

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

COMPUTER NETWORKS LAB

Submitted by

POORVIKA S K (1BM22CS412)

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 JUN-2023 to SEP-2023

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 LAB**" carried out by **POORVIKA S K (1BM22CS412)**, 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 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **COMPUTER NETWORKS - (22CS4PCCON)** work prescribed for the said degree.

Name of the Lab-Incharge: Sowmya T

Designation: Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak

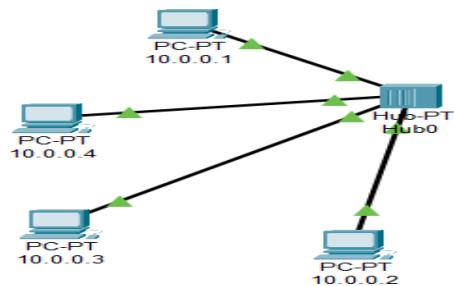
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

Sl. No.	Experiment Title	Page No.
	CYCLE 1	
1	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message	4-14
2	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	15-17
3	Configure default route, static route to the Router	18-28
4	Configure DHCP within a LAN and outside LAN	29-31
5	Configure RIP routing Protocol in Routers	32-39
6	Configure OSPF routing protocol	40-42
7	Demonstrate the TTL/ Life of a Packet	43-44
8	Configure Web Server, DNS within a LAN	45-46
9	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	47-51
10	To understand the operation of TELNET by accessing the router in server room from a PC in IT office	52-56
11	To construct a VLAN and make the PC's communicate among a VLAN	57-59
12	To construct a WLAN and make the nodes communicate wirelessly	60-62
	CYCLE 2	
13	Write a program for error detecting code using CRC CCITT (16-bits)	63-64
14	Write a program for congestion control using Leaky bucket algorithm	65-66
15	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	67-69
16	Using UDP sockets, write a client-server program to make clients sending the file name and the server to send back the contents of the requested file if present	70-71

LAB PROGRAM-01

Create a topology consisting of three or more device connected with help of a hub and a switch

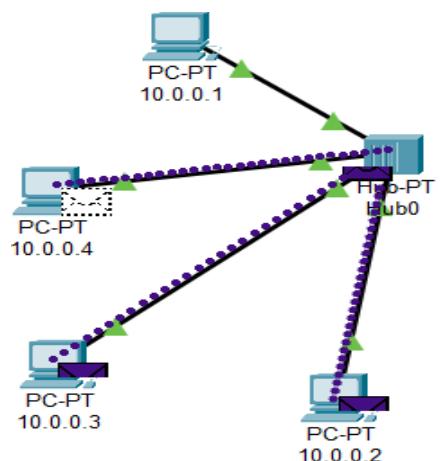
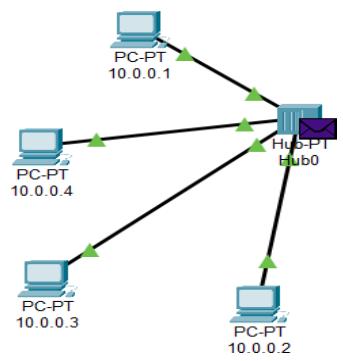


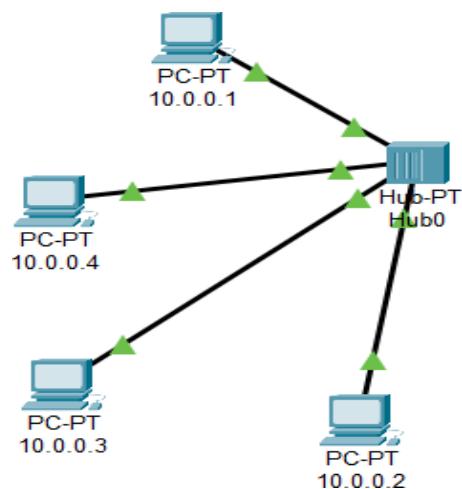
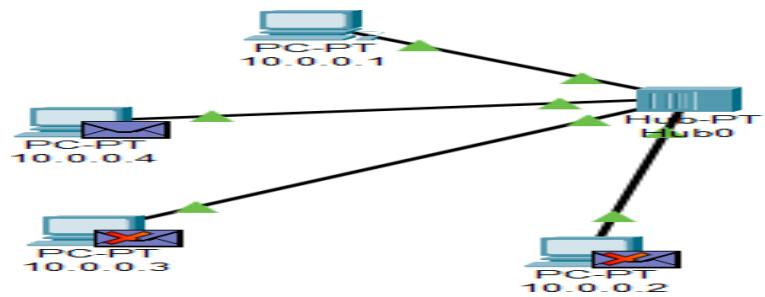
```
C:\>ping 10.0.0.2

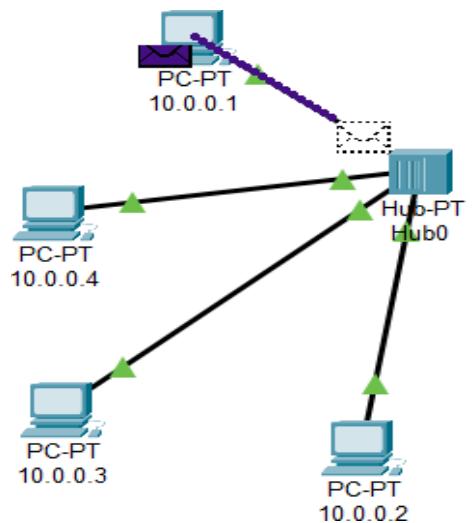
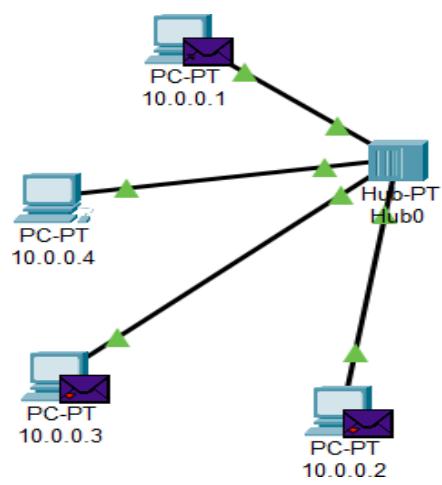
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=21ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

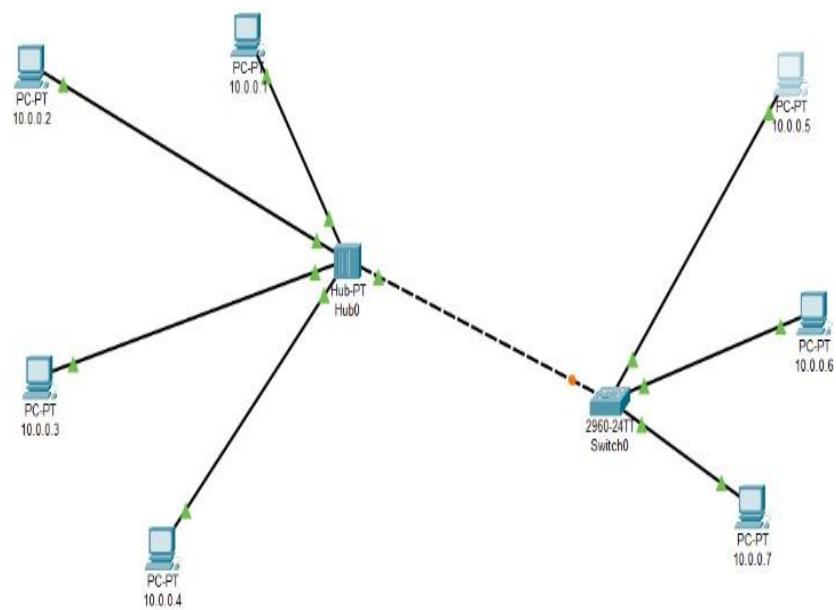
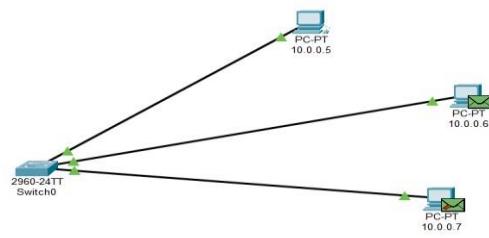
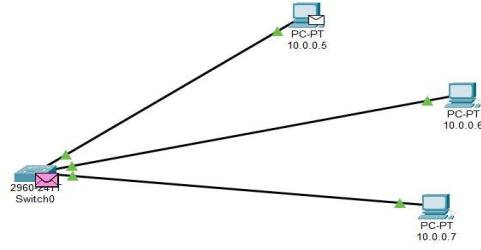
Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 21ms, Average = 5ms

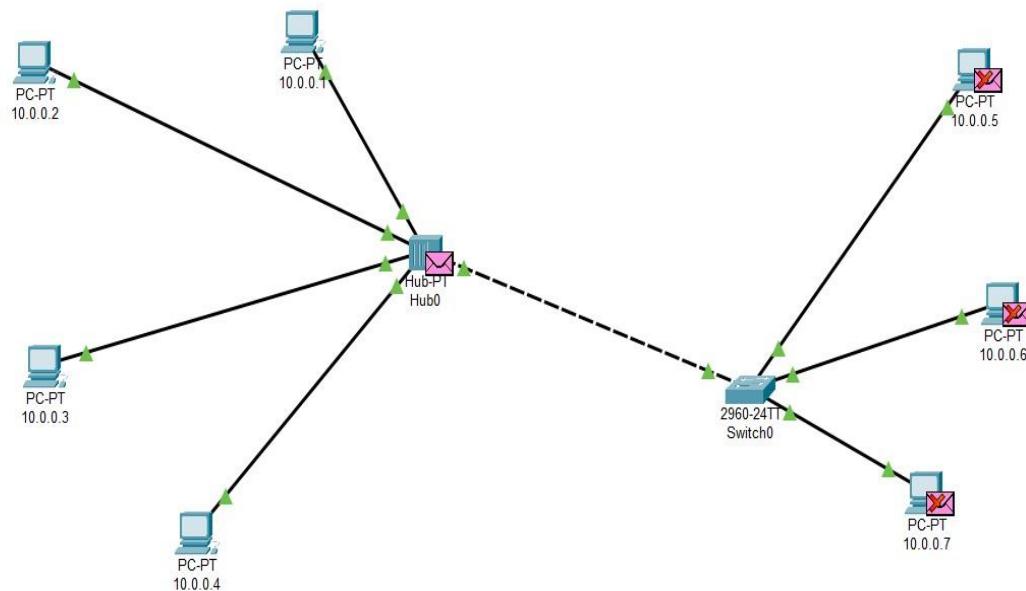
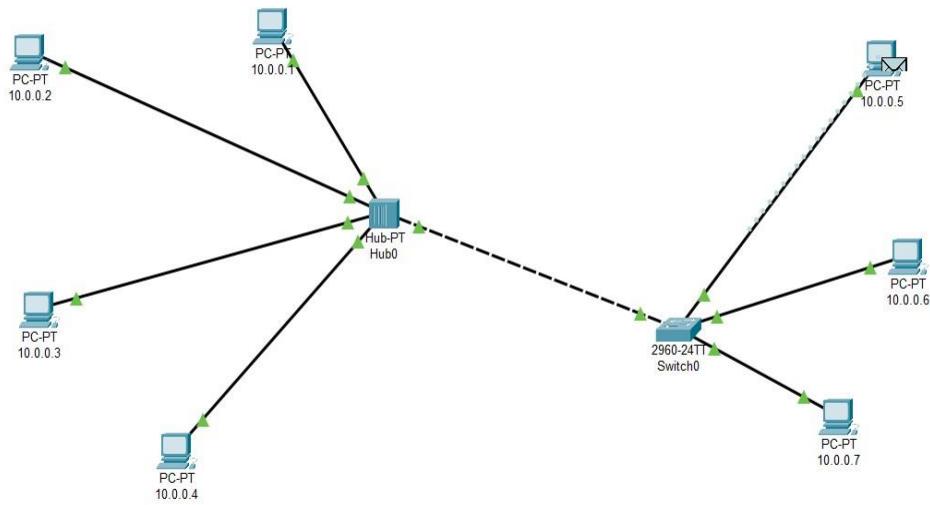
C:\>
```

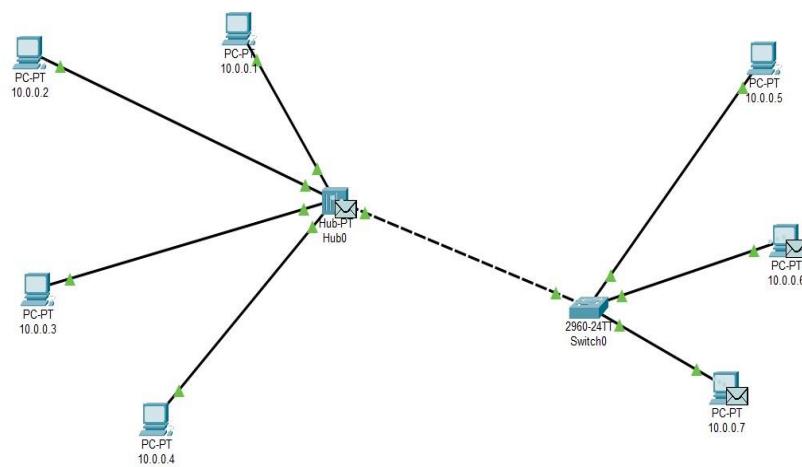
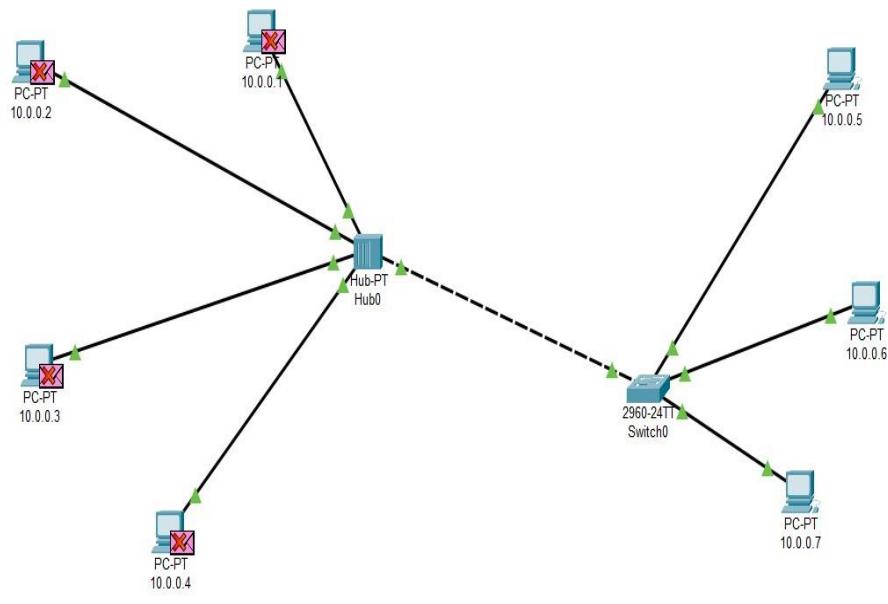












LAB - 01

LAN :- A series of computers linked together to form a network in a circumscribed location.

WAN :- A computer network that connects smaller networks that is not tied to a single location.

Ethernet :- A system for connecting a no of computers systems to form a LAN with protocols to control the passing of information b/w systems.

IP address :- A unique string of characters that identify each computer using the internet protocol to communicate over a network.

Hub :- Hub is a node that broadcasts data to every computer or Ethernet based device that is connected to it.

Switch :- It connects devices in a network to each other enabling them to talk by exchanging data packets.

