

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

## COMPUTER NETWORKS (23CS5PCCON)

*Submitted by*

**SAKSHI KALLUR (1BM22CS345)**

*in partial fulfillment 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**

**Sep 2024-Jan 2025**

**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 NETWORKS**" carried out by **SAKSHI KALLUR (1BM22CS345)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Computer Networks Lab - (23CS5PCCON)** work prescribed for the said degree.

**Dr. Nandhini Vineeth**

Associate Professor,  
Department of CSE,  
BMSCE, Bengaluru

**Dr. Kavitha Sooda**

Professor and Head,  
Department of CSE  
BMSCE, Bengaluru

# INDEX

## CYCLE 1

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	25-9-24	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message	1-4
2	9-10-24	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	5-11
3	23-10-24	Configure default route, static route to the Router	12-15
4	13-11-24	Configure DHCP within a LAN and outside LAN	16-20
5	21-11-24	Configure RIP routing Protocol in Routers	21-24
6	20-11-24	Demonstrate the TTL/ Life of a Packet	25-28
7	27-11-24	Configure OSPF routing protocol	29-34
8	18-12-24	Configure Web Server, DNS within a LAN	35-38
9	18-12-24	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	39-42
10	18-12-24	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	43-46
11	18-12-24	To construct a VLAN and make the PC's communicate among a VLAN	47-51
12	18-12-24	To construct a WLAN and make the nodes communicate wirelessly	52-55

# **INDEX**

## CYCLE 2

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	1-1-25	Write a program for error detecting code using CRC-CCITT (16-bits)	56-59
2	1-1-25	Write a program for congestion control using Leaky bucket algorithm.	60-61
3	1-1-25	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.	62-64
4	1-1-25	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.	65-67
5	3-1-25	Tool exploration - Wireshark	68-69

Github Link:

<https://github.com/Sakshi-Kallur/CN-LAB.git>

# CYCLE 1

## Program 1

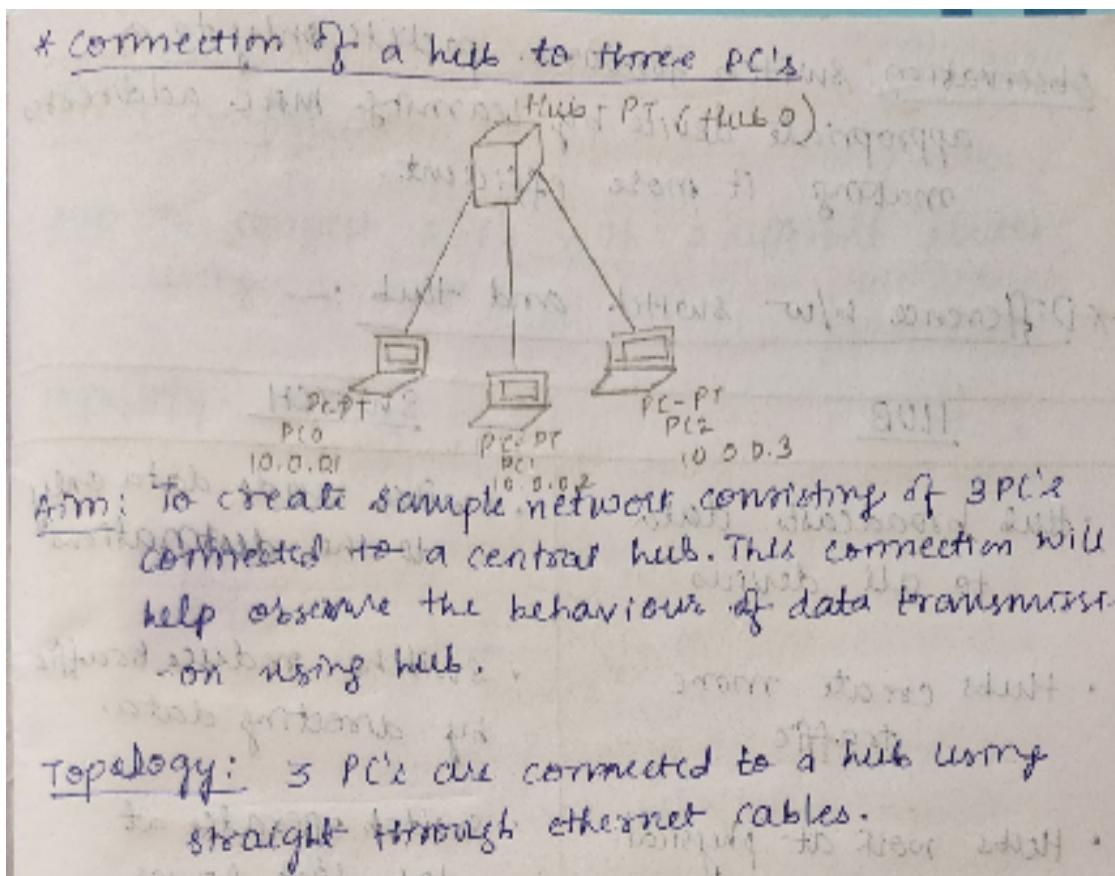
### **Question:**

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

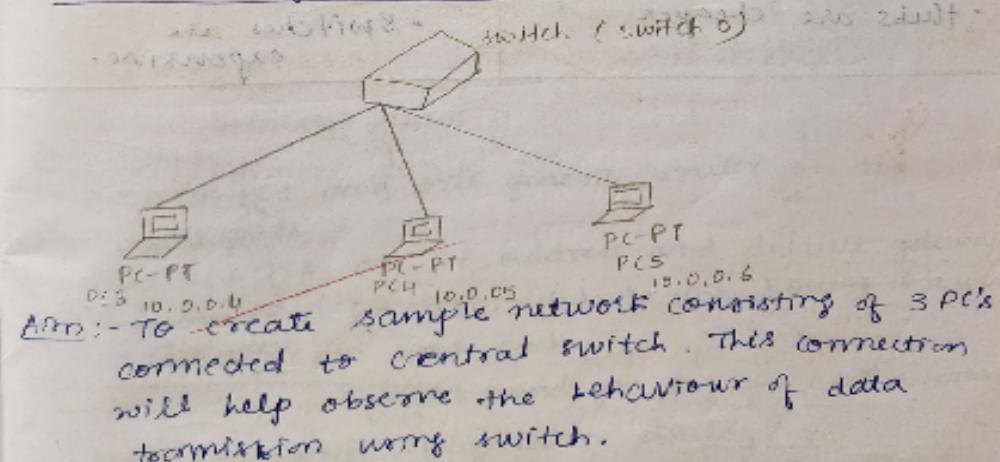
Aim: To create sample network consisting of 3 PC's connected to a central hub. This connection will help observe the behaviour of data transmission using hub.

~~- To create sample network consisting of 3 PC's connected to central switch. This connection will help observe the behaviour of data transmission using switch.~~

### **Observation writeup:**



\* Connection of switch to three PC's

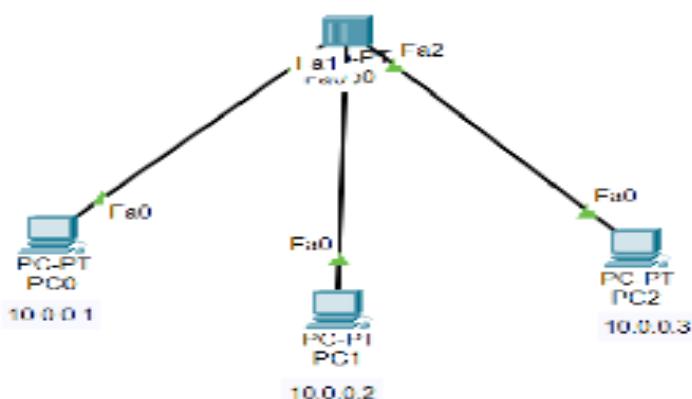


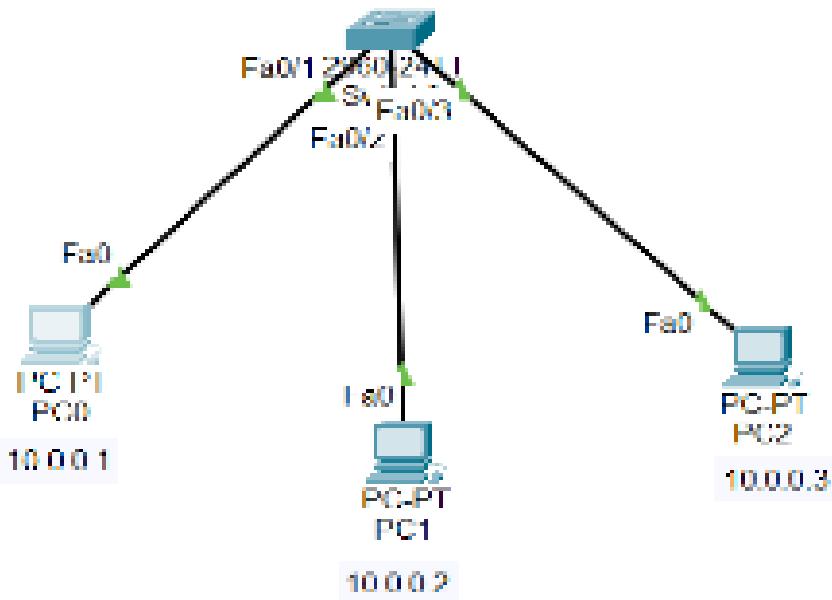
Topology: 3 PCs are connected to a switch using straight-through Ethernet cables.

Observation: Hub broadcasts packets to all devices which may cause unnecessary traffic.

Observation: Switch forwards packets only to an appropriate device by learning MAC addresses, making it more efficient.

**Topology:**



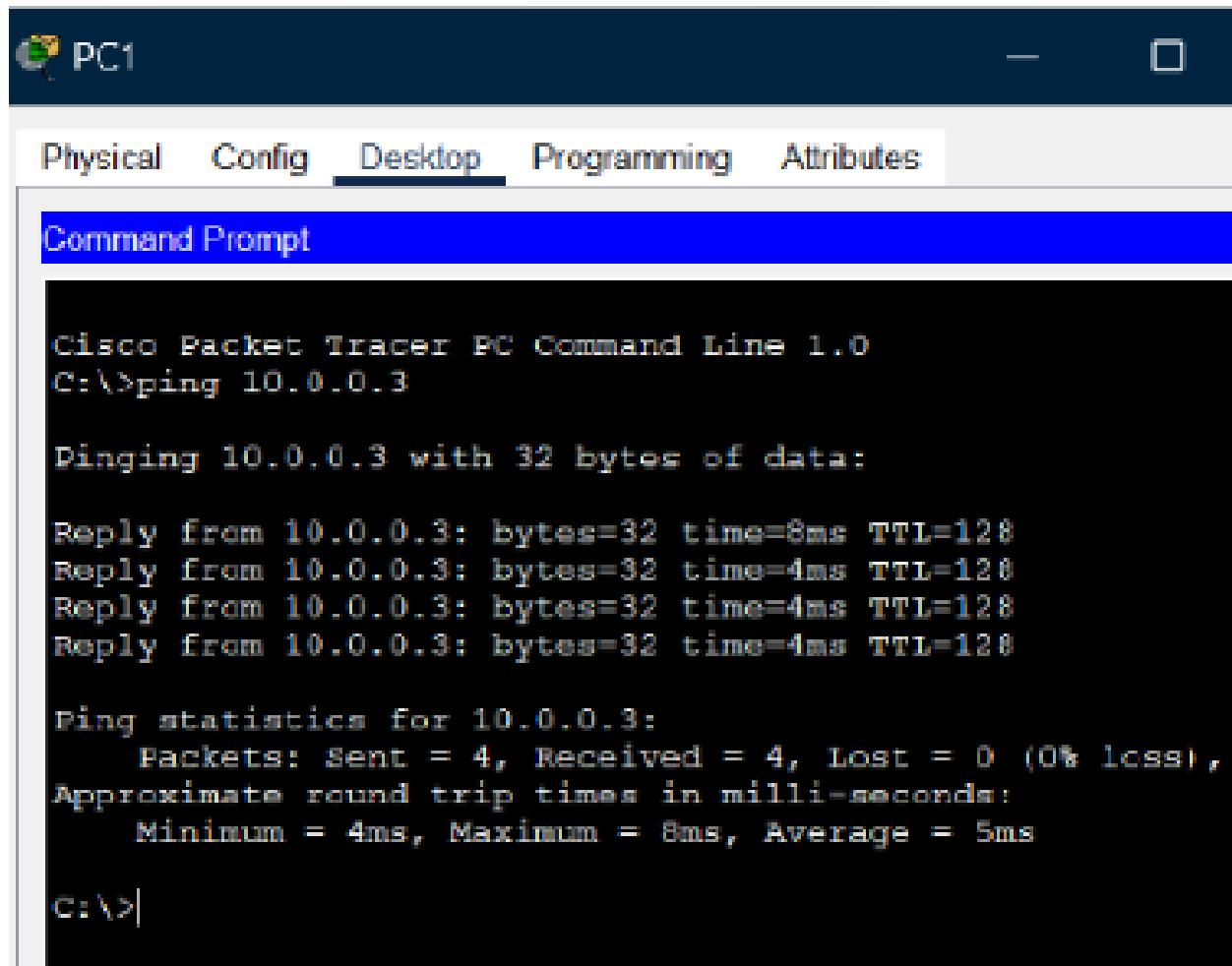


## Output:

SWITCH:

Physical	Config	<u>Router</u>	Programming	Attributes
Command Prompt				
<pre> Cisco switch# traceroute to command line 1.0 Destination: 10.0.0.1  Tracing route to 10.0.0.1 with 32 bytes of data:   Reply from 10.0.0.1: bytes=32 time=0ms ttl=128   Reply from 10.0.0.1: bytes=32 time=4ms ttl=128   Reply from 10.0.0.1: bytes=32 time=4ms ttl=128   Reply from 10.0.0.1: bytes=32 time=4ms ttl=128  Trace complete for 10.0.0.1:   Round-trip min= 4ms, max= 4ms, avg= 4ms approximate round trip times in milliseconds:   Minimum = 4ms, Maximum = 4ms, Average = 4ms  Cisco# </pre>				

HUB:



The screenshot shows a window titled "PC1" with a dark blue header bar. Below the header is a menu bar with tabs: "Physical", "Config", "Desktop" (which is underlined), "Programming", and "Attributes". A blue horizontal bar labeled "Command Prompt" spans across the window. The main area is a black terminal window displaying the following text:

```
Cisco Packet Tracer PC Command Line 1.0
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=8ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms 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 = 4ms, Maximum = 8ms, Average = 5ms

C:\>
```

## Program 2

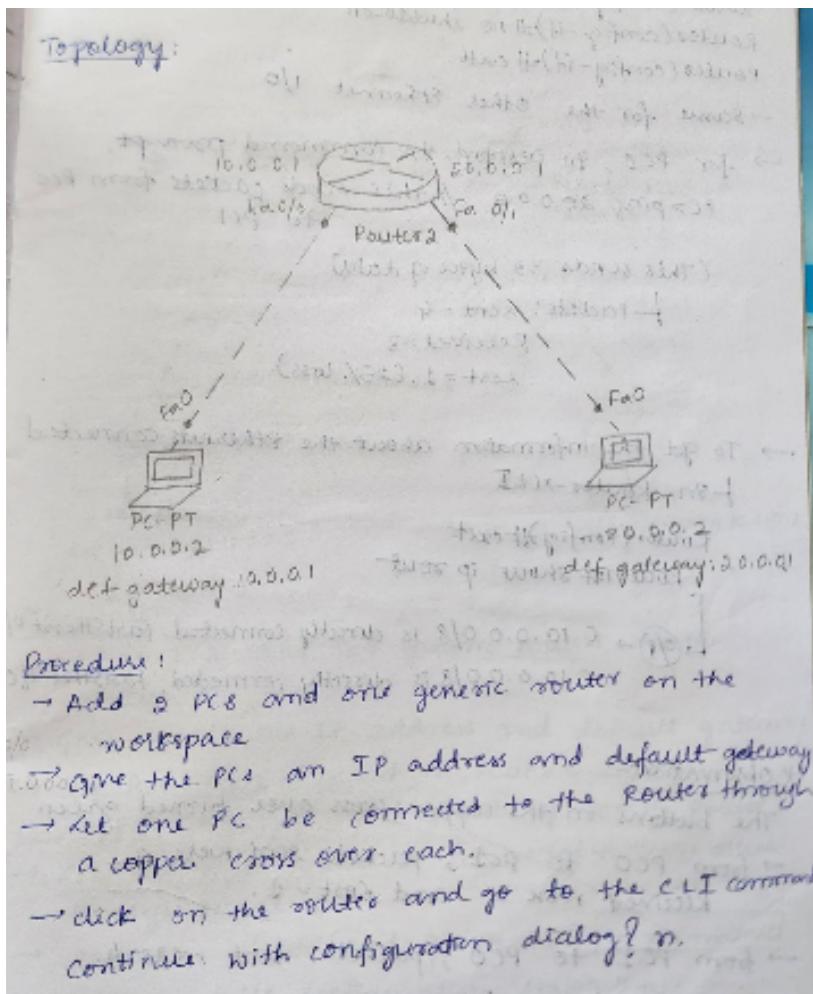
### Question:

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

~~ADD: To connect 2 PCs with 2 different subnets using a router.~~

~~ADD: To connect 2 PCs to 2 different routers and the routers connected to each other configuration of two routers.~~

### Observation writeup:



```

Router#enable
Router#config terminal
Router(config-if)#interface FastEthernet 0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit.

→ same for the other Ethernet I/O

→ for PC0, in Desktop, in command prompt,
PC> ping 20.0.0.2 // this sends packets from PC0
          to PC1
          (this sends 32 bytes of data)
  ↓ packets: send = 4
            Received = 3
            Lost = 1. (25% loss).

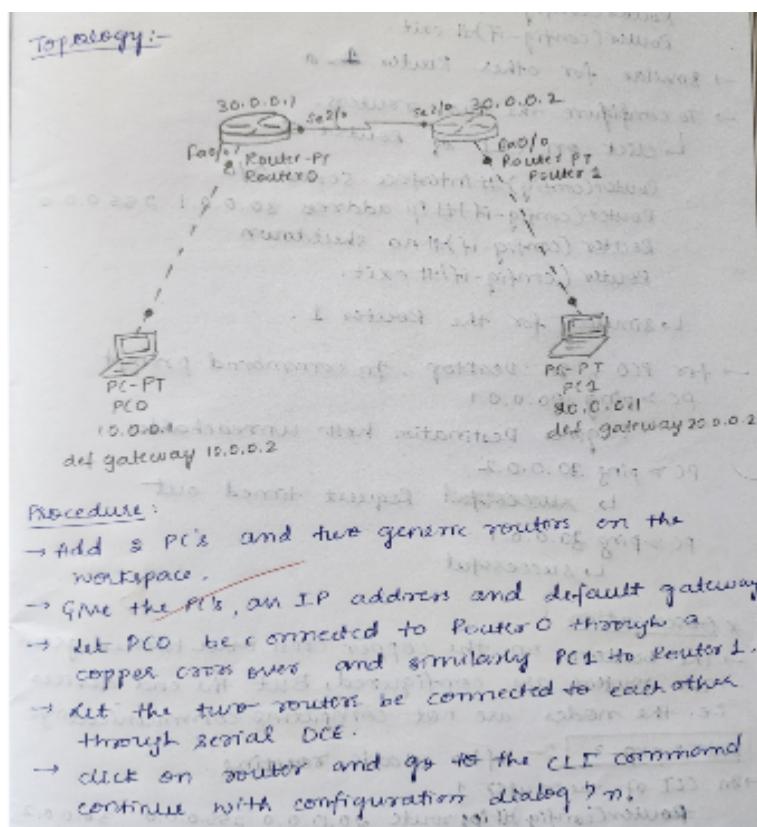
```

→ To get the information about the Ethernet connected  
from Router → CLI

```

Router(config)#exit
Router# show ip route
[①] → C 10.0.0.0/8 is directly connected, fastEthernet 0/0
      via interface 10.0.0.0/8 is directly connected, fastEthernet 0/0

```



```

Router# enable
Router# config terminal
Router(config)# interface fastethernet 0/0
Router(config-if)# ip address 10.0.0.8 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
→ similar for other Router 2 & 3
→ To configure the two routers.
  ↳ click on CLI of Router 0/-
    Router(config)# interface serial 2/0
    Router(config-if)# ip address 30.0.0.1 255.0.0.0
    Router(config-if)# no shutdown
    Router(config-if)# exit.
  ↳ similar for the Router 1.
→ for PC0, in desktop, in command prompt
  PC> ping 20.0.0.1
    ↳ gives Destination host unreachable.
  PC> ping 30.0.0.2,
    ↳ Request timed out
  PC> ping 30.0.0.1
    ↳ successful

```

10/10/2024

Aim: Configure default routes - static routes to Router. Configure static connection to router.

