0.0/8 is directly connected, serial 2/0
	C 20.0.0.18 is directly connected, serial 3/0
	S* 0.0.0.0/0 [1/0] via 20.0.0.2

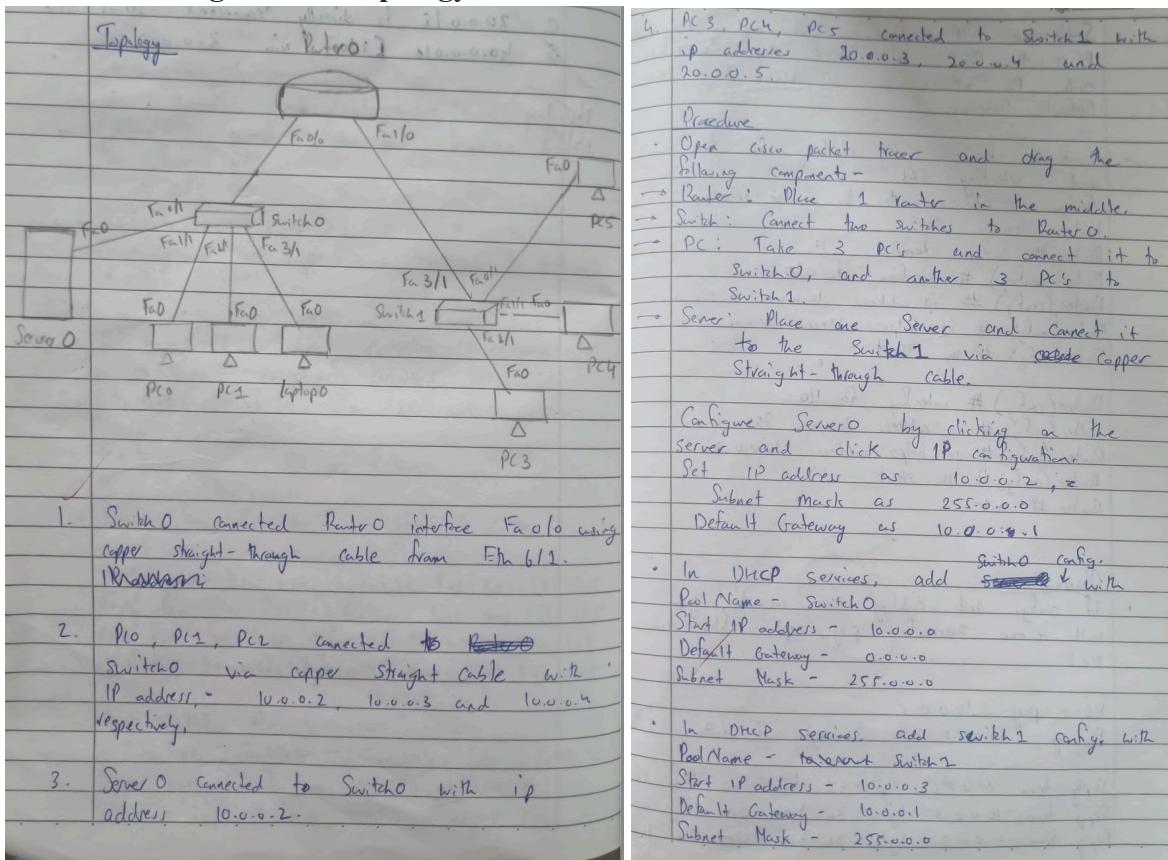
N
23/10/04

Program 4

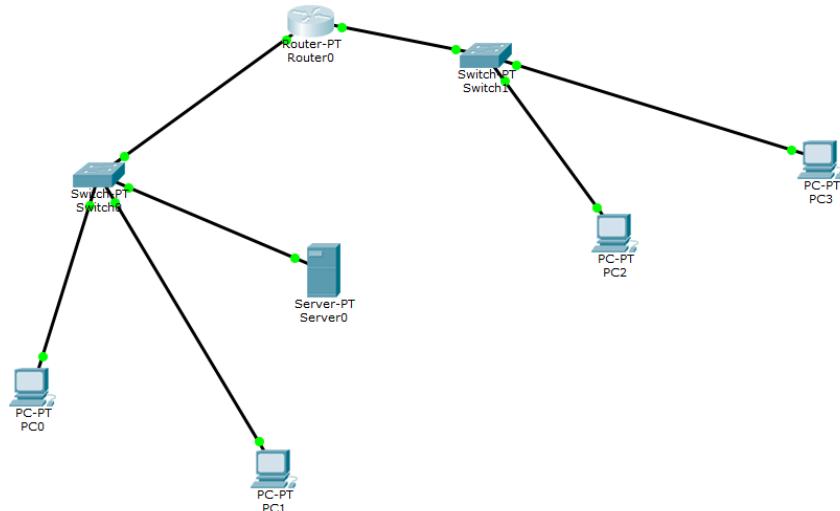
Aim of the program:

Configure IP address of the host using DHCP server within and outside a LAN.

Procedure along with the topology:



Screen shots/ output:



Server0

Physical Config Services Desktop Programming Attributes

SERVICES

- HTTP
- DHCP**
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

DHCP

Interface	FastEthernet0	<input checked="" type="radio"/> Service On	<input type="radio"/> Off
Pool Name	serverPool1		
Default Gateway	10.0.0.2		
DNS Server	10.0.0.1		
Start IP Address :	10	0	0
Subnet Mask:	255	0	0
Maximum Number of Users :	512		
TFTP Server:	0.0.0.0		
WLC Address:	0.0.0.0		

Add Save Remove

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool1	10.0.0.2	10.0.0.1	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool2	20.0.0.1	10.0.0.1	20.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
●	Successful	PC1	Laptop1	ICMP		0.004	N	1	(edit)	

PC0

Physical Config Desktop Programming Attributes

GLOBAL

- Settings
- Algorithm Settings

INTERFACE

- FastEthernet0
- Bluetooth

FastEthernet0

Port Status	<input checked="" type="checkbox"/> On
Bandwidth	<input type="radio"/> 100 Mbps <input type="radio"/> 10 Mbps <input checked="" type="checkbox"/> Auto
Duplex	<input type="radio"/> Half Duplex <input type="radio"/> Full Duplex <input checked="" type="checkbox"/> Auto
MAC Address	0005.5EA7.9778
IP Configuration	<input checked="" type="radio"/> DHCP <input type="radio"/> Static
IPv4 Address	10.0.0.3
Subnet Mask	255.0.0.0
IPv6 Configuration	<input type="radio"/> Automatic <input checked="" type="radio"/> Static
IPv6 Address	FE80::205:5EFF:FEA7:9778
Link Local Address	FE80::205:5EFF:FEA7:9778

Observation:

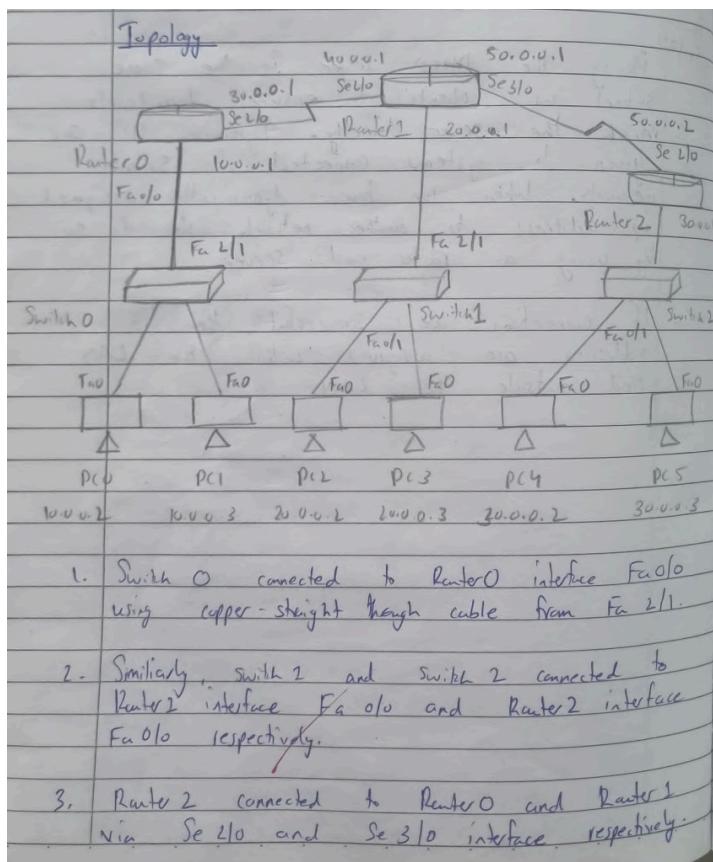
	Reply from 10.0.0.3: bytes = 32 time = 2 ms TTL = 120
	Ping statistics for 10.0.0.3: Packets: Sent = 4, Received = 4, Loss = 0 (0% loss) Approximate round trip times in milliseconds: Minimum = 0 ms, Maximum = 2 ms, Average = 0 ms
N 13/11/24	Placing the DHCP server in the same subnet as clients to ensure broadcasts reach the server directly. Dynamic IP are given to systems connected in same network. When we have dynamically assigned ip address to another network, we do so by using a router and server.
	. If connections are successful, the ip addresses are assigned within the LAN and outside the LAN.