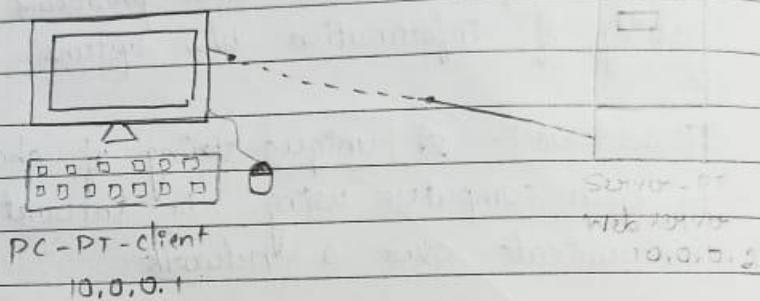
Server :- It connects is a computer program or device that provides a service to another computer program and its user known as client.

End device :- Are either the source or destination of data transmitted over the network.

Node :- The connection point among network devices such as routers, printers or switches that can receive and send data from one end point to another.

Packet tracer :-

1. Add PC and server from end devices
2. Connect them with copper cross over
3. Set PC ethernet IP address as 10.0.0.1 and DNS server address as 10.0.0.2
4. Set server ethernet IP address as 10.0.0.1
5. Services \rightarrow DNS \rightarrow Name: www first.com
Address : 10.0.0.2
Add



Observation :-

Click on pc in real time \rightarrow desktop \rightarrow Command

prompt command :

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 = 128

Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 128

Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 128

Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.2 :

packets : sent = 4, received = 4, loose = 0 (0% loss)

Approximate round trip times in milliseconds

Minimum = 0ms, Maximum = 0ms, Average = 0ms

Lab - 01

Create a topology consisting of three or more device connected with help of a hub and help of a switch.

Steps involved :-

Step 1 : Drag and drop 3 generic PC's and a generic switch. Connect 3 pc's as peripherals to the switch after setting the IP address as 10.0.0.1, 10.0.0.2 and 10.0.0.3 for PC1, PC2, PC3 respectively and connect them.

Step 2 : Drag and drop 3 more generic pc's and a generic hub. Set the IP address of PC4, PC5 and PC6 as 10.0.0.4, 10.0.0.5 and 10.0.0.6 respectively connect all the three 3pc's. to the hub.

Scenario 1 :

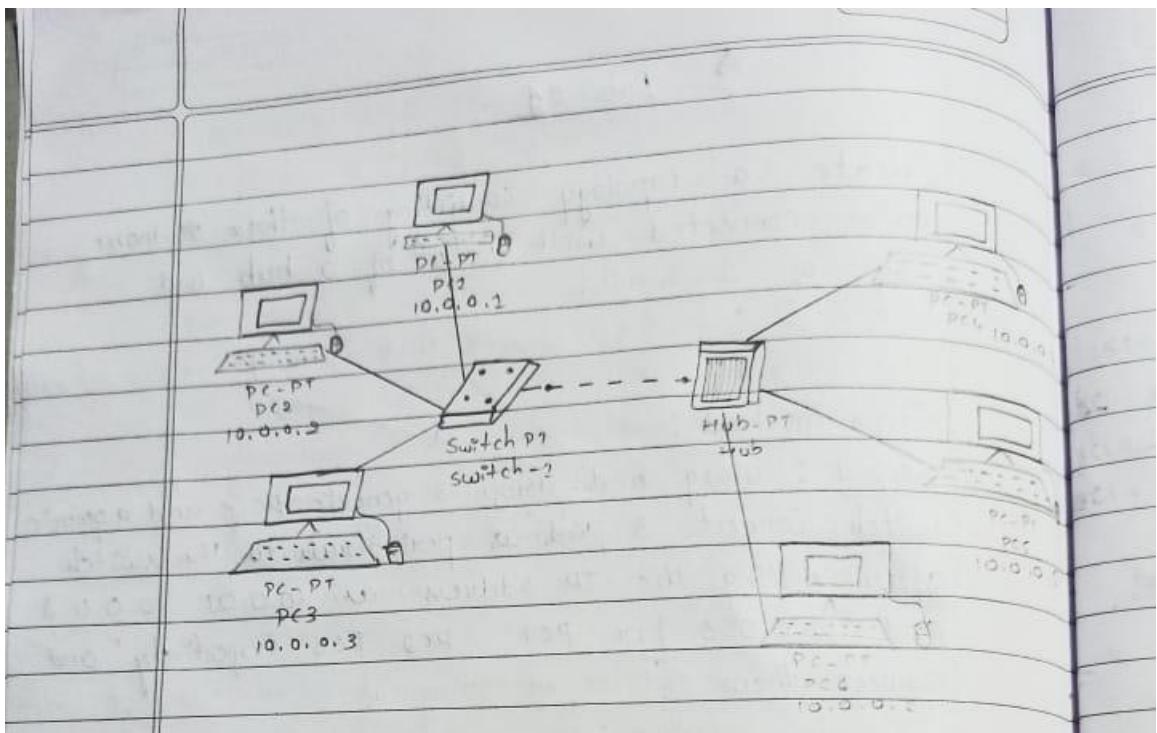
Step 3 : Turn on the switch and send a PDU from PC1 (10.0.0.1) to PC2 (10.0.0.2) via switch.

Scenario 2 :

Step 4 : Send a PDU from PC4 with ip address 10.0.0.4 to PC6 with IP address 10.0.0.6 via hub. Hub will send PDU to every pc connected to it. PC6 will acknowledge and receive it.

Scenario 3 :

Step 5 :- Connect switch and hub. Send a PDU from PC1 with IP address 10.0.0.1 to PC6 with IP address 10.0.0.6 via switch and hub.



Command prompt

PC> ping 10.0.0.6

pinging 10.0.0.6 with 32 bytes of data:

Reply from 10.0.0.6 : bytes = 32 time = 6ms TTL=128

Reply from 10.0.0.6 : bytes = 32 time = 6ms TTL=128

Reply from 10.0.0.6 : bytes = 32 time = 6ms TTL=128

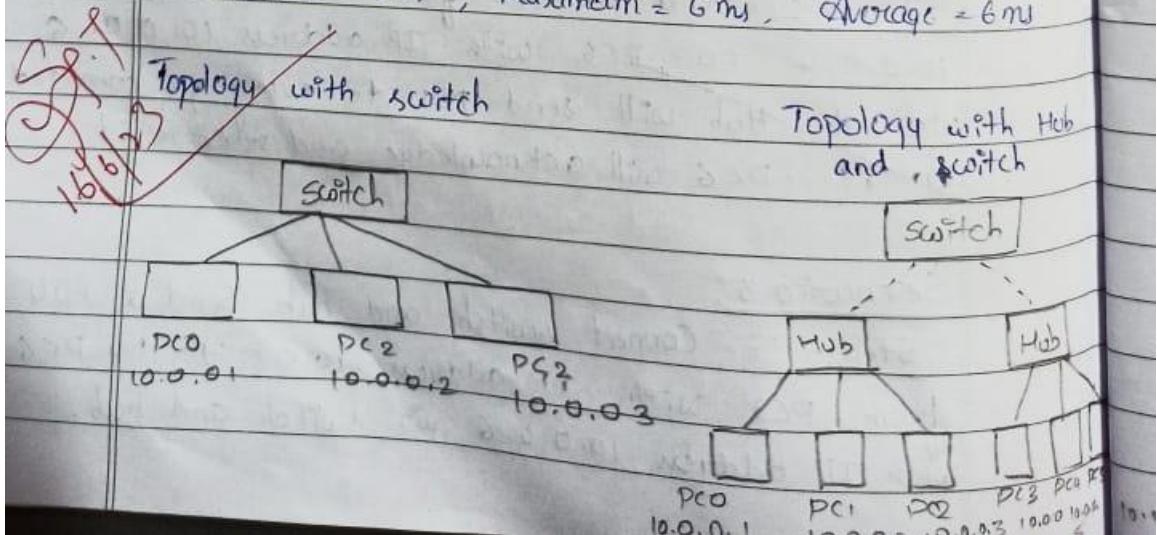
Reply from 10.0.0.6 : bytes = 32 time = 6ms TTL=128

ping statistics for 10.0.0.6

packets : Sent = 4, Received = 4, lost = 0 (0% loss)

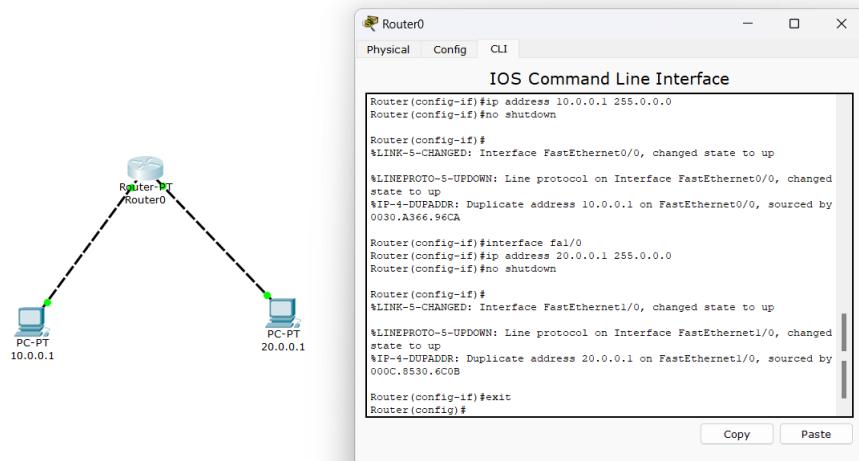
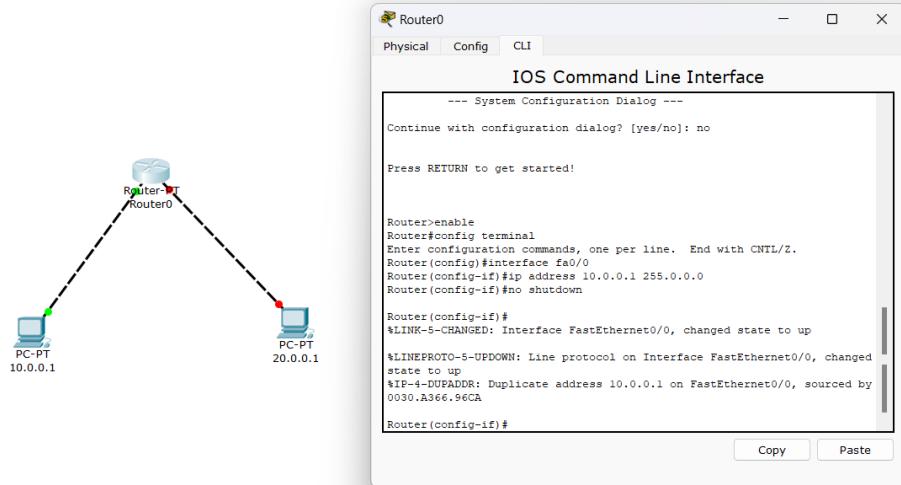
approximate round trip times in milliseconds:

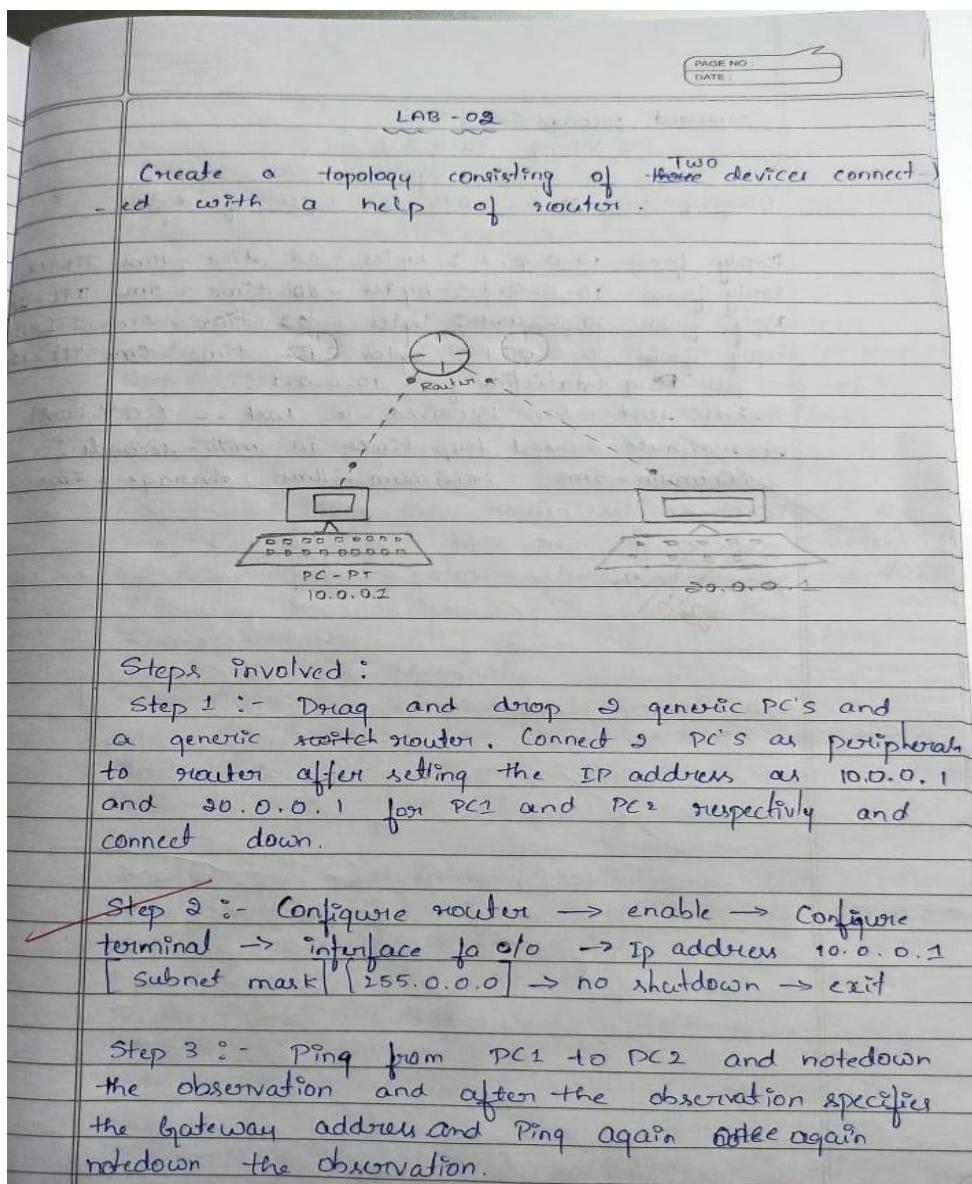
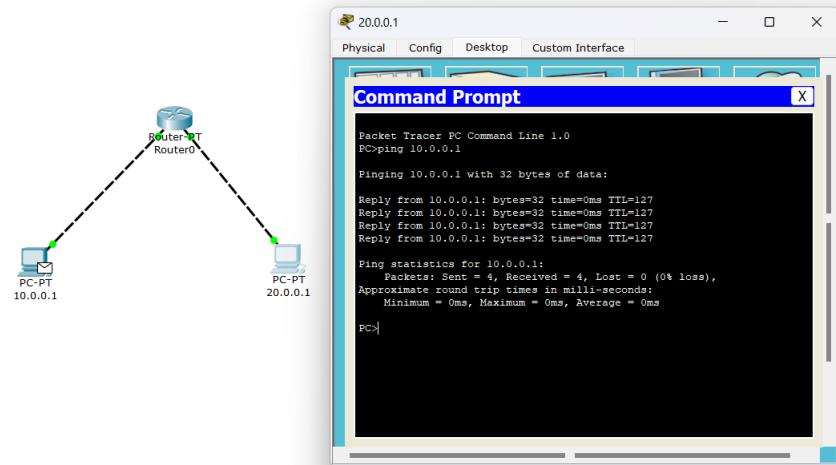
Minimum = 5ms, Maximum = 6ms, Average = 6ms



LAB PROGRAM-02

Create a topology consisting of two devices connected with a help of router.