Topology: same as experiment-02.

Procedure:

- \* go to CLI of router0 and in CLI enter
 

```

Router#enable
Router#config terminal
Router(config)# ip route 20.0.0.0
      ↳ 255.0.0.0 10.0.0.2
      ↳ Router(config)# exit.
      ↳ Repeat the same for router0 by changing 20 to 10 and 30.0.0.2 to 30.0.0.3.
    
```

Observation: In CLI

- # show ip route
 

```

      ↳ (for router 2)
      S 10.0.0.0/8 [1/0] via 30.0.0.1
      C 30.0.0.0/8 is directly connected,
          fast ethernet 0/0
      C 30.0.0.0/8 is directly connected,
          serial 0/0/0.
    
```

Output: On pinging end devices, the message sent is lost. Other routers are as well.

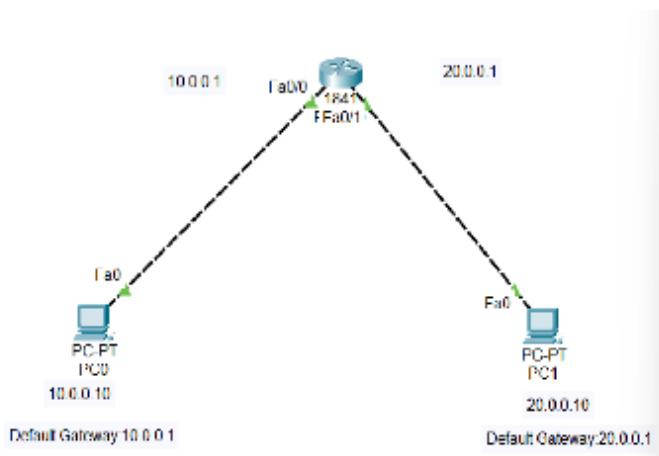
\* Observation:  
 The buttons on the copper cross over turned green  
 → from PC0 to PC1, packets sent were 4, and received were 3, and lost = 1.  
 → from PC1 to PC0, packets sent were 4, and received were 4, and lost = 0.

\* Observation:  
 → The buttons on the copper cross over turned green.  
 → The routers are configured, But the end devices i.e. the nodes are not communicating.

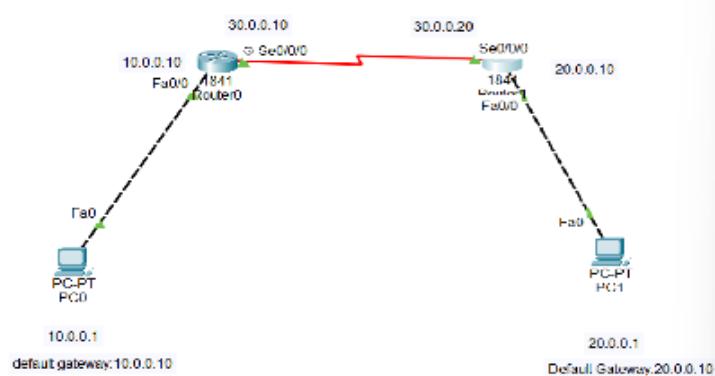
on C-LT  
 # show ip route  
 (for router 2)  
 S 10.0.0.0/8 [1/0] via 30.0.0.1  
 C 20.0.0.0/8 is directly connected  
     root ethernet 0/0  
 C 30.0.0.0/8 is directly connected  
     serial 0/0/0.

on pinging end devices, the message sent is lost.  
 after routers are as well.

Topology:



### Topology:



## Output:

```
PCO - Desktop - Programming - Attributes

Command Prompt [X]

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127

Ping statistics for 20.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 11ms, Average = 3ms

C:\>
```

### Output:(Before Static Routing):

```
Router>show ip route
Press RETURN to get started.

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 areas
      * - 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
C    80.0.0.0/8 is directly connected, Serial0/0/0
```

```
C:\>ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time<1ms TTL=255

Ping statistics for 10.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 30.0.0.20

Pinging 30.0.0.20 with 32 bytes of data:
request timed out,
request timed out,
request timed out,
request timed out,

Ping statistics for 30.0.0.20:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.

Ping statistics for 20.0.0.1:
Packets: sent = 4, received = 0, lost = 4 (100% loss),
```

(After Static Routing):

```

Router# show ip route
Gateway of last resort is not set

C 19.0.0.0/8 is directly connected, FastEthernet0/0
C 30.0.0.0/8 is directly connected, Serial0/0/0

Router# enable
Router# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip route 30.0.0.0 255.0.0.0 30.0.0.23
Router(config)# exit
Router#
*STB-5=CONFIG_I: Configured from console by console
enable
Router# show ip route
Codes: C - connected, S - static, I - ISRP, R - RIP, M - mobile, B - RGP
       O - EIGRP, E - EIGRP external, D - OSPF, T - 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 - OSPF
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, ? - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C 19.0.0.0/8 is directly connected, FastEthernet0/0
S 20.0.0.0/8 [1/0] via 30.0.0.23
C 30.0.0.0/8 is directly connected, Serial0/0/0

Router#

```

```

C:\> ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\> ping 10.0.0.10
Invalid Command.

C:\> ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time=1ms TTL=254

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\> ping 30.0.0.20

Pinging 30.0.0.20 with 32 bytes of data:
Reply from 30.0.0.20: bytes=32 time=1ms TTL=255

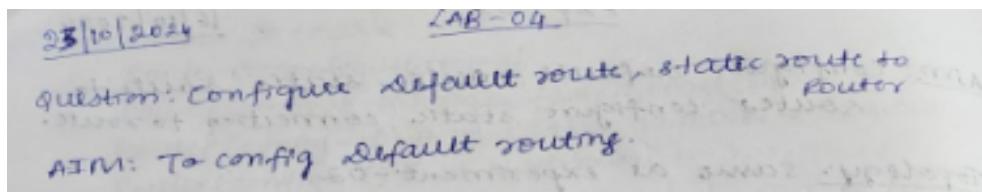
Ping statistics for 30.0.0.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

```

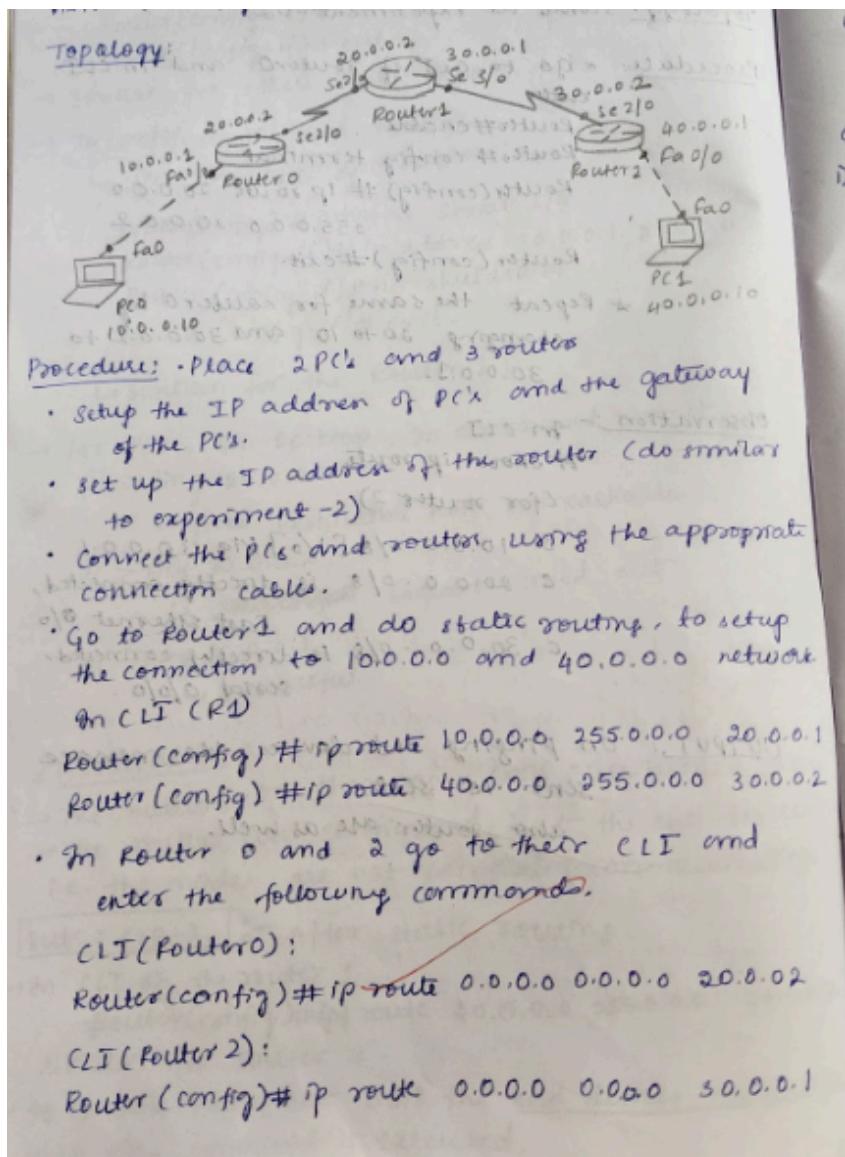
## Program 3

### Question:

Configure default route, static route to the Router



### Observation writeup:



This step is called default routing, in this if any network packet other than connected network will be passed to specified router.

Once the configuration is complete, we can now ping from the end device to other PC.

Ping 40.0.0.10

OBSERVATION :-

a) On pinging from one end device to other.

ping 40.0.0.10

Pingng 40.0.0.10 with 32 bytes of data.

Reply from 40.0.0.10 bytes = 32 time=2ms TTL=253

Reply from 40.0.0.10 bytes = 32 time=2ms TTL=253

Reply from 40.0.0.10 bytes = 32 time=8ms TTL=253

Reply from 40.0.0.10 bytes = 32 time=10ms TTL=253

Ping statistics for 40.0.0.2.

Packet: sent = 4, Received = 4, lost = 0 (0% loss)

b) In Router 0

Router# show ip route

Gateway of last resort is 20.0.0.3 to network 0.0.0.0

- c 10.0.0.0/8 is directly connected, FastEthernet 0/0
- c 20.0.0.0/8 is directly connected, serial 3/0
- s 0.0.0.0/0 [1/0]

(similar output for Router 2).

In Router 1, gateway of last resort is 20.0.0.1

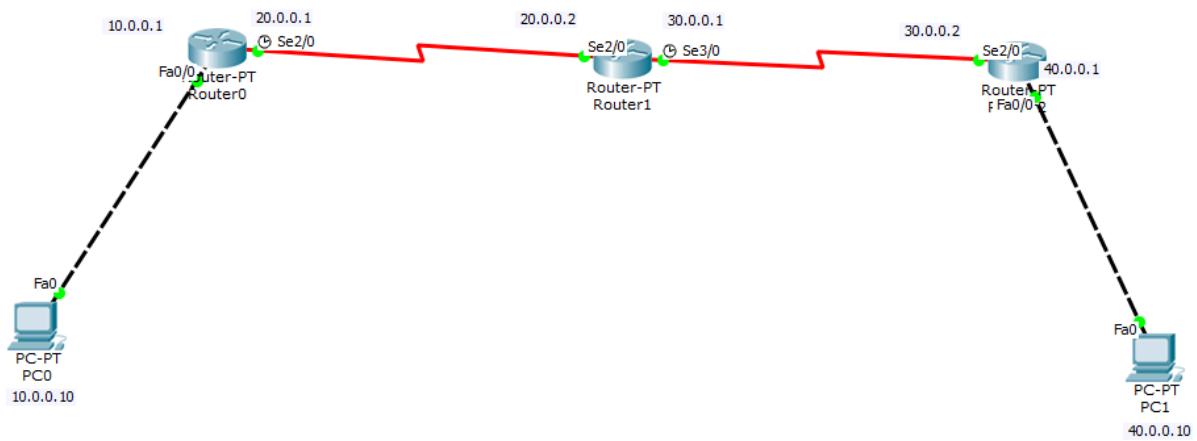
Router# show ip route

Gateway of last resort is not set

- s 10.0.0.0/8 [1/0] via 20.0.0.1
- c 20.0.0.0/8 is directly connected, serial 3/0
- c 30.0.0.0/8 is directly connected, serial 3/0
- s 40.0.0.0/8 [1/0] via 30.0.0.2

Through this experiment, we learnt, on how to connect 3 or more networks by the concept of static and default routing. and we also send messages from end device to other.

Topology:



### iii. Output:

PC> ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.

Reply from 40.0.0.10: bytes=32 time=10ms TTL=125

Reply from 40.0.0.10: bytes=32 time=7ms TTL=125

Reply from 40.0.0.10: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.10:

Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),

Approximate round trip times in milli-seconds:

Minimum = 7ms, Maximum = 10ms, Average = 8ms

PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time=2ms TTL=253

Reply from 40.0.0.1: bytes=32 time=2ms TTL=253

Reply from 40.0.0.1: bytes=32 time=8ms TTL=253

Reply from 40.0.0.1: bytes=32 time=10ms TTL=253

Ping statistics for 40.0.0.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 2ms, Maximum = 10ms, Average = 5ms

PC> SsS

Router0

Physical Config CLI

IOS Command Line Interface

```
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 20.0.0.2 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 20.0.0.2
Router#
```

Copy Paste

Router1

Physical Config CLI

IOS Command Line Interface

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

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#show ip route
^
* Invalid input detected at '^' marker.

Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
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