Program 5

Aim of the program:

Configure RIP routing Protocol in Routers

Procedure along with the topology:



- 4. PC0, PC1 connected to Switch 0 via copper straight cable with IP address 10.0.0.2, 10.0.0.3 respectively.
 - 5. PC2, PC3 connected to Switch 1 via copper straight cable with IP address 20.0.0.2 and 20.0.0.3 respectively.
 - 6. PC4, PC5 connected to Switch 2 via copper straight cable with IP address 30.0.0.2 and 30.0.0.3 respectively.
- Procedure:**
- Open cisco packet tracer and drag the following components:
 Router: Place 3 routers in the middle.
 Switch: Place 3 switches in and connect them to the routers with Fa 2/1 interface.
 PC: Place 6 PCs, two of them connected to each of the 3 switches via Fa 0/0 interface.
 - Configure all the 3 routers:
 Router 0:
 Router #> config terminal
 Router #>(config) # interface serial 2/0
 Router (config) # ip address 30.0.0.1 255.0.0.0
 Router (config) # no shutdown
 - Router 1:
 Router (config) # interface serial 2/0
 Router (config) # ip address 20.0.0.1 255.0.0.0

```

Router (config) # interface Fa 0/0
Router (config) # ip address 20.0.0.1 255.0.0.0
Router (config) # interface Se 3/0
Router (config) # int ip address 50.0.0.1 255.0.0.0
Router (config) # no shutdown

• Router 3:
Router (config) # interface Fa 0/0
Router (config) # ip address 30.0.0.1 255.0.0.0
Router (config) # interface Se 2/0
Router (config) # ip address 50.0.0.2 255.0.0.0

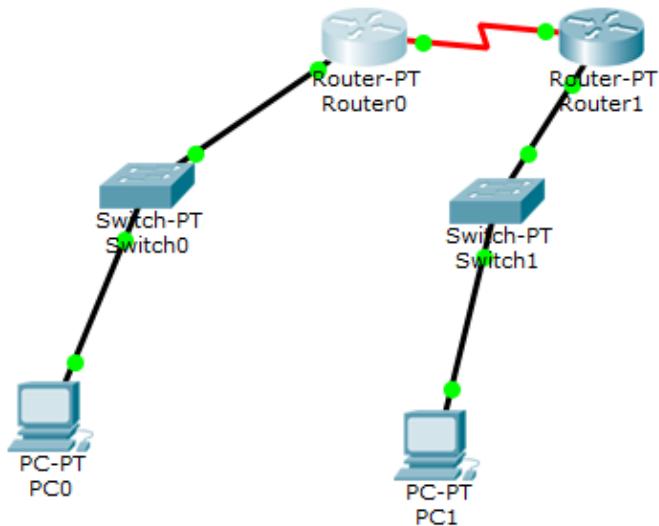
→ Configure the RIP for all routers
• Router 1:
Router (config) # router rip
Router (config) # network 10.0.0.0
Router (config) # network 30.0.0.0

• Router 2:
Router (config) # router rip
Router (config) # network 40.0.0.0
Router (config) # network 20.0.0.0
Router (config) # network 50.0.0.0

• Router 3:
Router (config) # router rip
Router (config) # network 30.0.0.0
Router (config) # network 50.0.0.0

```

Screen shots/ output:



Router0

GLOBAL		RIP Routing								
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Failed	PC0	PC1	ICMP	Green	0.000	N	0	(edit)	(delete)
●	Successful	Router0	Router1	ICMP	Blue	0.000	N	1	(edit)	(delete)
●	Successful	PC0	PC1	ICMP	Brown	0.000	N	2	(edit)	(delete)

Equivalent IOS Commands

```

Router(config-if)#exit
Router(config)#
Router(config)#router rip
Router(config-router)#
Router(config-router)#exit
Router(config)#
Router(config)#router rip
Router(config-router)#network 192.168.1.0
Router(config-router)#

```

Router1

GLOBAL		RIP Routing								
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Failed	PC0	PC1	ICMP	Green	0.000	N	0	(edit)	(delete)
●	Successful	Router0	Router1	ICMP	Blue	0.000	N	1	(edit)	(delete)
●	Successful	PC0	PC1	ICMP	Brown	0.000	N	2	(edit)	(delete)

Equivalent IOS Commands

```

$LINK-S=CHANGED: Interface Serial2/0, changed state to up
*LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
ip address 10.10.0.3 255.0.0.0
Router(config-if)#
Router(config-if)#exit
Router(config)#
Router(config)#router rip
Router(config-router)#
Router(config-router)#
Router(config-router)#
Router(config-router)#
Router(config-router)#
Router(config-router)#

```

IOS Command Line Interface

```

Router>enable
Router>configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
R    20.0.0.0/8 [120/1] via 40.0.0.2, 00:00:12, Serial2/0
R    30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:12, Serial2/0
C    40.0.0.0/8 is directly connected, Serial2/0
R    50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:12, Serial2/0
Router#

```

PC0

Command Prompt

```

Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.2: bytes=32 time=9ms TTL=125
Reply from 30.0.0.2: bytes=32 time=8ms TTL=125
Reply from 30.0.0.2: bytes=32 time=10ms TTL=125

Ping statistics for 30.0.0.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 10ms, Average = 9ms
PC>

```

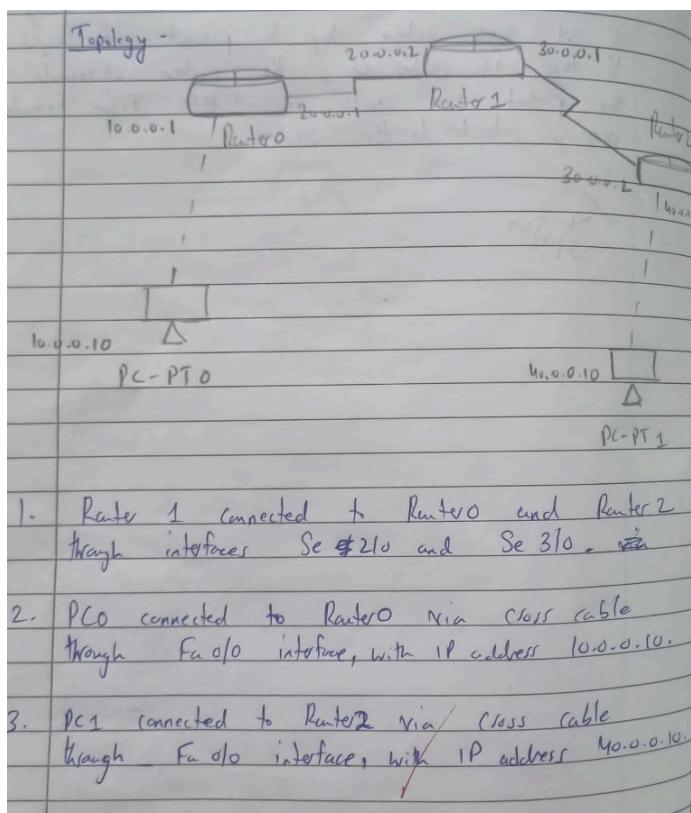
Observation:

- | | |
|--|---|
| | <u>Observation:</u> |
| | <ul style="list-style-type: none">The routers communicate with each other and share a their routing table
each of their |
| | <p style="text-align: center;">↙
26 12 24</p> |

Program 6

Configure OSPF routing protocol

Procedure along with the topology



Procedure

Open Cisco packet tracer and drag the following:

Router: Place 3 routers in the middle.

PC: Place 2 PCs, each connected to Router 0 and Router 1 via Fa 0/0 interface.

Configure all 3 routers:

Router 0:

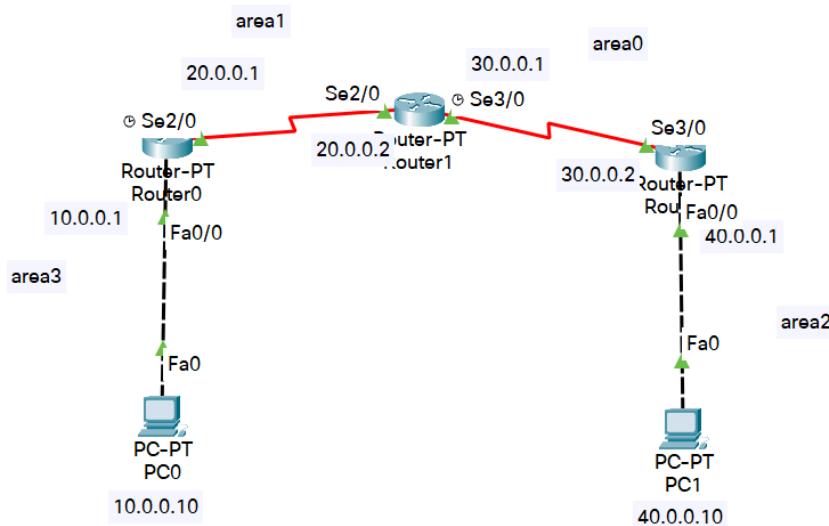
- Router > enable
- Router (config) # > config terminal
- Router (config) > interface fastethernet 0/0
- Router (config-if) > ip address 10.0.0.1 255.0.0.0
- Router (config-if) > no shut
- Router (config) > interface serial 0/2/0
- Router (config-if) > ip address 20.0.0.1 255.0.0.0
- Router (config-if) > encapsulation ppp
- Router (config-if) > clock rate 64000
- Router (config-if) > no shut

Router 1:

- Router (config) > interface serial 2/0
- Router (config-if) > encapsulation ppp
- Router (config-if) > ip address 20.0.0.2 255.0.0.0
- Router (config) > interface serial 3/0
- Router (config-if) > ip address 30.0.0.1 255.0.0.0
- Router (config-if) > encapsulation ppp
- Router (config-if) > clock rate 64000
- Router (config-if) > no shut

<pre> Router 2 - Router (config) > interface serial 2/0 Router (config-if) > ip address 80.0.0.2 255.0.0.0 Router (config-if) > encapsulation PPP Router (config-if) > clock rate 64000 Router (config-if) > no shut Router (config) > interface fastethernet 0/0 Router (config-if) > ip address 40.0.0.1 255.0.0.0 Router (config-if) > no shut Router (config) > exit PC1 Set IP address = 10.0.0.10 Subnet mask = 255.0.0.0 Gateway = 10.0.0.1 PC2 Set IP address = 40.0.0.10 Subnet mask = 255.0.0.0 Gateway = 40.0.0.1 Enable ip routing for configuring ospf routing protocol in all routers- </pre>	<pre> Router 2 - R2 (config) # router ospf 1 " # router-id 2.2.2.2 " # network 20.0.0.0 0.255.255.255 area 1 " # network 30.0.0.0 0.255.255.255 area 0 " # exit Router 2 - R2 (config) # router ospf 1 " # router-id 3.3.3.3 " # network 30.0.0.0 0.255.255.255 area 0 " # network 40.0.0.0 0.255.255.255 area 2 " # exit → Configure loopback address to routers R2(config)# interface loopback 0 R2(" " ip add 172.16.1.252 255.255.255.0 " " no shut R1 (config) # interface loopback 0 " # ip address 172.16.1.253 255.255.255.0 " # no shut R2 (config) # interface loopback 0 R2 (config) # ip add 172.16.1.254 255.255.255.0 R2 (config) # ip route 12.4.6.254 no shut → Create virtual link b/w R0, R2 . Router 0 R0 (config) # router ospf 1 R0 (config) # area 1 virtual-link 2.2.2.2 </pre>
--	---

Screen shots/ output



Router0

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up

Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#exit
Router(config)#+
```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to
up

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
^
% Invalid input detected at '^' marker.

Router(config-if)#no shutdown
```

Router2

```

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
%IP-4-DUPADDR: Duplicate address 40.0.0.1 on FastEthernet0/0, sourced by
000D.BDDA.0123

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to
up

```

OSPF Routing Protocol

Router0

```

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 1.1.1.1
Router(config-router)#network 10.0.0.0 0.255.255.255 area 3
Router(config-router)#network 20.0.0.0 0.255.255.255 area 1
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#sho
00:27:19: *OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FULL, Loading Done
w ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
      20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          20.0.0.0/8 is directly connected, Serial2/0
C          20.0.0.2/32 is directly connected, Serial2/0
O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:00:02, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:02, Serial2/0