Command prompt:-

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 = 11ms TTL=128

Reply from 10.0.0.1: bytes = 32 time = 5ms TTL=128

Reply from 10.0.0.1: bytes = 32 time = 7ms TTL=128

Reply from 10.0.0.1: bytes = 32 time = 6ms TTL=128

Ping statistics for 10.0.0.1:

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

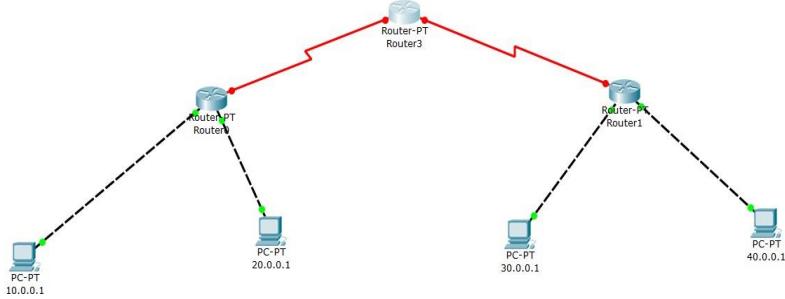
Approximate round trip times in milli-seconds:

Minimum = 5ms, Maximum = 11ms, Average = 7ms

S.S.
23/6/23

LAB PROGRAM-03

Configure default route, static route to router.



```
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=2ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
    PC>
```

Packet Tracer PC Command Line 1.0

PC>ping 10.0.0.1

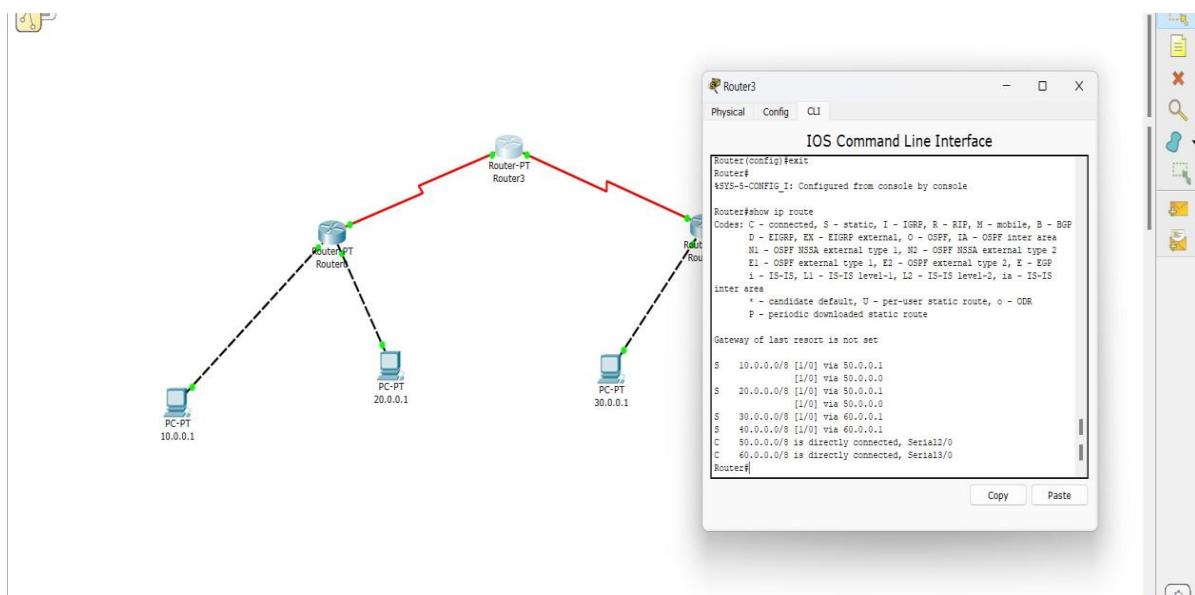
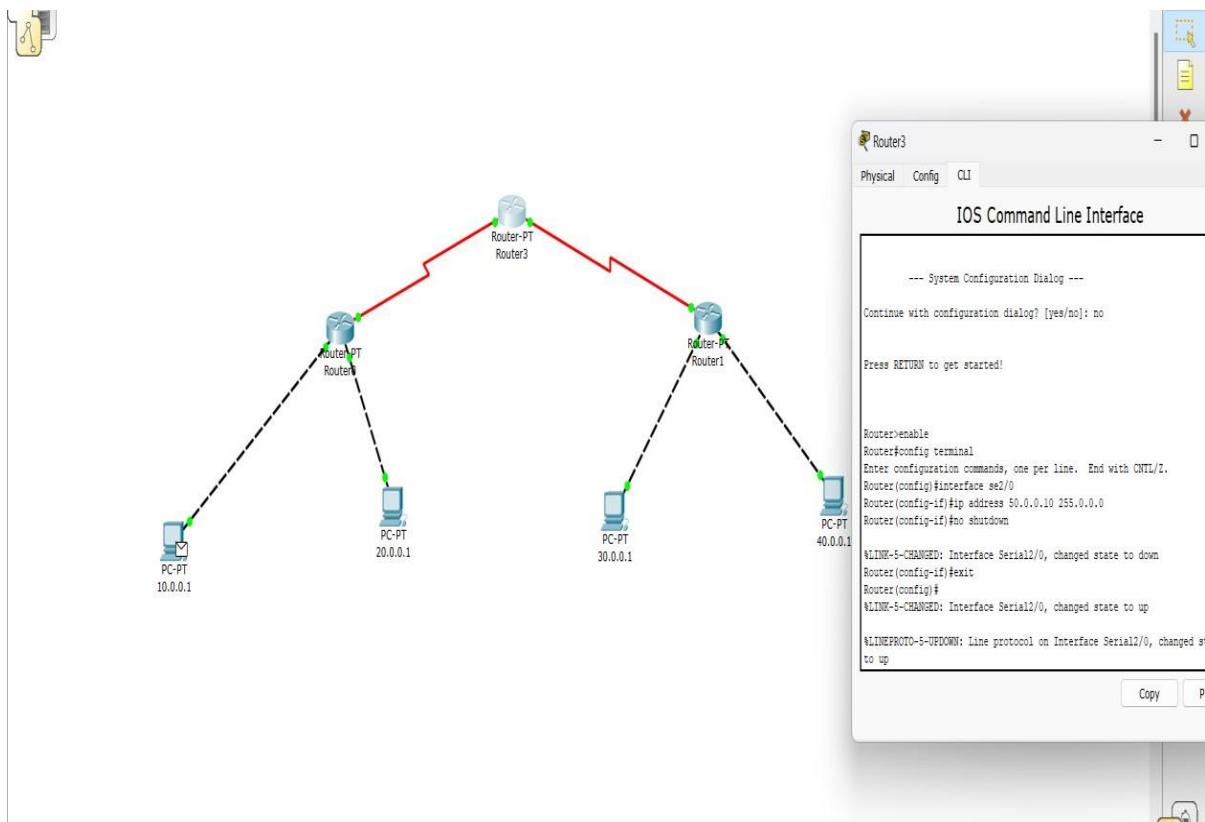
Pinging 10.0.0.1 with 32 bytes of data:

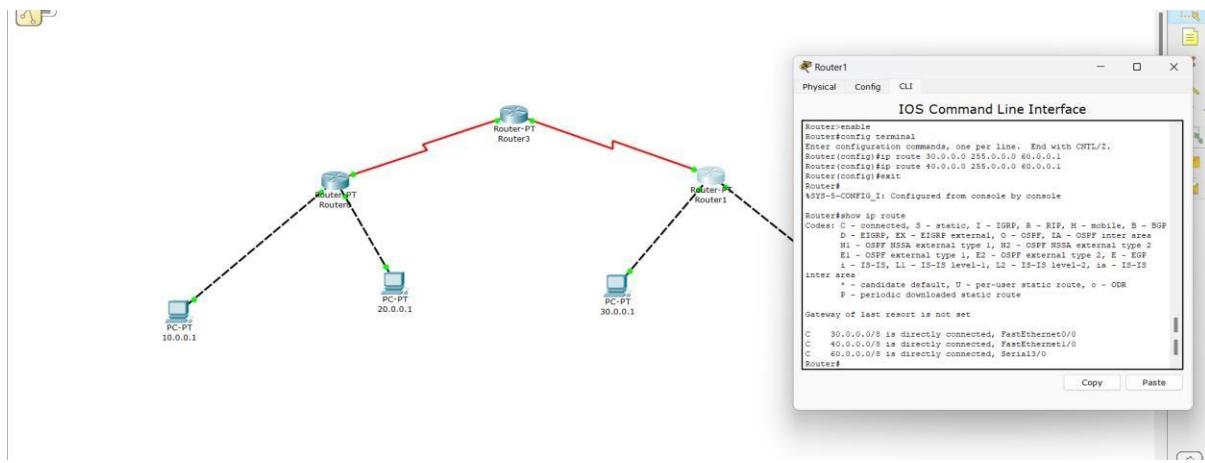
Reply from 40.0.0.10: Destination host unreachable.

Ping statistics for 10.0.0.1:

 Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>





20.0.0.1

Physical Config Desktop Custom Interface

Command Prompt

```

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 20.0.0.10: Destination host unreachable.
Reply from 20.0.0.10: Destination host unreachable.
Reply from 20.0.0.10: Destination host unreachable.
Request timed out.

Ping statistics for 40.0.0.1:
  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:

Reply from 20.0.0.1: bytes=32 time=3ms TTL=128
Reply from 20.0.0.1: bytes=32 time=0ms TTL=128
Reply from 20.0.0.1: bytes=32 time=0ms TTL=128
Reply from 20.0.0.1: bytes=32 time=11ms TTL=128

Ping statistics for 20.0.0.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 11ms, Average = 3ms
PC>

```

LAB - 03

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

Steps involved:-

Step 1 : Drag and drop 2 pc's and a generic router set the IP addresses of 2 pc's as 10.0.0.1 and 20.0.0.1 respectively. Set the gateway of 2 pc's as 10.0.0.3 and 20.0.0.3 respectively and connect them to the router.

Step 2 : Configure the router settings to connect the two networks (i.e: two pc's of different n/w) by using following steps after making the connections

Router > enable

Router # config terminal

Router (config) # interface fastEthernet 0/0

Router (config-if) # ip address 10.0.0.3 255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # exit

Router (config) # interface fastethernet 1/0

Router (config-if) # ip address 20.0.0.3 255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # no exit

Router (config) # exit

Router #

Step 3: Send a simple PDU from PC0 with IP address 10.0.0.1 to PC1 with IP address 30.0.0.1 and confirm how many packets sent by using ping command.

Step 4: Similarly, connect two more PC's with routers and configure by following above mentioned steps. Introduce one more router and connect it to the existing two routers of different network and configure it.

Step 5: Now, if you ping from the PC with IP address 10.0.0.1 as > ping 40.0.0.1 the response will be destination unreachable. Although, it seems there's a connection b/w these two PC's indirect via routers, but every router may not have information regarding every network present in the topology so these PC's cannot communicate. To eliminate this, we should use static routing to teach every router manually.

Step 6: We can do static routing for router by the following steps

Router# config t

```
Router (config)# ip route 10.0.0.0 255.0.0.0 50.0.0  
Router (config)# ip route 30.0.0.0 255.0.0.0 50.0.0  
Router (config)# ip route 30.0.0.0 255.0.0.0 60.0.0  
Router (config)# ip route 40.0.0.0 255.0.0.0 60.0.0  
Router (config) # exit
```

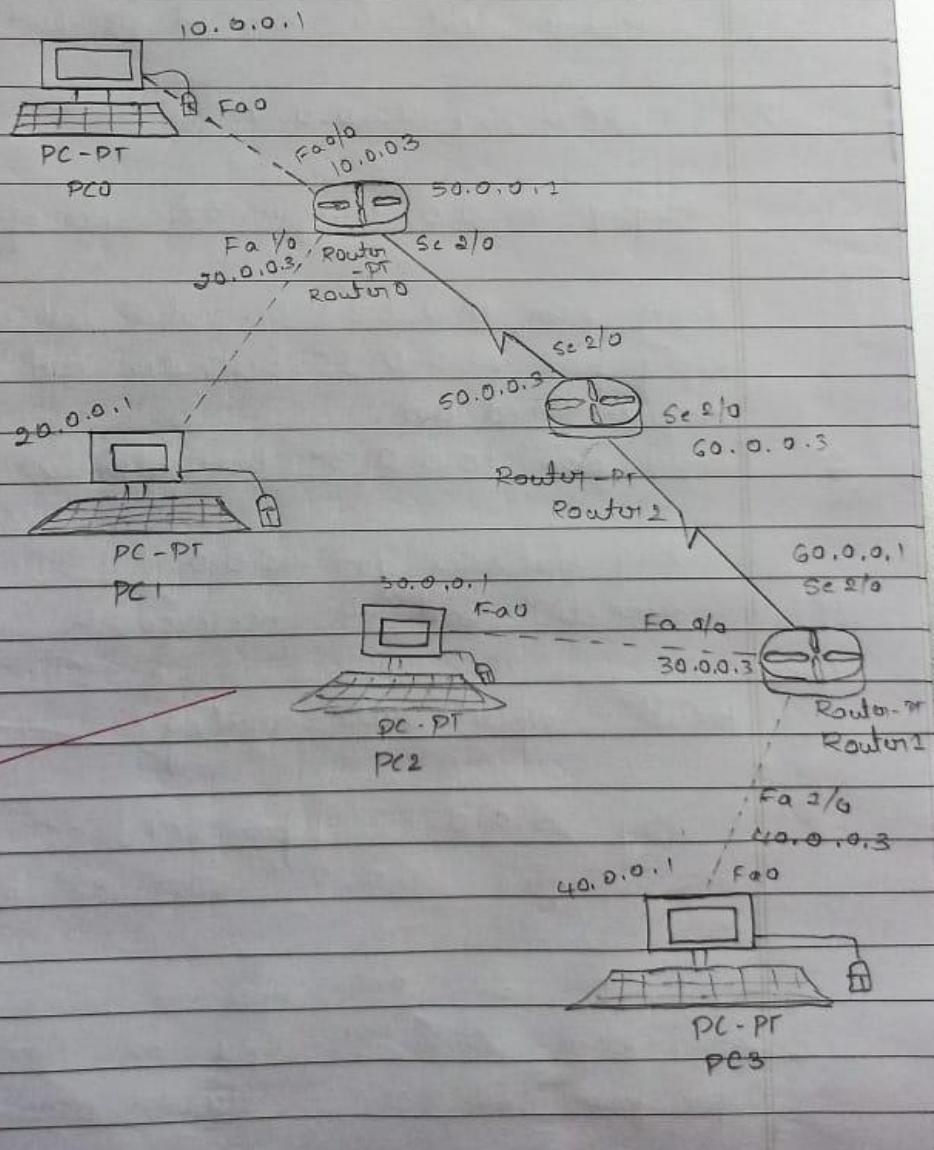
Router#

Step 7: We can view all the networks connected to a router as follows:

Router # show ip routes

Codes : C - connected , S - static

S	10.0.0.0/8	[1/0]	Via	50.0.0.1
S	20.0.0.0/8	[1/0]	Via	50.0.0.1
S	30.0.0.0/8	[1/0]	Via	60.0.0.1
S	40.0.0.0/8	[1/0]	Via	60.0.0.1
C	50.0.0.0/8	is directly connected, Serial 1/0		
C	60.0.0.0/8	is directly connected, Serial 1/0		



1. Command prompt :-

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 = 0ms TTL = 127

Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127

Ping statistics for 20.0.0.1

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

2. PC > Ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 10.0.0.3 : Destination host unreachable.

Reply from 10.0.0.3 : Destination host unreachable.

Request timed out.

Reply from 10.0.0.3 : Destination host unreachable.