S    10.0.0.0/8 [1/0] via 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
Router#
```

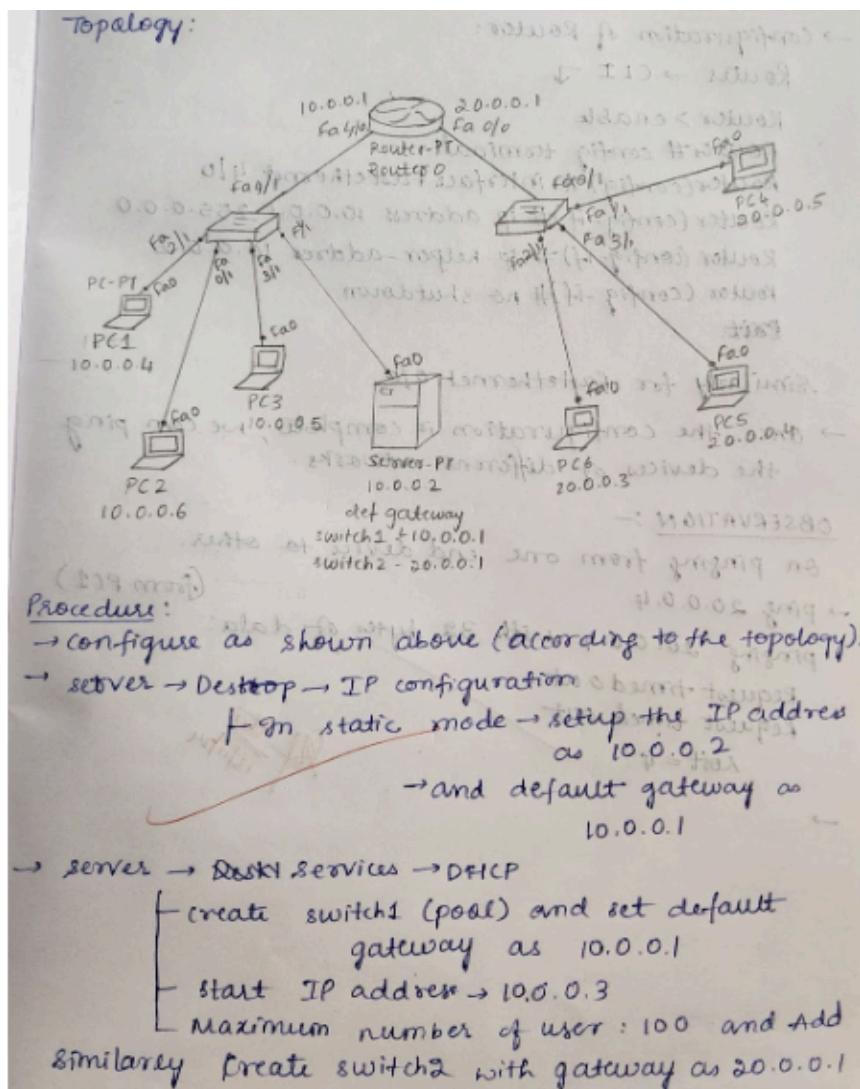
## Program 4

### Question:

Configure DHCP within a LAN and outside LAN

Question: Configure DHCP within a LAN and outside LAN  
 Aim: To configure DHCP both the networks.

### Observation writeup:



→ setup the IP addresses of the PC's by DHCP  
PC → Desktop → IP Configuration → DHCP configuration  
(dynamically IP addresses are allocated).

similar for all the rest PCs.

→ Configuration of Router:

Router → CLI →

Router > enable

Router# config terminal

Router(config)# interface FastEthernet 4/0

Router(config-if)# ip address 10.0.0.1 255.0.0.0

Router(config-if)# ip helper-address 10.0.0.2

Router(config-if)# no shutdown

exit.

Similarly for fastethernet 0/0.

→ Once the configuration is complete, we can ping the devices of different networks.

### OBSERVATION :-

on pinging from one end device to other.

→ ping 20.0.0.4

(from PC1)

pinging 20.0.0.4 with 32 bytes of data:

Request timed out after exp 9.1 ms posted 31 - 131831

Request timed out

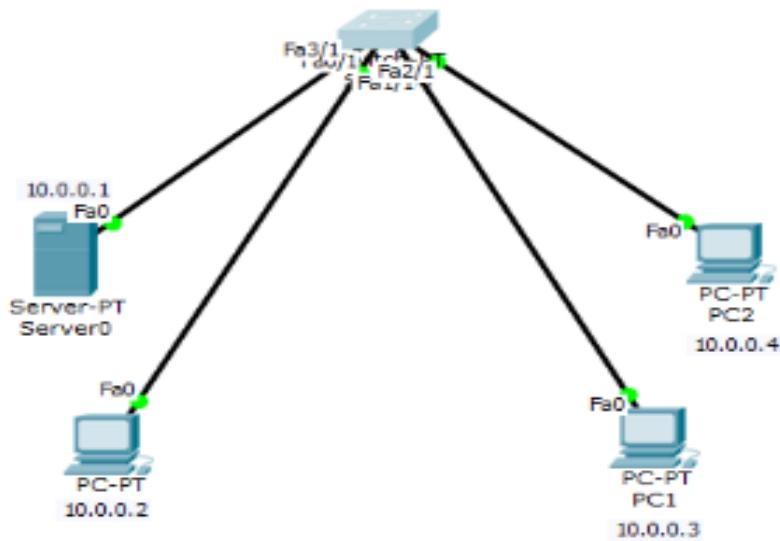
Request timed out along with 131831

lost = 4.01

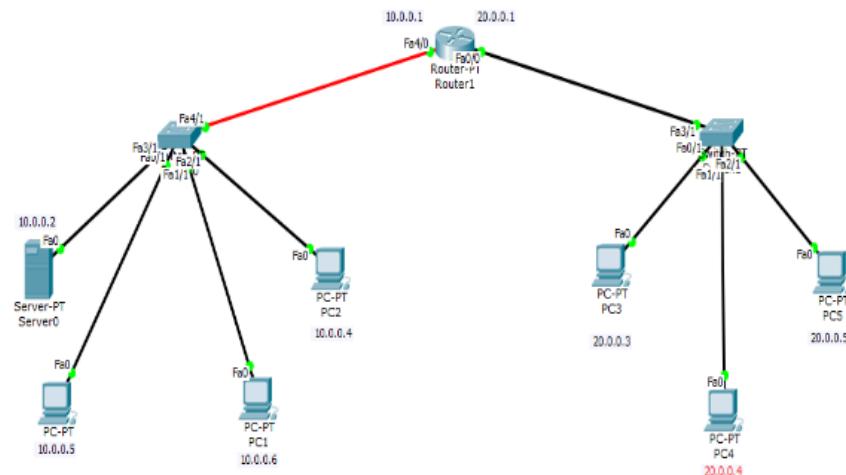
no routers timeout known

Request  
timed out  
131831

## Topology: (within Lan)

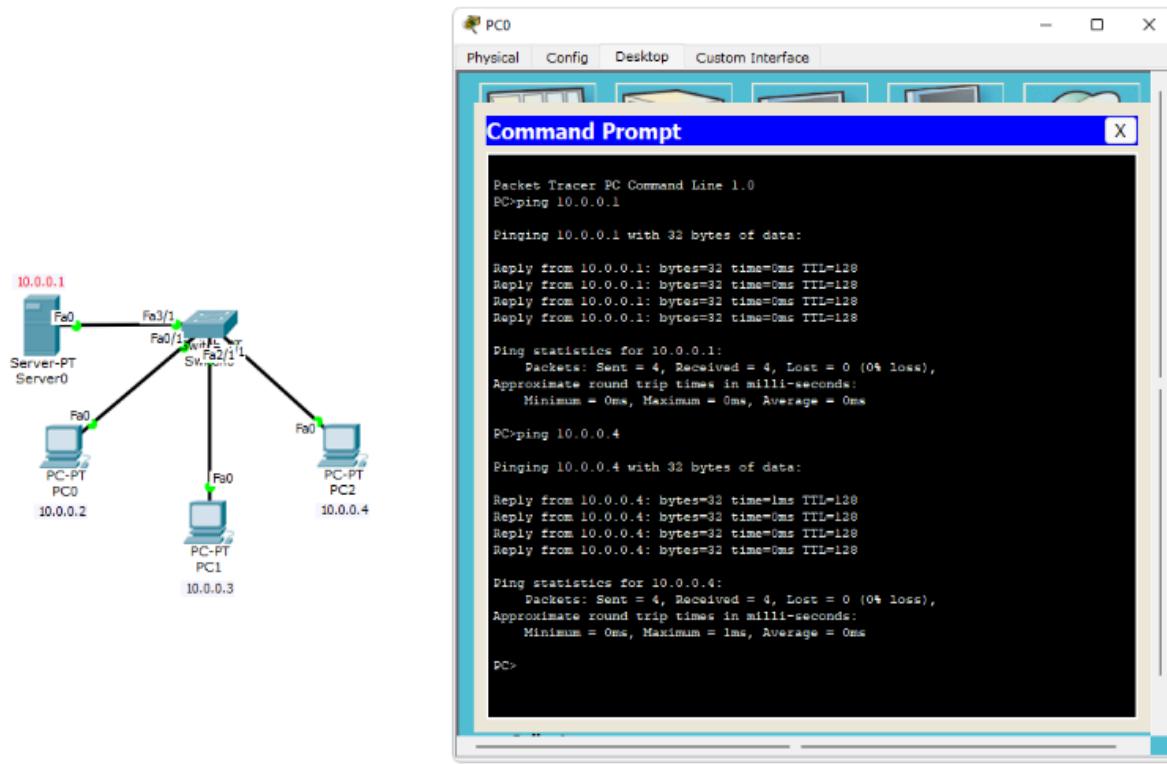


## (outside Lan)

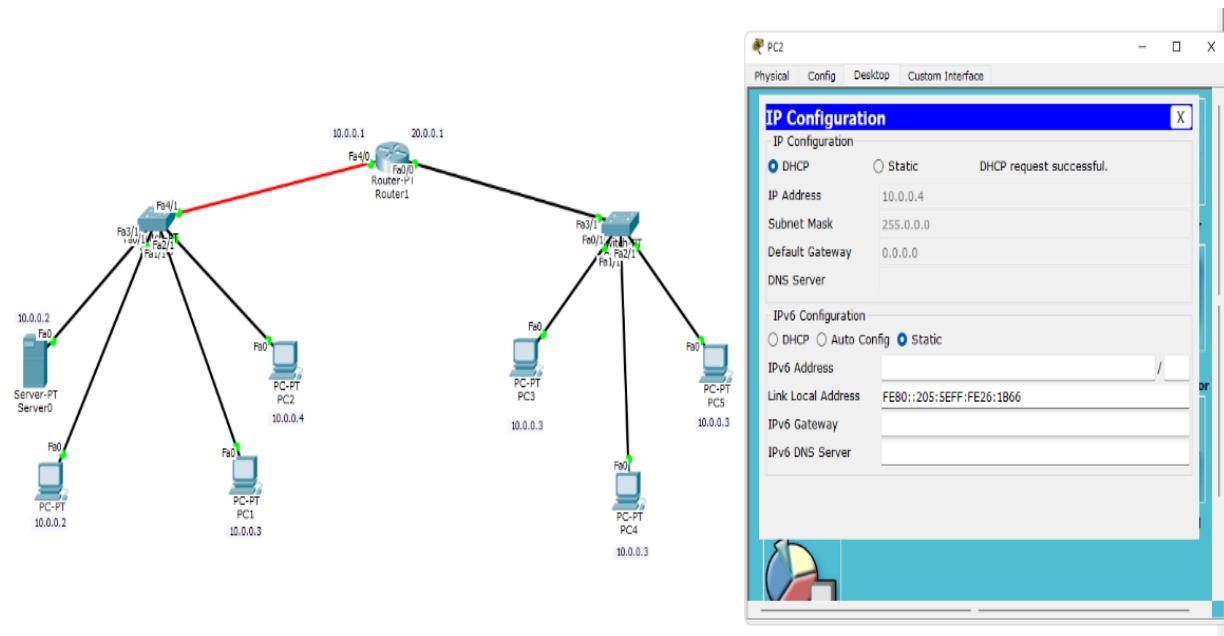


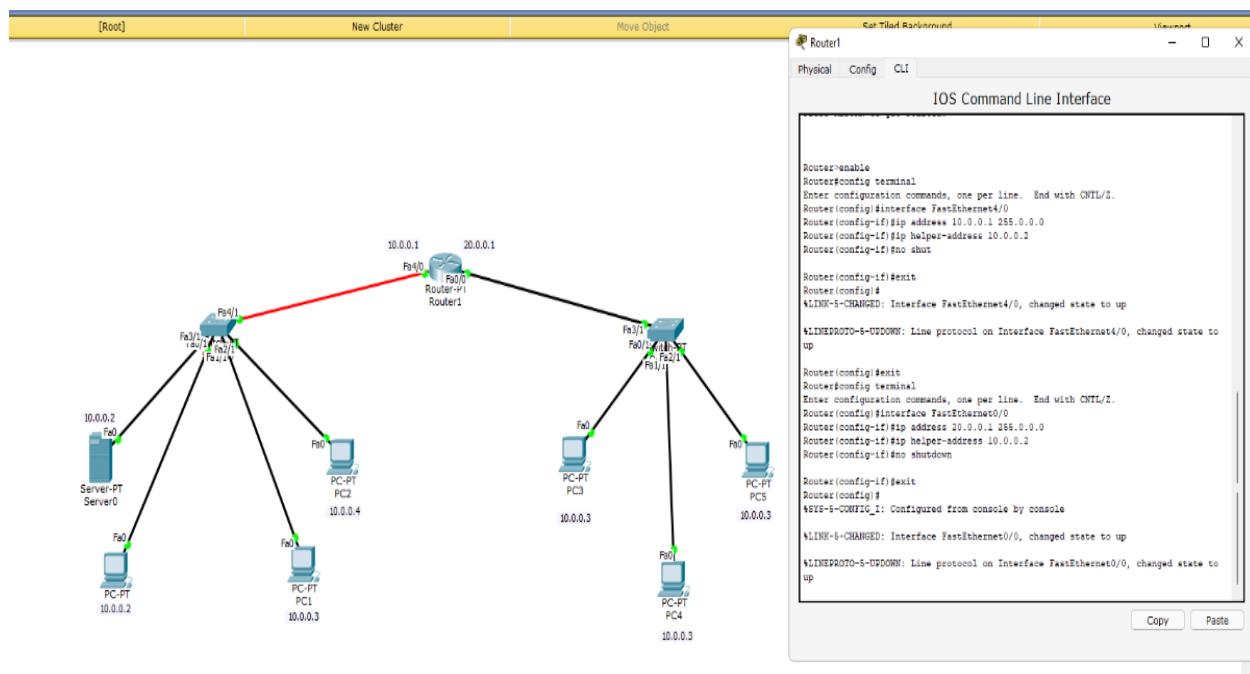
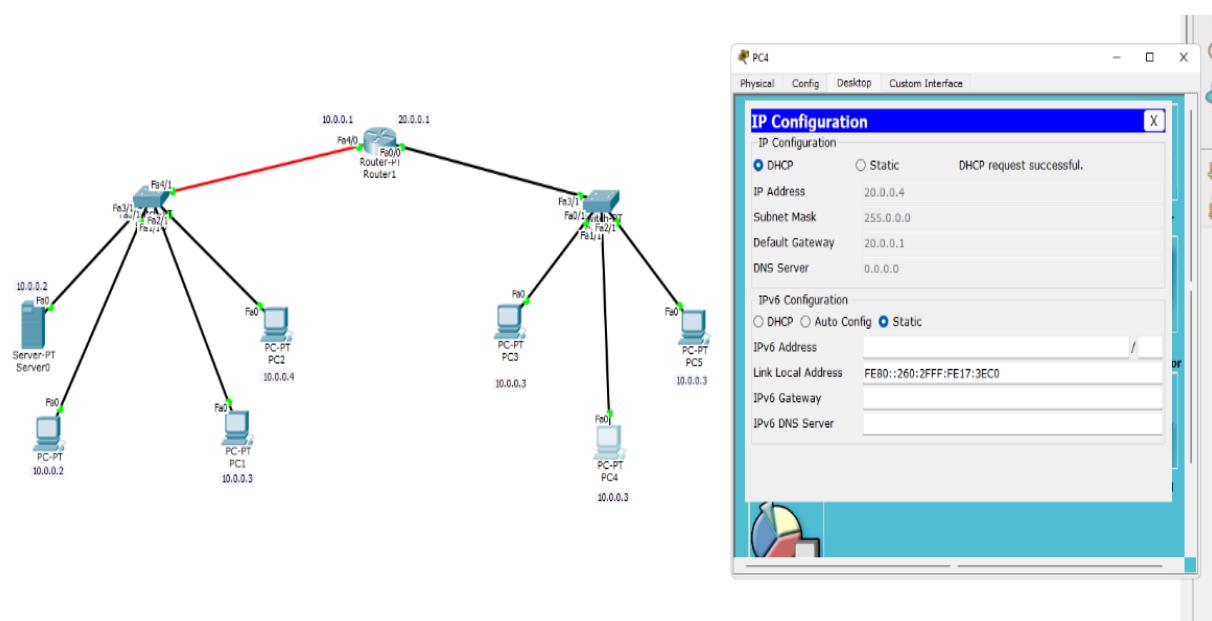
## Output:

(within Lan)



(outside Lan)





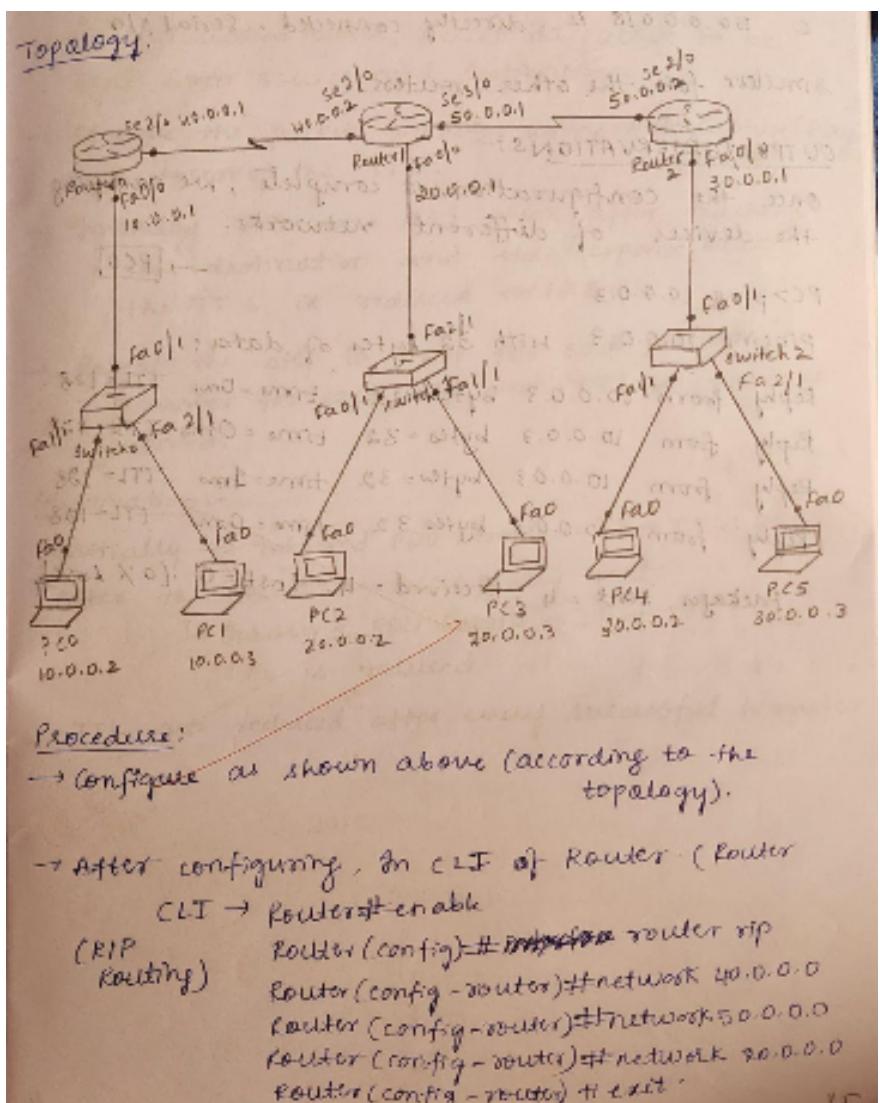
## Program 5

### Question:

Configure RIP routing Protocol in Routers

QUESTION: Configure Routing Information Protocol in Routers (RIP) & up give details of 0.0.0.0/24  
AIM: To configure RIP routing in routers 0.0.0/24

### Observation writeup:



```
# Router > show ip route
```

display:2

R 10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:24, Serial 0/0  
C 20.0.0.0/8 is directly connected, fastethernet 0/0  
C 40.0.0.0/8 is directly connected, serial 2/0  
C 50.0.0.0/8 is directly connected, serial 3/0

similar for the other routers.

### OUTPUT/OBSERVATIONS:-

Once the configuration is complete, we can ping the devices of different networks.

PC> ping 10.0.0.3

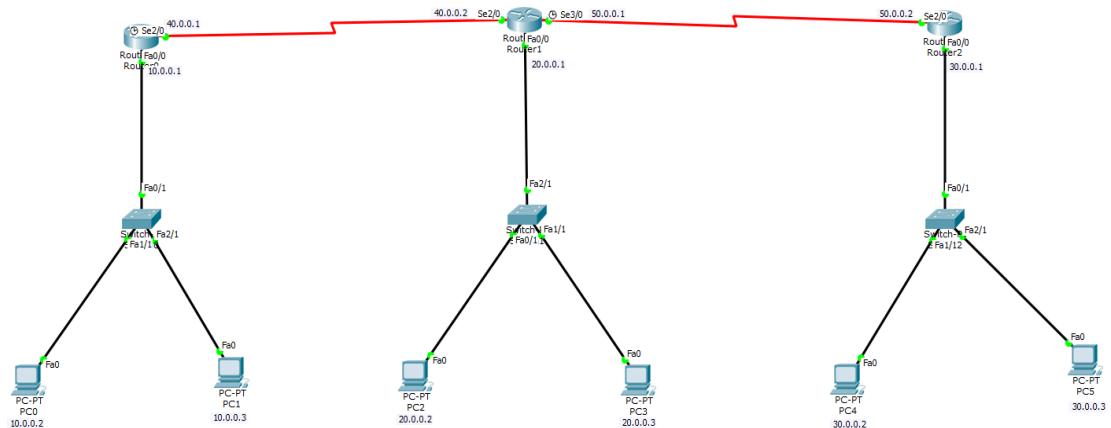
→ [PC0]

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3 bytes=32 time=0ms TTL=128

Bytes sent = 4, Received = 4, Lost = 0 (0% Loss).

### Topology:



## Output:

**Router2**

Physical Config CLI

IOS Command Line Interface

```

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 50.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#
^
* Invalid input detected at '^' marker.

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

R  10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:19, Serial2/0
R  20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:19, Serial2/0
C  30.0.0.0/8 is directly connected, FastEthernet0/0
R  40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:19, Serial2/0
C  50.0.0.0/8 is directly connected, Serial2/0
Router#

```

The network diagram shows Router2 connected to a Switch via port Fa0/1. The Switch is also connected to two PCs labeled PC-PT PC4 (IP 30.0.0.2) and PC-PT PC5 (IP 30.0.0.3). Router2 has two interfaces: Se2/0 (IP 50.0.0.2) and Fa0/0 (IP 30.0.0.1).

**Router1**

Physical Config CLI

IOS Command Line Interface

```

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#
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

R  10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:24, Serial2/0
C  20.0.0.0/8 is directly connected, FastEthernet0/0
C  40.0.0.0/8 is directly connected, Serial2/0
C  50.0.0.0/8 is directly connected, Serial3/0
Router#

```

The network diagram shows Router1 connected to a Switch via port Fa0/0. The Switch is also connected to two PCs labeled PC-PT PC2 (IP 20.0.0.2) and PC-PT PC3 (IP 20.0.0.3). Router1 has three interfaces: Se2/0 (IP 40.0.0.2), Se3/0 (IP 50.0.0.1), and Fa0/0 (IP 20.0.0.1).

PC0

Physical Config Desktop Custom Interface

**Command Prompt**

```

Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.3

Pinging 30.0.0.3 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.3: bytes=32 time=2ms TTL=125
Reply from 30.0.0.3: bytes=32 time=13ms TTL=125
Reply from 30.0.0.3: bytes=32 time=6ms TTL=125

Ping statistics for 30.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 13ms, Average = 7ms

PC>ping 10.0.0.3

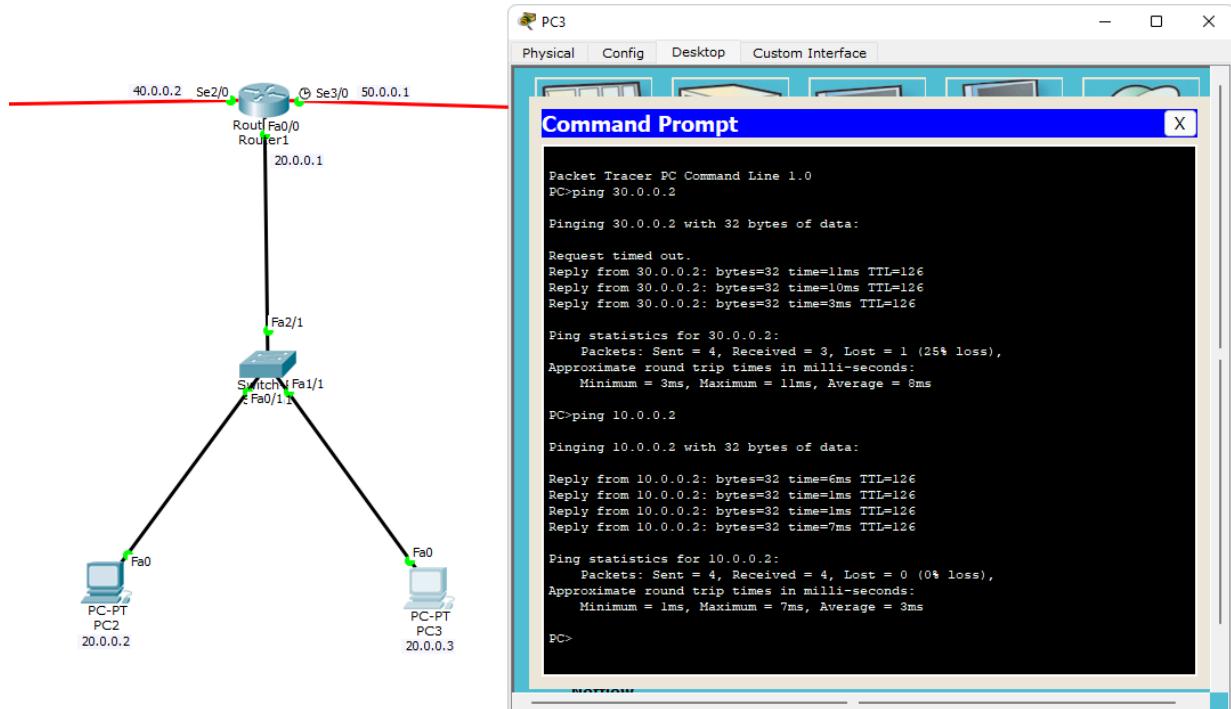
Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms 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 = 1ms, Average = 0ms