```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 2.2.2.2
Router(config-router)#network 20.0.0.0 0.255.255.255 area 1
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

00:26:21: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on Serial3/0 from LOADING to FULL, Loading Done
00:27:18: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial2/0 from LOADING to FULL, Loading Done

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        20.0.0.0/8 is directly connected, Serial2/0
C        20.0.0.1/32 is directly connected, Serial2/0
      30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        30.0.0.0/8 is directly connected, Serial3/0
C        30.0.0.2/32 is directly connected, Serial3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:02:00, Serial3/0
```

Router2

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 3.3.3.3
Router(config-router)#network 40.0.0.0 0.255.255.255 area 2
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
00:26:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial3/0 from LOADING to FULL, Loading Done

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:02:45, Serial3/0
      30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        30.0.0.0/8 is directly connected, Serial3/0
C        30.0.0.1/32 is directly connected, Serial3/0
C        40.0.0.0/8 is directly connected, FastEthernet0/0
```

Pinging

```
C:\>ping 40.0.0.10  
Pinging 40.0.0.10 with 32 bytes of data:  
Reply from 40.0.0.10: bytes=32 time=24ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=20ms TTL=125  
  
Ping statistics for 40.0.0.10:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 18ms, Maximum = 24ms, Average = 20ms
```

Observation

Observation

The experiment demonstrates how OSPF dynamically learns and advertises routes, enabling efficient and scalable routing across multiple areas.

Routing tables on all routers must display networks from all areas with O IA indicating inter-area router.

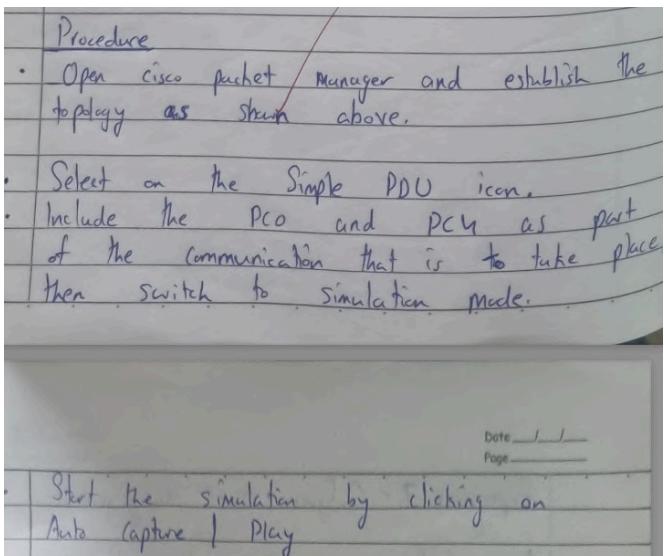
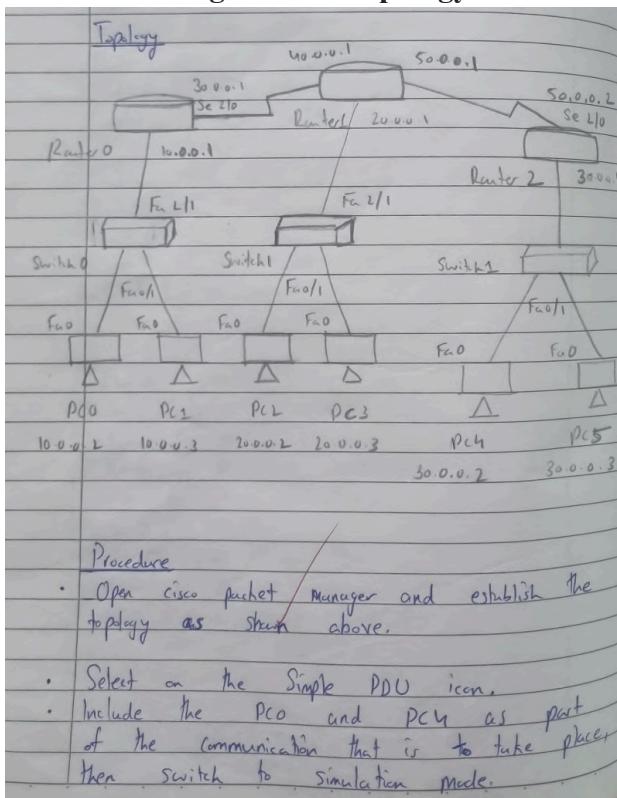
N 1st

Program 7

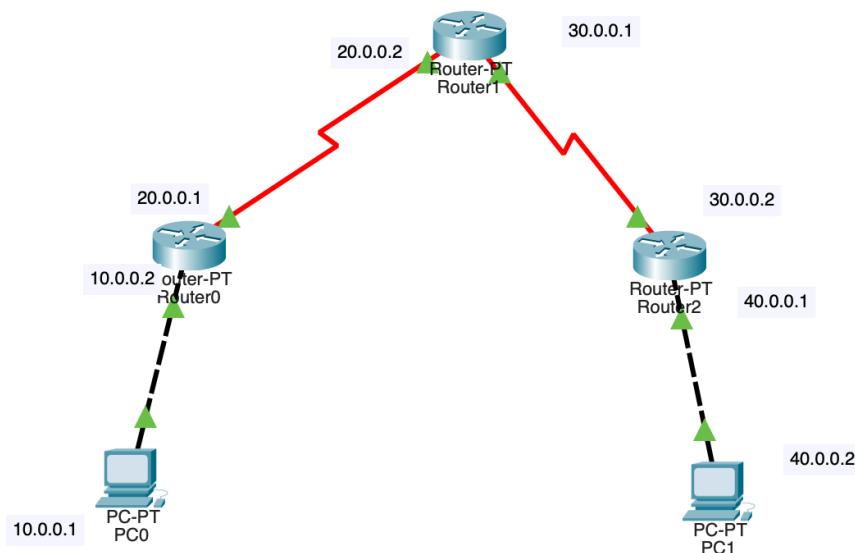
Aim of the program:

Demonstrate the TTL/ Life of a Packet

Procedure along with the topology:



Screen shots/ output:



PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19	Bytes
PREAMBLE: 101010...1011		DEST MAC: 0001.C780.D8C7	SRC MAC: 00E0.F711.E727		
TYPE: 0x800		DATA (VARIABLE LENGTH)		FCS: 0x0	

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x4		0x0	0x0		
TTL: 255	PRO: 0x1	CHKSUM			
SRC IP: 10.0.0.1					
DST IP: 40.0.0.2					
OPT: 0x0		0x0			
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x8		CODE: 0x0	CHECKSUM
ID: 0x7		SEQ NUMBER: 4	

PDU Information at Device: Router1

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19	Bytes
PREAMBLE: 101010...1011		DEST MAC: 0001.6497.4E46	SRC MAC: 0005.5ECB.7262		
TYPE: 0x800		DATA (VARIABLE LENGTH)		FCS: 0x0	

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x4		0x0	0x0		
TTL: 128	PRO: 0x1	CHKSUM			
SRC IP: 40.0.0.2					
DST IP: 10.0.0.1					
OPT: 0x0		0x0			
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x0		CODE: 0x0	CHECKSUM
ID: 0x7		SEQ NUMBER: 6	

PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

HDLC

0	8	16	32	32+x	48+x 56+ Bits
FLG: 0111	ADR: 0x8f	CONTROL: 0x0	DATA: (VARIABLE LENGTH)		FCS: 0x0
1110					FLG: 0111

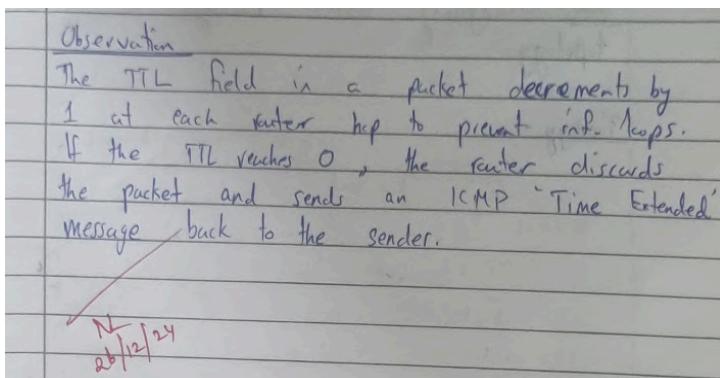
IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x6		0x0	0x0		
TTL: 253	PRO: 0x1	CHKSUM			
SRC IP: 10.0.0.1					
DST IP: 40.0.0.2					
OPT: 0x0		0x0			
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x8		CODE: 0x0	CHECKSUM
ID: 0x7		SEQ NUMBER: 6	

Observation:

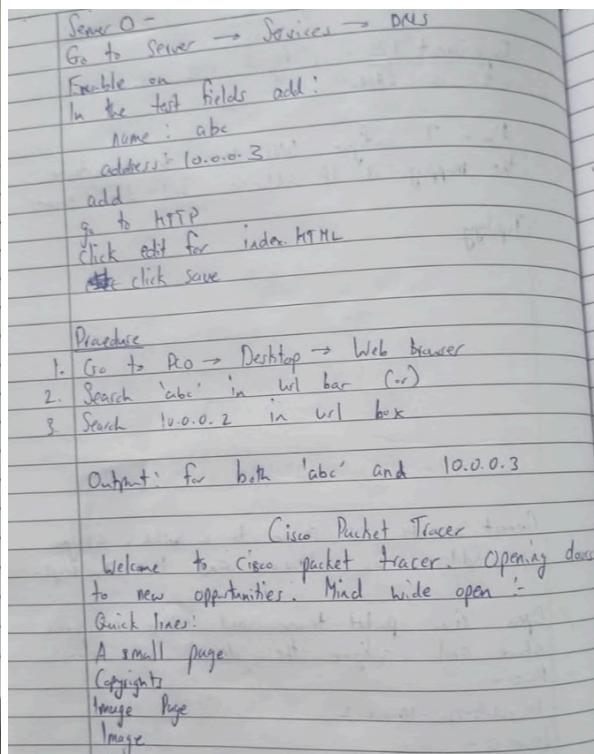
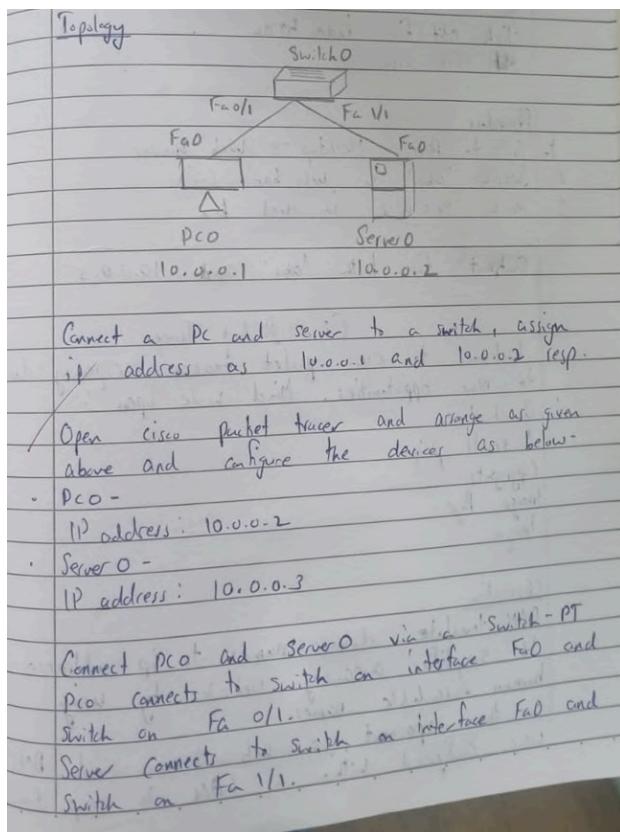