Ping statistics for 40.0.0.1

Packet: Sent = 4, Received = 0, Lost = 4 (100%)

Result:- After static routing

1. Ping 20.0.0.1 [from PC1]

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 127

Reply from 90.0.0.1 bytes = 32 time = 0ms TTL = 127

Ping statistics for 90.0.0.1

Packets : Sent = 4, Received = 4, Lost = 0 (0.00%)

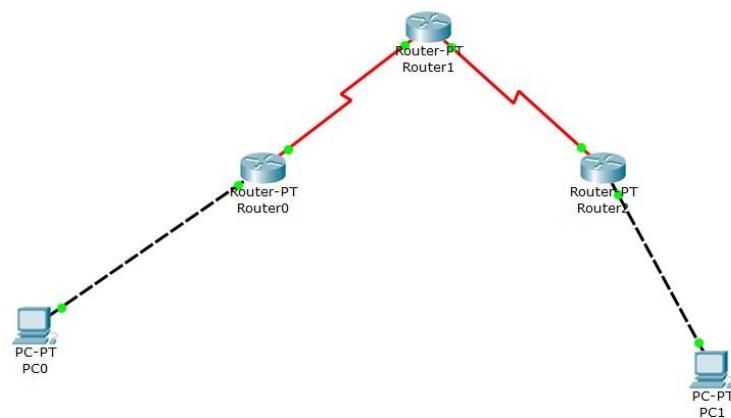
~~Approximate round trip times in milli-seconds~~

Minimum = 0ms, Maximum = 1ms, Average = 0ms

6.1
20/6/23

LAB PROGRAM-03

Default routing:-



Router1

Physical Config CLI

IOS Command Line Interface

```
Press RETURN to get started!

Router>enable
Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface se 2/0
Router(config-if)#ip address 30.0.0.10 255.0.0.0
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)#interface se 3/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shutdown

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

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

Router(config)#ip route 0.0.0.0 255.0.0.0 30.0.0.0
Router(config)#ip route 0.0.0.0 255.0.0.0 40.0.0.0
Router(config)#
Router#*ST1-5-CONFIS_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 30.0.0.0 to network 0.0.0.0

* 0.0.0.0/8 is subnetted, 1 subnets
S* 0.0.0.0 [1/0] via 40.0.0.0
C 30.0.0.0/8 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, Serial3/0
Router#
```

Copy Paste

Search ENG IN 2322 18-07-2023

Router2

Physical Config CLI

IOS Command Line Interface

Press RETURN to get started!

```

Router>enable
Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fa 0/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
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
exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface se 3/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shutdown

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

Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

Router(config)#ip route 0.0.0.0 0.0.0.0 40.0.0.0
Router(config)#
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 40.0.0.0 to network 0.0.0.0

C 20.0.0.0/8 is directly connected, FastEthernet0/0
C 40.0.0.0/8 is directly connected, Serial3/0
S* 0.0.0.0/0 [1/0] via 40.0.0.0
Router#
```

[Copy](#) [Paste](#)

ENG IN 23:21 18-07-2023

PC1

Physical Config Desktop Custom Interface

Command Prompt

```

Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

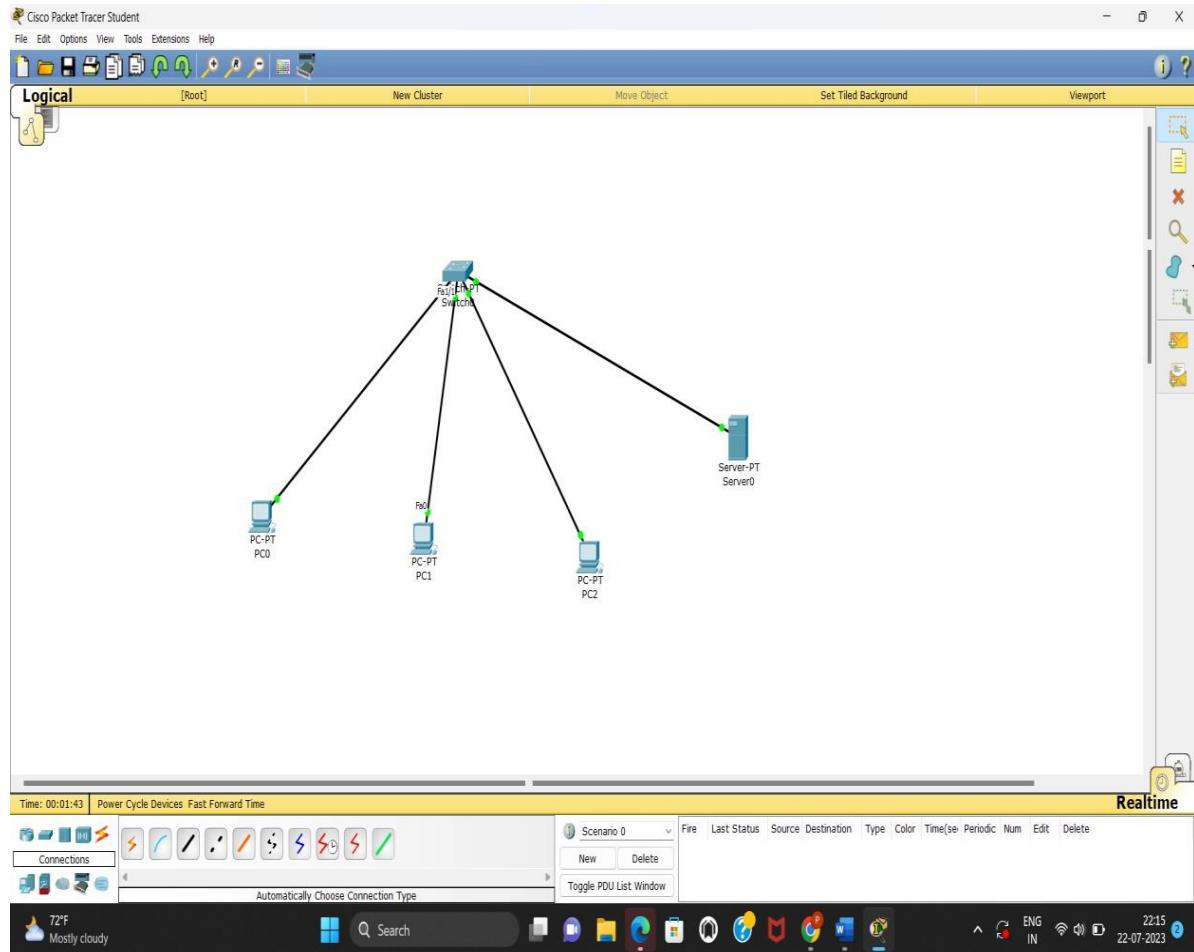
Reply from 10.0.0.1: bytes=32 time=3ms TTL=125
Reply from 10.0.0.1: bytes=32 time=4ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125

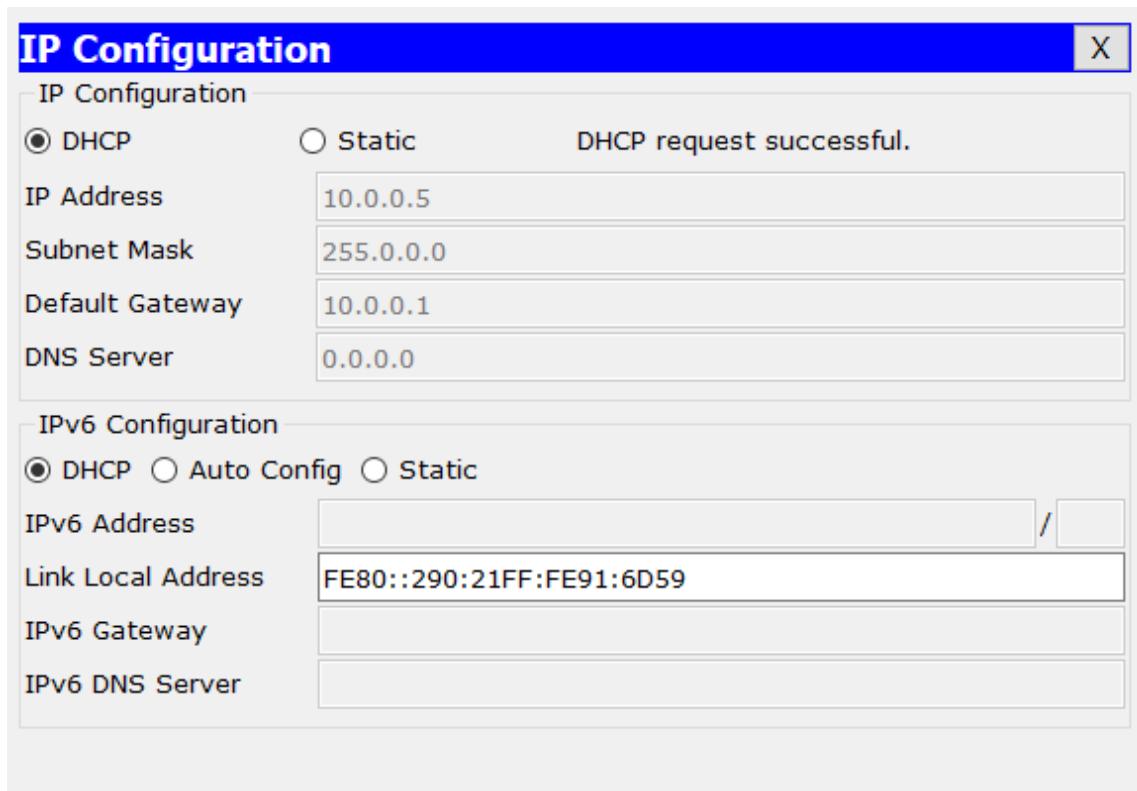
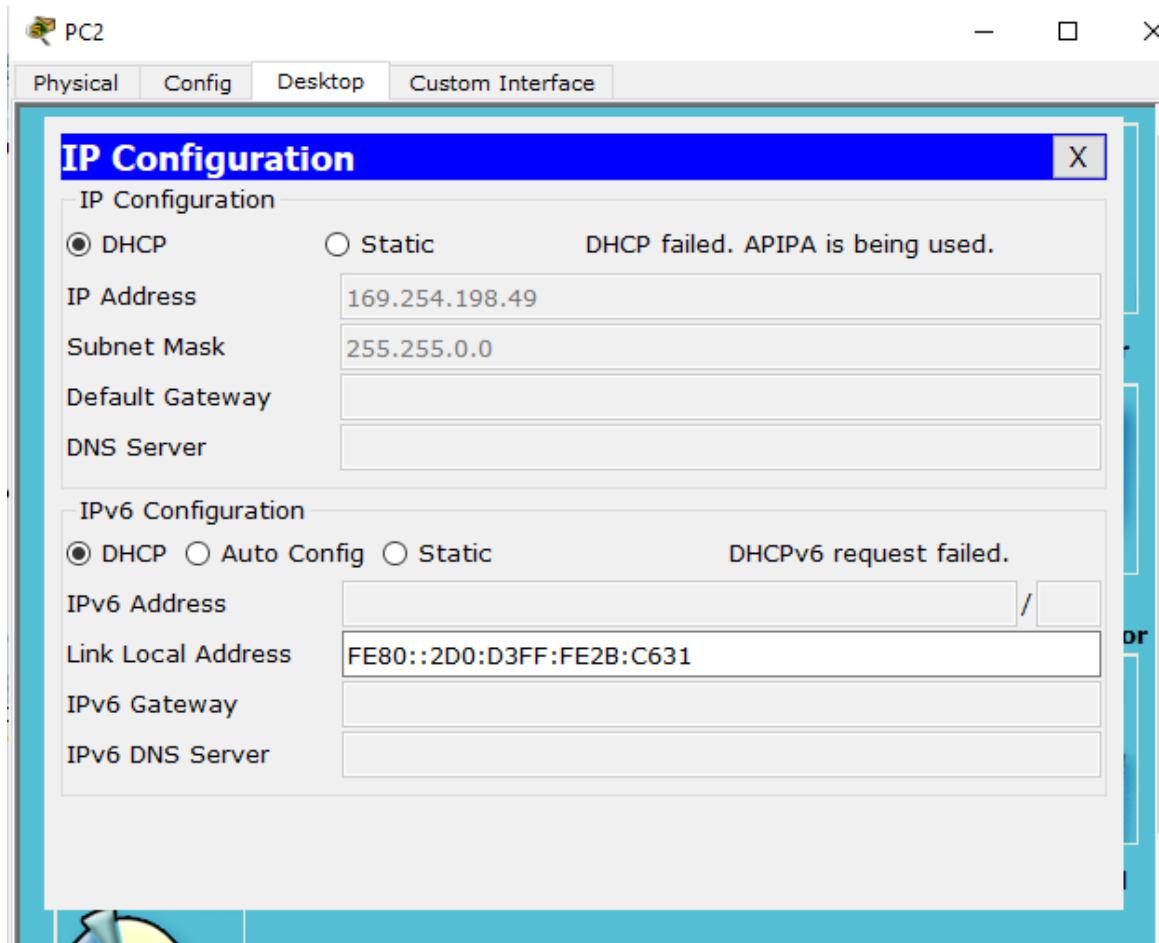
Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 4ms

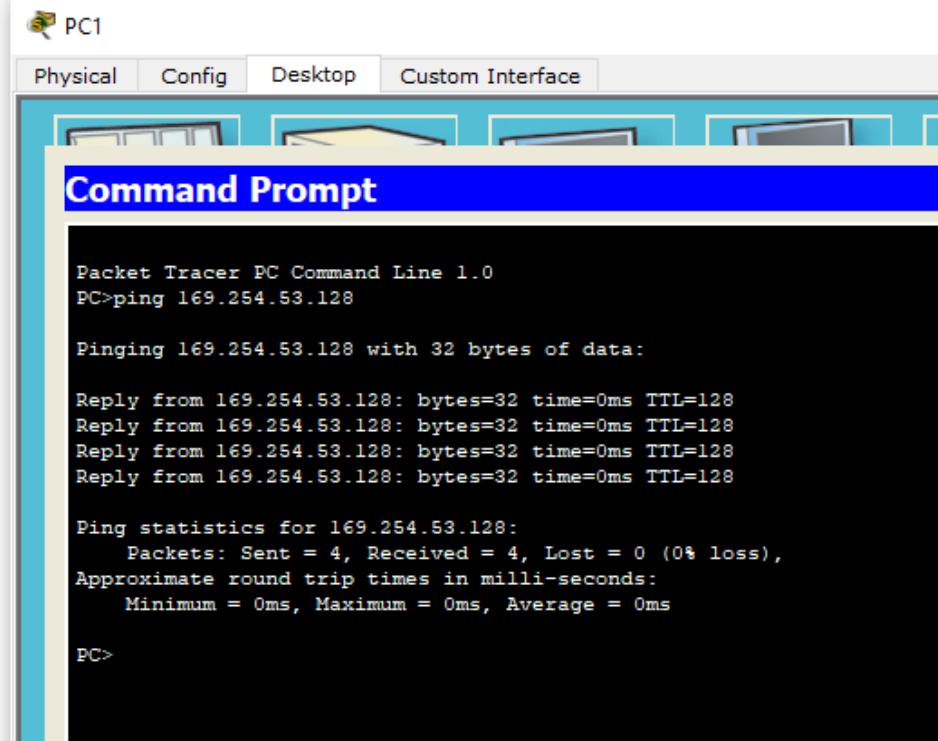
PC>
```

LAB PROGRAM-04

To configure DHCP routing for internal and external LAN.