PC>

```



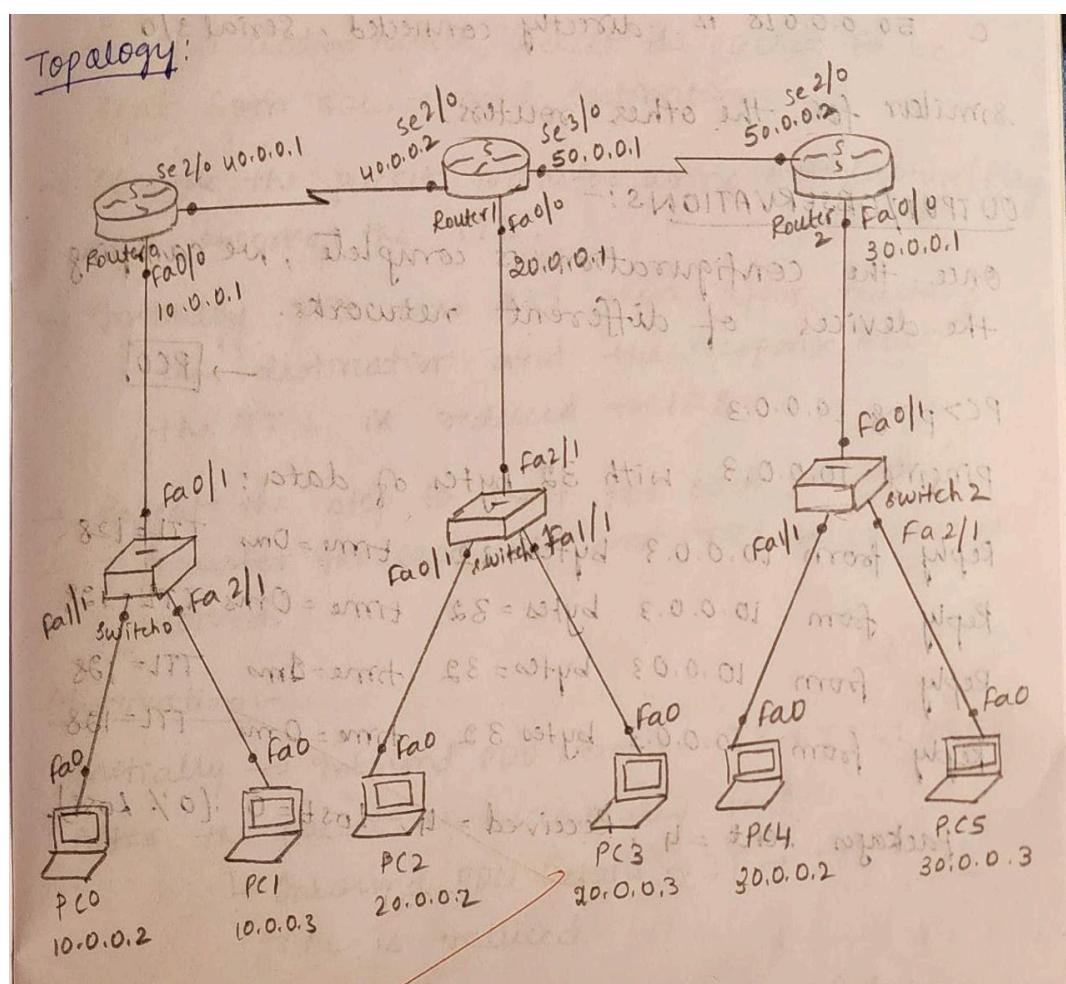
## Program 6

### Question:

Demonstrate the TTL/ Life of a Packet

AIM:  
Demonstrate TTL/Life of Packet  
Topology: Same as experiment-5

### Observation writeup:



Procedure:

- In simulation Mode, select the packet to be sent from source and destination.
- Observe the packet transfer using Auto Capture/Play and observe the TTL.
- Initially TTL was 255 then after reaching the destination and the response back the TTL is reduced to 128.
- Repeat the step to check for each packet transfer for each step and TTL decrease is observed.

Observation:

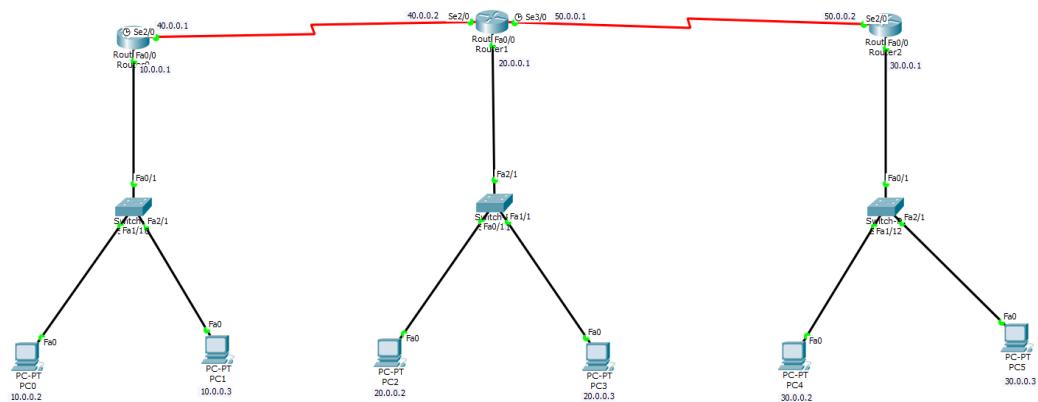
Initially → Inbound PDU Details → IP → TTL: 255

After the packet is successfully sent

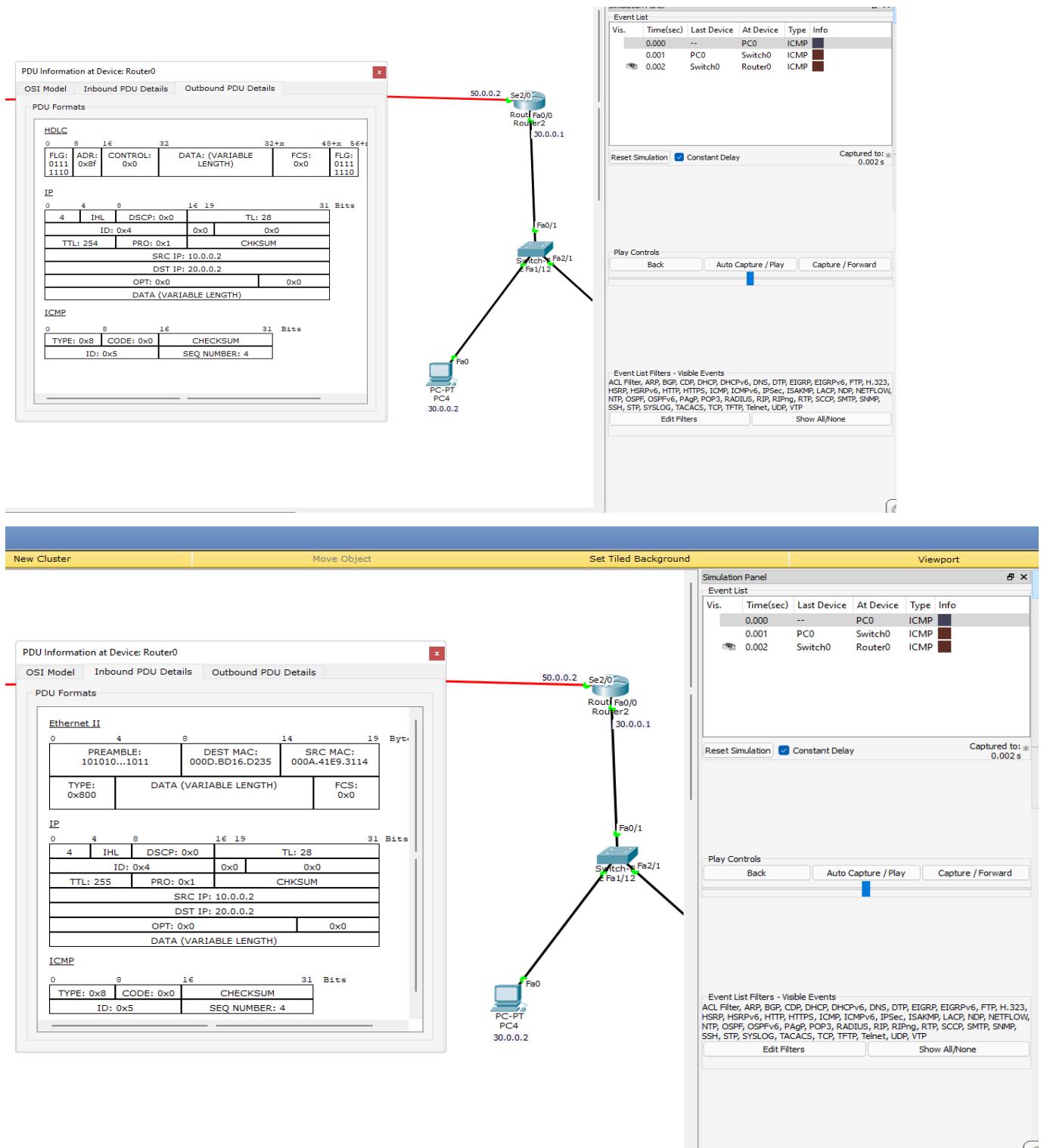
→ Inbound PDU Details → IP → TTL: 126  
TTL is reduced.

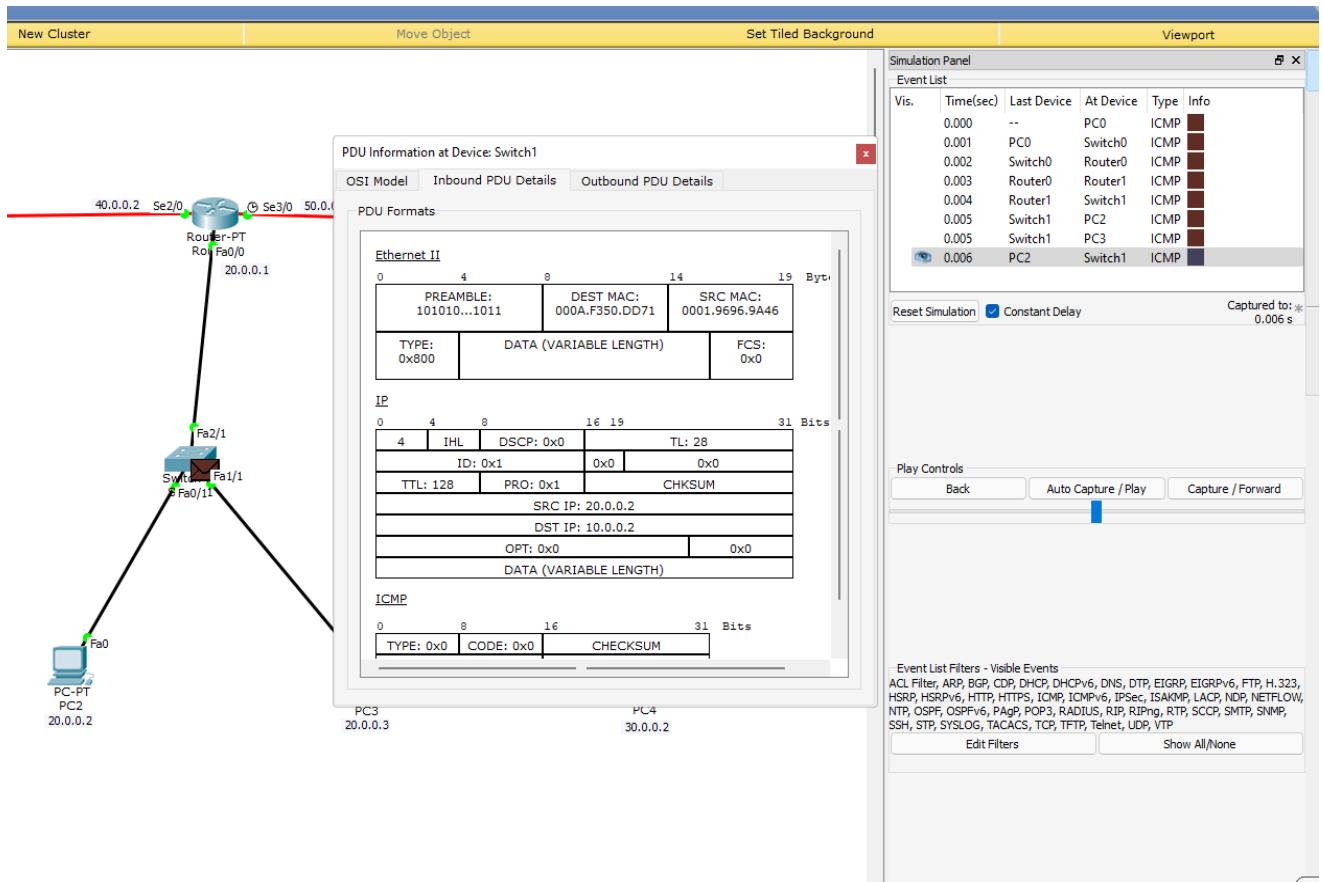
TTL gets reduced after every successful transfer

Ques  
27/1/24



## Output:





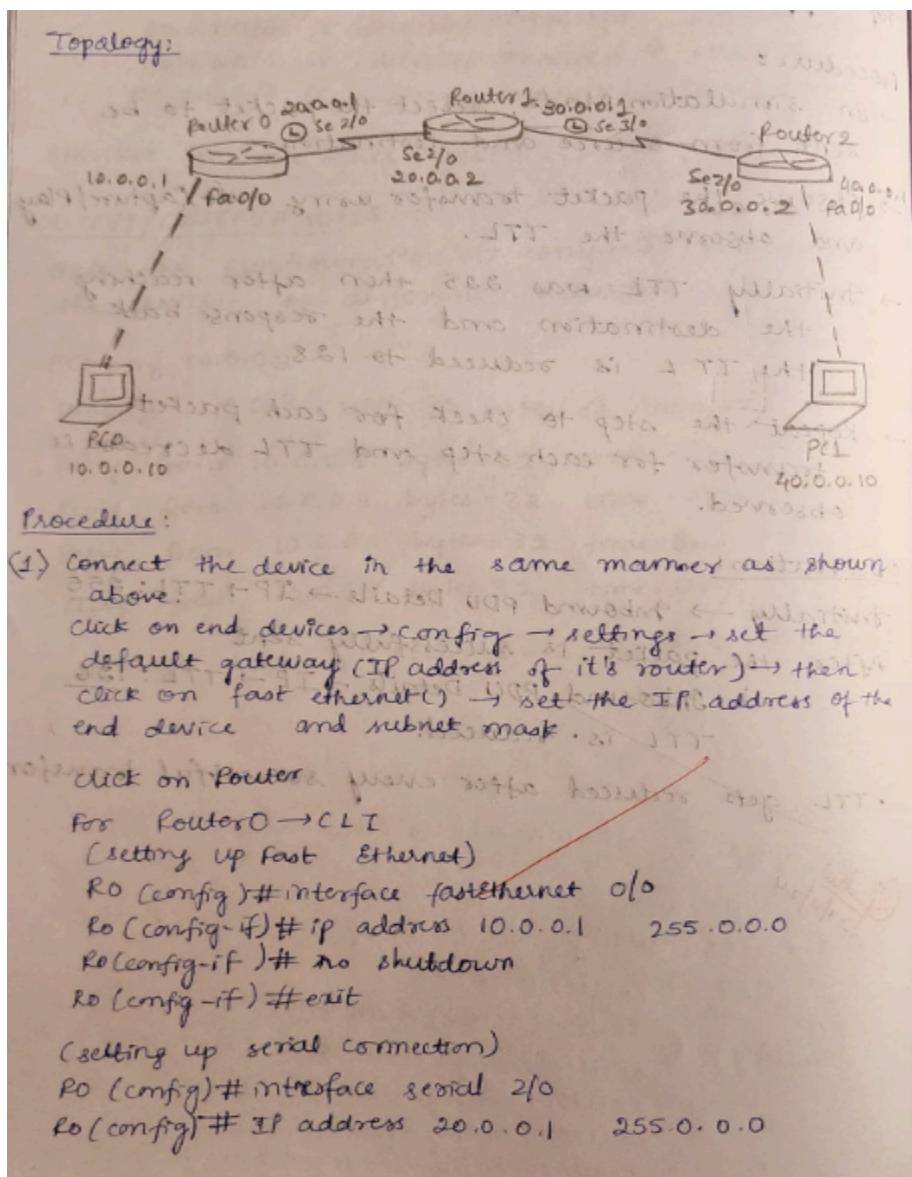
## Program 7

### Question:

Configure OSPF routing protocol

Question: OSPF routing protocol configuration  
Aim: To configure OSPF routing protocol

### Observation writeup:



```
R0(config-if)# encapsulation PPP  
R0(config-if)# clock rate 64000  
R0(config-if)# no shutdown  
R0(config-if)# exit.
```

Similarly we set up the IPs of R1 and R2 while the setup of fast ethernet remains, the setting up of serial connections has 2 extra lines Encapsulation PPP, clock rate 64000) clock rate 6400 must only be written if the serially connected port shows a  $\oplus$  symbol.

Here, we write clock rate command for R0 serial 2/0, R1 serial 3/0

After this step all the connected must have turned green.

(2) To enable IP routing by configuring OSPF routing protocol in all routers.

Router R0  $\rightarrow$  CLI  
R0(config)#router OSPF 1  
R0(config-router)#router\_id 1.1.1.1  
R0(config-router)#network 10.0.0.0 0.255.255.255 area3  
R0(config-router)#network 20.0.0.0 0.255.255.255 area1  
R0(config-router)#exit

Router R1 - CLI

```
R1(config)#router ospf 1  
R1(config-router)#router_id 2.2.2.2  
R1(config-router)#network 20.0.0.0 0.255.255.255 area1  
R1(config-router)#network 30.0.0.0 0.255.255.255 area0  
R2(config-router)#exit
```

In router R2 → CLI

```
R2(config)#router ospf 1
R2(config-router)#router-id 3.3.3.3
R2(config-router)#network 30.0.0.0 0.255.255.255 area0
R2(config-router)#network 20.0.0.0 0.255.255.255 area2
R2(config-router)#exit
```

(3) Once the setting up of networking area is done use configure loopback address to routers.

```
R0(Config-if)#interface loopback 0
R0(config-if)#ip add 172.16.1.252 255.255.0.0
R0(config-if)#no shutdown
R1(config-if)#interface loopback 0
R1(config-if)#ip add 172.16.1.253 255.255.0.0
R1(config-if)#no shutdown
```

```
R2(config-if)#interface loopback 0
R2(config-if)#ip add 172.16.1.254 255.255.0.0
R2(config-if)#no shutdown
```

(4) On checking routing table of R2 using show ip route we can see that R2 doesn't know about area3 gateway of last resort is not set.