Program 8

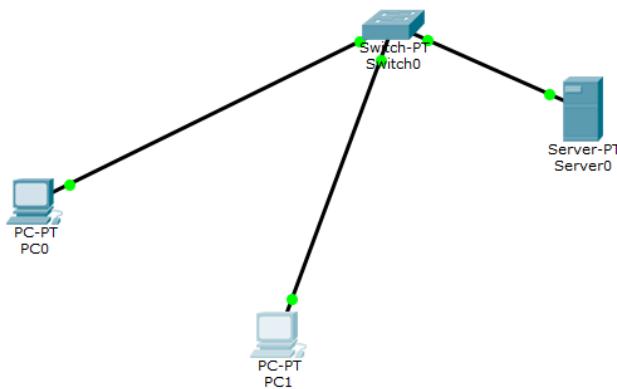
Aim of the program:

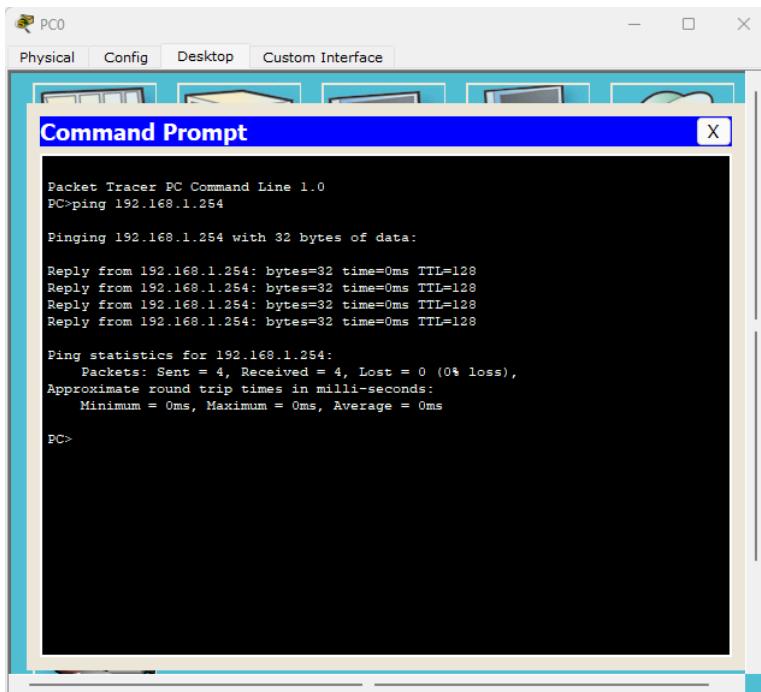
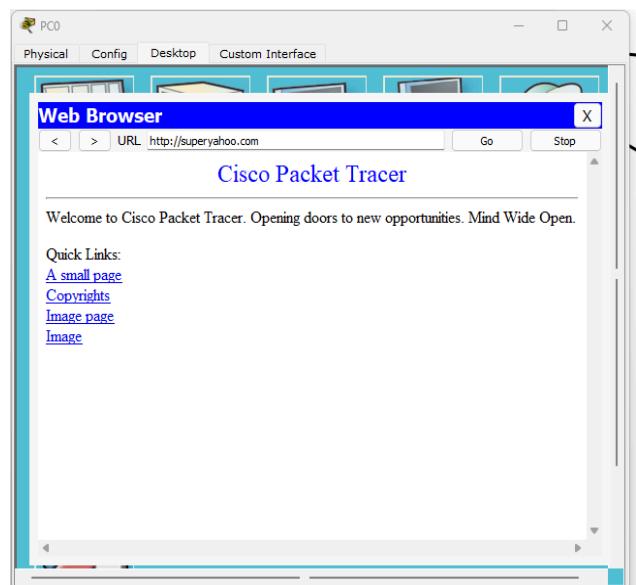
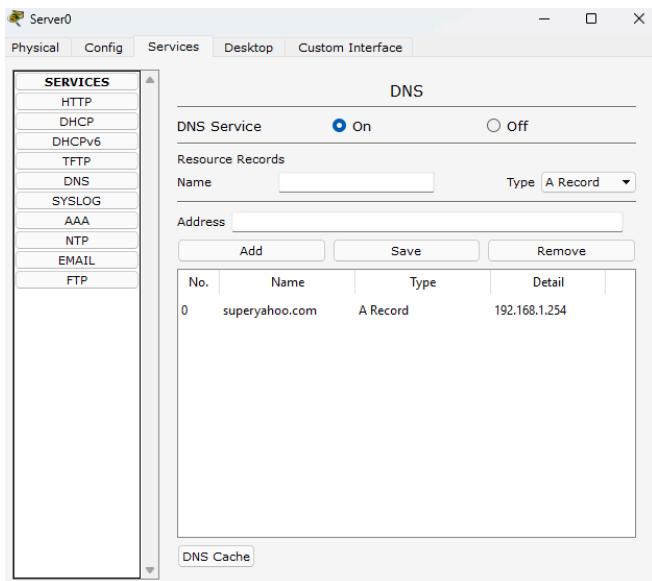
Configure Web Server, DNS within a LAN.

Procedure along with the topology:



Screen shots/ output:





Observation:

Observation

DNS translates domain names to ip addresses. It simplifies accessing websites by using human-readable names.

In this experiment, a web server and DNS were configured within a LAN to map domain names to ip addresses. The PC successfully accessed the server by both its ip addresses and the configured domain name 'abc'. The configuration was successful allowing the web page to be accessed via both methods.

Date / /
Page _____

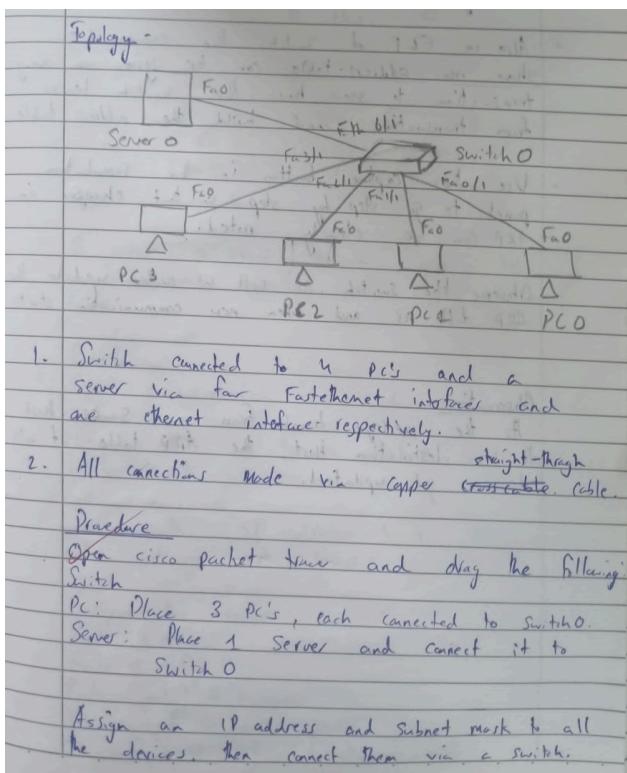
N
26/12/24

Program 9

Aim of the program:

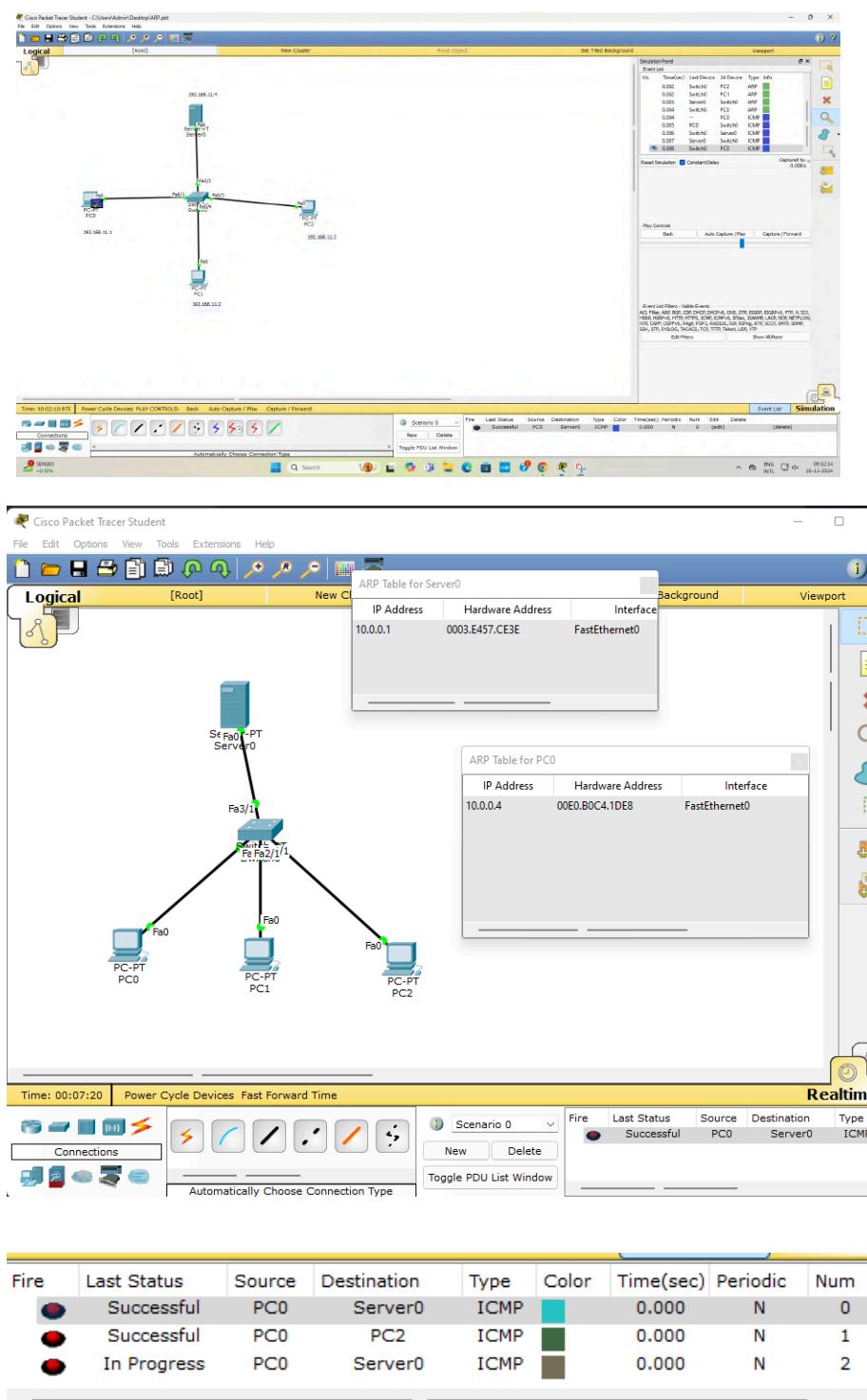
To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