The image shows a screenshot of the Cisco Packet Tracer software. At the top, there is a menu bar with tabs: Physical, Config, Desktop, and Custom Interface. Below the menu bar, there is a toolbar with several icons. The main area is a window titled "Command Prompt" with a blue header bar. The window contains a black text area displaying the output of a ping command. The text reads:

```
Packet Tracer PC Command Line 1.0
PC>ping 169.254.53.128

Pinging 169.254.53.128 with 32 bytes of data:

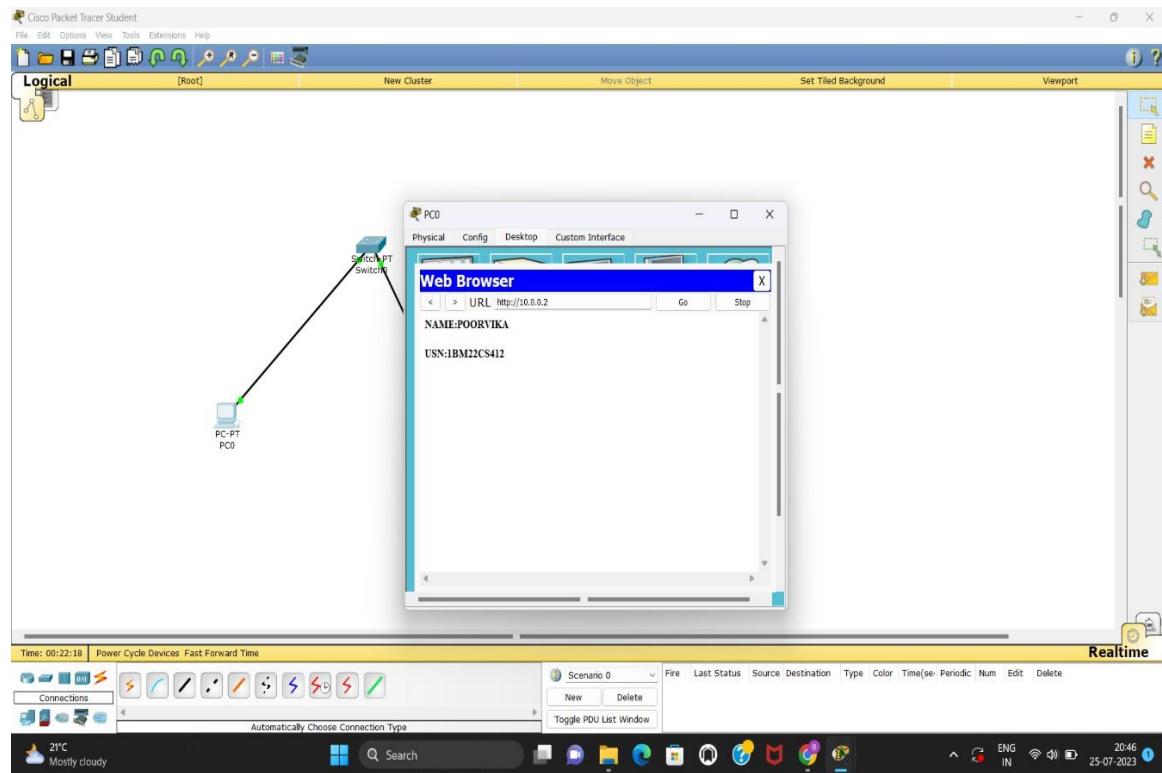
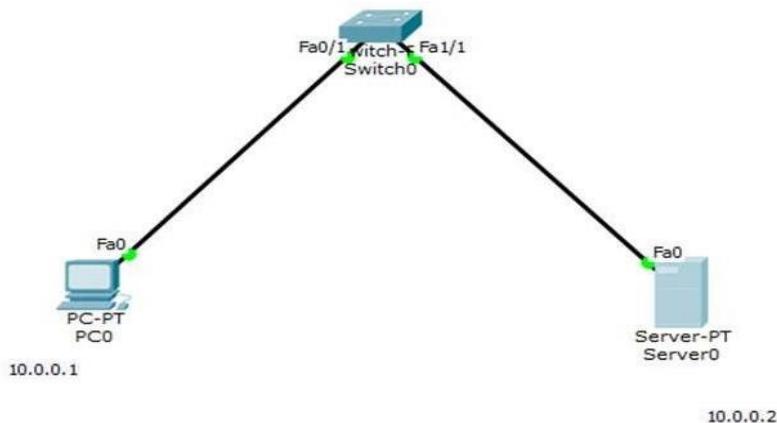
Reply from 169.254.53.128: bytes=32 time=0ms TTL=128

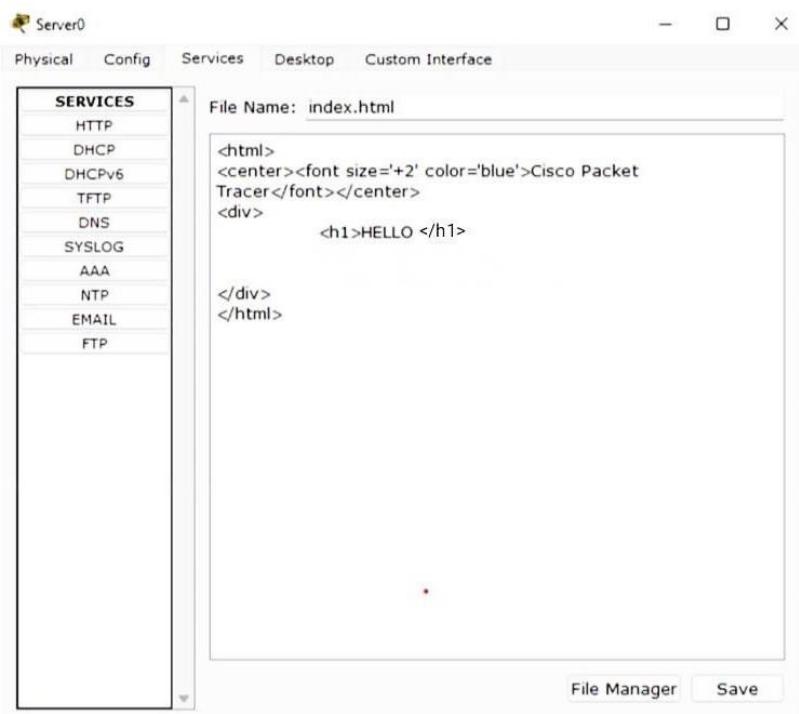
Ping statistics for 169.254.53.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

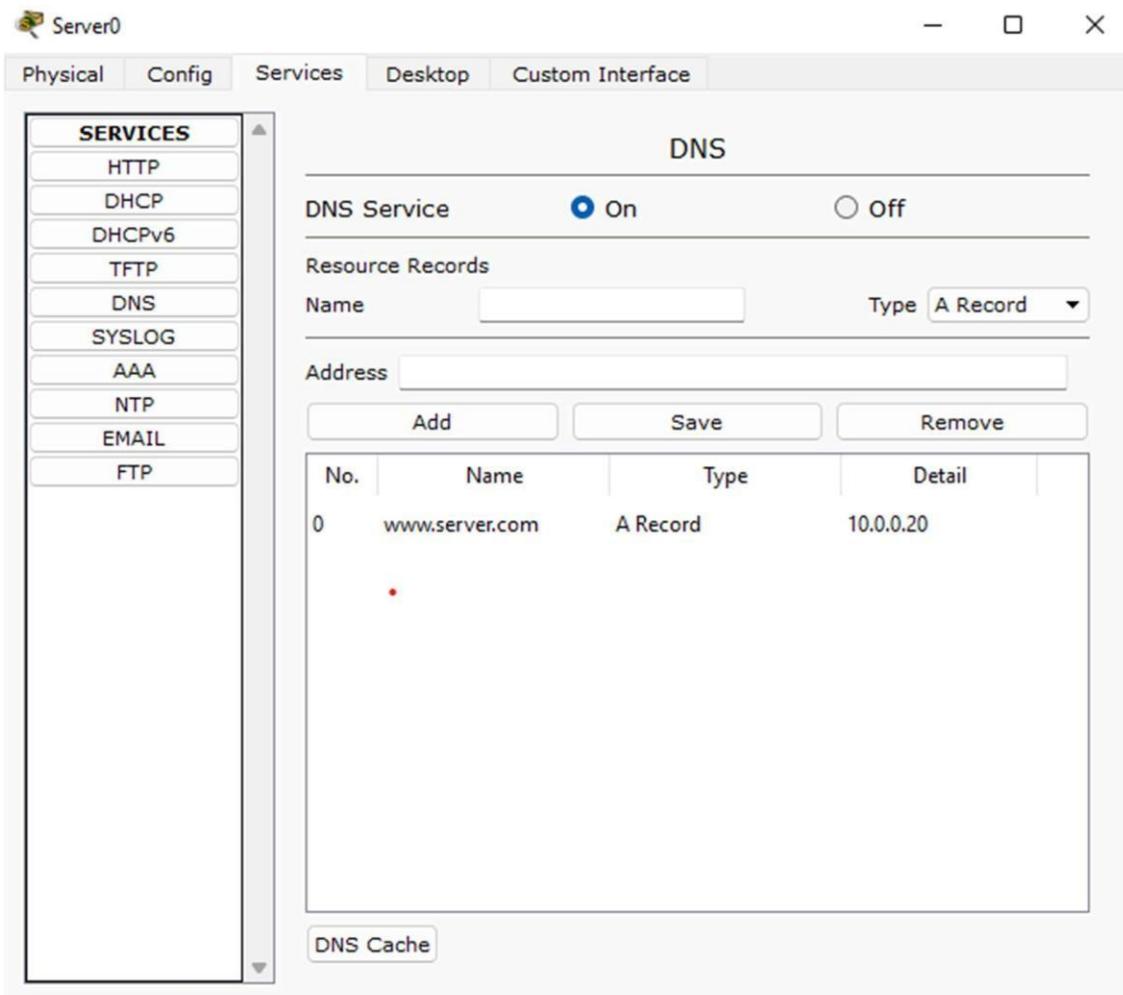
PC>
```

LAB PROGRAM-05

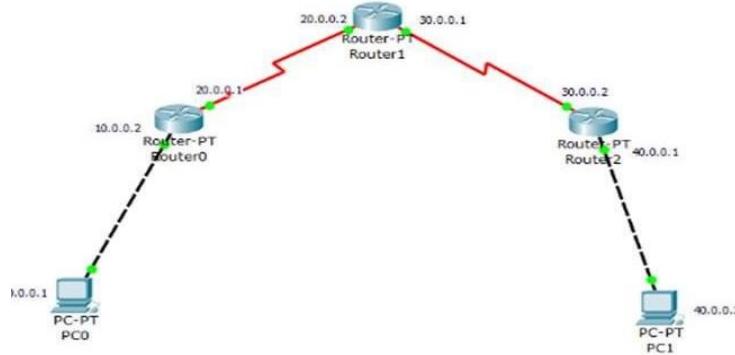
DNS(Using one pc and server):-







RIP(Using 3 routers and 2 pcs):-



PC2

Physical Config Desktop Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.2

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>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 6ms, Maximum = 8ms, Average = 6ms

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=10ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=10ms TTL=125

Ping statistics for 40.0.0.2:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 7ms, Maximum = 10ms, Average = 8ms

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=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=4ms TTL=125
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125
Reply from 40.0.0.2: bytes=32 time=13ms TTL=125

Ping statistics for 40.0.0.2:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 4ms, Maximum = 13ms, Average = 7ms

PC>
```

Router3

Physical Config CLI

IOS Command Line Interface

```
Router(config)#interface Serial2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
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 Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0 20.0.0.0
^
% Invalid input detected at '^' marker.

Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#encapsulation ppp
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

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

Copy Paste

 Router4

- □ ×

Physical Config CLI

IOS Command Line Interface

```
Serial2/0 is DOWN. Line protocol on interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#clock rate 64000
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

R    10.0.0.0/8 [120/1] via 20.0.0.1, 00:00:18, Serial2/0
      20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      20.0.0.0/8 is directly connected, Serial2/0
C      20.0.0.1/32 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
R    40.0.0.0/8 [120/1] via 30.0.0.2, 00:00:02, Serial3/0
Router#
```

 Router5

- □ ×

Physical Config CLI

IOS Command Line Interface

```
Router(config)#clock*rate 64000
               ^
% Invalid input detected at '^' marker.

Router(config)#
Router(config)#interface Serial3/0
Router(config-if)#clock rate 64000
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

R    10.0.0.0/8 [120/2] via 30.0.0.1, 00:00:28, Serial3/0
R    20.0.0.0/8 [120/1] via 30.0.0.1, 00:00:28, 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
Router#
```

Physical Config CLI

IOS Command Line Interface

```
Router(config)#interface Serial2/0
Router(config-if)#encapsulation ppp
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

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

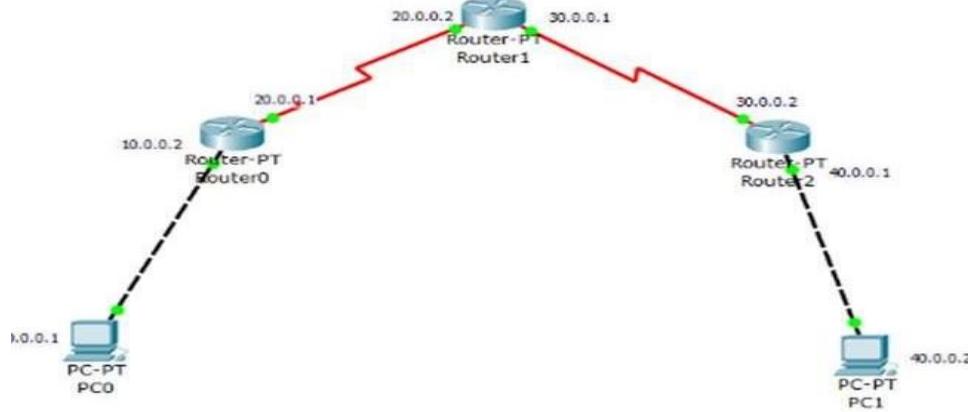
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      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
     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
R    30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:11, Serial2/0
R    40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:11, Serial2/0
Router#
```

LAB PROGRAM-06

RIP(Using 3 routers and 2 pcs):-



```
PC2
Physical Config Desktop Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.2

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>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 8ms, Average = 6ms
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=10ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=10ms TTL=125
Reply from 40.0.0.2: bytes=32 time=10ms TTL=125

Ping statistics for 40.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 10ms, Average = 8ms
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=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=4ms TTL=125
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125
Reply from 40.0.0.2: bytes=32 time=13ms TTL=125

Ping statistics for 40.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 13ms, Average = 7ms
PC>
```

 Router3

Physical Config CLI

IOS Command Line Interface

```

Router(config)#interface Serial2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

*LINK-5-CHANGED: Interface Serial2/0, changed state to down
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 Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0 20.0.0.0
                           ^
* Invalid input detected at '^' marker.

Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#encapsulation ppp
Router(config-if)#
*LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

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

```

 Router4

Physical Config CLI

IOS Command Line Interface

```

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

Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#clock rate 64000
Router(config-if)#
Router(config)#
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 [120/1] via 20.0.0.1, 00:00:18, Serial2/0
      20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    20.0.0.0/8 is directly connected, Serial2/0
C    20.0.0.1/32 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
R    40.0.0.0/8 [120/1] via 30.0.0.2, 00:00:02, Serial3/0
Router#

