VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

Gurrala Naga Pragnathmik (1BM22CS103)

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 **Gurrala Naga Pragnathmik (1BM22CS103)** 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 - I

Cycle-II

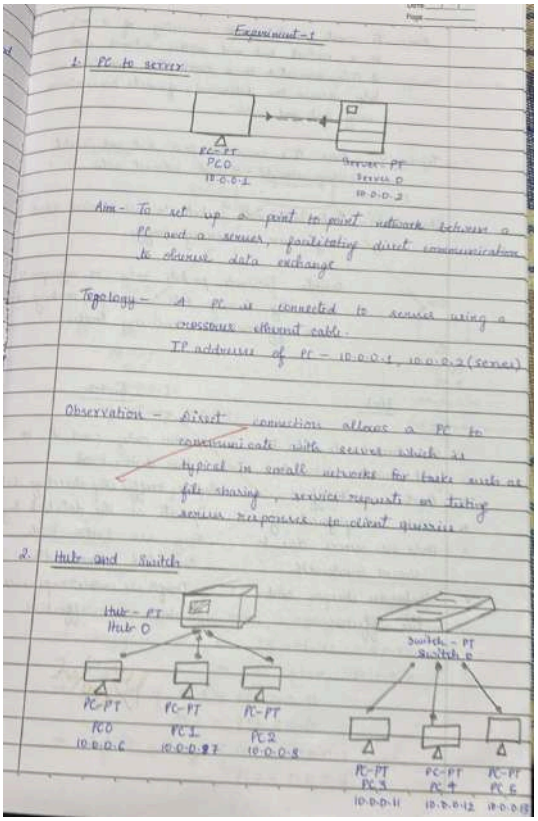
Github Link:
https://github.com/pragna-gn/CN-Observation

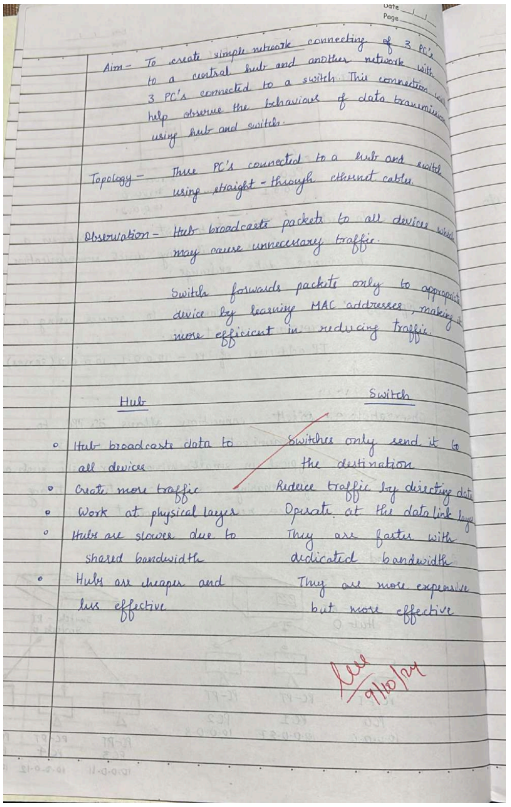
Program 1

Aim of the program:

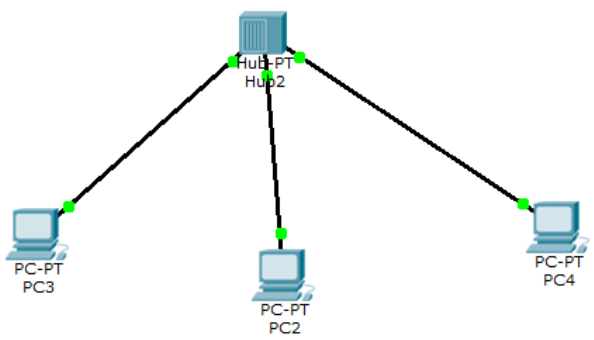
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

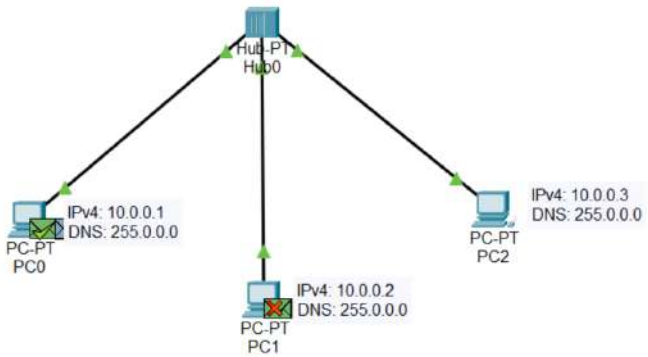
Procedure along with the topology:



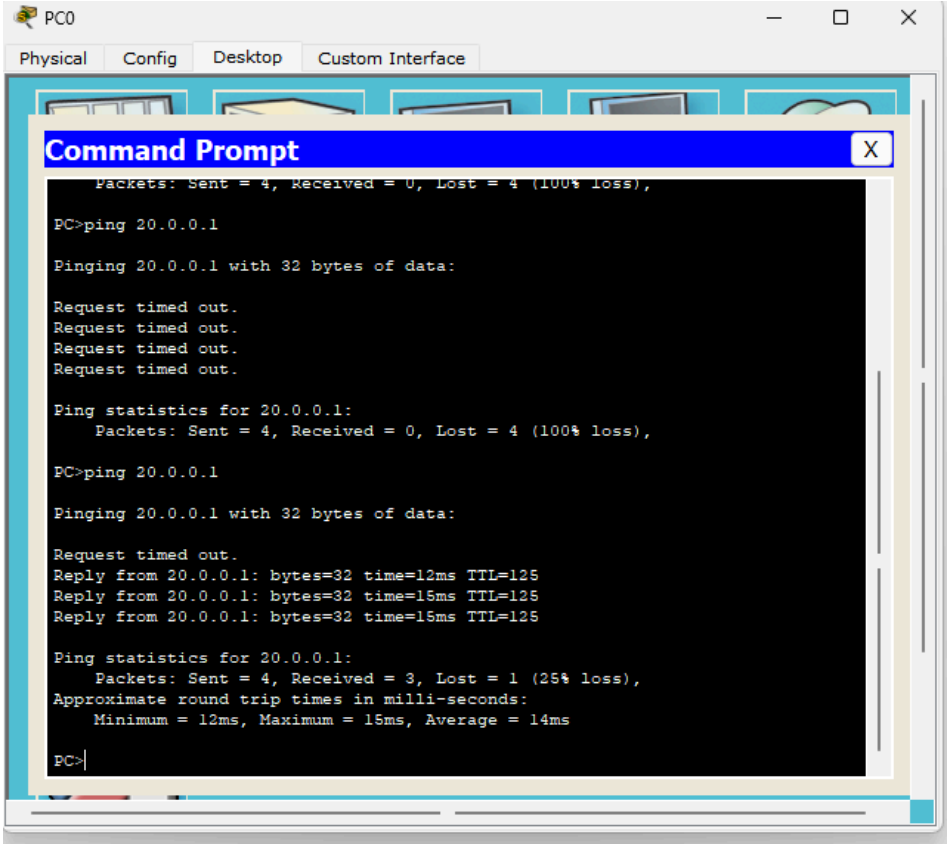


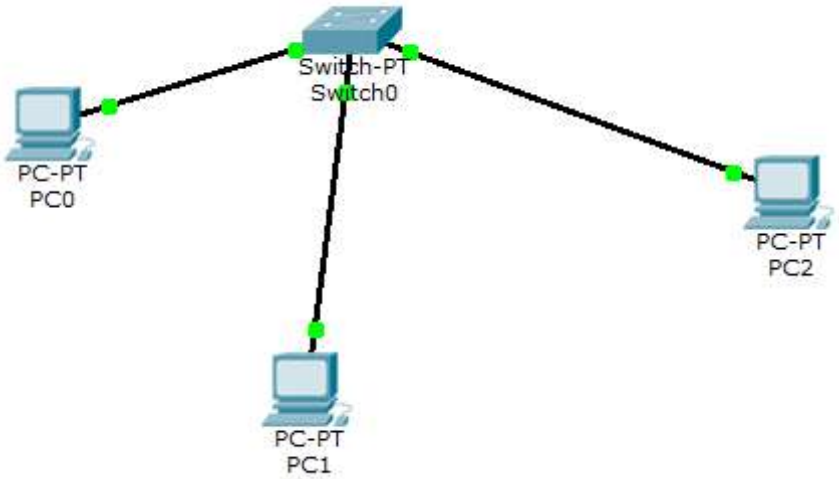
Screen shots/ output:





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





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

```
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

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

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

Observation:

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

Switch forwards packets only to appropriate device by learning MAC addresses, making it more efficient in reducing traffic.

Program 2

Aim of the program:

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

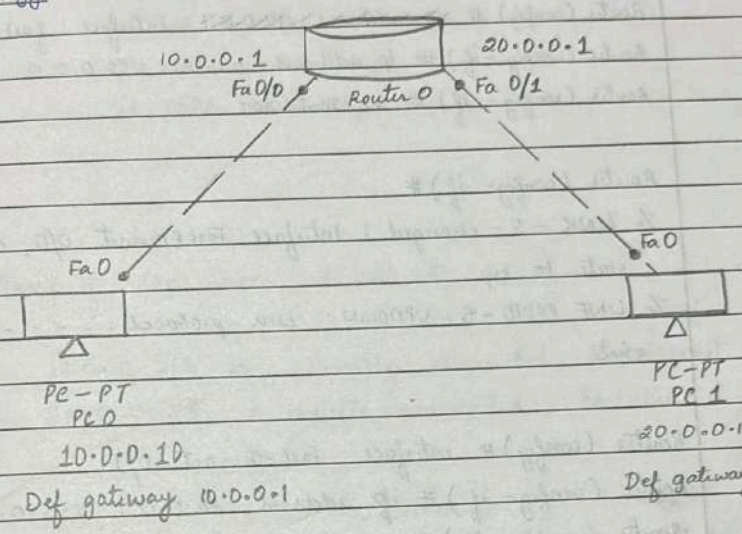
Procedure along with the topology:

Date 09/10/24
Page

Experiment-2A

Aim – To connect two PC's belonging to different networks using a router

Topology:



Two PC's are connected to each other using a router using copper cross-over cable.

Procedure –

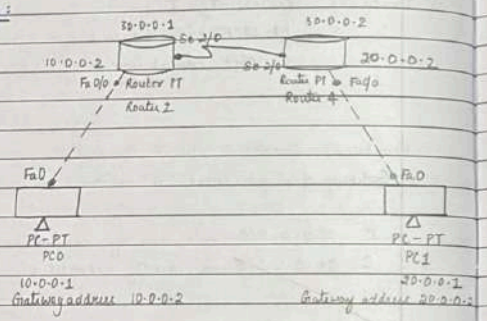
→ Place 2 PC's and 1 router on the workspace
→ Use a copper cross over to connect PC's to router
→ Configure PC 1 to IP address 10.0.0.10
→ Configure PC2 to IP address 20.0.0.10
→ Subnet of PC 1 = 255.0.0.0
PC 2 = 255.0.0.0
→ Default gateway of PC 1 = 10.0.0.1
PC 2 = 20.0.0.1

Date 16/10/24
Page

Experiment - 2B

Aim: To observe the various errors messages displayed while connecting 2 routers of different networks

Topology:



1) PC0 - connected to router's interface Fa 0/0 using a cross over cable
2) PC1 - connected to router's interface Fa 1/0 using a copper cross over cable
3) Router - there are 2 routers connected via serial cable to each other

Date / /
Page

Router > enable
Router # configure terminal
Router (config) # interface fast ethernet 1/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # exit

Router (config) # interface serial 2/0
" " # ip address 30.0.0.2 255.0.0.0
Router (config) # no shutdown

Configure the PC's

PC0: Ip address - 10.0.0.1
subnet - 255.0.0.0
default gateway - 10.0.0.2

PC1: Ip address - 20.0.0.1
subnet - 255.0.0.0
default gateway - 20.0.0.2

Test Connectivity by opening command prompt on PC0. Use the ping command to check connectivity from PC0, ping PC1's IP address (20.0.0.1)

Observation:

The configuration and cabling are correct will recieve successful ping replies between the two PC's

Date / /
Page

Router 0:

Interface Fa 0/0 connected to PC-0
Interface Se 2/0 connected to Router-1
Ip address of Fa 0/0 : 10.0.0.2
Ip address of Se 2/0 : 30.0.0.1

Router 1:

Interface Fa 1/0 connected to PC-1
Interface Se 2/0 connected to Router 0
Ip address of Fa 1/0 : 20.0.0.2
Ip address of Se 2/0 : 30.0.0.2

Procedure:

- Open cisco packet tracer and drag the following components onto workspace
  Place the two routers in the middle
  Place the two PC's on either side of the routers
- Use cross over cable to connect the devices as follows
  PC0: Router 0 Fa 0/0 interface
  PC1: Router 1 Fa 1/0 interface
- Configure Router 0 by clicking on the router and enter CLI
  Assign Ip addresses to the router interfaces
  Router > enable
  Router # config terminal
  Router (config)# interface ethernet 0/0
  Router (config-if)# ip address 10.0.0.2 255.0.0.0
  Router (config-if) # exit

Date / /
Page

Router # interface serial 2/0
Router (config-if)# ip address 30.0.0.1 255.0.0.0
Router (config-if) # no shut down

The ping results are as follows:

Pinging 10.0.0.1 with 32 bytes of data
Request timed out

Ping statistics for 10.0.0.1
Packet 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 10.0.0.2

~~Ping & Route~~

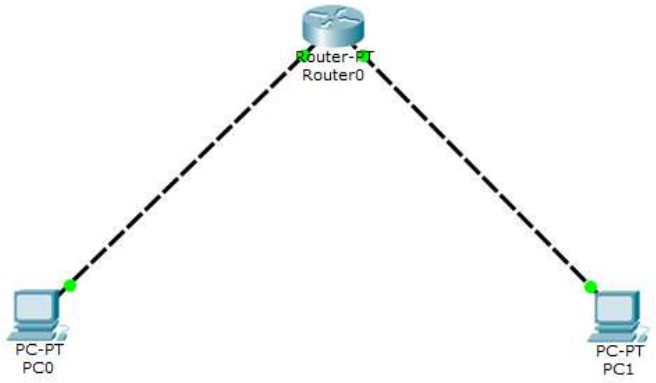
Show Ip Route before static Routing Router 1

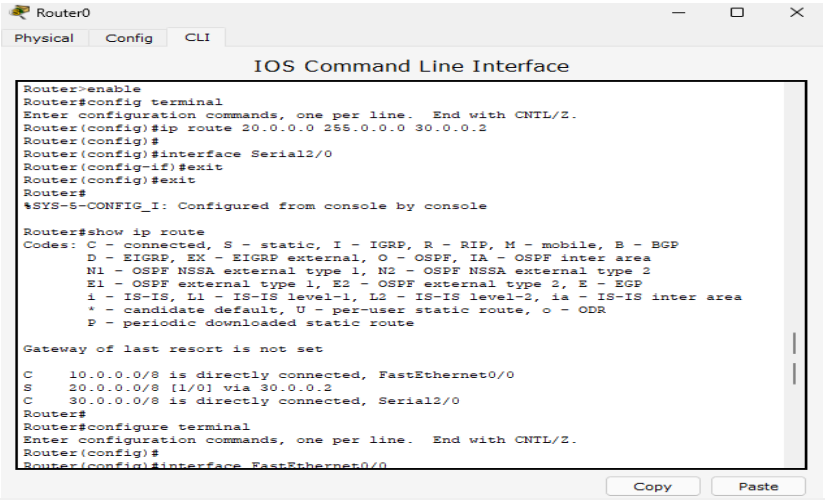
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 30.0.0.0/8 is directly connected, Serial 2/0

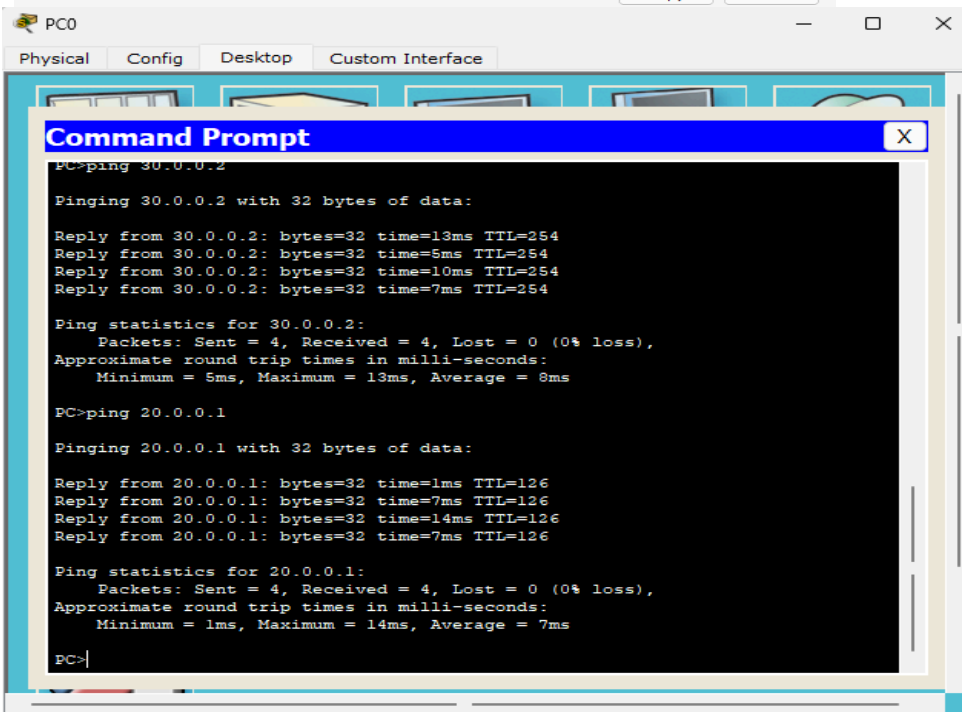