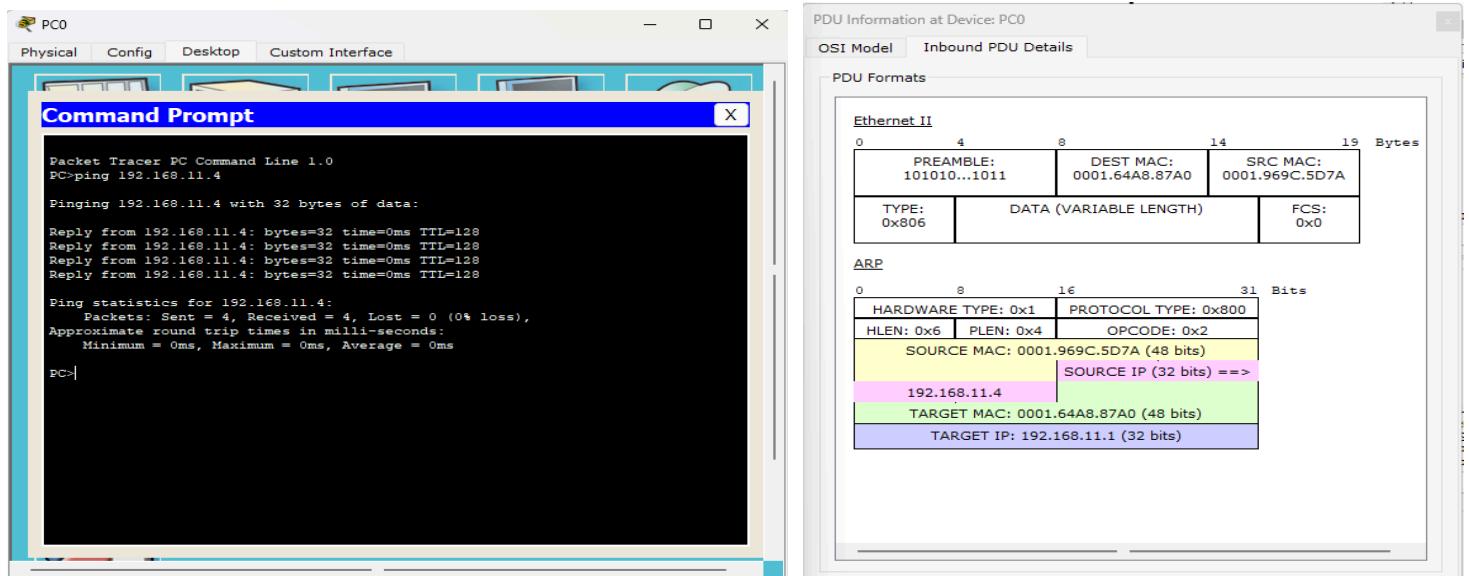
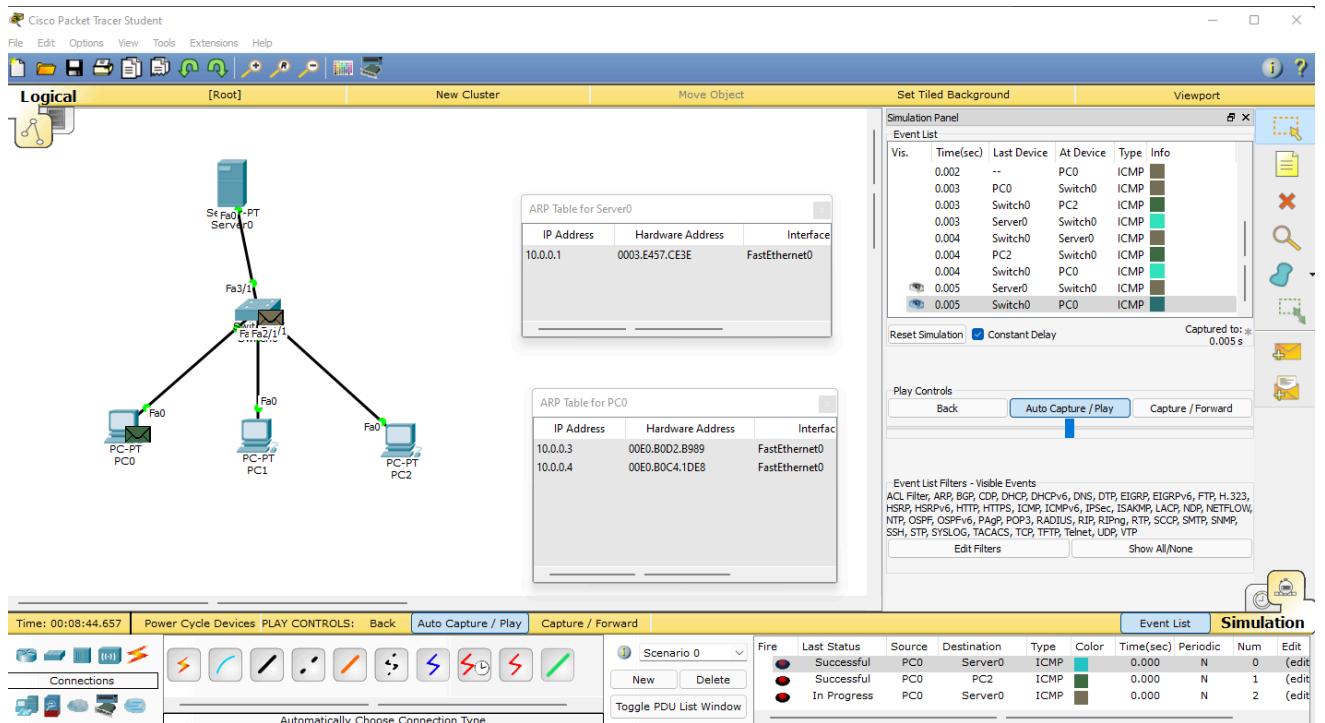
Procedure along with the topology:



- Use the inspect tool to click on a PC to view ARP table.
- Select Display the ARP table of all the devices.
- Initially ARP is empty for all.
- Also in CLI of switch, the command `show mac address-table` can be given in any transaction to see how the switch learns from transactions and build the address table.
- Use the capture button in the simulation panel to go step by step so that changes in ARP can be clearly noted.
- Observe the switch as well as nodes update the ARP table as and when new communication starts.

Screen shots/ output:





Observation:

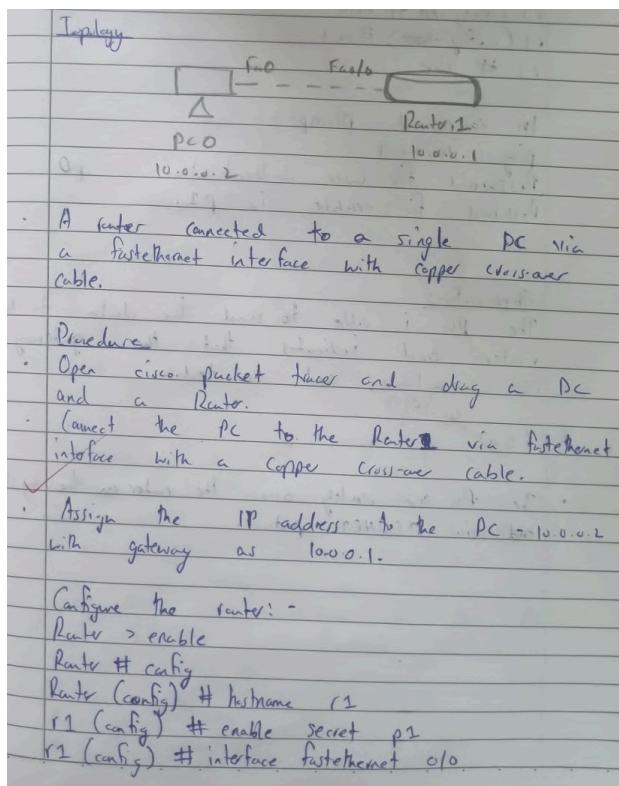
Observation	
	<ul style="list-style-type: none">As the message travels from one source host to its destination host, the ARP table of all devices get updated.

Program 10

Aim of the program:

To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.

Procedure along with the topology:



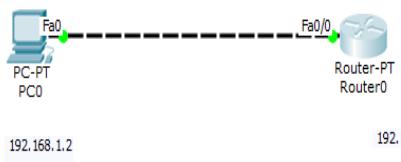
```

r1 (config-if) # ip address 192.168.1.1 255.0.0.0
r1 (config-if) # no shutdown
r1 (config-if) # line vty 0 5
r1 (config-if) # login
r1 (config-line) # password p0
r1 (config-line) # exit
r1 (config) # exit
r1 # wr
  
```

In command prompt -

ping 192.168.1.1
Password for user authentication is p0
Password for enable is p1.