```
o 1A . 20.0.0.0/8 [110/125] via 30.0.0.1 serial 1/0
  C 40.0.0.0/8 is directly connected fastethernet 0/0
  C 30.0.0.0/8 is directly connected serial 2/0
```

Since R2 doesn't know about area3, we have to create a virtual link between R0 and R1.

(5) creating virtual link between R1, R0

```
in Router R0
R0(config)#router ospf 1
R0(config-router)#area 1 virtual-link 2.2.2.2
R0(config-router)#exit
R0(config)#exit

in Router R1
R1(config)#router ospf 1
R1(config-router)#area 1 virtual-link 1.1.1.1
R1(config-router)#exit
```

(6) Now, check routing table of R2

Once all these steps are completed, the message can now be pinged from 1 end device to other.

OBSERVATION:

8m p 2

Power # show if works

OIA 20.0.0.0/8 [110/128] via 30.0.0.1 09:57:23, serial 2/0  
C 40.0.0.0/8 is directly connected, FastEthernet 0/0  
OIA 10.0.0.0/8 [110/128] via 30.0.0.1 00:57:05, serial 2/0  
C 30.0.0.0/8 is directly connected, Serial 2/0  
C 172.16.0.0/16 is directly connected, Loopback 0

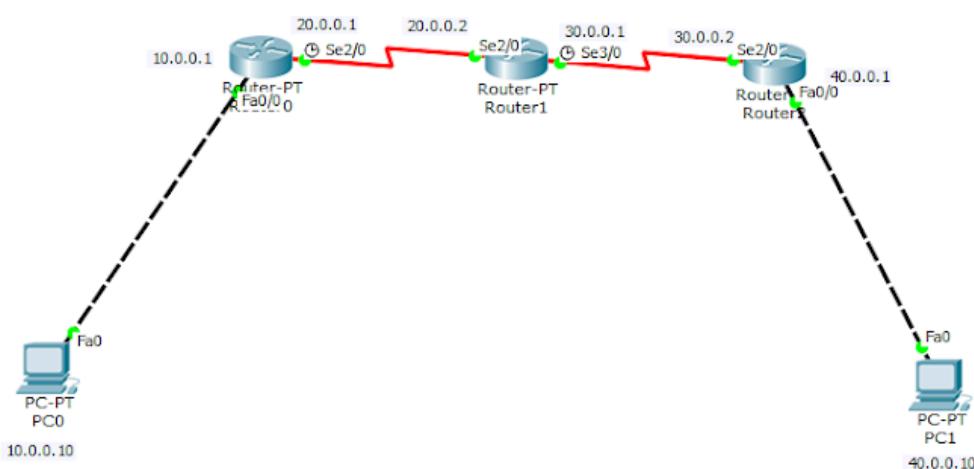
similarly the output is shown for factors and 1

### Ping output:

(From  $P(c)$  to  $P(c_1)$ )

PC0 → command prompt

c:\> ping 40.0.0.10 -n 3 -l 32 -f -w 1000  
Pinging 40.0.0.10 with 32 bytes of data:  
Request timed out.  
Reply from 40.0.0.10 bytes=32 time=21ms TTL=125  
Reply from 40.0.0.10 bytes=32 time=21ms TTL=125  
Reply from 40.0.0.10 bytes=32 time=21ms TTL=125  
Ping statistics for 40.0.0.10  
Packets: sent=4, received=3, lost=1 (25% loss).



## Output:

Router0

Physical Config CLI

### IOS Command Line Interface

```
* Invalid input detected at '^' marker.

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet0/0
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 serial2/0
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)#
*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)#router ospf 1
Router(config-router)#router-id 1.1.1.1
```

Router0

Physical Config CLI

## IOS Command Line Interface

```
*SYS-5-CONFIG_I: Configured from console by console

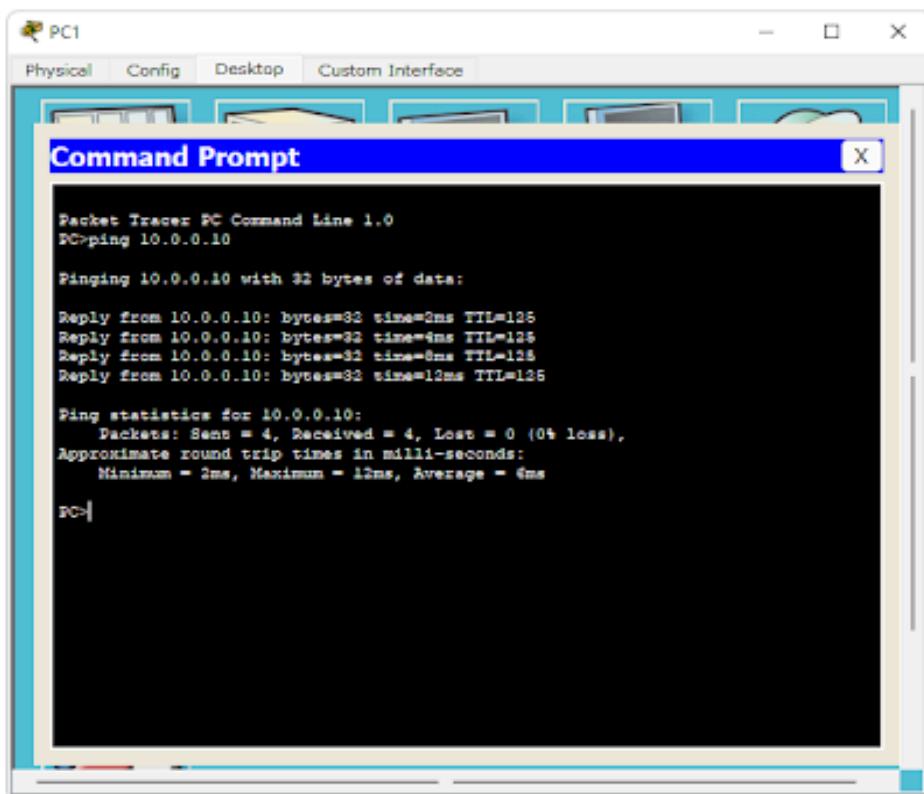
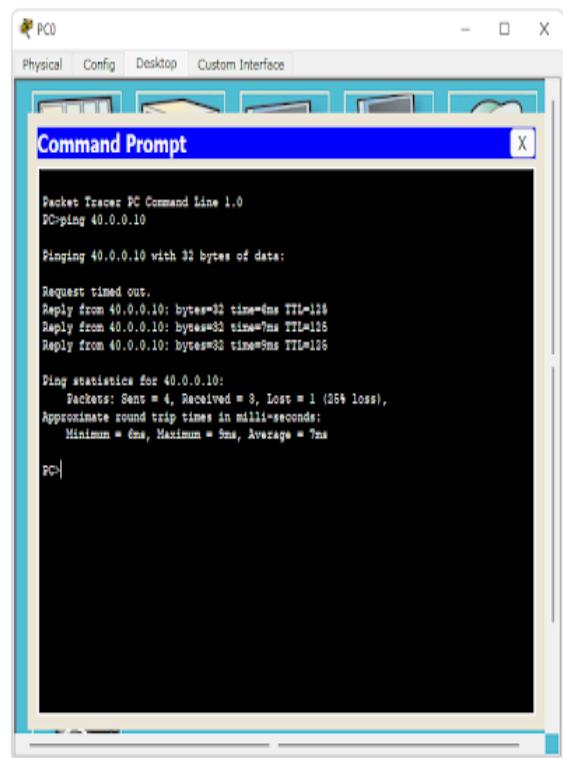
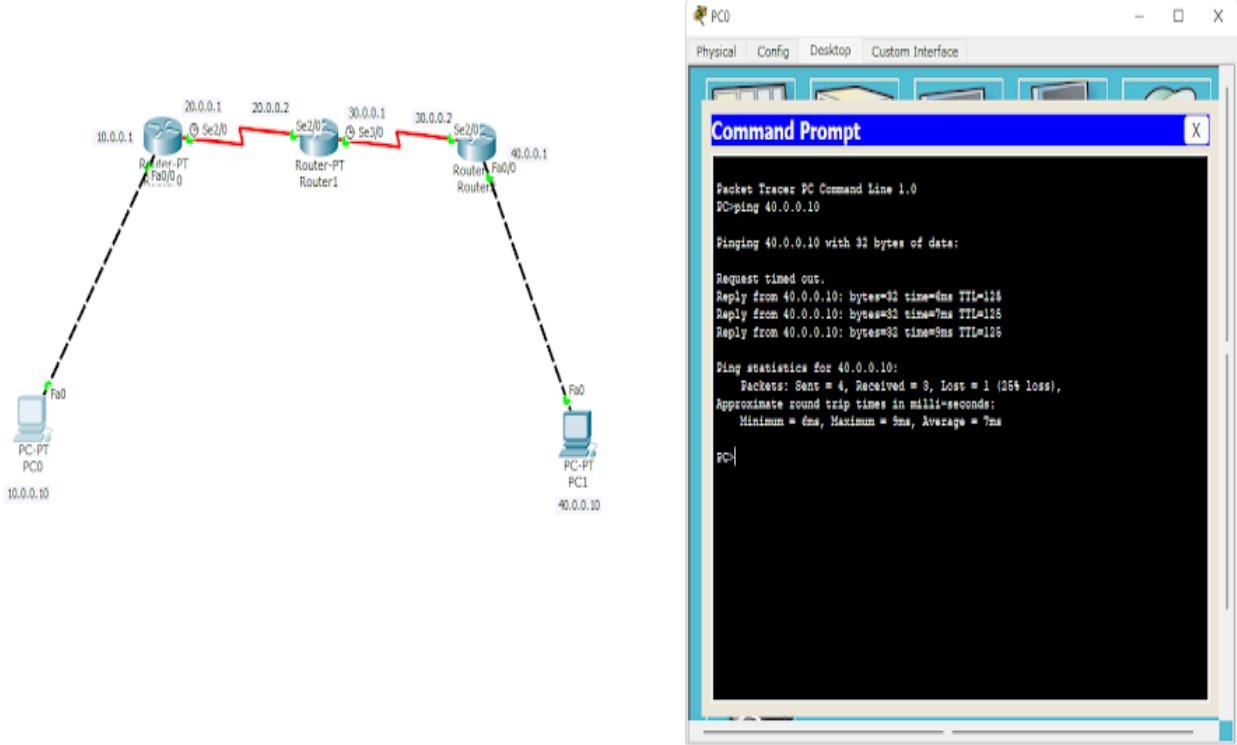
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      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
      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, 00:16:57, Serial2/0
O  IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:15:31, Serial2/0
C      172.16.0.0/16 is directly connected, Loopback0
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#area 1 virtual-link 2.2.2.2
Router(config-router)#
Router(config-router)#
00:42:03: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on OSPF_VL0 from LOADING to FULL,
Loading Done