Show Ip Route after static routing Router 1

C 10.0.0.0/8 is directly connected, Fast ethernet 0/0
S 20.0.0.0/8 [1/0] via 30.0.0.2
C 30.0.0.0/8 is directly connected, Serial 2/0

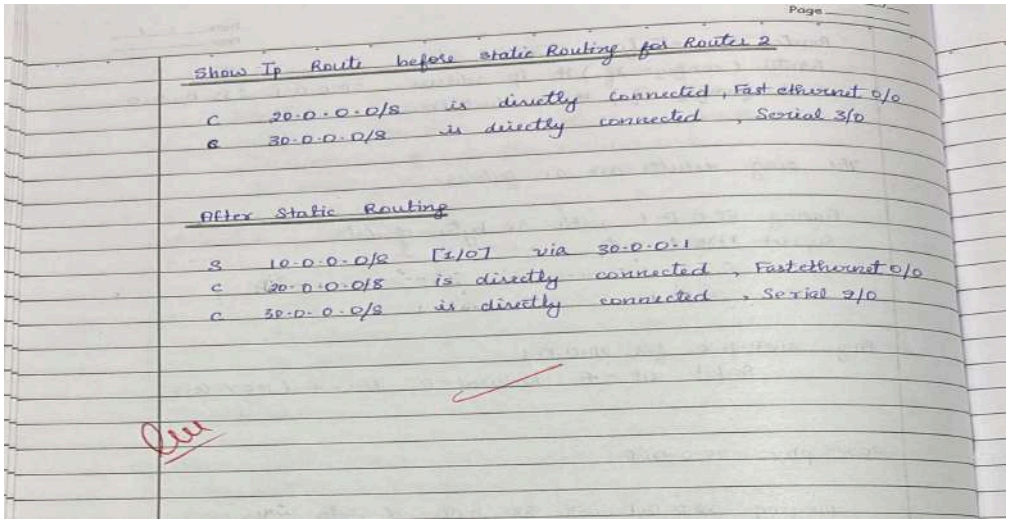
Screen shots/ output:







Observation:



Show Ip Route before static Routing for Router 2

C 20.0.0.0/8 is directly connected, Fast ethernet 0/0
C 30.0.0.0/8 is directly connected, Serial 3/0

After Static Routing

S 10.0.0.0/8 [1/0] via 30.0.0.1
C 20.0.0.0/8 is directly connected, Fastethernet 0/0
C 30.0.0.0/8 is directly connected, Serial 3/0

<u>Program 3</u>

Aim of the program:
Configure default route, static route to the Router

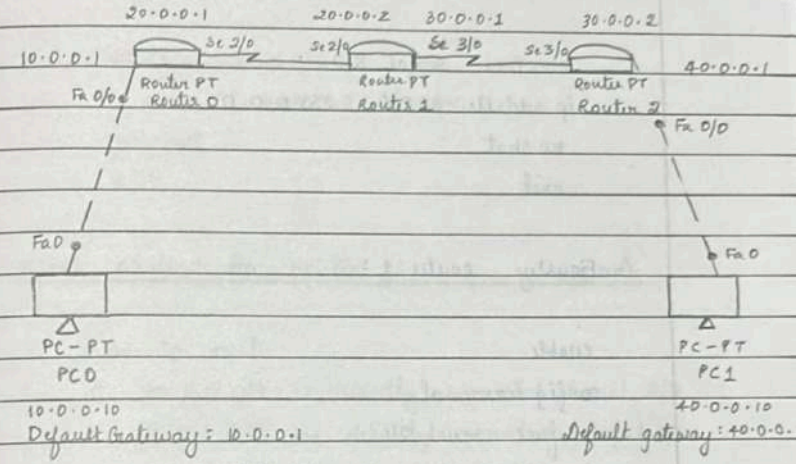
Procedure along with the topology:

Date 23/10/24
Page

Experiment - 3

Aim: Configure Default Route, Static Route to the router

Topology: 3 Routers are connected, 2 PC's are connected to one router each.



Procedure:

Configuring PC0: Ip address : 10.0.0.10
Default Gateway : 10.0.0.1

Configuring PC1 : Ip address : 40.0.0.10
Default gateway : 40.0.0.1

P.T.O

Configuring Router 0:

enable
config terminal
interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
no shut
exit

interface serial 2/0
ip address 20.0.0.1 255.0.0.0
no shut
exit

Configuring Router 1:

enable
config terminal
interface serial 2/0
ip address 20.0.0.2 255.0.0.0
no shut
exit

interface serial 3/0
ip address 30.0.0.1 255.0.0.0
no shut
exit

configuring Router 2:

enable
config terminal
interface fastethernet 0/0
ip address 40.0.0.1 255.0.0.0
no shut
exit

interface serial 3/0
ip address 30.0.0.2 255.0.0.0
no shut
exit

Static Routing for Router 1:

Show ip route
C 20.0.0.0/8 is directly connected Serial 2/0
C 30.0.0.0/8 is directly connected Serial 3/0

config terminal
ip route 10.0.0.0 255.0.0.0 20.0.0.2
ip route 40.0.0.0 255.0.0.0 30.0.0.2
exit

Show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected Serial 2/0
C 30.0.0.0/8 is directly connected serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

Default Routing for Router 0:

enable
config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.2
exit

show ip route
C 10.0.0.0/8 is directly connected Fast ethernet 0/0
C 20.0.0.0/8 is directly connected serial 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2

Default Routing for Router 2:

enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1
exit

show ip route
C 30.0.0.0/8 is directly connected, Serial 3/0
C 40.0.0.0/8 is directly connected, Fastethernet 0/0
S* 0.0.0.0/8 [1/0] via 30.0.0.1

Observations:

Ping command from PC0 to PC1

ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:

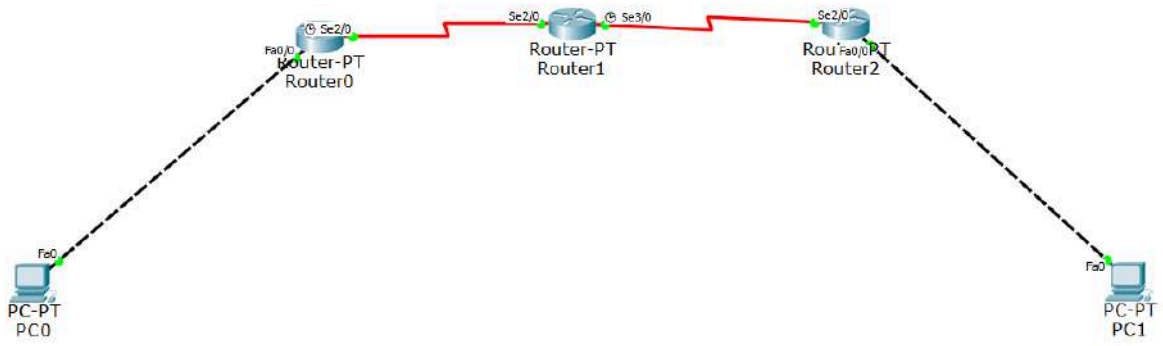
Request timed out

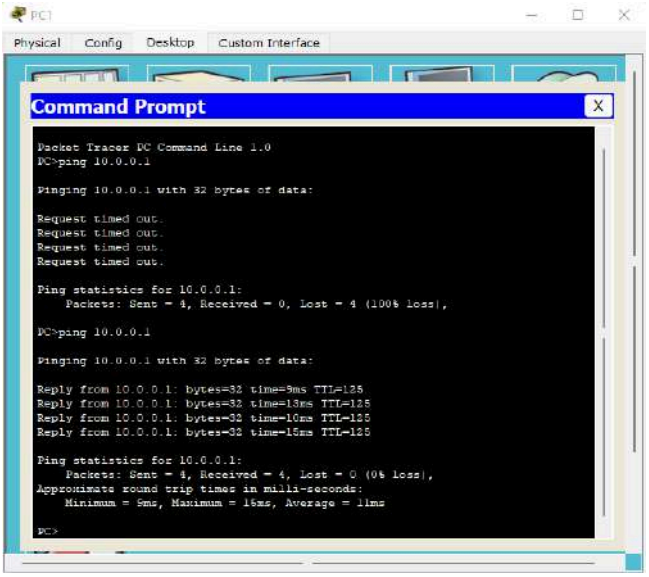
Ping command from PC1 to PC0

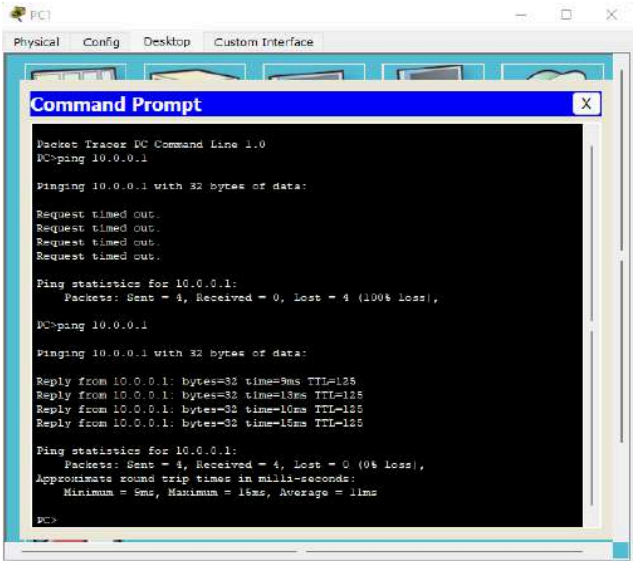
ping 10.0.0.10
Pinging 10.0.0.10 with 32 bytes of data

23/10/24

Screen shots/ output:


Se2/0
Fa0/0
Router-PT
Router0
Se2/0
Se3/0
Router-PT
Router1
Se2/0
Fa0/0
Router-PT
Router2
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1


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:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=9ms TTL=125
Reply from 10.0.0.1: bytes=32 time=13ms TTL=125
Reply from 10.0.0.1: bytes=32 time=10ms TTL=125
Reply from 10.0.0.1: bytes=32 time=15ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 9ms, Maximum = 15ms, Average = 11ms
PC>


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=9ms TTL=125
Reply from 10.0.0.1: bytes=32 time=13ms TTL=125
Reply from 10.0.0.1: bytes=32 time=10ms TTL=125
Reply from 10.0.0.1: bytes=32 time=15ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 9ms, Maximum = 15ms, Average = 11ms
PC>

PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