Screen shots/ output:



Router0 Physical Config CLI

IOS Command Line Interface

```
Router(config)#ip address 192.168.1.1 255.255.255.0
Router(config)#exit
Router(config)#hostname R1
R1(config)#enable secret rp
R1(config)#int g0/0/0
R1(config)#line vty 0 5
R1(config-line)#login
# Login disabled on line 132, until 'password' is set
# Login disabled on line 133, until 'password' is set
# Login disabled on line 134, until 'password' is set
# Login disabled on line 135, until 'password' is set
# Login disabled on line 136, until 'password' is set
# Login disabled on line 137, until 'password' is set
R1(config-line)#password tp
R1(config-line)#exit
R1(config)#
R1(config)#exit
R1#
SYS-5-CONFIG_I: Configured from console by console

R1#wr
Building configuration...
[OK]
R1#
R1#
```

Copy Paste

PC0 Physical Config Desktop Custom Interface

Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=2ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open

User Access Verification

Password:
R1>enable
R1>enable
Password:
R1#
```

Observation:

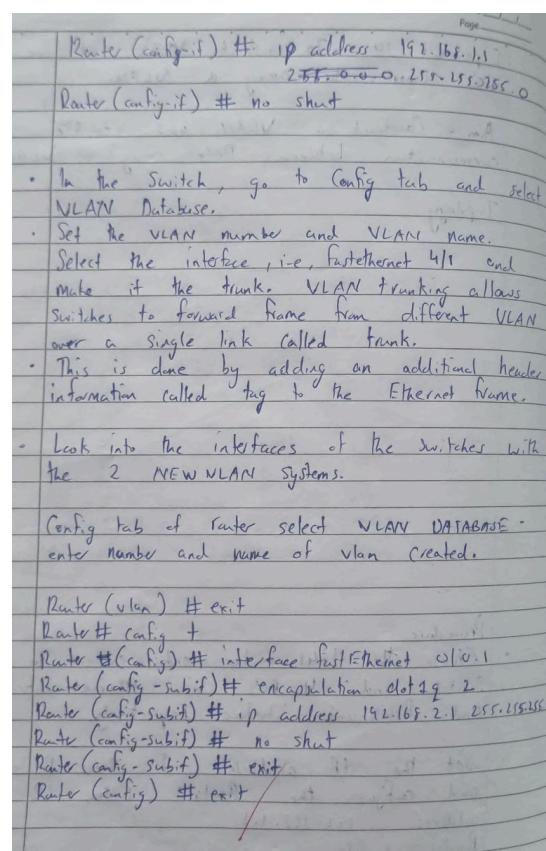
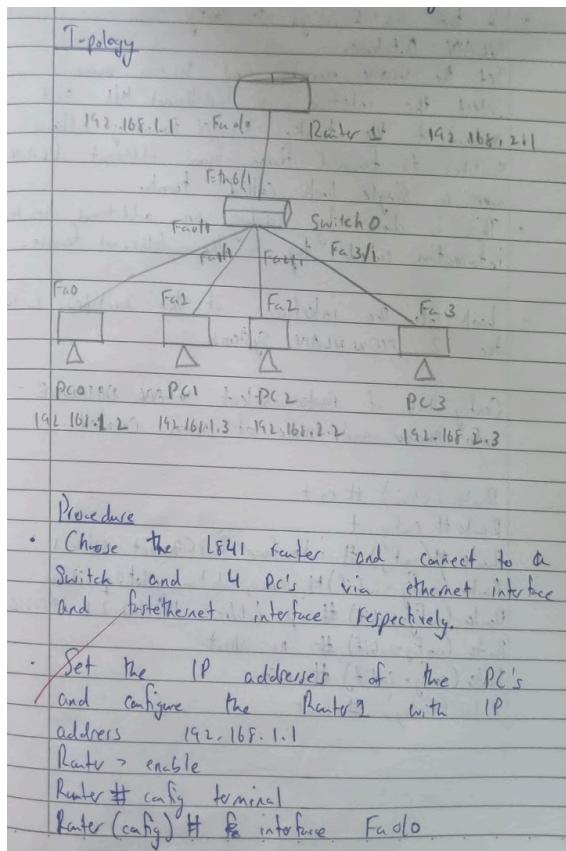
<u>Observations</u>	
	The pc is able to send the data to the router and indicates that the gateway is available, and connected.
• The PC can remotely access the router on the system via TELNET protocol.	✓

Program 11

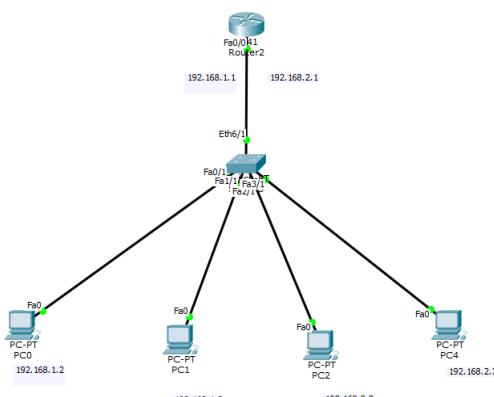
Aim of the program:

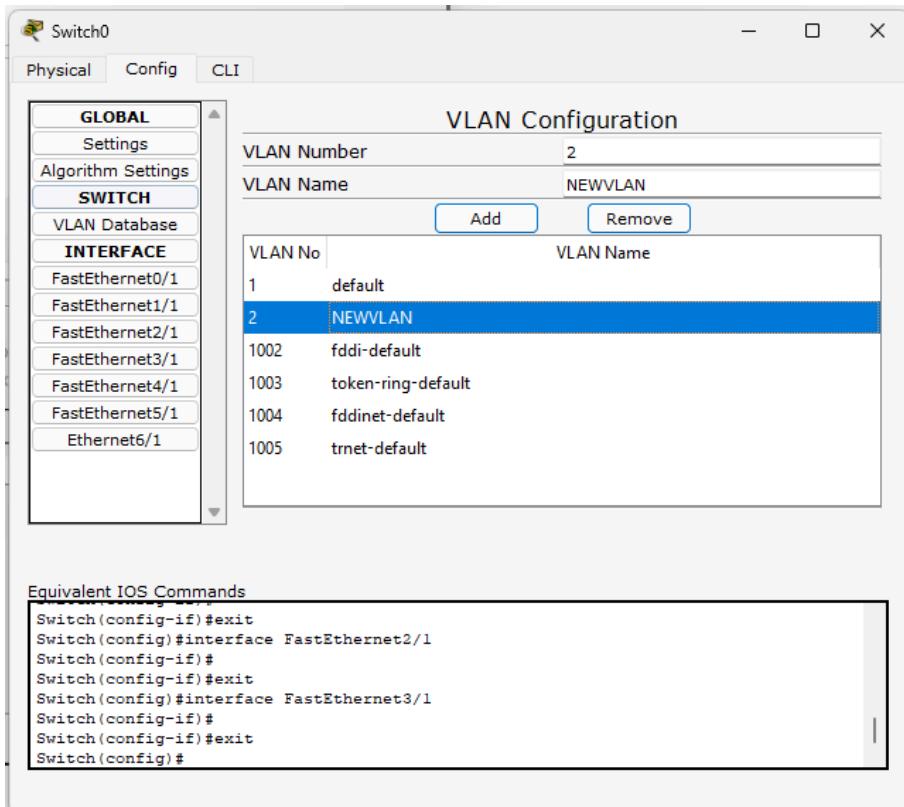
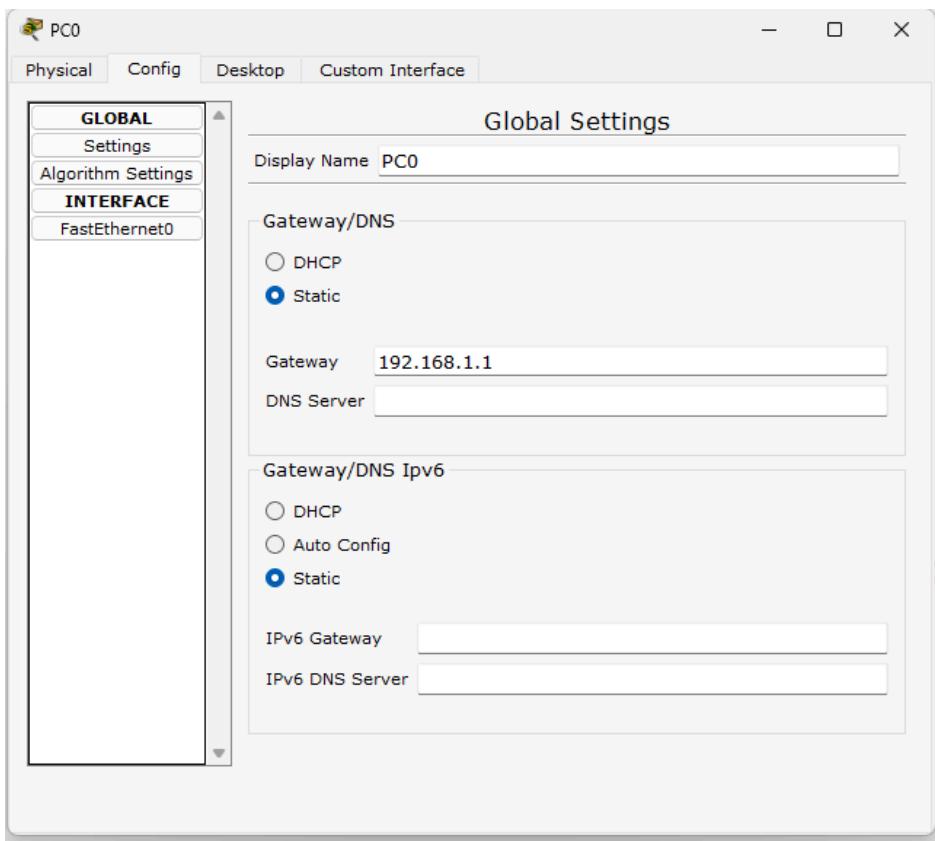
To construct a VLAN and make the PC's communicate among a VLAN

Procedure along with the topology:



Screen shots/ output:





```
Usage: ping [-n count | -v Ios | -t ] target
PC>ping 192.168.2.3
Pinging 192.168.2.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.3:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 192.168.2.3
Pinging 192.168.2.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.2.3: bytes=32 time=5ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.3:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 5ms, Average = 1ms
PC>
```

Observation:

Page _____

Observation

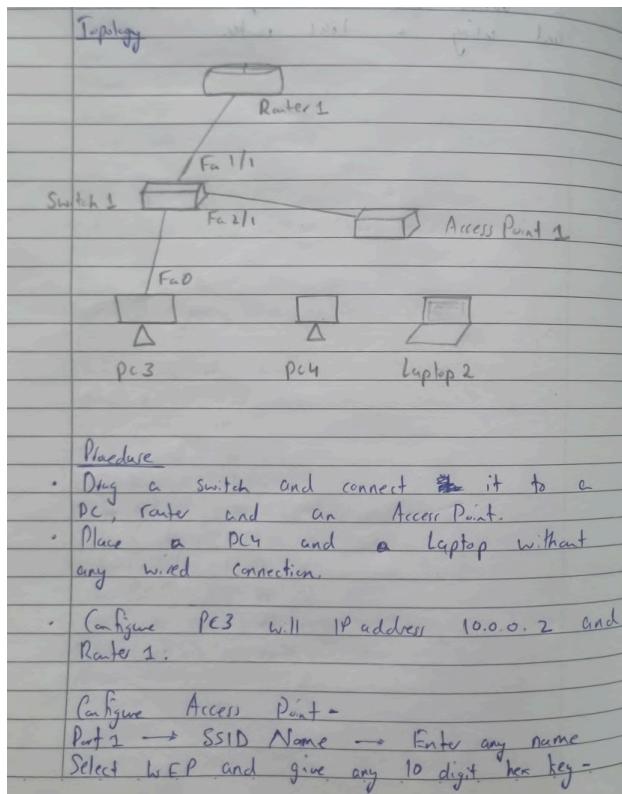
- On pinging over the VLAN, the PC's are able to communicate. Over the
- The physical LAN has been divided into a VLAN with the help of VLAN database and using a 1841 router.

Program 12

Aim of the program:

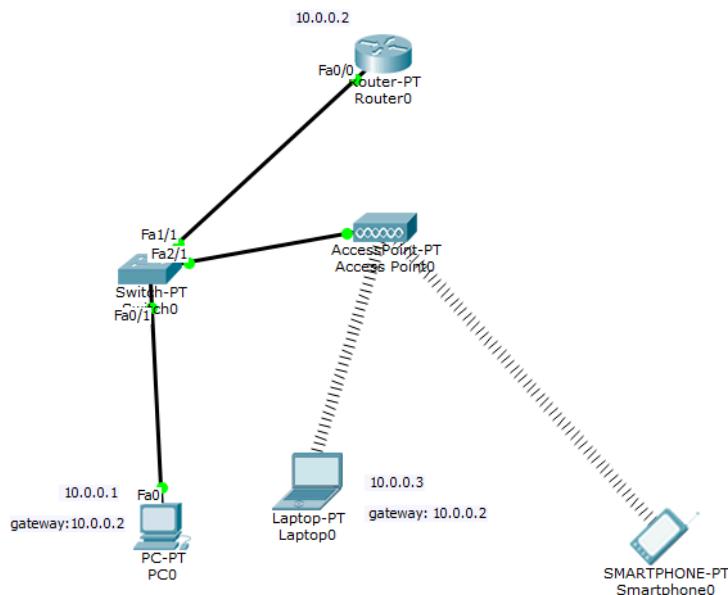
To construct a WLAN and make the nodes communicate wirelessly

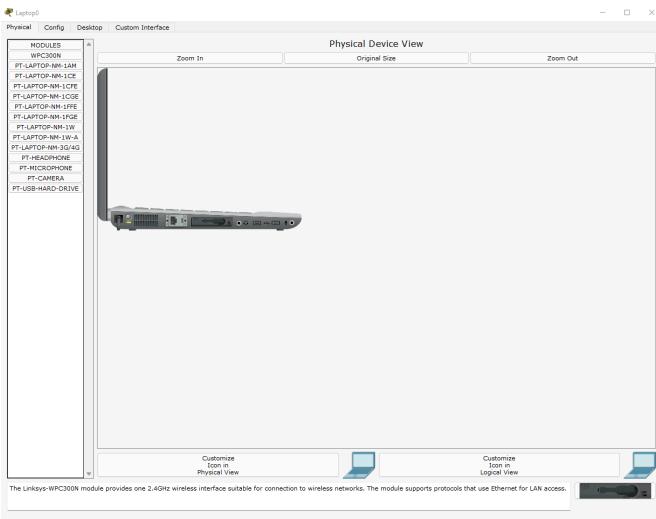
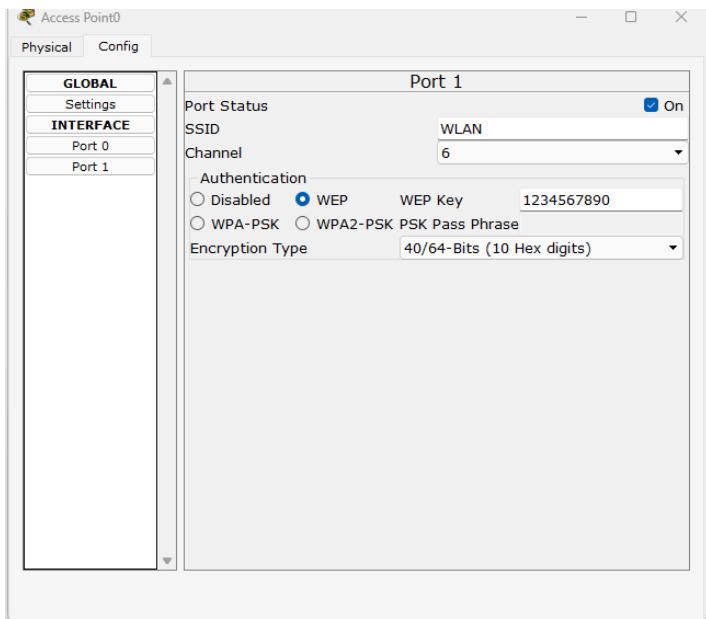
Procedure along with the topology:

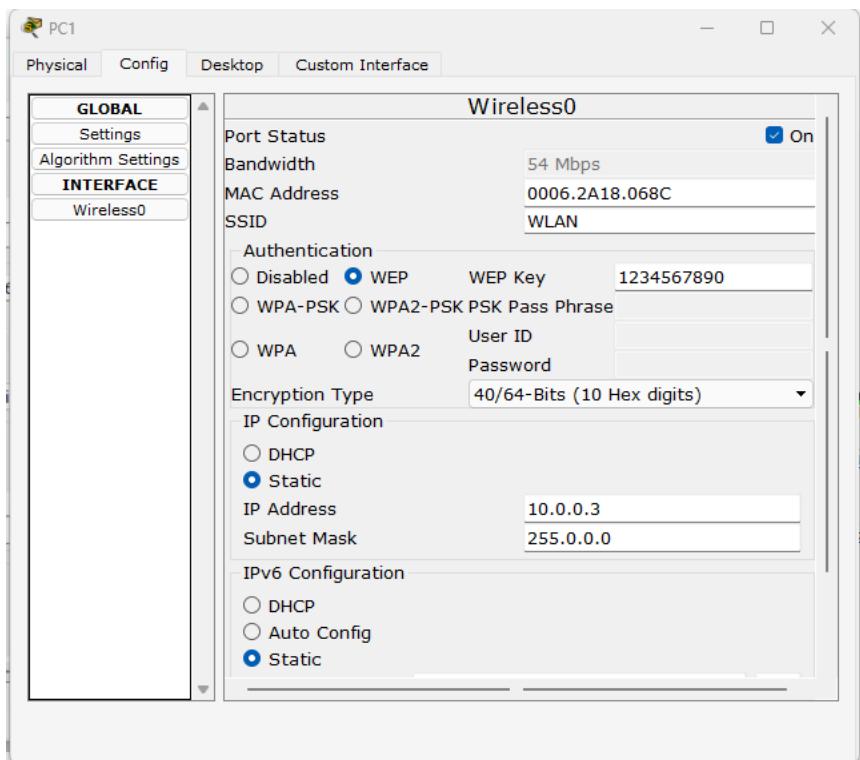


- 1234567890.
- Configure PC4 and Laptop with Wireless standards:
 - Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHI. Drag WMP300N wireless interface to the empty port. Switch On the device.
 - In the config tab, a new wireless interface would have been added. Now, configure SSID, WEP, WEP key, IP address and Gateway to the device.
 - Ding every device to every other device and see the results.

Screen shots/ output :







The screenshot shows the Command Prompt window on PC0. The user has run ping commands to test connectivity between PC1 and PC0.

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=20ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=8ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 20ms, Average = 10ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=18ms TTL=128
Reply from 10.0.0.4: bytes=32 time=10ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms TTL=128
Reply from 10.0.0.4: bytes=32 time=11ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 7ms, Maximum = 18ms, Average = 11ms
```

Observation:

<u>Observations</u>
WLAN enables wireless network communication. It uses radio waves for connectivity. WLAN connects devices wirelessly within a local area with the help of WEP for encryption and SSID for respective identification.

CYCLE-2

Program 1

Aim of the program:

Write a program for error detecting code using CRC-CCITT (16-bits).

Observation Book:

```

Sd. def calculate_crc(data: bytes, polynomial:
    int = 0x1021, initial_value: int = 0xFFFF):
    crc = initial_value
    for byte in data:
        crc = byte << 8
        for i in range(8):
            if crc & 0x8000:
                crc = (crc << 1) ^ polynomial
            else:
                crc <<= 1
    crc &= 0xFFFF
    return crc.

def append_crc(data: bytes) -> bytes:
    crc = calculate_crc(data)
    return data + crc.to_bytes(2, byteorder='big')

def verify_crc(data_with_crc: bytes) -> bool:
    if len(data_with_crc) < 2:
        return False
    data = data_with_crc[2:]
    received = int.from_bytes(data[-2:], byteorder='big')
    calculated = calculate_crc(data)
    return received == calculated
  
```

```

message = b"Hello, CRC!"
data = append_crc(message)
print(f"Data with CRC: {data.hex()}")
is_valid = verify_crc(data)
print(f"Is the CRC valid? {is_valid}")

corrupted = data[:-1] + bytes([data[-1]^0xFF])
print(f"Corrupted data: {corrupted.hex()}")
is_valid = verify_crc(corrupted)
print(f"Is the CRC valid for corrupted data? {is_valid}")

Output 333102
Ans 20/12/2024
  
```

Code:

```

#include<stdio.h>
#include<string.h>
#define N strlen(gen_poly)

char data[28], check_value[28], gen_poly[10];
int data_length,i,j;

void XOR(){
    for(j = 1;j < N; j++)
        check_value[j] = ((check_value[j] == gen_poly[j])?'0':'1');
}
  
```

```

void receiver(){

    printf("Enter the received data: ");
    scanf("%s", data);
    printf("\n-----\n");
    printf("Data received: %s", data);

    crc();

    for(i=0;(i<N-1) && (check_value[i]!='1');i++){
        if(i<N-1)
            printf("\nError detected\n\n");
        else
            printf("\nNo error detected\n\n");
    }

}

void crc(){

    for(i=0;i<N;i++)
        check_value[i]=data[i];
    do{
        if(check_value[0]=='1')
            XOR();
        for(j=0;j<N-1;j++)
            check_value[j]=check_value[j+1];
        check_value[j]=data[i++];
    }while(i<=data_length+N-1);
}

int main()
{
    printf("\nEnter data to be transmitted: ");
    scanf("%s",data);
    printf("\nEnter the Generating polynomial: ");
    scanf("%s",gen_poly);
    data_length=strlen(data);
    for(i=data_length;i<data_length+N-1;i++)
        data[i]='0';
    printf("\n-----");
    printf("\n Data padded with n-1 zeros : %s",data);
    printf("\n-----");
    crc();
    printf("\nCRC or Check value is : %s",check_value);
    for(i=data_length;i<data_length+N-1;i++)
        data[i]=check_value[i-data_length];
    printf("\n-----");
    printf("\n Final data to be sent : %s",data);
    printf("\n-----\n");
    receiver();
    return 0;
}

```

OUTPUT:

```
2.c -o CRC2 && "/Users/avyukthinna/Documents/CN programs/"CRC2  
Enter Data: 1001  
Enter generator: 101  
Data with zeroes: 100100  
CRC: 11  
Data sent: 100111  
Data received:100110  
Error detected!%
```

Program 2

Aim of the program:

Write a program for congestion control using Leaky bucket algorithm.