Router(config-router)#

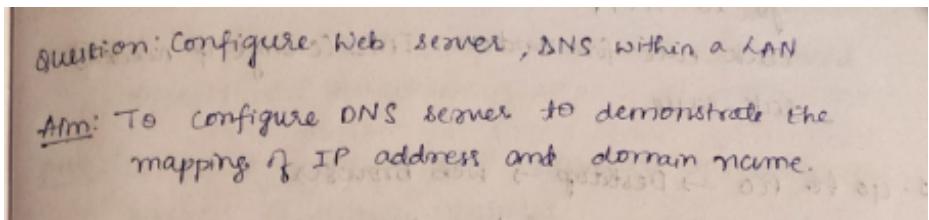
```



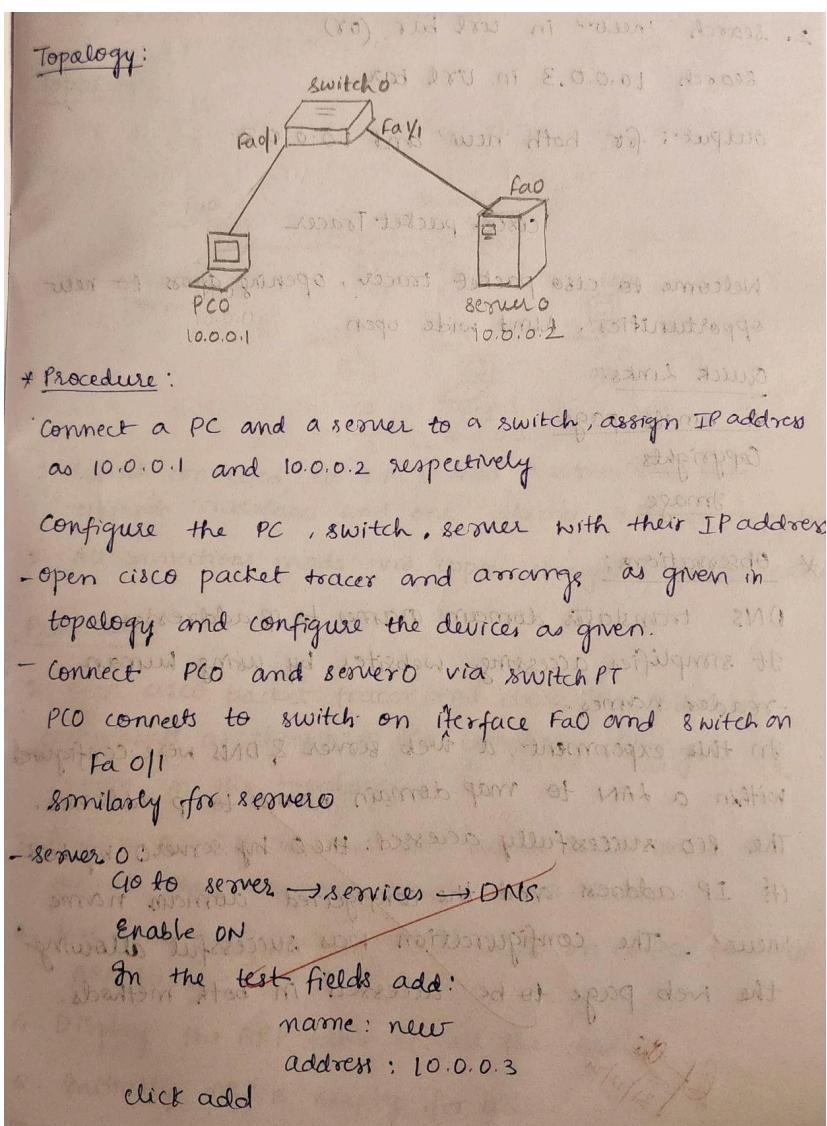
## Program 8

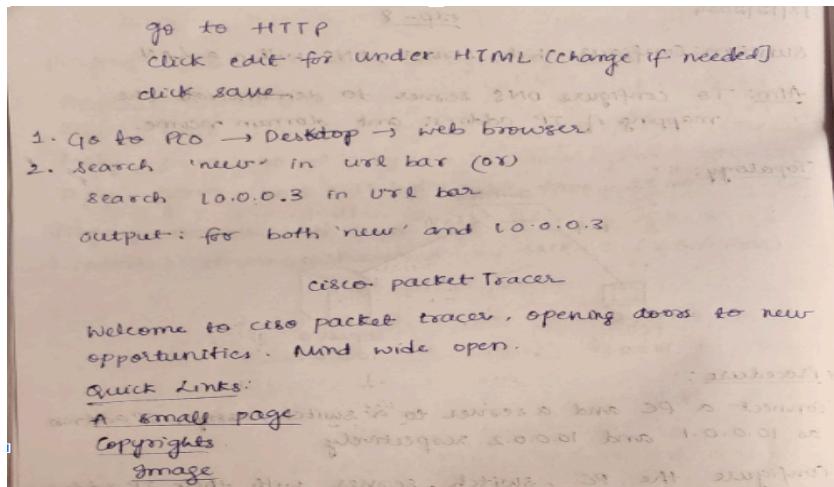
### Question:

Configure Web Server, DNS within a LAN



### Observation writeup:

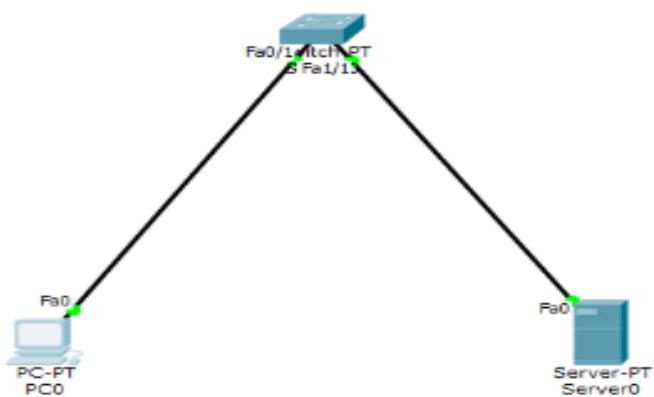




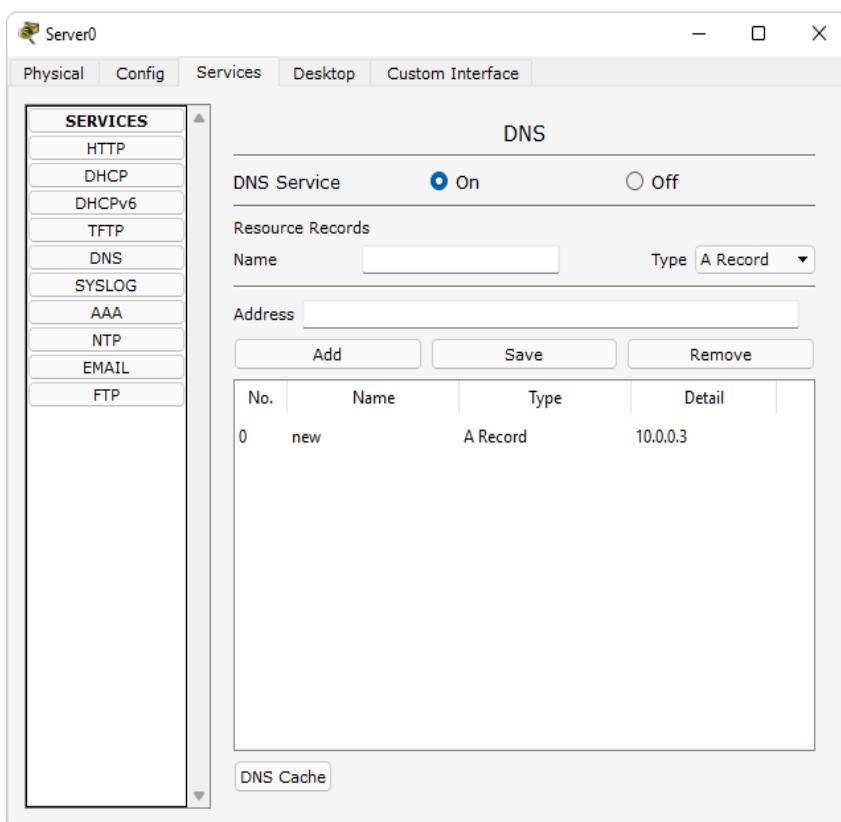
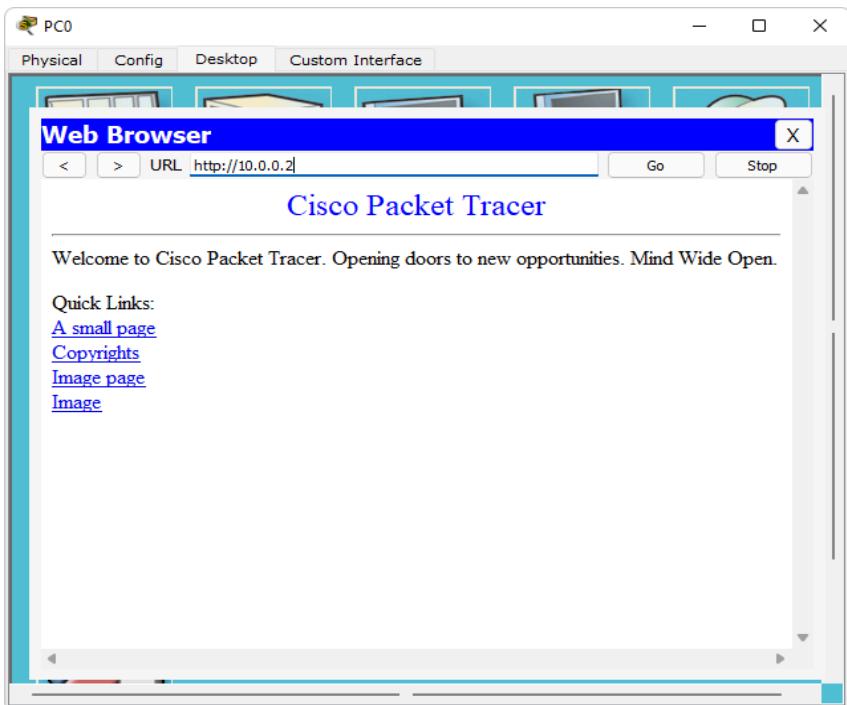
\* Observations: ~~generated from report following Cisco PT.~~  
 DNS translates domain names to IP addresses ~~and~~  
 It simplifies accessing websites by using human  
 reader names.  
 In this experiment, a web server & DNS were configured  
 within a LAN to map domain names to IP addresses.  
 The PC0 successfully accessed the server by both  
 its IP address and the configured domain name  
 'new'. The configuration was successful allowing  
 the web page to be accessed in both methods.

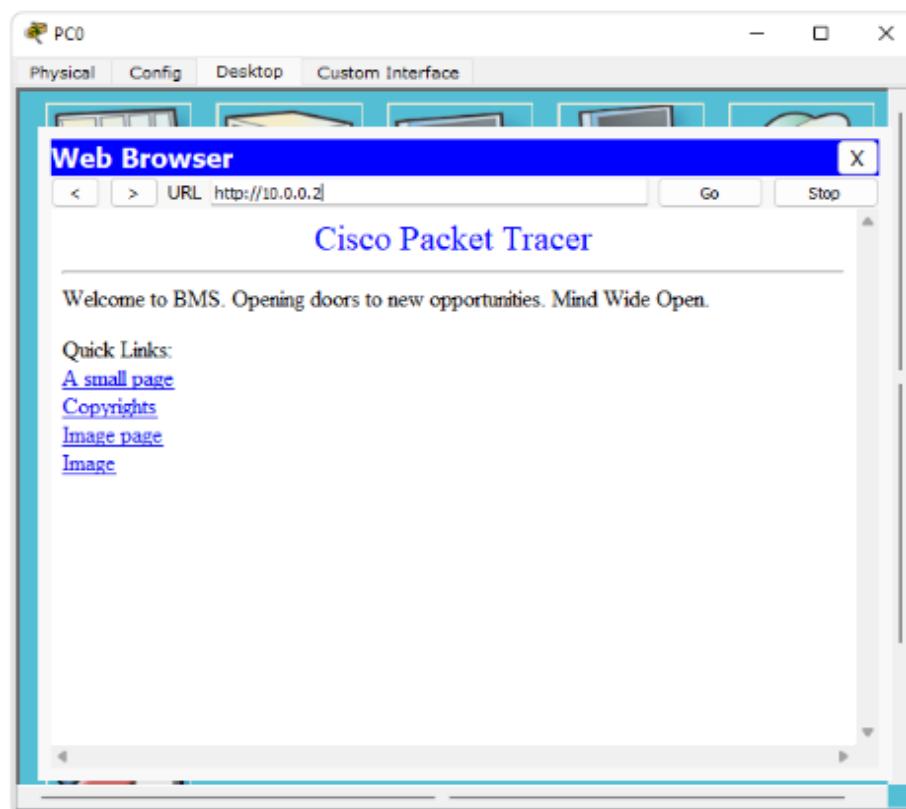
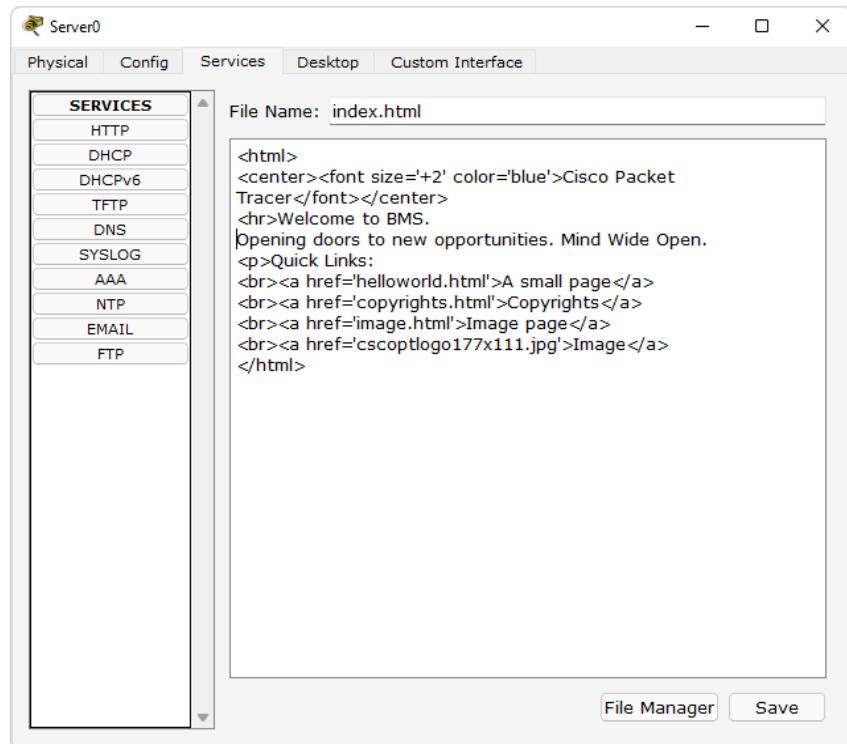
31/12/2018  
 8.0.0.0 : Network  
 labo 12/2018

## Topology:



## Output:





## Program 9

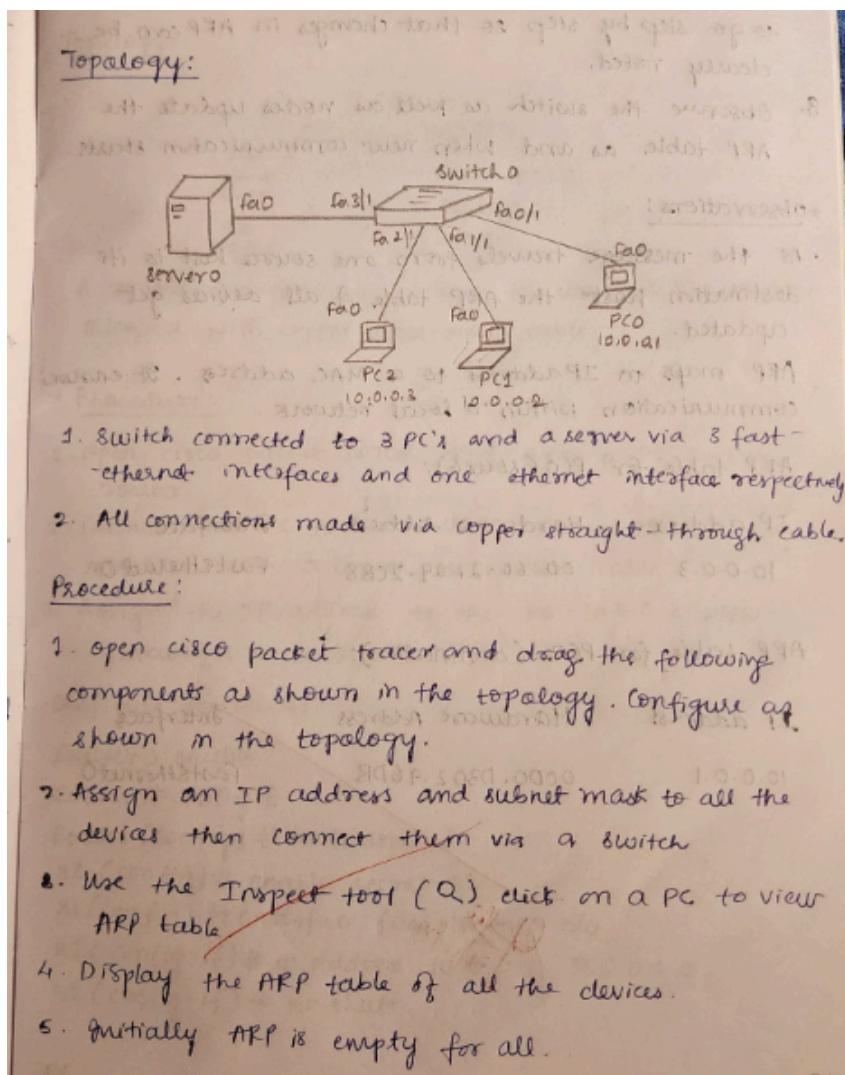
### Question:

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

question: To construct a simple LAN and understand concept and operation of ARP.

Aim: Construct a simple LAN & simulate operation of Address Resolution Protocol.

### Observation writeup:



6. Also in CLI of switch the command `show mac address-table` can be given on every transaction to see how the switch learns from transactions and build the address table.
7. Use the capture button in the simulation panel to go step by step so that changes in ARP can be clearly noted.
8. Observe the switch as well as nodes update the ARP table as and when new communication starts.

\*Observations:

As the message travels from one source host to its destination host the ARP table of all devices get updated.

ARP maps an IP address to a MAC address. It ensures communication within a local network.

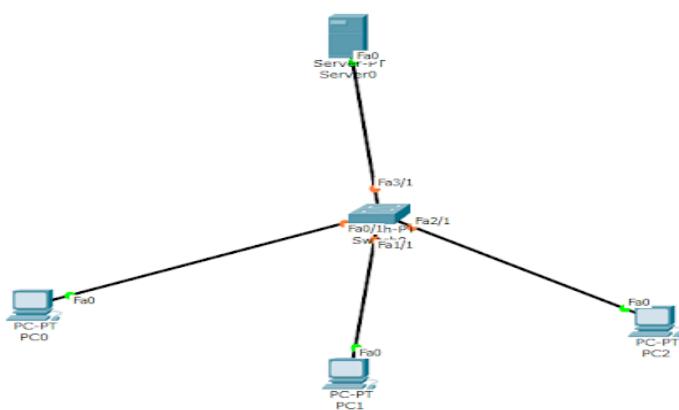
ARP table for PC0 (source):

IP address	Hardware Address	Interface
10.0.0.3	00:60-2F29-2CB8	FastEthernet0

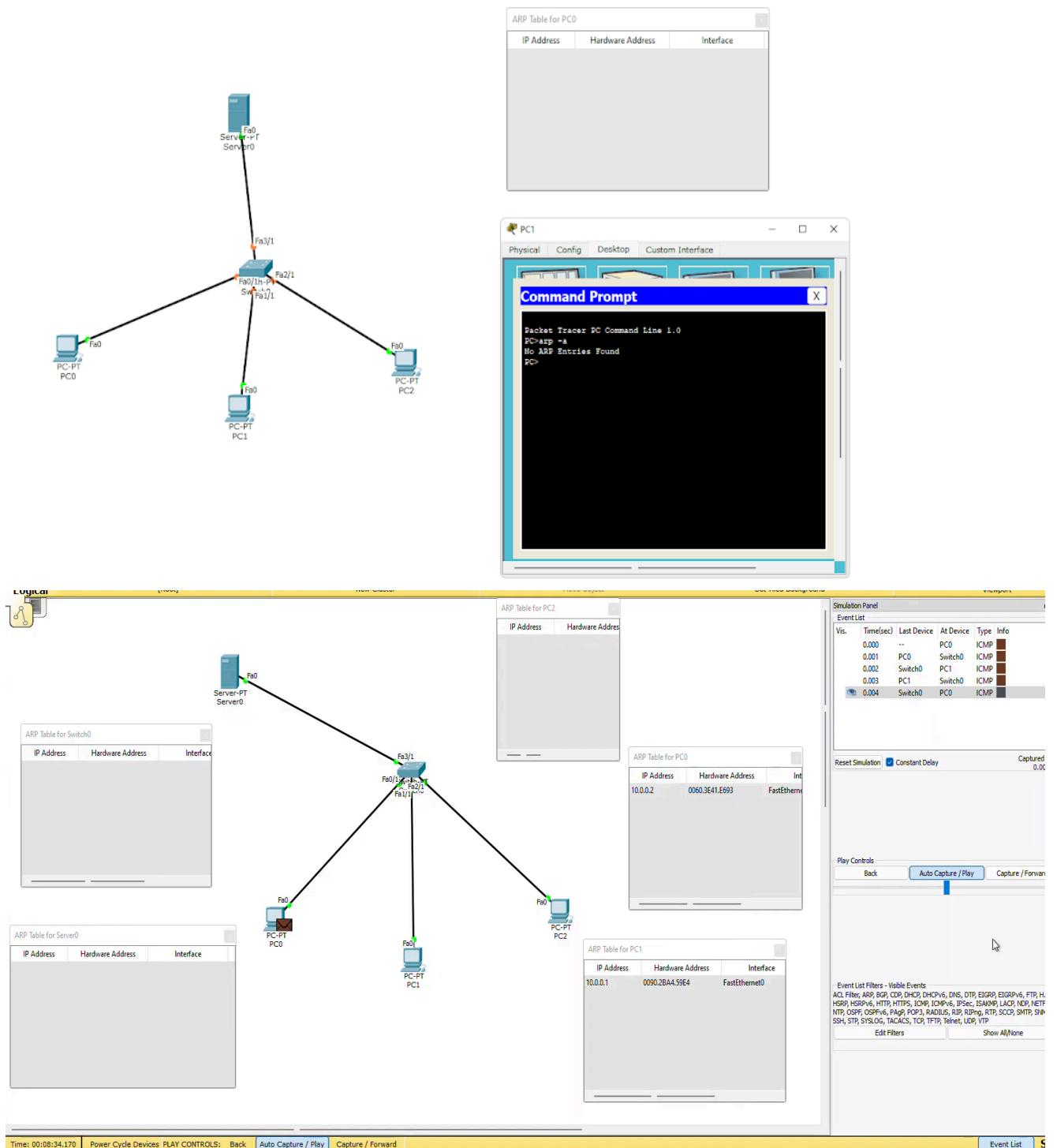
ARP table for PC2 (destination):

IP address	Hardware Address	Interface
10.0.0.1	00:00.D302-96DB	FastEthernet0

## Topology:



## Output:



Switch0

Physical Config CLI

### IOS Command Line Interface

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state to up

%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state to up

Switch>show mac address-table
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              -----      -----
  1        000a.41e0.130b    DYNAMIC   Fa0/1
  1        0060.47e1.1058    DYNAMIC   Fa1/1
  1        00d0.d337.698e    DYNAMIC   Fa2/1
Switch>show mac address-table
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              -----      -----
  1        000a.41e0.130b    DYNAMIC   Fa0/1
  1        0060.47e1.1058    DYNAMIC   Fa1/1
  1        00d0.d337.698e    DYNAMIC   Fa2/1
Switch>
```

Copy Paste

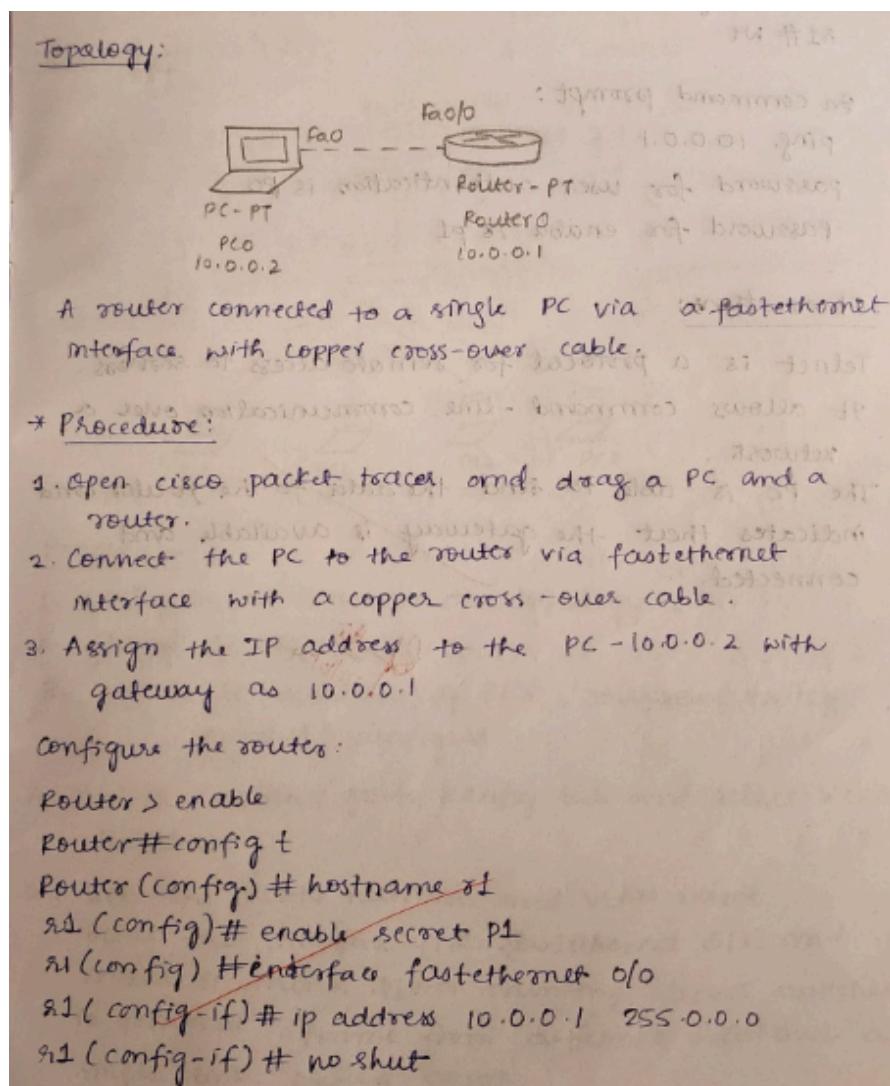
## Program 10

### Question:

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

question: To construct/understand the operation of TELNET  
Aim: To understand the operation of TELNET by accessing the router in server room from a pc in IT office.