```
Pinging 40.0.0.2 with 32 bytes of data:

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

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>show ip route
Invalid Command.

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

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

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 13ms, Average = 9ms

PC>
```

Realtime

Scenario 0

New | Delete

Toggle PDU List Window

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

Router0

Physical | Config | CLI

IOS Command Line Interface

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

Gateway of last resort is not set

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

Gateway of last resort is not set

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

Copy | Paste

Observation:

Date __/__/__
Page _____

Observation :

Ping command from PC0 to PC1

ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data :

Request timed out

Ping command from PC1 to PC0

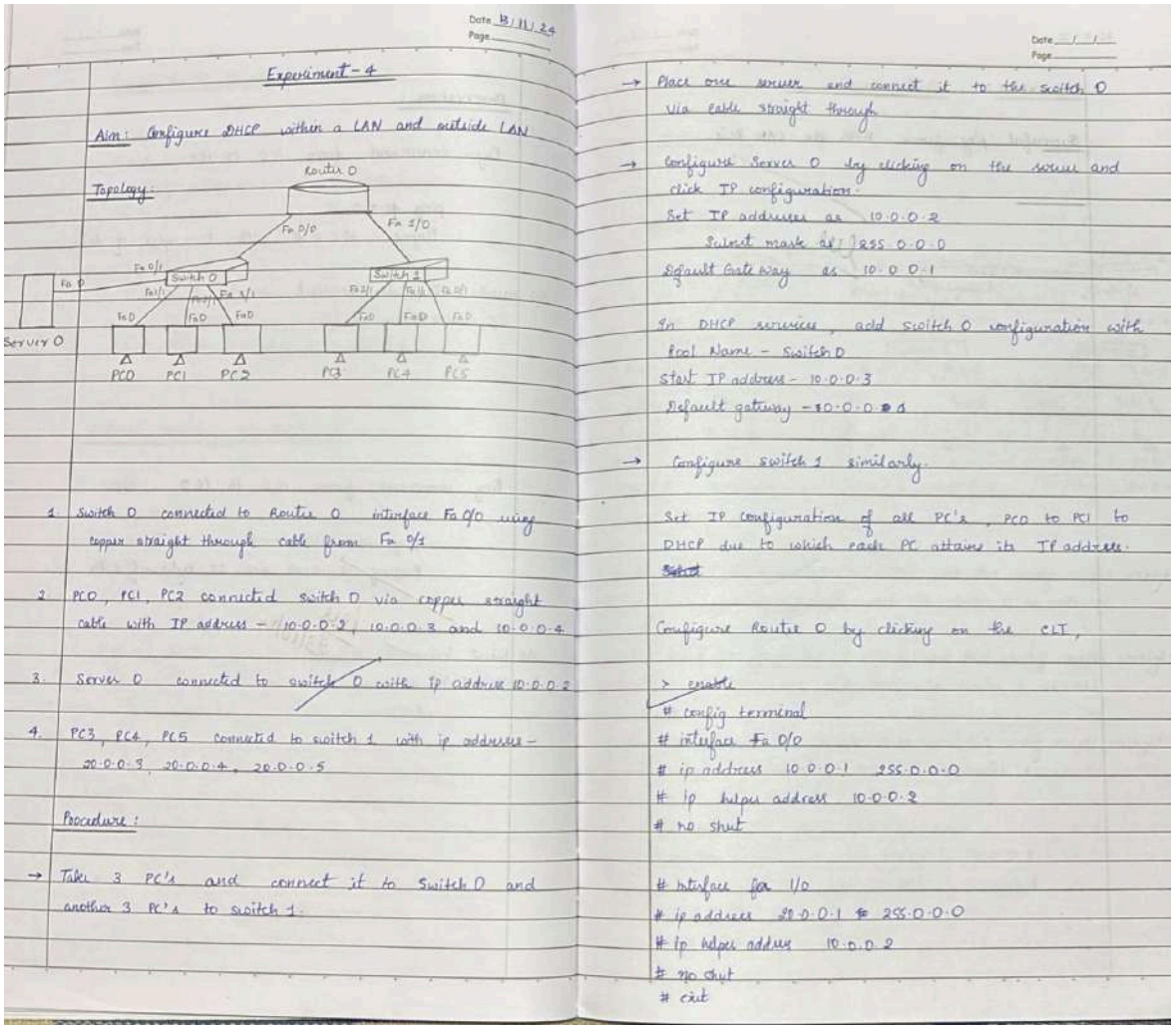
ping 10.0.0.10
Pinging 10.0.0.10 with 32 bytes of data

lee
23/10/24

Program 4

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

Procedure along with the topology:



Date 12/11/24

Experiment - 4

Aim: Configure DHCP within a LAN and outside LAN

Topology:

Router 0
Fa 0/0 Fa 1/0
Switch 0 Switch 1
Server 0
PC0 PC1 PC2 PC3 PC4 PC5

1. Switch 0 connected to Router 0 interface Fa 0/0 using copper straight through cable from Fa 0/1
2. PC0, PC1, PC2 connected switch 0 via copper straight cable with IP address - 10.0.0.2, 10.0.0.3 and 10.0.0.4
3. Server 0 connected to switch 0 with ip address 10.0.0.2
4. PC3, PC4, PC5 connected to switch 1 with ip addresses - 20.0.0.3, 20.0.0.4, 20.0.0.5

Procedure:

→ Take 3 PC's and connect it to Switch 0 and another 3 PC's to switch 1.

→ Place one server and connect it to the switch 0 via cable straight through

→ Configure Server 0 by clicking on the server and click IP configuration.
Set IP address as 10.0.0.2