Observation Book:

```

Sol. #include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define NOF_PACKETS 5

int rand(int a) {
    int rn = (random() % 10) % a;
    return rn == 0 ? 1 : rn;
}

int Main() {
    int packet_sz[NOF_PACKETS], i, clk, b-size,
        o-rate, p-sz-m=0, p-sz, p-time, op;
    for (i=0, i< NOF_PACKETS; i++) {
        packet_sz[i] = random() % 100;
    }
    for (i=0; i< NOF_PACKETS; i++) {
        printf("In packet[%d]: l.d bytes (%d)", i, packet_sz[i]);
        printf(" Enter Output rate");
        scanf("%f", &o-rate);
        printf("%f", o-rate);
        for (i=0; i< NOF_PACKETS; i++) {
            if (packet_sz[i] + p-sz-m > b-size)
                if (packet_sz[i] > b-size)
                    printf(" Incoming packet size (%.2f bytes) is
                           greater than bucket capacity (%.2f bytes),
                           PACKET REJECTED", packet_sz[i],
                           b-size);
            else
                printf(" In Bucket (capacity exceeded!)");
        }
    }
}

```

```

else {
    p-sz-m += packet_sz[i];
    printf("Size", packet_sz[i]);
    printf(" Bytes remaining to Transmit:", p-sz-m);
    while (p-sz-m > 0) {
        sleep(1);
        if (p-sz-m < o-rate)
            op = p-sz-m, p-sz-m = 0;
        else
            op = o-rate, p-sz-m -= o-rate;
        printf(op);
        printf(" Bytes remaining:", p-sz-m);
    }
    else {
        printf("No packets to transmit!");
    }
}

```

Code:

```
# initial packets in the bucket  
storage = 0
```

```
# total no. of times bucket content is checked  
no_of_queries = 4
```

```
# total no. of packets that can  
# be accommodated in the bucket  
bucket_size = 10
```

```
# no. of packets that enters the bucket at a time  
input_pkt_size = 4
```

```
# no. of packets that exits the bucket at a time
```

```
output_pkt_size = 1
for i in range(0, no_of_queries): # space left

    size_left = bucket_size - storage
    if input_pkt_size <= size_left:
        # update storage
        storage += input_pkt_size
    else:
        print("Packet loss = ", input_pkt_size)

print(f"Buffer size= {storage} out of bucket size = {bucket_size}")

# as packets are sent out into the network, the size of the storage decreases
storage -= output_pkt_size
```

OUTPUT:

```
Buffer size = 4 out of capacity 10
Buffer size = 7 out of capacity 10
Buffer size = 10 out of capacity 10
Packet lost: 4
Buffer size = 9 out of capacity 10
(base) avyukthinnna@Avyukths-MBP CN programs %
```

Program 3

Aim of the program:

Using TCP/IP sockets, write a client-server program to make the client sending the file name and the server to send back the contents of the requested file if present.

Observation Book:

Sol.	<pre>CLIENT-TCP.py from socket import * serverName = '127.0.0.1' serverPort = 12000 clientSocket = socket(AF_INET, SOCK_STREAM) clientSocket.connect((serverName, serverPort)) sentence = input("\n Enter file name:") clientSocket.send(sentence.encode()) filecontents = clientSocket.recv(1024).decode() print("\n From Server:\n") print(filecontents) clientSocket.close() SERVER-TCP.py from socket import * serverName = "127.0.0.1" serverPort = 12000 serverSocket = socket(AF_INET, SOCK_STREAM) serverSocket.bind((serverName, serverPort)) serverSocket.listen(1) while 1: print("Server is ready to receive") connectionSocket, addr = serverSocket.accept() sentence = connectionSocket.recv(1024).decode() file = open(sentence, 'r') l = file.read(1024) file.close() connectionSocket.send(l.encode()) print("Sent contents of " + sentence) connectionSocket.close()</pre>
------	---

Code:

```
ClientTCP.py
from socket import *
```

```
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("\nFrom Server:\n")
print(filecontents)
clientSocket.close()
```

```
ServerTCP.py
from socket import *
```

```
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```

```
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")

    l = file.read(1024)
    connectionSocket.send(l.encode())
    print("\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

OUTPUT:

The screenshot shows the VS Code interface with the following details:

- EXPLORER** view: Shows files in the CN LAB folder: ClientTCP.py, ServerTCP.py, VLAN.pkt, WLAN.pkt.
- ServerTCP.py** editor tab: Displays the Python code for a TCP server. The code creates a socket, binds it to port 12000, and listens for incoming connections. It then reads data from the connection, decodes it, and writes it to a file named 'sentence'.
- TERMINAL** tab: Shows the command PS C:\Users\dell\Desktop\CN_LAB & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ServerTCP.py being run. The output indicates that the server is ready to receive data.
- OUTPUT** tab: Shows the contents of the 'sentence' file as sent by the client.

The screenshot shows the VS Code interface with the following details:

- EXPLORER** view: Shows files in the CN LAB folder: ClientTCP.py, ServerTCP.py, ServerTCP.png, VLAN.pkt, WLAN.pkt.
- ClientTCP.py** editor tab: Displays the Python code for a TCP client. It connects to the server at 127.0.0.1 port 12000, sends a message, and receives a response from the server.
- TERMINAL** tab: Shows the command PS C:\Users\dell\Desktop\CN_LAB & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ClientTCP.py being run. The output prompts the user to enter a file name and then displays the received message from the server.
- OUTPUT** tab: Shows the contents of the 'sentence' file as received from the server.

Program 4

Aim of the program:

Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

Observation Book:

Sol. CLIENT- UDP. py

```
from socket import *
Server Name = "127.0.0.1"
server Port = 12000
client Socket = socket (AF_INET, SOCK_DGRAM)
sentence = input ("In Enter file name:")
client Socket.sendto (bytes(sentence, "utf-8"), (Server Name, server Port))
filecontents, server Address = client Socket.recvfrom (2048)
print ("Reply - from Server")
print (filecontents.decode ('utf-8'))
client Socket.close()
client Socket.close()

SERVER- UDP. py
from socket import *
```

Date _____
Page _____

```
Server Port = 12000
server Socket = socket (AF_INET, SOCK_DGRAM)
server Socket.bind ((127.0.0.1, Server Port))
print ("The server is ready to receive")
while 1:
    sentence, client Address = server Socket.recvfrom (2048)
    sentence = sentence.decode ('utf-8')
    file = open (sentence, "r")
    con = file.read (2048)
    server Socket.sendto (bytes(con, 'utf-8'), (client Address))
    print ('In Sent contents of ', end = '')
    print (sentence)
    file.close()
```

Code:

ClientUDP.py

```
from socket import *
```

```
serverName = "127.0.0.1"
```

```
serverPort = 12000
```

```
clientSocket = socket(AF_INET, SOCK_DGRAM)
```

```
sentence = input("\nEnter file name: ")
```

```
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))
```

```
filecontents, serverAddress = clientSocket.recvfrom(2048)
```

```
print("\nReply from Server:\n")
```

```
print(filecontents.decode("utf-8"))
```

```
clientSocket.close()
```

ServerUDP.py

```
from socket import *
```

```
serverPort = 12000
```

```
serverSocket = socket(AF_INET, SOCK_DGRAM)
```

```
serverSocket.bind(("127.0.0.1", serverPort))
```

```
print("The server is ready to receive")
```

```
while 1:
```

```
    sentence, clientAddress = serverSocket.recvfrom(2048)
```

```
    sentence = sentence.decode("utf-8")
```

```
    file = open(sentence, "r")
```

```
    con = file.read(2048)
```

```
    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)
```

```
    print("\nSent contents of ", end=' ')
```

```
    print(sentence)
```

```
    file.close()
```

OUTPUT:

The image shows two terminal windows side-by-side. Both windows have tabs for ClientUDP.py, ServerUDP.py, and ClientTCP.py. The left window (CN_LAB) is titled 'Python.ClientUDP' and shows the output of running ClientUDP.py. The right window (CH_LAB) is titled 'Python.ServerUDP' and shows the output of running ServerUDP.py.

Terminal CN_LAB (Python.ClientUDP):

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ClientUDP.py
Enter file name:ServerUDP.py
Reply from Server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('127.0.0.1', serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
    print ('Sent contents of', end = '')
    print (sentence)
    # For i in sentence:
    #     # print (str(i), end = " ")
    file.close()
```

Terminal CH_LAB (Python.ServerUDP):

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ServerUDP.py
The server is ready to receive
Sent contents ofServerUDP.py
```

Took Exploration - Wireshark

Wireshark - Tool Exploration	
Wireshark is a powerful and widely used network protocol analyzer. It allows you to capture and inspect data packets travelling over a network in real-time, making it a crucial tool for studying computer networks, troubleshooting network issues and understanding.	
→ Key Features	
• Packet Capture:	Captures live network traffic from various interfaces such as ethernet, wifi etc.
• Protocol Analysis:	Supports hundreds of protocols such as TCP, UDP, HTTP, etc.
• Filtering:	Offers powerful filters to isolate specific packets or traffic types.
• Visualization:	Displays packet details with hierarchical layers.
→ Use Cases	
• Network Troubleshooting -	
↳ Diagnosing slow network speeds.	
↳ Identifying bottlenecks or misconfigurations.	
• Security Analysis -	
↳ Detecting malicious traffic or intrusions.	
• Protocol Study -	
↳ Understanding packet structures and communication flows.	

Common Filters -	
HTTP:	Show only HTTP traffic
tcp port == 80:	Show traffic on TCP port 80
ip addr == 192.168.1.1:	Show packets to or from a specific IP address.
udp:	Show only UDP traffic.