### Observation writeup:



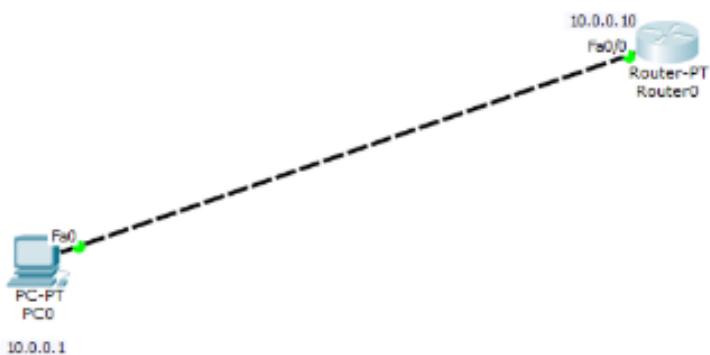
```

R1 (config-if)# line vty 0 5
R1 (config-if)# login
R1 (config-line)# password PO
R1 (config-line)# exit
R1 (config)# exit
R1# wr

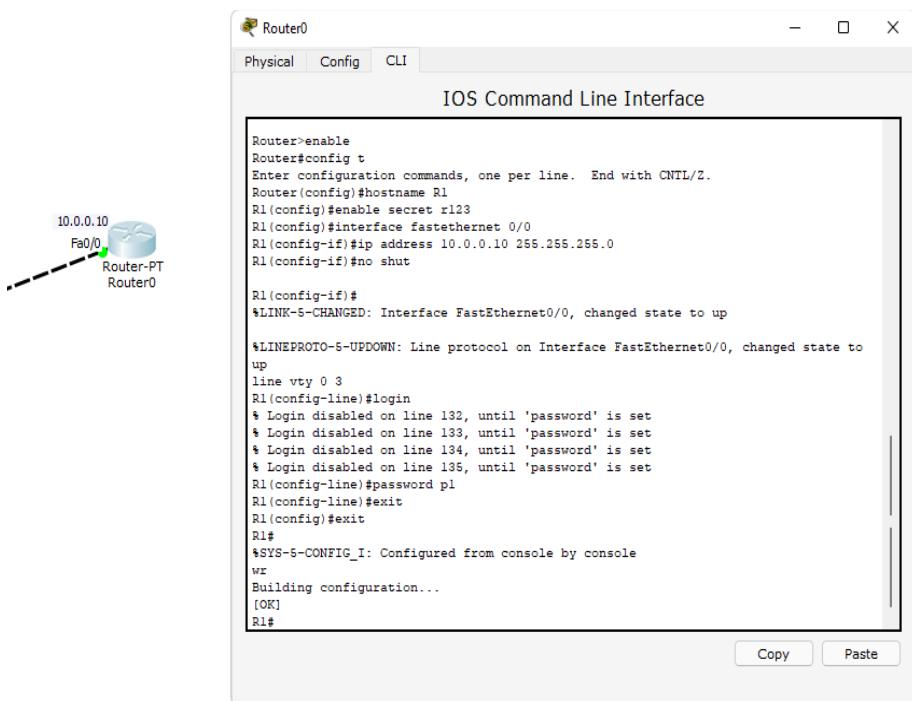
In command prompt:
ping 10.0.0.1
password for user authentication is PO
password for enable is p1

```

## Topology:



## Output:



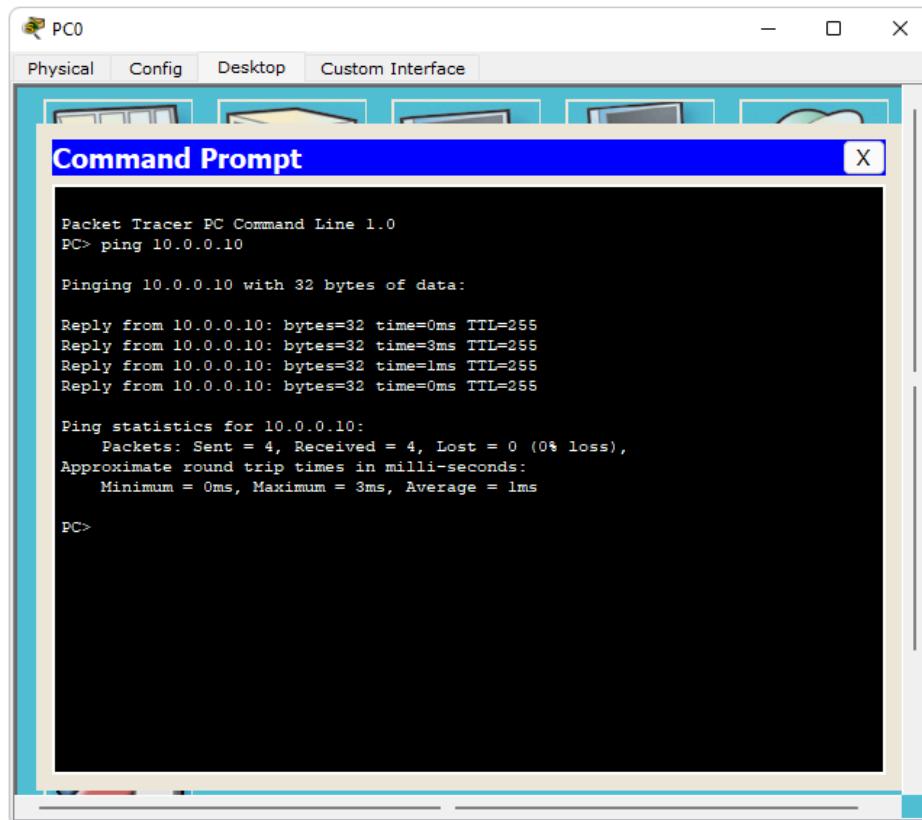
The window title is "Router0". The tab bar shows "Physical", "Config", and "CLI". The main area is titled "IOS Command Line Interface". It displays the following configuration commands:

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#enable secret r123
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.10 255.255.255.0
R1(config-if)#no shut

R1(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
line vty 0 3
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
R1(config-line)#password p1
R1(config-line)#exit
R1(config)#
%SYS-5-CONFIG_I: Configured from console by console
wr
Building configuration...
[OK]
R1#
```

Buttons at the bottom right are "Copy" and "Paste".



The window title is "PC0". The tab bar shows "Physical", "Config", "Desktop", and "Custom Interface". A toolbar with icons for File, Edit, View, Tools, Help, and a search bar is visible. The main area is titled "Command Prompt" and contains the following ping command output:

```
Packet Tracer PC Command Line 1.0
PC> ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Reply from 10.0.0.10: bytes=32 time=3ms TTL=255
Reply from 10.0.0.10: bytes=32 time=1ms TTL=255
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 3ms, Average = 1ms

PC>
```

**Router0**

Physical Config CLI

**IOS Command Line Interface**

```
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#enable secret r123
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.10 255.255.255.0
R1(config-if)#no shut

R1(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
```

Copy Paste

**PC0**

Physical Config Desktop Custom Interface

**Command Prompt**

```
Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Reply from 10.0.0.10: bytes=32 time=3ms TTL=255
Reply from 10.0.0.10: bytes=32 time=1ms TTL=255
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 3ms, Average = 1ms

PC>telnet 10.0.0.10
Trying 10.0.0.10 ...Open

User Access Verification

Password:
Password:
R1>pl
Translating "pl" .. domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address

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

## Program 11

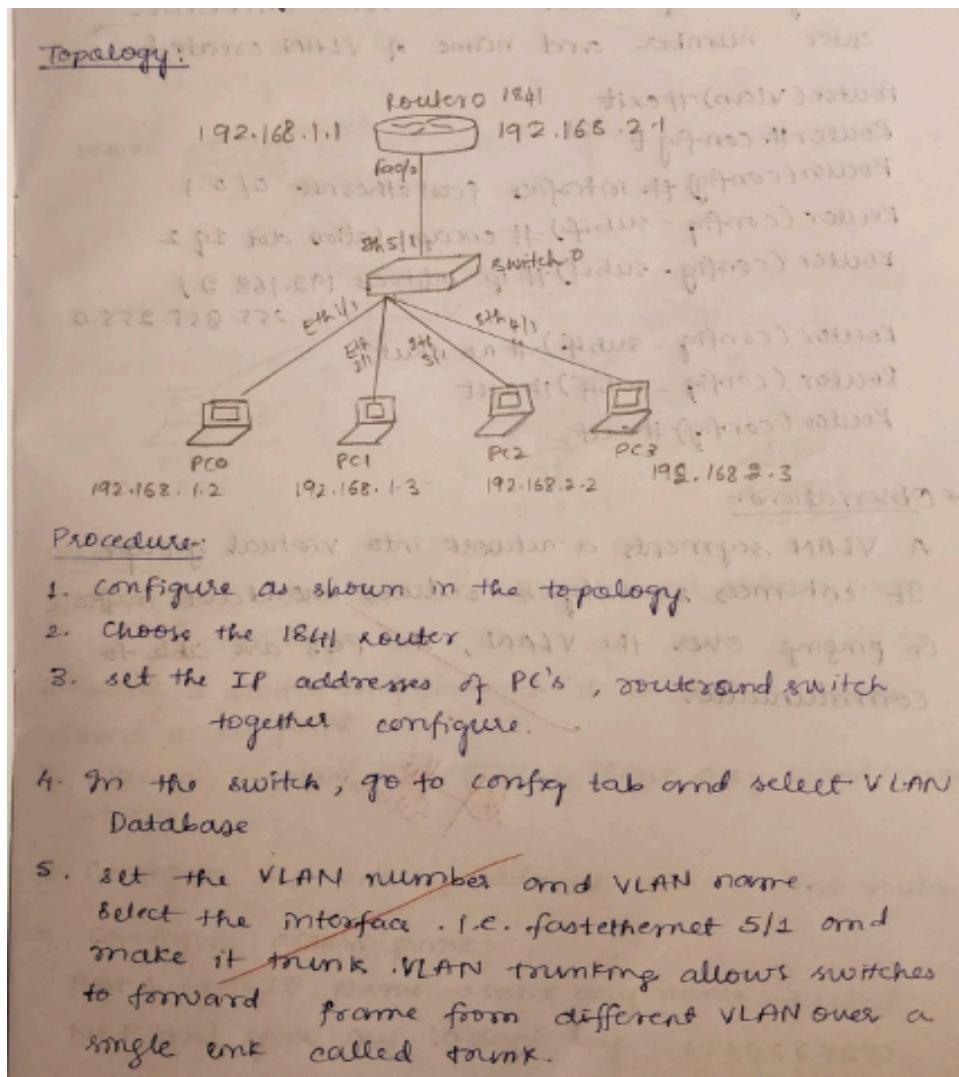
### Question:

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

question to construct VLAN and enable communication among a VLAN.

Aim: Construct VLAN and enable PCs communication among a VLAN.

### Observation writeup:



6. This is done by adding an additional header information called tag to the ethernet frame.
7. Look into the interface IP of the switcher with a NEW VLAN systems.
- config tab of router select <sup>VLAN</sup> VLAN DATABASE - enter number and name of VLAN created.

```

Router(Vlan)#exit
Router#config t
Router(config)# interface fastethernet 0/0/1
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ip address 192.168.2.1
                           255.255.255.0
Router(config-subif)# no shutdown
Router(config-subif)#exit
Router(config)#exit

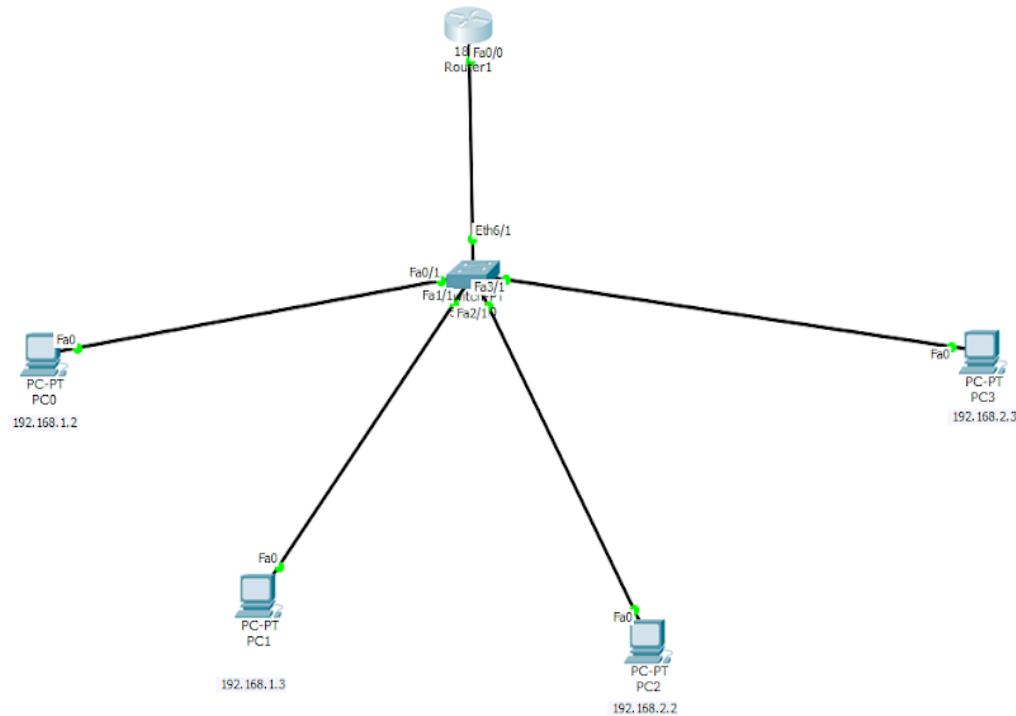
```

#### \* Observations:-

A VLAN segments a network into virtual groups. It enhances security & reduces broadcast traffic. On pinging over the VLAN, the PCs are able to communicate.

~~With VTP server, it is able to automatically update configuration across all switches.~~

## Topology:

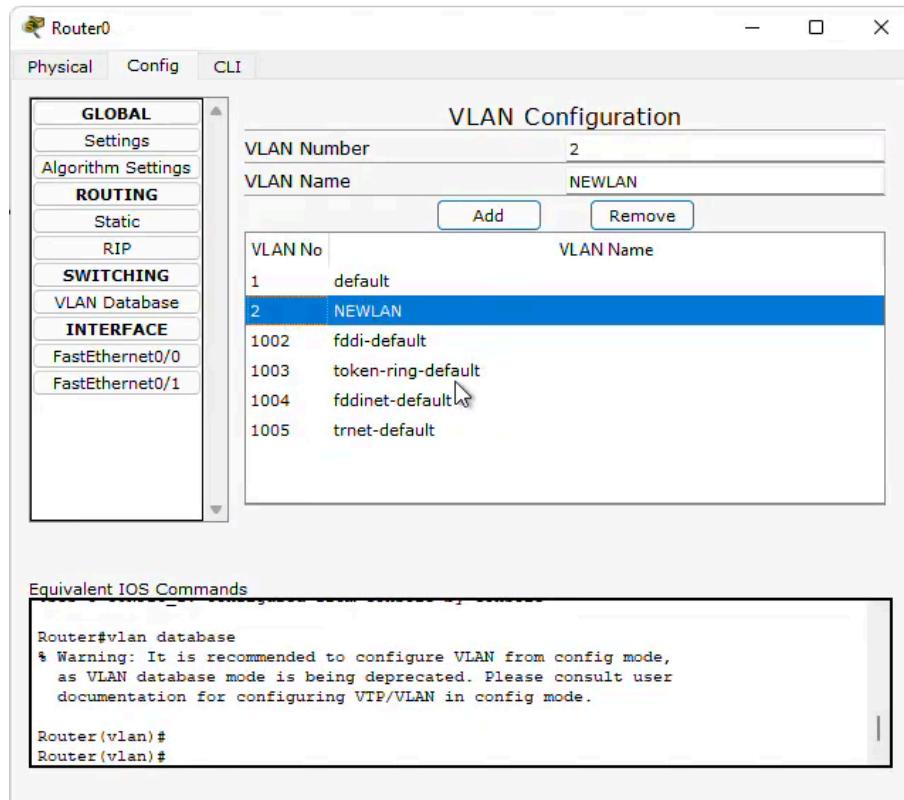
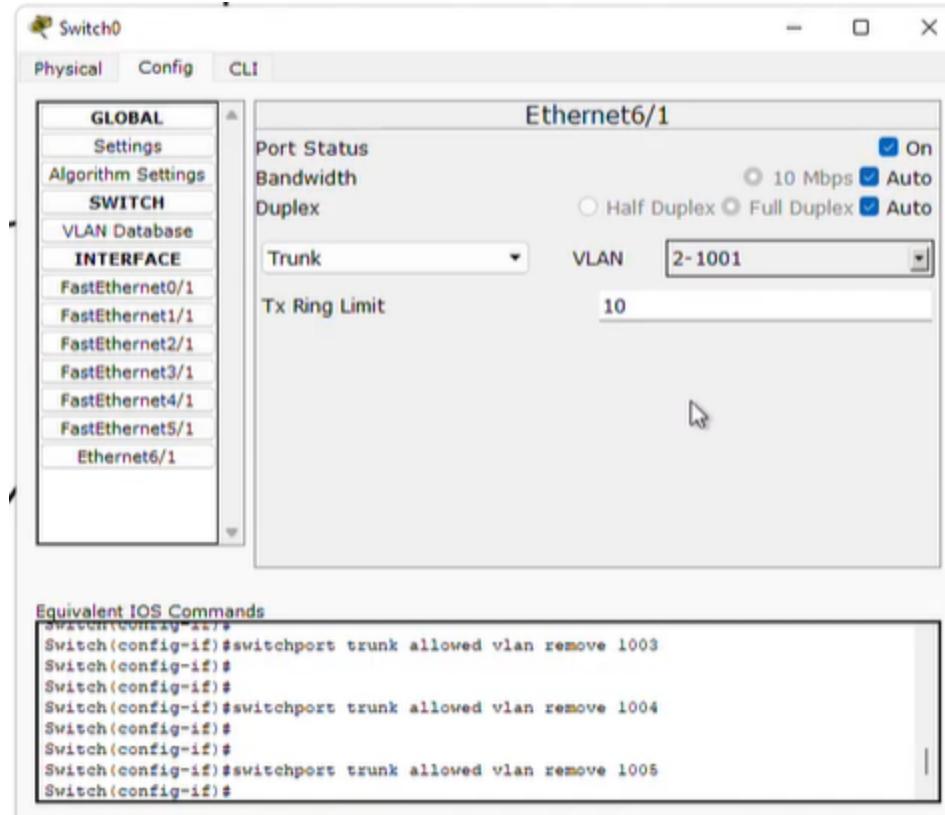


## Output:

```
Router1
Physical Config CLI
IOS Command Line Interface
documentation for configuring VTP/VLAN in config mode.

Router(vlan)#
*SYS-5-CONFIG_I: Configured from console by console
vlan 2 name NEWLAN
VLAN 2 modified:
  Name: NEWLAN
Router(vlan)#EXIT
APPLY completed.
Exiting...
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet 0/0.1
Router(config-subif)#
*LINK-5-CHANGED: Interface FastEthernet0/0.1, changed state to up
*LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.1, changed state
to up

Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 192.168.2.1 255.255.255.0
Router(config-subif)#no shut
Router(config-subif)#exit
Router(config)#exit
Router#
*SYS-5-CONFIG_I: Configured from console by console
```



Router1

Physical Config CLI

### IOS Command Line Interface

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 192.168.1.1
% Incomplete command.
Router(config-if)#ip address 192.168.1.1 255.255.255.0
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
exit
Router(config)#
Router(config)#exit
Router#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

Router(vlan)#
%SYS-5-CONFIG_I: Configured from console by console
vlan 2 name NEWLAN
VLAN 2 modified:
  Name: NEWLAN
Router(vlan)#EXIT
```

Copy Paste

PC0

Physical Config Desktop Custom Interface

### Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=3ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.3:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 0ms

PC>
```

## Program 12

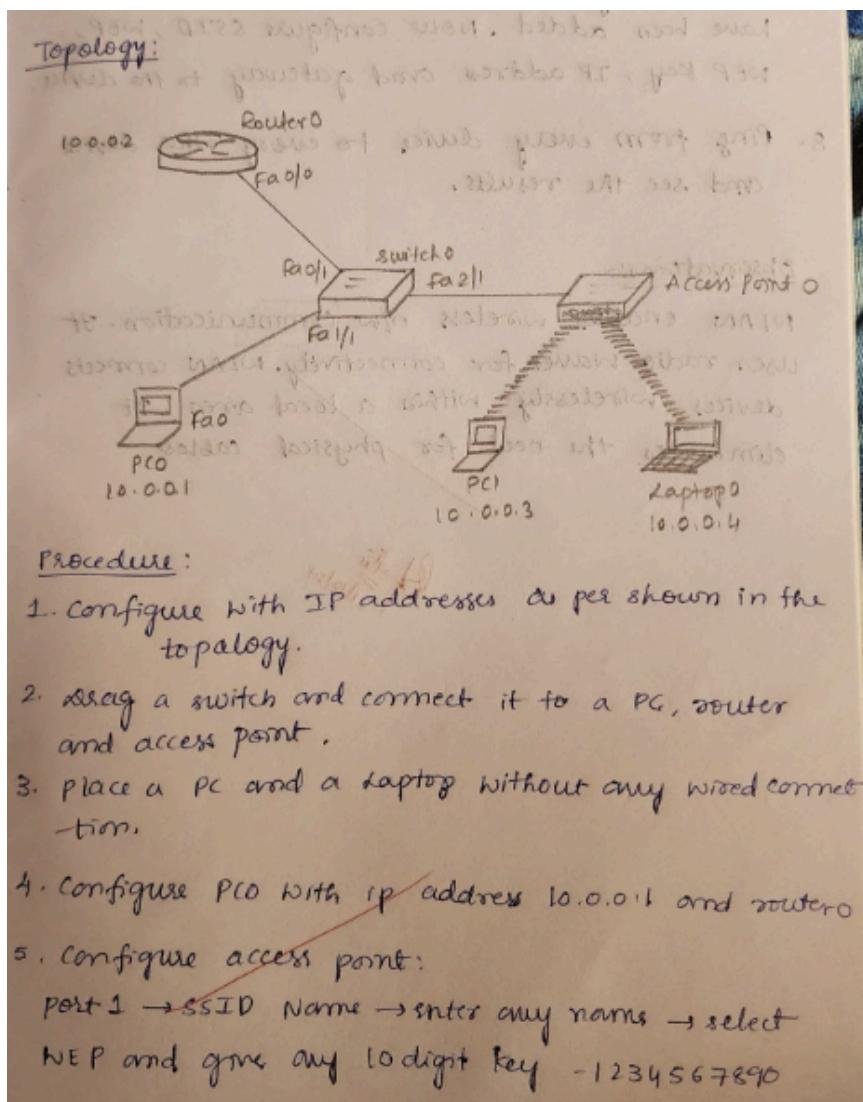
### Question:

To construct a WLAN and make the nodes communicate wirelessly

Question: To construct a WLAN and make the nodes communicate wirelessly.