Subnet mask as 255.0.0.0
Default Gateway as 10.0.0.1

In DHCP services, add switch 0 configuration with
Pool Name - Switch0
Start IP address - 10.0.0.3
Default gateway - 10.0.0.1

→ Configure switch 1 similarly.

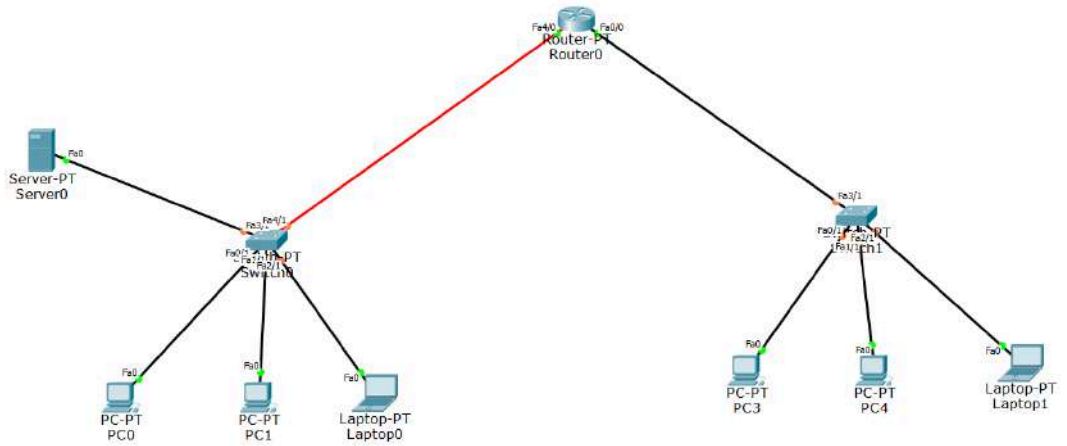
Set IP configuration of all PC's, PC0 to PC1 to DHCP due to which each PC attains its IP address.

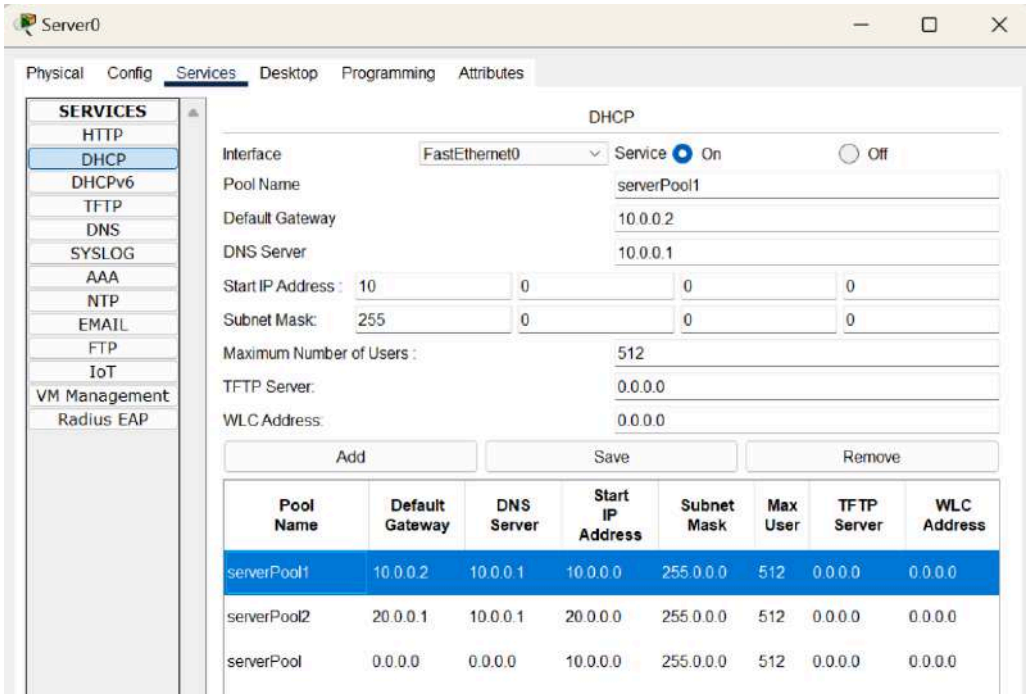
Configure Router 0 by clicking on the CLI,

> enable
config terminal
interface Fa 0/0
ip address 10.0.0.1 255.0.0.0
ip helper address 10.0.0.2
no shut

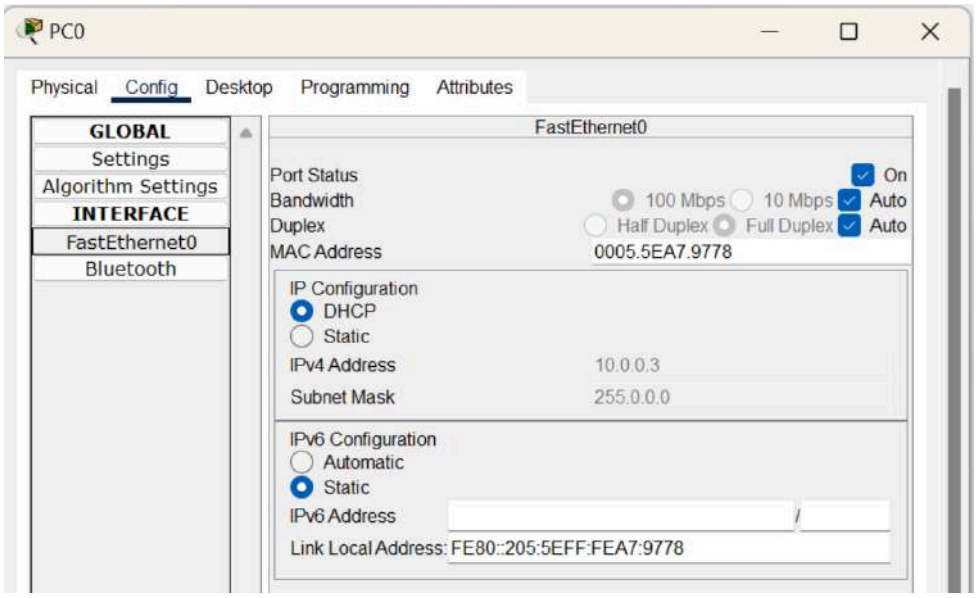
interface fa 1/0
ip address 20.0.0.1 255.0.0.0
ip helper address 10.0.0.2
no shut
exit

Screen shots/ output:

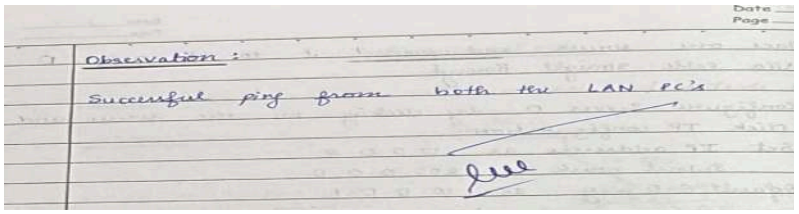




Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
	Successful	PC1	Laptop1	ICMP		0.004	N	1	(edit)	



Observation:



Program 5

Aim of the program:
Configure RIP routing Protocol in Routers

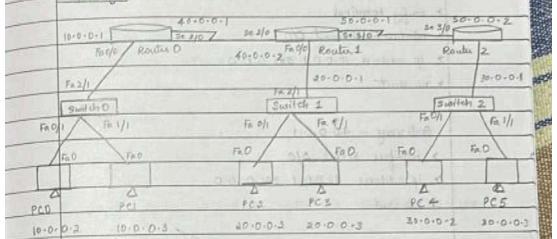
Procedure along with the topology:

Date 20/11/24
Page

Experiment - 5

Aim: Configure using RIP

Topology:



1. Switch 0 - connected to PC0 and PC1 using copper straight through cable and to Router 0 as well.
2. Switch 1 - connected to PC 2 and PC3 using copper straight through cable and to Router 1 as well
3. Switch 2 - connected to PC4 and PC5 using copper straight through cable and to Router 2.

IP addresses:

PC0 - 10.0.0.2
PC1 - 10.0.0.3
PC2 - 20.0.0.2
PC3 - 20.0.0.3
PC4 - 30.0.0.2
PC5 - 30.0.0.3

Page

Procedure:

Configure Router 0:

Default Gateway - 10.0.0.1 (Fast Ethernet)

> enable
> config terminal
> interface FastEthernet 0/0
> ip address 10.0.0.1 255.0.0.0
> no shut

Gateway - 40.0.0.1

> interface Serial 2/0
> ip address 40.0.0.1 255.0.0.0
> no shut

Configure Router 1

> enable
> config terminal
> interface FastEthernet 0/0
> ip address 20.0.0.1
> no shut
> interface Serial 2/0
> ip address 40.0.0.2
> no shut
> interface Serial 3/0
> ip address 50.0.0.1
> no shut

Page

Configure Router 2:

> enable