```

Router3

Physical Config CLI

IOS Command Line Interface

```
Router(config)#interface Serial2/0
Router(config-if)#encapsulation ppp
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
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

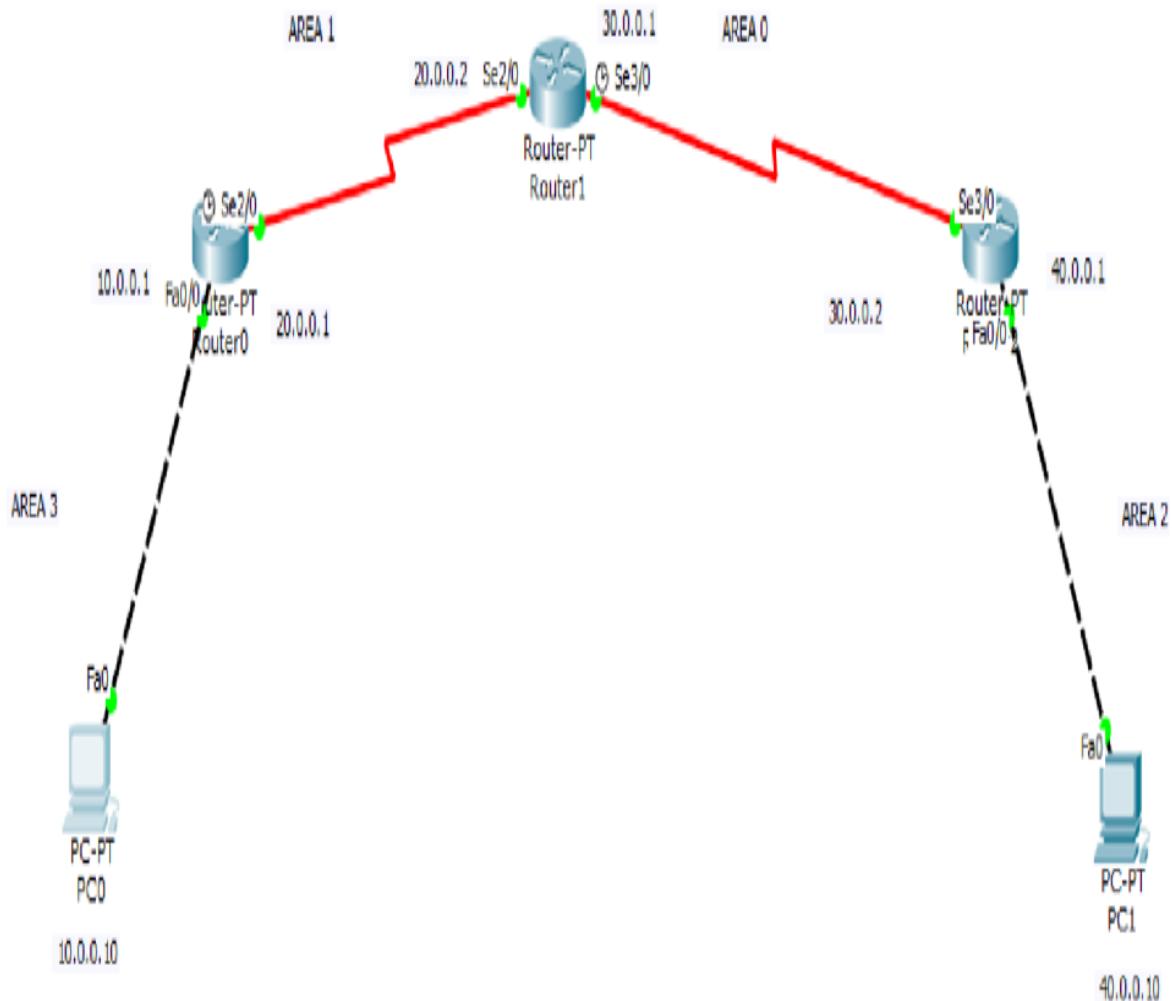
Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
     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
R        30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:11, Serial2/0
R        40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:11, Serial2/0
Router#
```

LAB PROGRAM -07

Configure OSPF routing protocol.

TOPOLOGY:



OUTPUT:

PC0

Physical Config Desktop Custom Interface

Command Prompt

```

Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
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=4ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=12ms 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 = 4ms, Maximum = 12ms, Average = 7ms

PC>

```

Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS047\ospf.pkt

File Edit Options View Tools Extensions Help

Logical [Root] New Cluster Move Object Set Tiled Background Viewport

Router1 Router2 Router0 PC0 PC1

AREA 1 AREA 0 AREA 2

PC0 Router1 Router0 PC1

Event List Simulation

Vis.	Time(sec)	Last Device	All Device	Type	Info
20.002	...	Router1	Router1	OSPF	
20.003	...	Router2	Router2	OSPF	
22.376	...	Router0	Router0	OSPF	
22.377	...	Router0	Router0	OSPF	
22.378	...	Router0	Router0	OSPF	
22.379	...	Router0	Router0	OSPF	
22.380	...	Router2	Router2	OSPF	
22.381	...	Router2	PC1	OSPF	
22.383	...	Router1	Router1	OSPF	

Reset Simulation Constant Delay Captured to: 22.383s

Play Controls Back Auto Capture / Play Capture / Forward

Event List Filters - Visible Events: ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRP-6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IS-IS, IS-ISv6, LACP, NDS, NETFLOW, OSPF, OSPFv3, PIM, RIPv1, RIPv2, RIPv2v6, RIP, RMON, RTSP, SCP, SFTP, SSH, SMTP, SYSLOG, TACACS, TCG, TFTP, Telnet, UDP, VTP

Edit Filters Show All(None)

Time: 00:09:30.703 Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward

Kulators

Scenario 0 New Delete

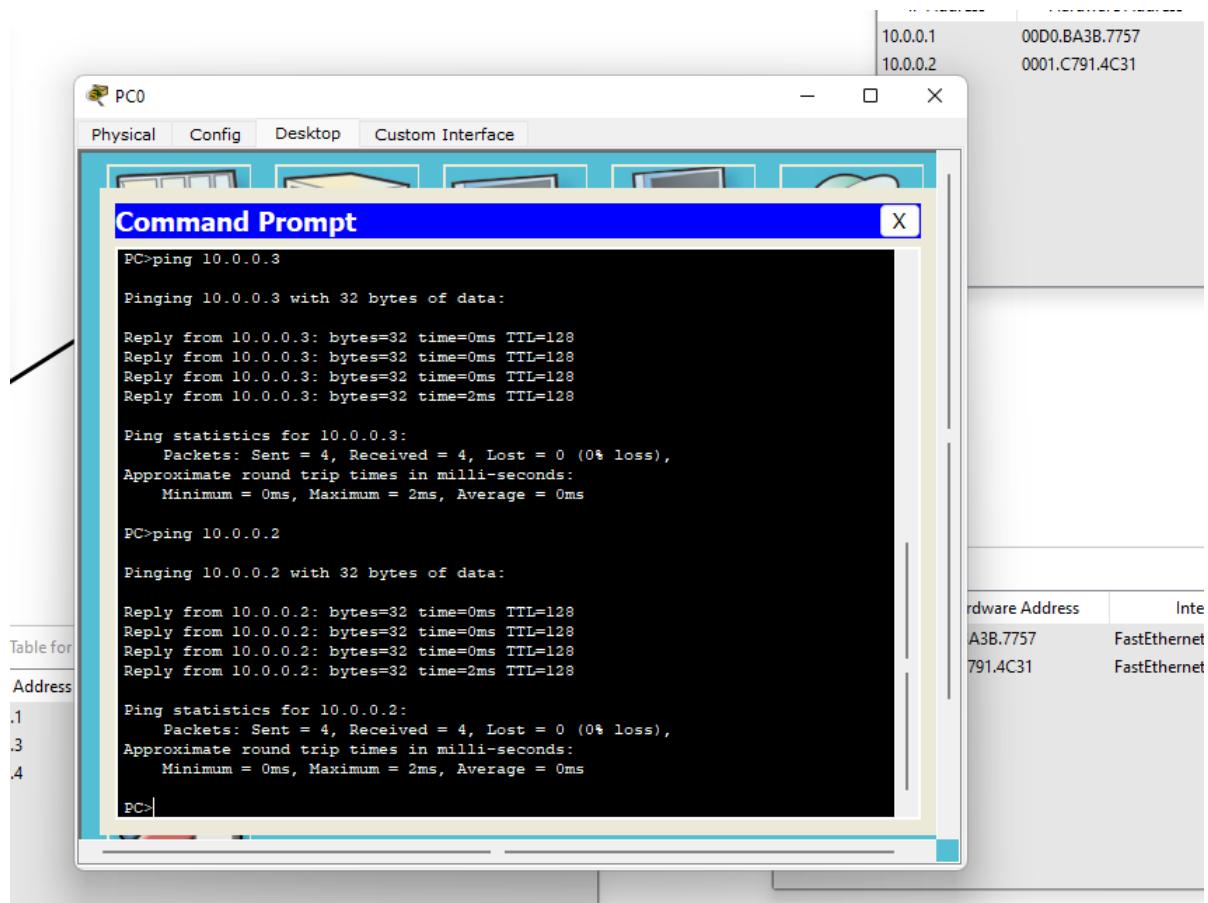
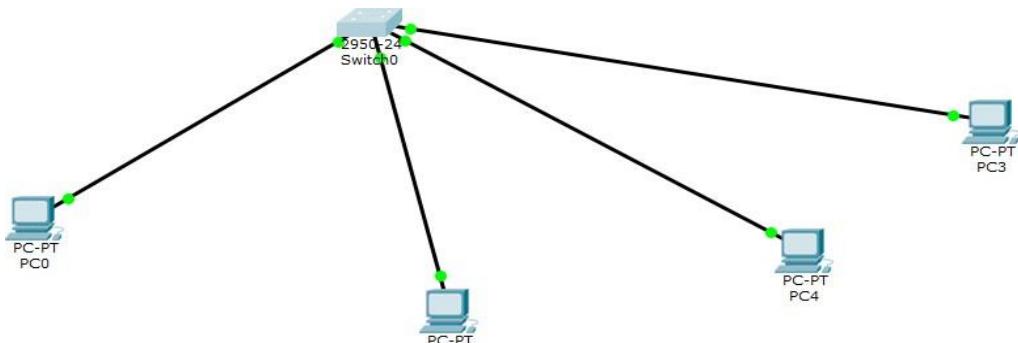
Toggle PDU List Window

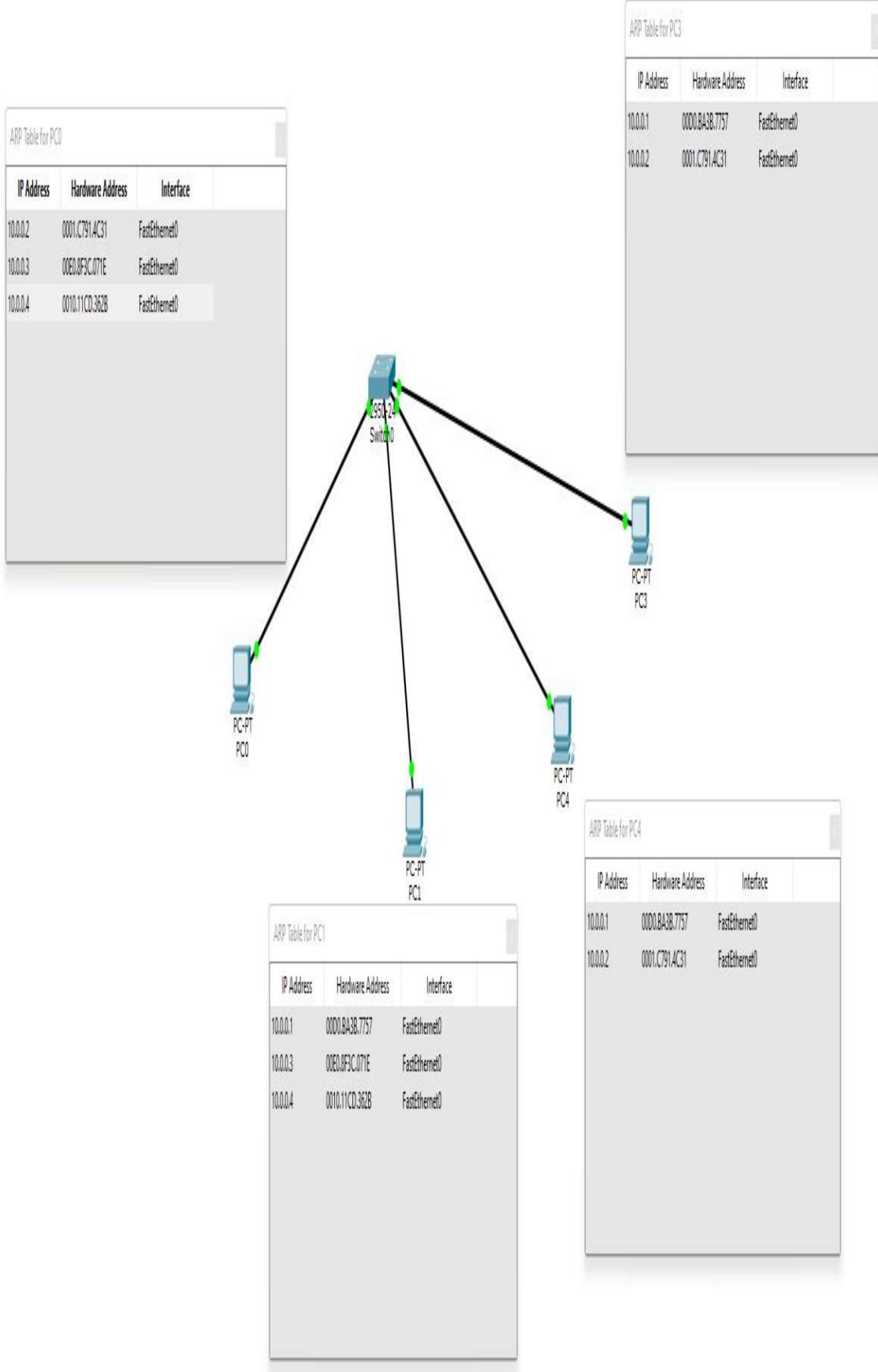
(Select a Device to Drag and Drop to the Workspace)

BIG IN 11/08 27-07-2023

LAB PROGRAM-08

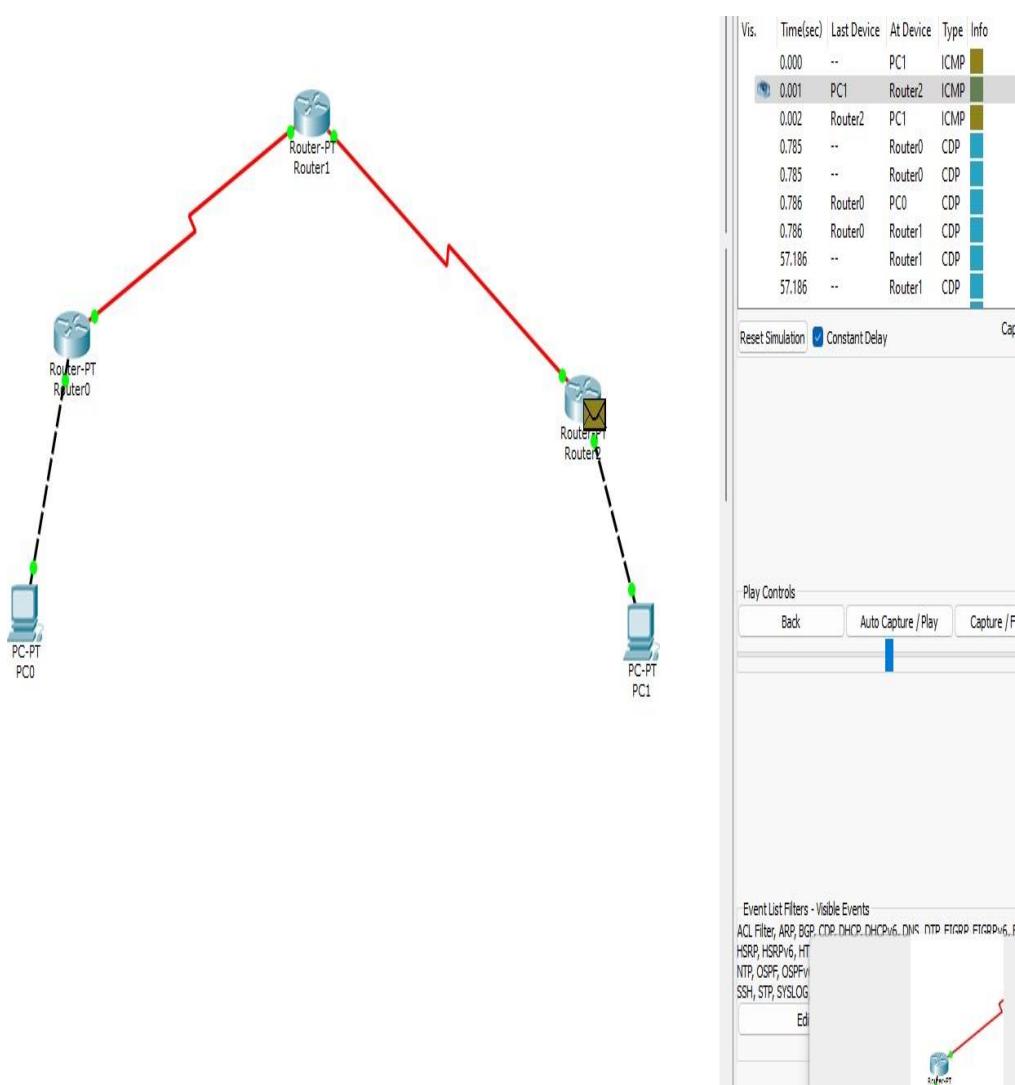
To construct simple LAN and understand the concept and operation of address resolution protocol (ARP)