Aim: Construct WLAN and make nodes communicate wirelessly.

### Observation writeup:

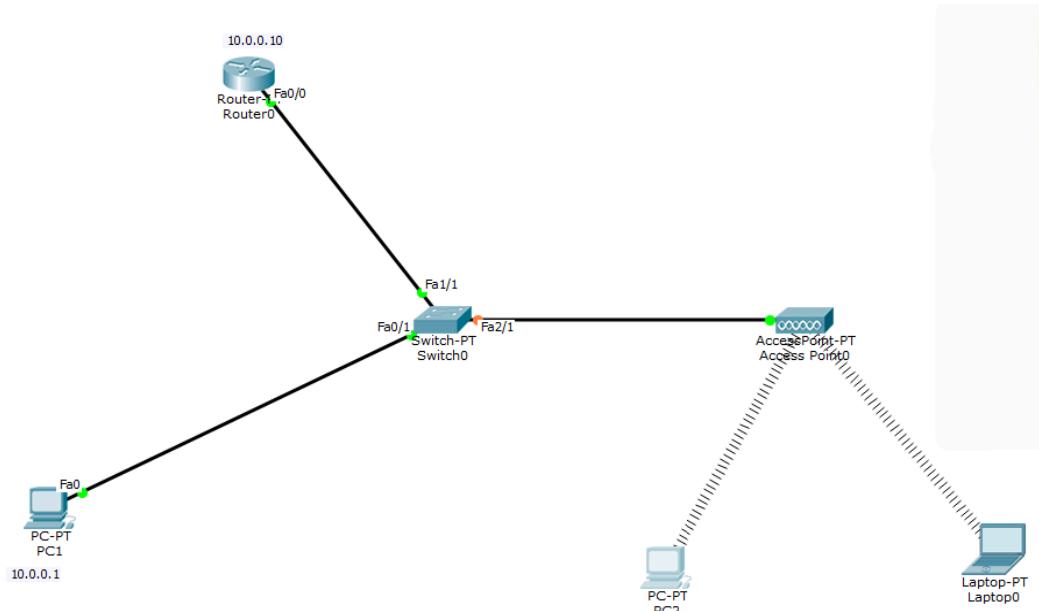


6. Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag WMP300N wireless interface to the empty port. Switch on the device.
7. In the config tab, a new wireless interface would have been added. Now configure SSID, WEP, WEP Key, IP address and gateway to the device.
8. Ping from every device to every other device and see the results.

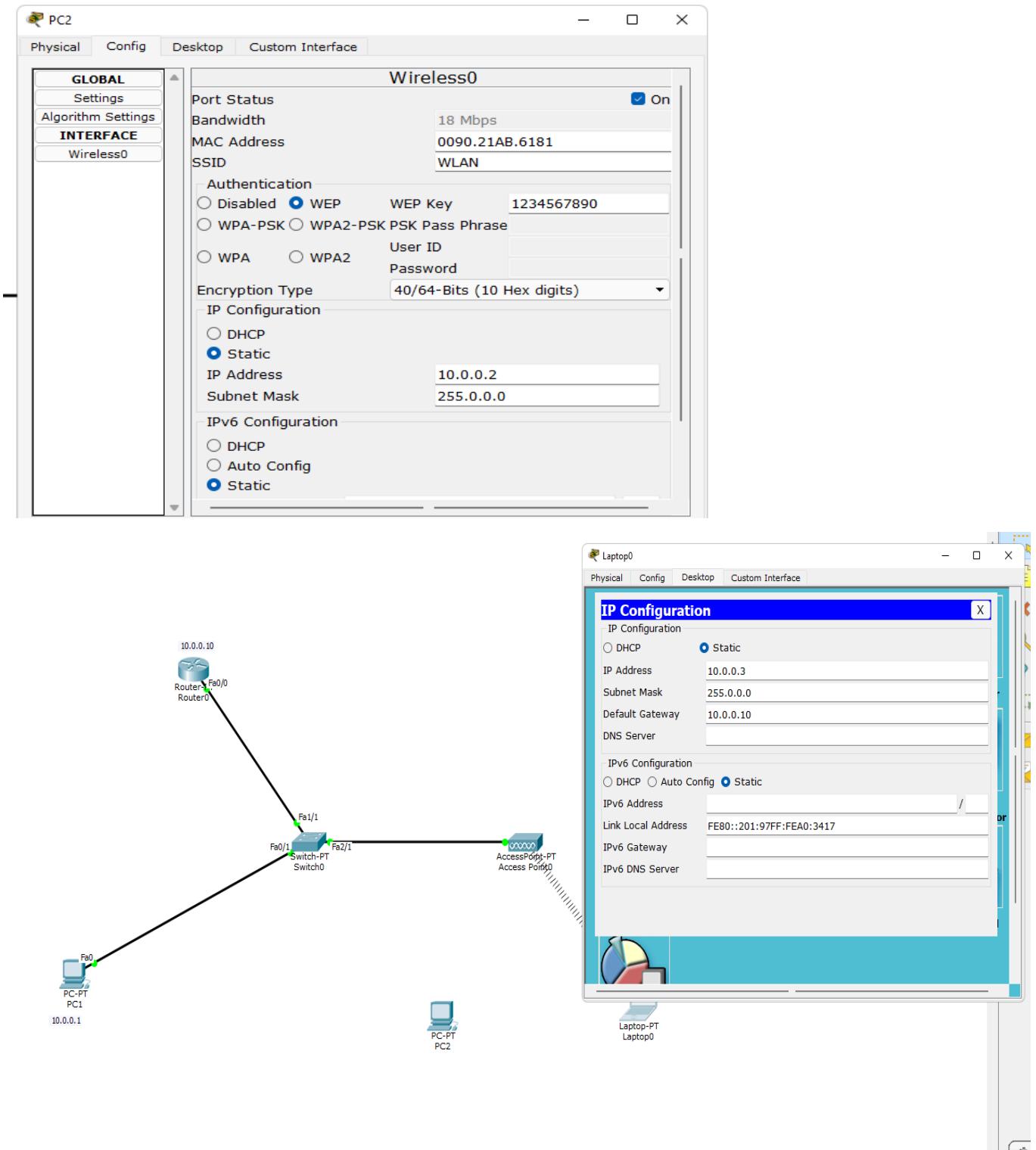
### Observations:-

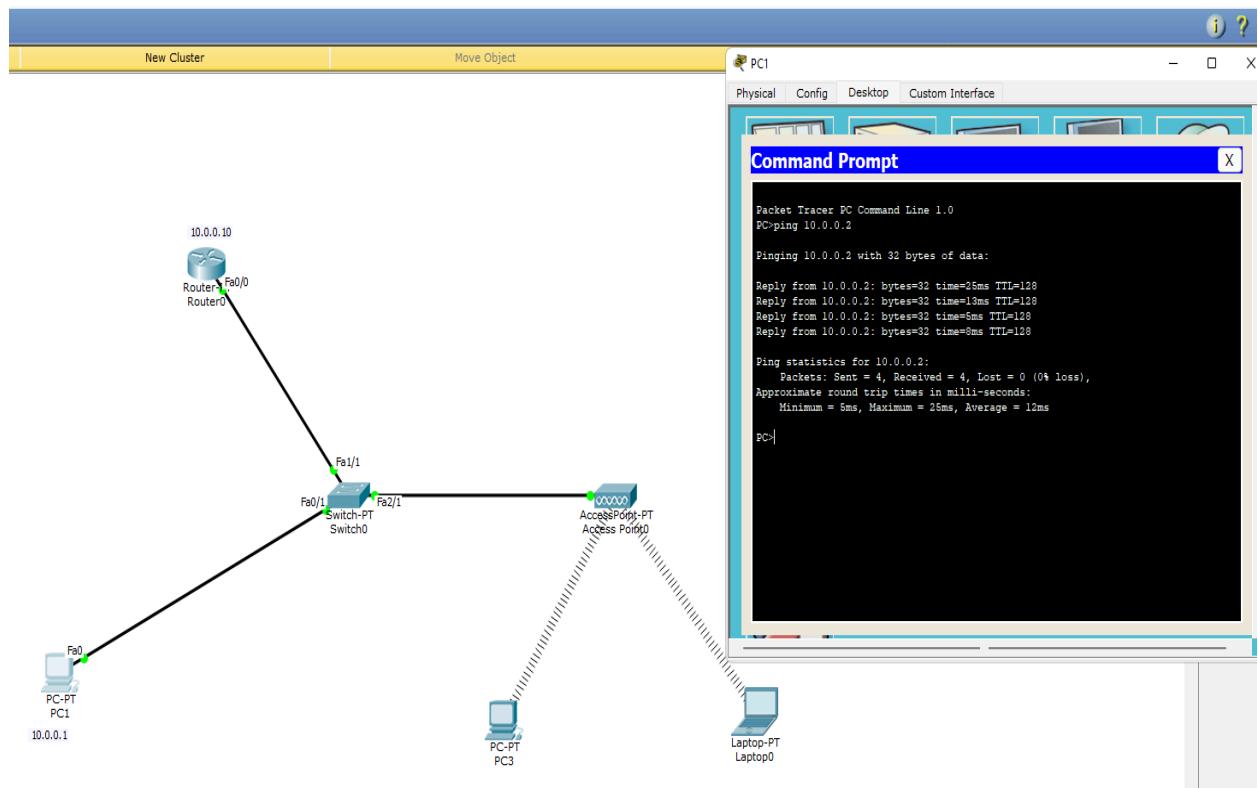
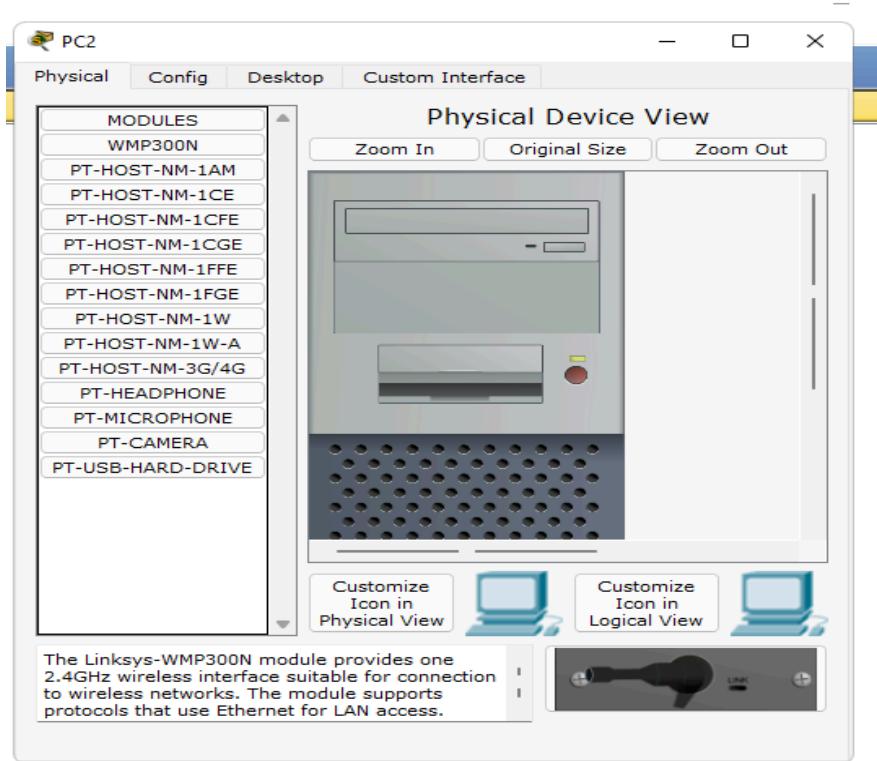
WLAN enables wireless n/w communication. It uses radio waves for connectivity. WLAN connects devices wirelessly within a local area. It eliminates the need for physical cables.

### Topology:



## Output:





## CYCLE 2

## Program 1:

## Question:

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

```

print ("Transmitted bitstream with CRC: ")
print (bitstream_with_CRC)

user_bitstream = input("Enter the received  

bitstream for verification: ").strip()

if not all (bit in "01" for bit in user_bitstream):
    print ("Invalid input. Please enter a valid  

binary bitstream.")

elif len(user_bitstream) < 16:
    print ("Invalid input. Received bitstream  

must include at least 16 bits for CRC.")

else:
    is_valid = Verify_CRC_bitstream(user_bitstream)

    if is_valid:
        print ("No errors detected! CRC valid")
    else:
        print ("Error detected! CRC invalid")

```

### OUTPUT:

Enter the original bitstream : 10100011110010100  
Transmitted bitstream with CRC: 10100011110010100  
10100011110010100

Enter the received bitstream for verification:

10100111100101001000

Error detected! CRC invalid.

Code:

```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    crc = init_crc
    for bit in bitstream:
        crc ^= int(bit) << 15
        for _ in range(8):
            if crc & 0x8000:
                crc = (crc << 1) ^ poly
            else:
                crc <<= 1
        crc &= 0xFFFF
    return crc

def append_crc_to_bitstream(bitstream: str) -> str:
    crc = crc_ccitt_16_bitstream(bitstream)
    crc_bits = f'{crc:016b}'
    return bitstream + crc_bits

def verify_crc_bitstream(bitstream_with_crc: str) -> bool:
    if len(bitstream_with_crc) < 16:
        return False
    data, received_crc = bitstream_with_crc[:-16], bitstream_with_crc[-16:]
    calculated_crc = crc_ccitt_16_bitstream(data)
    return calculated_crc == int(received_crc, 2)

if __name__ == "__main__":
    message_bits = input("Enter the original bitstream (e.g., 11010011101100): ").strip()
    if not all(bit in "01" for bit in message_bits):
        print("Invalid input. Please enter a binary bitstream (e.g., 11010011101100).")
    else:
        bitstream_with_crc = append_crc_to_bitstream(message_bits)
        print(f"Transmitted bitstream with CRC: {bitstream_with_crc}")
        user_bitstream = input("Enter the received bitstream for verification: ").strip()
        if not all(bit in "01" for bit in user_bitstream):
```

```
print("Invalid input. Please enter a valid binary bitstream.")  
elif len(user_bitstream) < 16:  
    print("Invalid input. Received bitstream must include at least 16 bits for CRC.")  
else:  
    is_valid = verify_crc_bitstream(user_bitstream)  
    if is_valid:  
        print("No errors detected. CRC valid.")  
    else:  
        print("Error detected! CRC invalid.")
```

Output:

```
PS C:\Users\OneDrive\Desktop\code> python crc-ccitt.py  
Enter the original bitstream (e.g., 11010011101100): 1010001111  
Transmitted bitstream with CRC: 1010001111100101001111110  
Enter the received bitstream for verification: 1010001111100101001000  
Error detected! CRC invalid.
```

## Program 2:

Question:

Write a program for congestion control using Leaky bucket algorithm.

1/1/25      PROGRAMS

(1) Write a program for congestion control using leaky bucket algorithm.

```

storage = 0
noofqueries = int(input("Enter no. of queries: "))
bucketsize = int(input("Enter bucket size: "))
inputpktsize = int(input("Enter input packet size: "))
outputpktsize = int(input("Enter output packet size: "))

for i in range(0, noofqueries):
    sizeleft = bucketsize - storage
    if inputpktsize <= sizeleft:
        storage += inputpktsize
    else:
        print("Packet loss = ", inputpktsize)
        print(f"Bucket size = {storage} out of bucket size {bucketsize}")
        storage -= outputpktsize + unmetd_pkts

```

OUTPUT:-

```

Enter no of queries : 10
Enter bucket size: 5
Enter input packet size: 4
Enter output packet size: 6
Bucket size = 4 out of bucket size = 5
Bucket size = 2 out of bucket size = 5
Bucket size = 0 out of bucket size = 5
Bucket size = -2 out of bucket size = 5
Bucket size = -4 out of bucket size = 5
Bucket size = -6 out of bucket size = 5
Bucket size = -8 out of bucket size = 5
Bucket size = -10 out of bucket size = 5
Bucket size = -12 out of bucket size = 5
Bucket size = -14 out of bucket size = 5

```

Code:

```
storage=0
noofqueries=int(input("Enter no of queries:"))
bucketsize=int(input("Enter bucket size:"))
inputpktsize=int(input("Enter input packet size:"))
outputpktsize=int(input("Enter output packet size:"))
for i in range(0,noofqueries):
    sizeleft=bucketsize-storage
    if inputpktsize<=sizeleft:
        storage+=inputpktsize
    else:
        print("Packet loss=", inputpktsize)
    print(f"Bucket size={storage} out of bucket size={bucketsize}")
    storage-=outputpktsize
```

Output:

```
PS C:\Users\DELL\OneDrive\Desktop\code> python leakybucketalgorithm.py
Enter no of queries:10
Enter bucket size:5
Enter input packet size:4
Enter output packet size:6
Bucket size=4 out of bucket size=5
Bucket size=2 out of bucket size=5
Bucket size=0 out of bucket size=5
Bucket size=-2 out of bucket size=5
Bucket size=-4 out of bucket size=5
Bucket size=-6 out of bucket size=5
Bucket size=-14 out of bucket size=5
```

### Program 3:

Question:

**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.**

(3) 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.

tcpclient.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('In From server: \n')
print(filecontents)
clientSocket.close()
```

tcpserver.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
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('In sent contents of ' + sentence)
```

file.close()  
 connectionSocket.close()

OUTPUT:-

The server is ready to receive }  
 sent contents of serverTCP.py } server side  
 The server is ready to receive  
 Enter file name : tcpserver.py } client side.  
 Reply from server

Code:

tcpserver.py file:

```

from socket import *
serverName="127.0.0.1"
serverPort = 14000
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()
  
```

tcpclient.py file:

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
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()
```

Output:

```
PS C:\Users\DELL\OneDrive\Desktop\code> python tcpserver.py
The server is ready to receive
```

```
Sent contents of example.txt
The server is ready to receive
```

```
PS C:\Users\DELL\OneDrive\Desktop\code> python tcpclient.py
```

```
Enter file name: example.txt
```

```
From Server:
```

```
Hello, this is a sample file.
It is used for testing the TCP server.
```

## Program 4:

Question:

**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.**

<4> 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.

udpcient.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(sentence.encode("utf-8"),
                    (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom(2048)
print('In Reply from server : \n')
print(fileContents.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

udpserver.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")
```

```

serverSocket.sendto (bytes (con, "utf-8"), clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)
file.close()

OUTPUT :-
The server is ready to receive } server side
sent contents of serverudp.py } client side
server is ready to receive

Enter file name: serverudp.py
Reply from server: (47, 111, 100, 117, 100)

```

Code:

udpserver.py file:

```

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)
    # for i in sentence:
    #     print (str(i), end = " ")
    file.close()

```

udpclient.py file:

```
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"))

# for i in filecontents:
#     print(str(i), end = '')
clientSocket.close()
clientSocket.close()
```

Output:

```
PS C:\Users\DELL\OneDrive\Desktop\code> python udpserver.py
The server is ready to receive
```

```
Sent contents of example.txt
```

```
█
```

```
PS C:\Users\DELL\OneDrive\Desktop\code> python udpclient.py
```

```
Enter file name: example.txt
```

```
Reply from Server:
```

```
Hello, this is a sample file.
It is used for testing the TCP server.
```

## 5. Tool exploration - Wireshark:

### (5) Tool exploration - Wireshark

Wireshark is a powerful and widely used network protocol analysis tool. 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 protocols.

#### key features:

1. **Packet capture:** Captures live network traffic from various interfaces (ex: ethernet, wifi).
2. **Protocol Analysis:** supports hundreds of protocols (ex: TCP, UDP, HTTP, FTP).
3. **Filtering:** offers powerful filters to isolate specific packets or traffic types.
4. **Visualization:** displays packet details with hierarchical layers (Ethernet, IP, TCP/UDP).

#### Use cases of Wireshark:

1. **Network Troubleshooting:**
  - \* Diagnosing slow network speeds.
  - \* Identifying bottlenecks or misconfigurations.
2. **Security Analysis:**
  - \* Detecting malicious traffic or intrusions.
3. **Protocol study:**
  - \* Understanding packet structures and communication flow.

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.

*Off the record*