> config terminal
> interface FastEthernet 0/0
> ip address 30.0.0.1

> interface Serial 3/0
> ip address 50.0.0.2

Observation:

> ping 30.0.0.2

Reply from 30.0.0.2 bytes = 32 time = 6ms TTL = 125

Ping statistics for 30.0.0.2:

Packets : Sent = 4, Recieved = 3, Loss = 1 (25% loss)

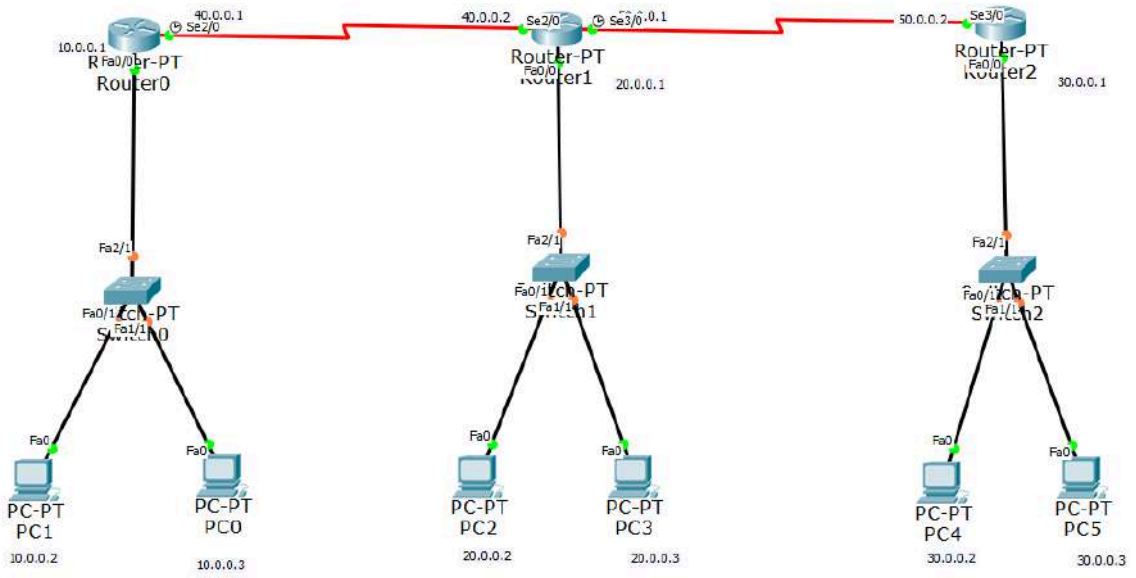
Approximate Round trip times in milliseconds:

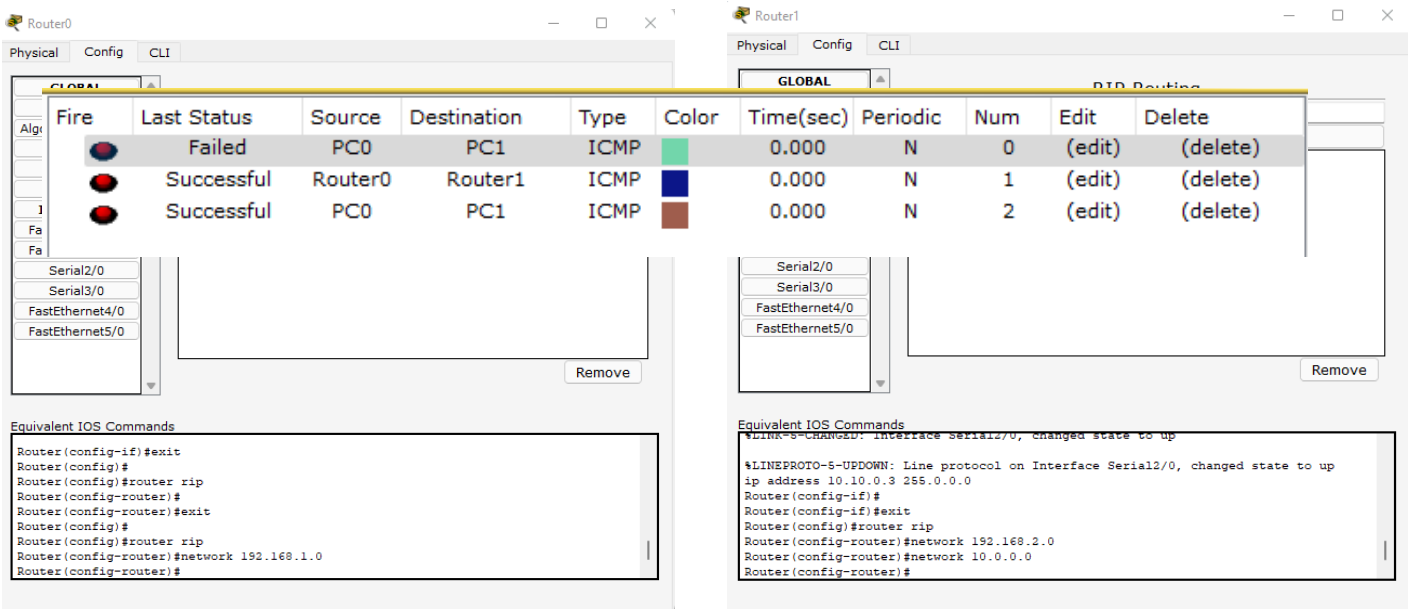
Min = 6ms, Max = 7ms, Average = 6ms

Routing using RIP for all routers

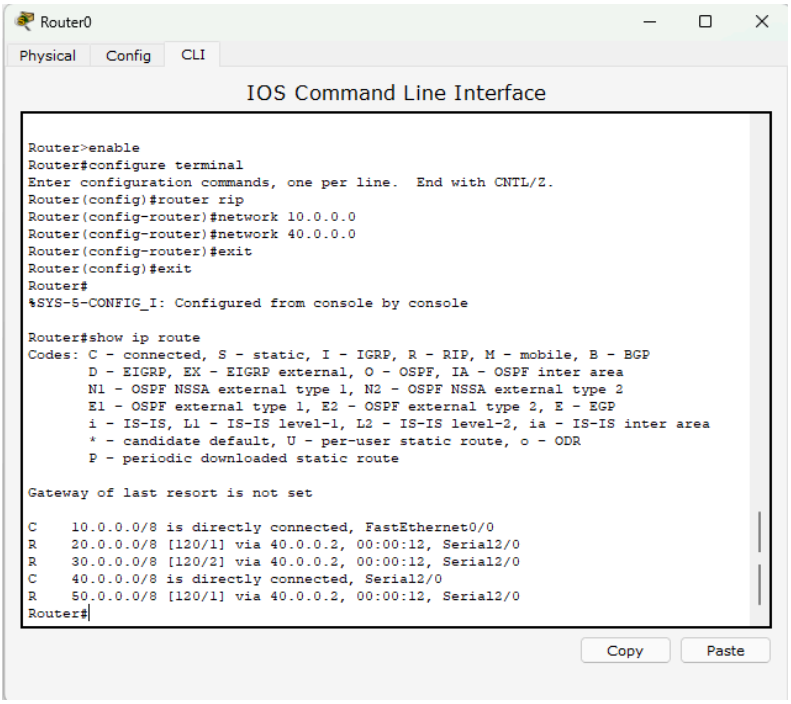
	Router 0	Router 1	Router 2
Router(config)	# router rip	# router rip	# router rip
	# network 10.0.0.0	# network 40.0.0.0	# network 30.0.0.
	# network 30.0.0.0	# network 20.0.0.0	# network 50.0.0.
		# network 50.0.0.0	

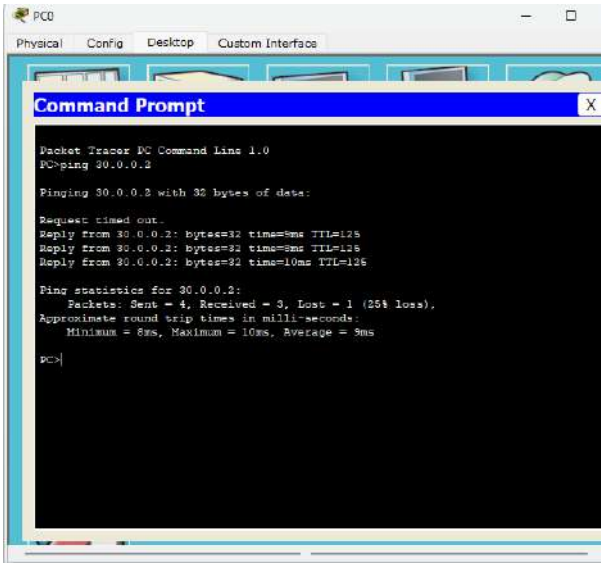
Screen shots/ output:



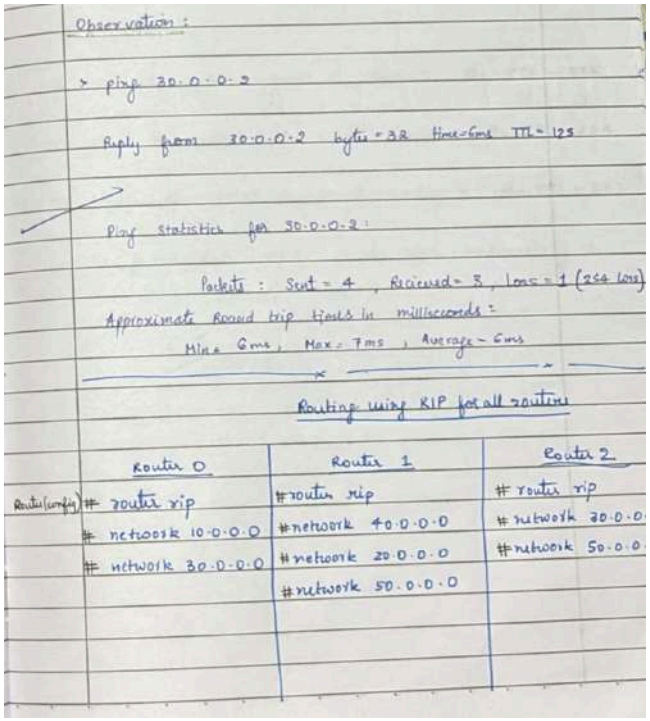


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Failed	PC0	PC1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	Router0	Router1	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC0	PC1	ICMP		0.000	N	2	(edit)	(delete)





Observation:



Observation :

> ping 30.0.0.2

Reply from 30.0.0.2 byte = 32 time = 6ms TTL = 125

Ping statistics for 30.0.0.2 :

Packets : Sent = 4 , Recieved = 3 , Loss = 1 (25% loss)

Approximate Round trip times in milliseconds :

Min = 6ms , Max = 7ms , Average = 6ms

Routing using RIP for all routers

	Router 0	Router 1	Router 2
Router(config)	# router rip	# router rip	# router rip
	# network 10.0.0.0	# network 40.0.0.0	# network 20.0.0.
	# network 30.0.0.0	# network 20.0.0.0	# network 50.0.0.
		# network 50.0.0.0	

Program 6

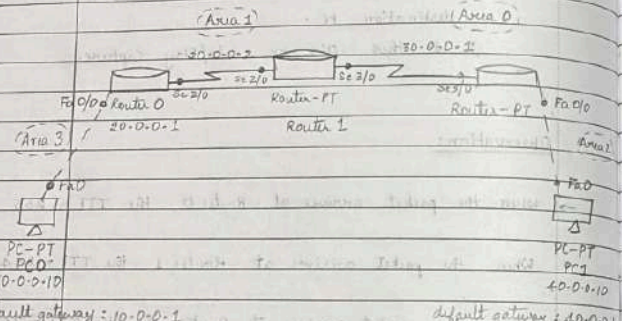
Configure OSPF routing protocol

Procedure along with the topology

Experiment - 7

Aim: To configure OSPF routing protocol

Topology



PC0

→ connected to router 0 via the fastethernet 0/0 interface

→ default gateway 10.0.0.1

PC1

→ connected to router 2 via the Fastethernet 0/0 interface

→ default gateway 40.0.0.1

Procedure

For Router 0,