LAB PROGRAM-09

Demonstrate the TTL/Life of a Packet



PDU Information at Device: Router0

OSI Model **Inbound PDU Details**

PDU Formats

HDLC

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

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x2		0x0	0x0		
TTL: 254	PRO: 0x1	CHKSUM			
SRC IP: 30.0.0.1					
DST IP: 20.0.0.10					
OPT: 0x0			0x0		
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x2		SEQ NUMBER: 1	



PDU Information at Device: Router2

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

Ethernet II

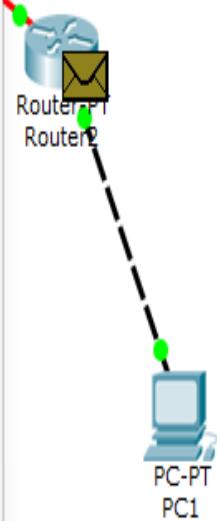
0	4	8	14	19 Bytes
PREAMBLE: 101010...1011		DEST MAC: 0060.4755.A5E5	SRC MAC: 0002.17EE.D7DA	
TYPE: 0x800	DATA (VARIABLE LENGTH)			FCS: 0x0

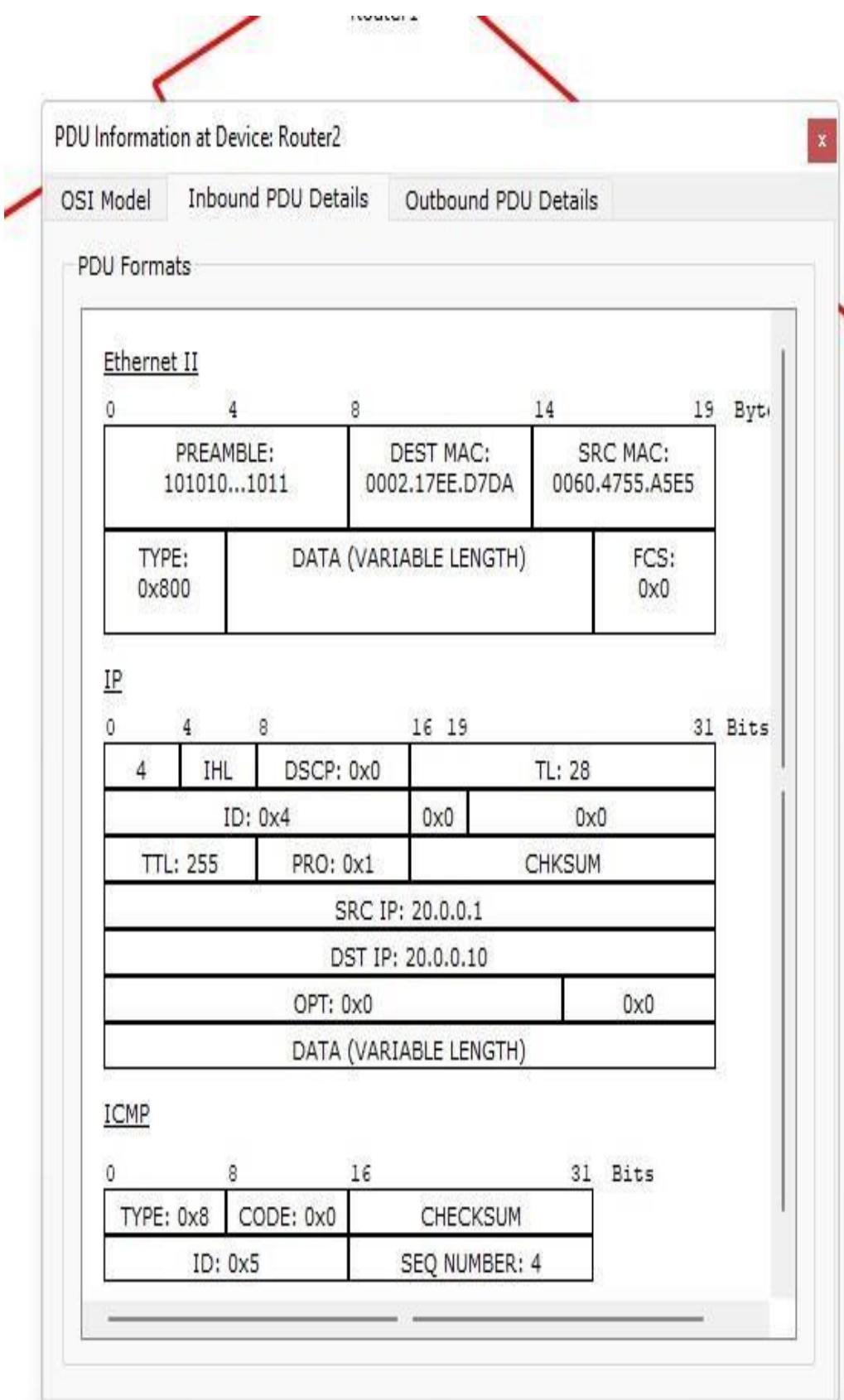
IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x4		0x0	0x0		
TTL: 255	PRO: 0x1	CHKSUM			
SRC IP: 20.0.0.10					
DST IP: 20.0.0.1					
OPT: 0x0			0x0		
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x0	CODE: 0x0	CHECKSUM	
ID: 0x5		SEQ NUMBER: 4	





PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

HDLC

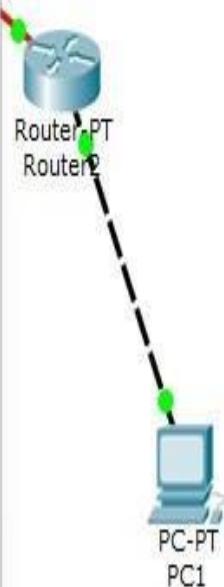
0	8	16	32	32+x	48+x	56+x
FLG: 0111 1110	ADR: 0x8f	CONTROL: 0x0	DATA: (VARIABLE LENGTH)		FCS: 0x0	FLG: 0111 1110

IP

0	4	8	16	19	31 Bits
	4	IHL	DSCP: 0x0	TL: 28	
			ID: 0x1	0x0	0x0
	TTL: 255	PRO: 0x1		CHKSUM	
			SRC IP: 30.0.0.1		
			DST IP: 30.0.0.10		
			OPT: 0x0	0x0	
			DATA (VARIABLE LENGTH)		

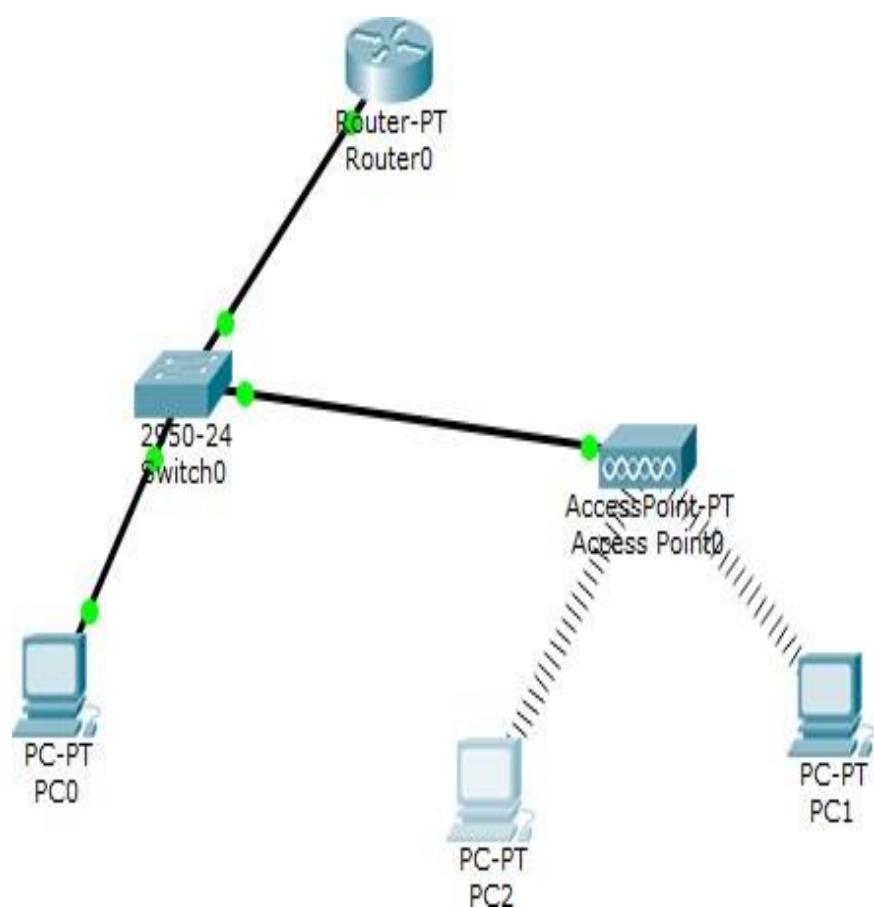
ICMP

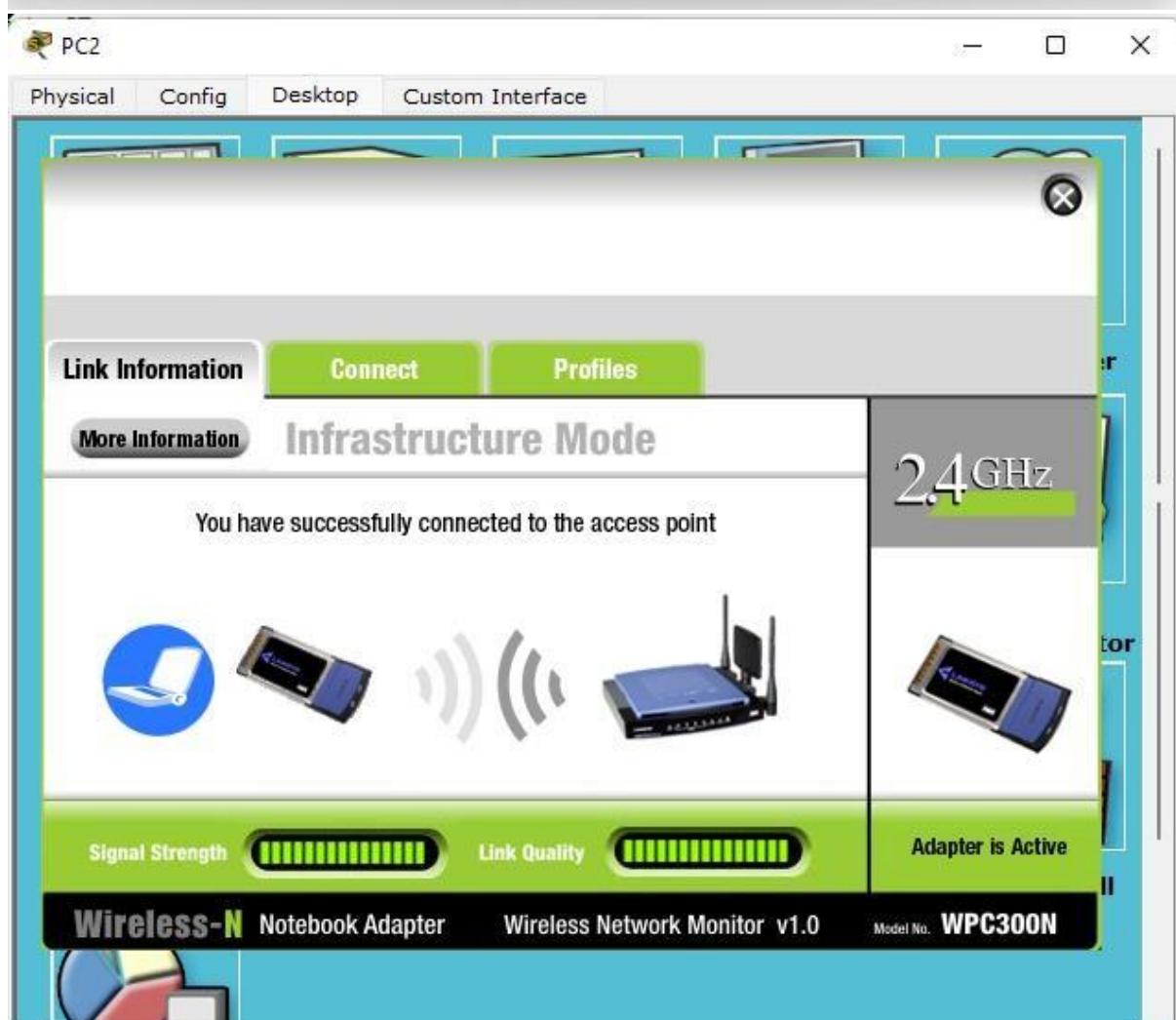
0	8	16	31 Bits
TYPE: 0x0	CODE: 0x0	CHECKSUM	
ID: 0x2		SEQ NUMBER: 1	

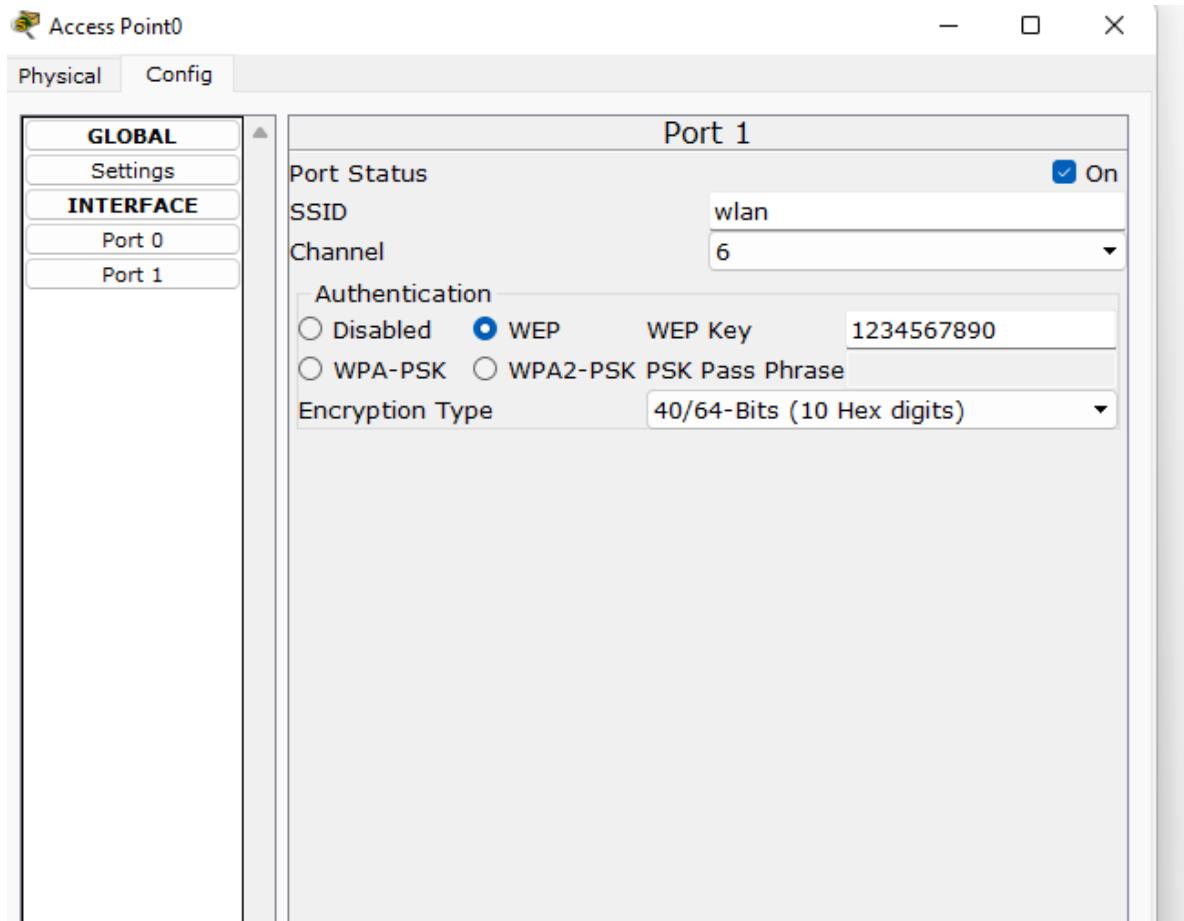


LAB PROGRAM:10

WLAN:









PC2

Physical Config Desktop Custom Interface



Command Prompt

```
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=31ms TTL=128
Reply from 192.168.1.3: bytes=32 time=12ms TTL=128
Reply from 192.168.1.3: bytes=32 time=11ms TTL=128
Reply from 192.168.1.3: bytes=32 time=13ms 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 = 11ms, Maximum = 31ms, Average = 16ms
```

```
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=15ms TTL=128
Reply from 192.168.1.3: bytes=32 time=13ms TTL=128
Reply from 192.168.1.3: bytes=32 time=13ms TTL=128
Reply from 192.168.1.3: bytes=32 time=16ms 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 = 13ms, Maximum = 16ms, Average = 14ms
```

```
PC>
```

PC0

Physical Config Desktop Custom Interface

Command Prompt

```
PC>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:

Reply from 192.168.1.4: bytes=32 time=16ms TTL=128
Reply from 192.168.1.4: bytes=32 time=6ms TTL=128
Reply from 192.168.1.4: bytes=32 time=4ms TTL=128
Reply from 192.168.1.4: bytes=32 time=9ms TTL=128

Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 16ms, Average = 8ms

PC>
```

LAB PROGRAM-11

TELNET:



```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface termial FA0/0
^
% Invalid input detected at '^' marker.

Router(config)#interface termial fa0/0
^
% Invalid input detected at '^' marker.

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)#hostname r1
r1(config)#enable secret p1
r1(config)#interface fastethernet 0/0
r1(config-if)#ip address 10.0.0.1 255.0.0.0
r1(config-if)#no shutdown
r1(config-if)#
r1(config-if)#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 p0
r1(config-line)#
r1(config-line)#exit
r1(config)#exit
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#wr
Building configuration...
[OK]
r1#
```

PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
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=0ms TTL=255

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

PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open

User Access Verification

Password:
Password:
Password:

[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open

User Access Verification

Password:
rl>enable
Password:
Password:
rl#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

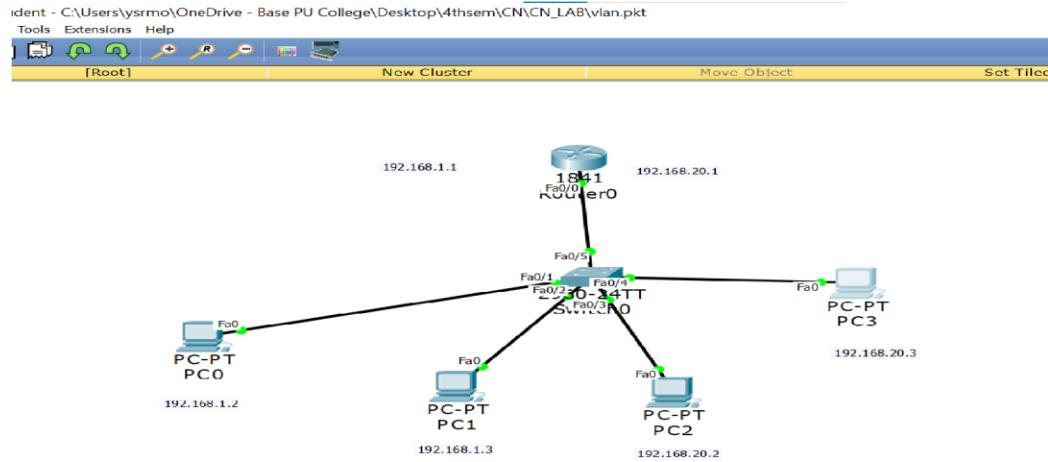
Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
rl#
```

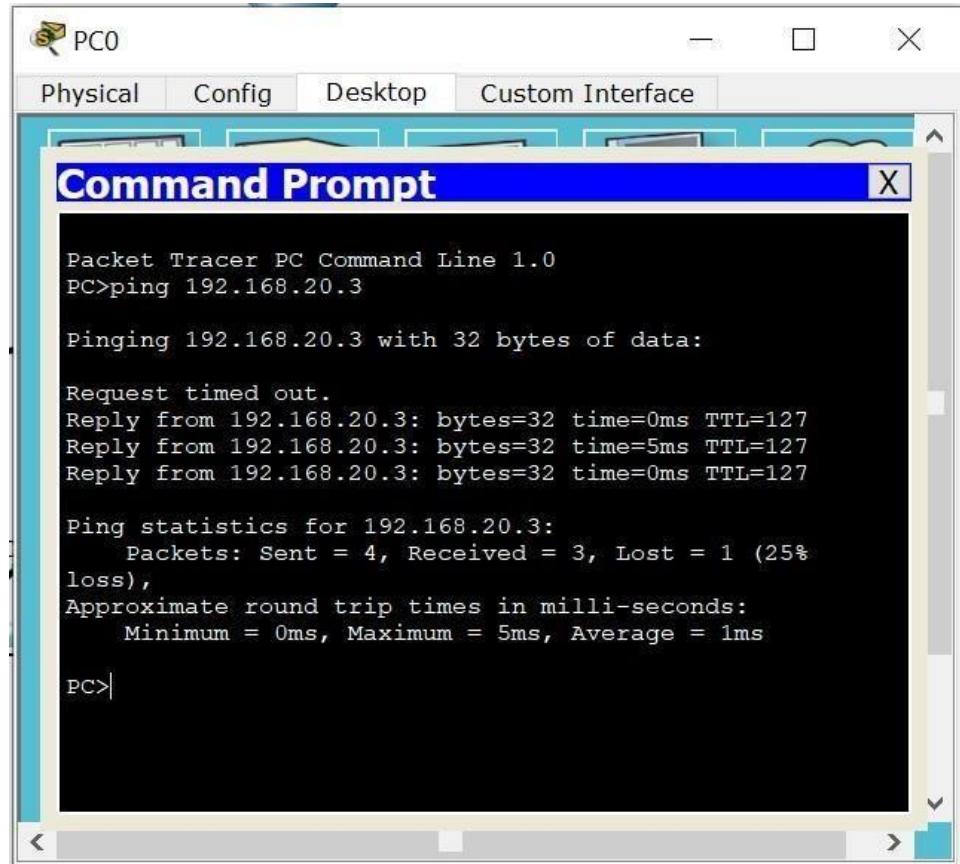
LAB PROGRAM-12

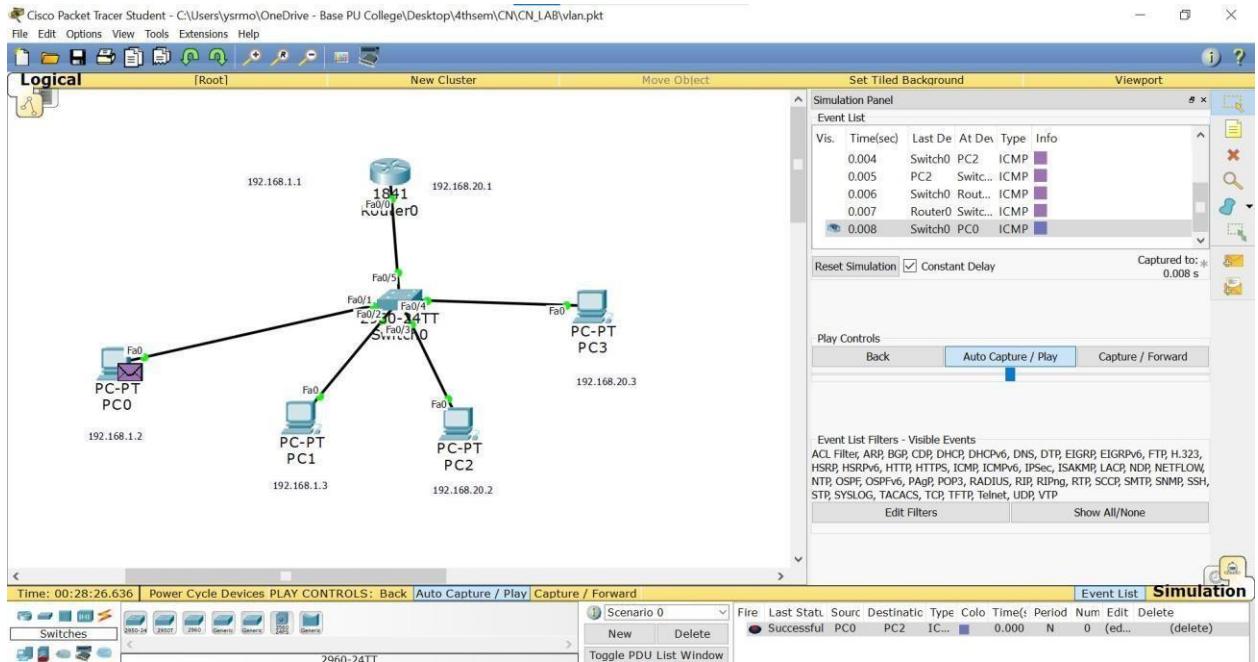
To construct a VLAN and make a pc communicate among VLAN.

TOPOLOGY:



OUTPUT:





LAB PROGRAM-13

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

CODE:

```
#include<stdio.h>#include<string.h>

#define N strlen(gen_poly) char data[28]; char check_value[28]; char
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("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("\n Enter 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 Data padded with n-1 zeros : %s",data); 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 Final
  data to be sent : %s",data); receiver();
  return 0;
}

```

OUTPUT:

```

Enter data to be transmitted: 10001000000100001

Enter the Generating polynomial: 1011

Data padded with n-1 zeros : 10001000000100001000
CRC or Check value is : 100
Final data to be sent : 10001000000100001100
Enter the received data: 10001000000100001100
Data received: 10001000000100001100
No error detected

```

```

Enter data to be transmitted: 10001000000100001

Enter the Generating polynomial: 1011

Data padded with n-1 zeros : 10001000000100001000
CRC or Check value is : 100
Final data to be sent : 10001000000100001100
Enter the received data: 10010000000100001100
Data received: 10010000000100001100
Error detected

```

LAB PROGRAM-14

Write a program for congestion control using Leaky bucket algorithm.

CODE:

```
#include<stdio.h> void  
main()  
{ int b_size,d_rate,in_d_rate,rem_b_size;  
printf("Enter the bucket size:\n");  
scanf("%d",&b_size); rem_b_size=b_size;  
printf("Enter the outgoing data rate:\n");  
scanf("%d",&d_rate); while(1) {  
printf("Enter the size of incoming packet\n");  
scanf("%d",&in_d_rate); if(in_d_rate<=b_size)  
{ if(in_d_rate<=rem_b_size) { rem_b_size=rem_b_size-  
in_d_rate; rem_b_size=rem_b_size+d_rate;  
printf("Data packet is accepted\n"); printf("Remaining  
space in bucket is....  
%d\n",rem_b_size); printf("\n");  
} else{ printf("Data packet is dropped because the bucket size is less than  
the packet  
size\n"); printf("\n");  
}  
}  
}  
}
```

OUTPUT:

```
Enter the bucket size:  
5000  
Enter the outgoing data rate:  
200  
Enter the size of incoming packet  
3000  
Data packet is accepted  
Remaining space in bucket is.... 2200  
  
Enter the size of incoming packet  
2500  
Data packet is dropped because the bucket size is less than the packet size  
  
Enter the size of incoming packet
```

LAB PROGRAM-15

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.

SOLUTION:

```
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:

Client:



IDLE Shell 3.10.8

```
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =
Enter file name:serverTCP.py

From Server:

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 recieve")
    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()
```

```

>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =
Enter file name:aab.py
From Server:
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
class Node:
    def __init__(self,data):
        self.data=data
        self.left=None
        self.right=None
        self.height=1

class AVL Tree:
    def getHeight(self,root):
        if not root:
            return 0
        return root.height

    def getBalance(self,root):
        if not root:
            return 0
        return self.getHeight(root.left)-self.getHeight(root.right)

    def rightRotate(self,z):
        y=z.left
        T3=y.right

        y.right=z
        z.left=T3

        z.height=1+max(self.getHeight(z.left),self.getHeight(z.right))
        y.height=1+max(self.getHeight(y.left),self.getHeight(y.right))

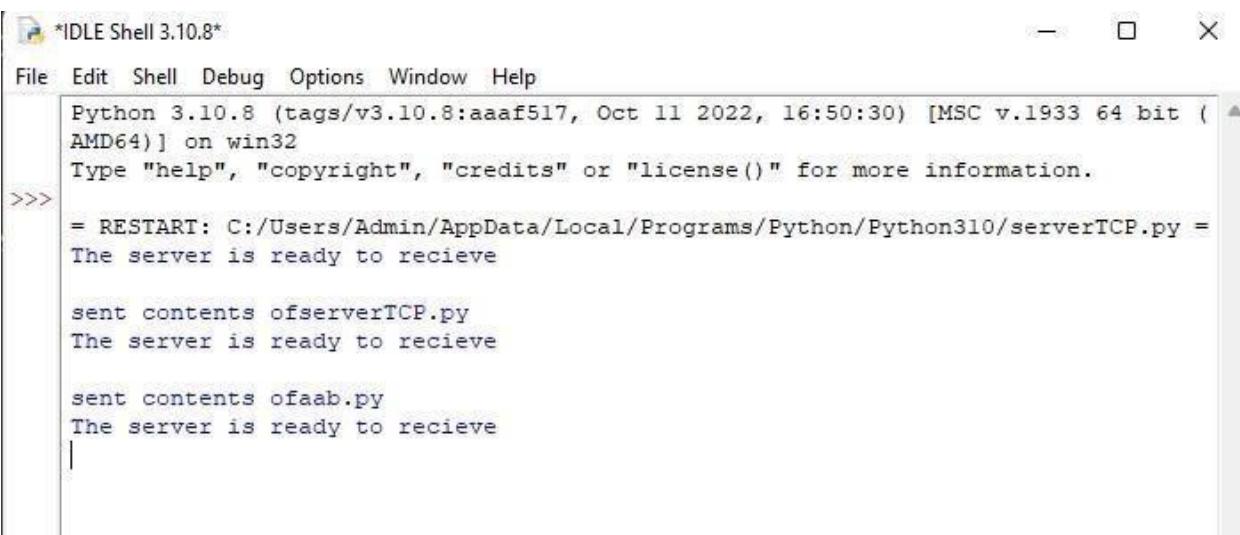
        return y

    def insert(self,root,data):
        if not root:
            return Node(data)
        if data < root.data:
            root.left=self.insert(root.left,data)
        else:
            root.right=self.insert(root.right,data)

>>>

```

Server:



The screenshot shows an IDLE Shell window titled '*IDLE Shell 3.10.8*'. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window displays the following Python code:

```

File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverTCP.py =
The server is ready to recieve

sent contents ofserverTCP.py
The server is ready to recieve

sent contents ofaab.py
The server is ready to recieve
|
```

LAB PROGRAM-16

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.

SOLUTION:

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

```
# print (str(i), end = '') file.close()
```

OUTPUT:

Client:

```
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/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 ('\nSent contents of ', end = ' ')  
    print (sentence)  
    # for i in sentence:  
    #     print (str(i), end = '')  
    file.close()  
  
>>>
```

Server:

```
>>>  
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverUDP.py =  
The server is ready to receive  
  
Sent contents of serverUDP.py
```