```
config terminal
interface fastethernet 2/0
ip address 10.0.0.1 255.0.0.0
no shut down
exit
interface serial 2/0
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shut down
exit
```

For Router 1,

```
interface serial 2/0
ip address 20.0.0.2 255.0.0.0
encapsulation ppp
no shutdown
exit
interface serial 3/0
ip address 30.0.0.1 255.0.0.0
encapsulation ppp
no shutdown
exit
```

For Router 2

interface Serial 3/0
ip address 30.0.0.2 255.0.0.0
encapsulation ppp
no shutdown
exit
interface fastEthernet 3/0
ip address 40.0.0.1 255.0.0.0
no shutdown
exit

Now, enable ip routing by configuration ospf routing protocol in all routers.

Router 0

router ospf 1
router-id 1.1.1.1
network 10.0.0.0 0.255.255.255 area 3
network 20.0.0.0 0.255.255.255 area 1
exit

Router 1

router ospf 1
router-id 2.2.2.2
network 20.0.0.0 0.255.255.255 area 1
network 30.0.0.0 0.255.255.255 area 0
exit

Router 3

router ospf 1
router-id : 3.3.3.3
network 30.0.0.0 0.255.255.255 area 0
network 40.0.0.0 0.255.255.255 area 2
exit

Now check routing table of Router 1

C 10.0.0.0/8 is directly connected, Fa 2/0
C 20.0.0.0/8 is directly connected, Serial 2/0
O IA 40.0.0.0/8 [40/129] via 30.0.0.2, 00:04:23,
O IA 30.0.0.0/8 [110/129] via 30.0.0.2, 00:07:29, Serial

STEPS - Configure loopback

Router 0

interface loopback 0
ip add 172.16.1.252 255.255.0.0

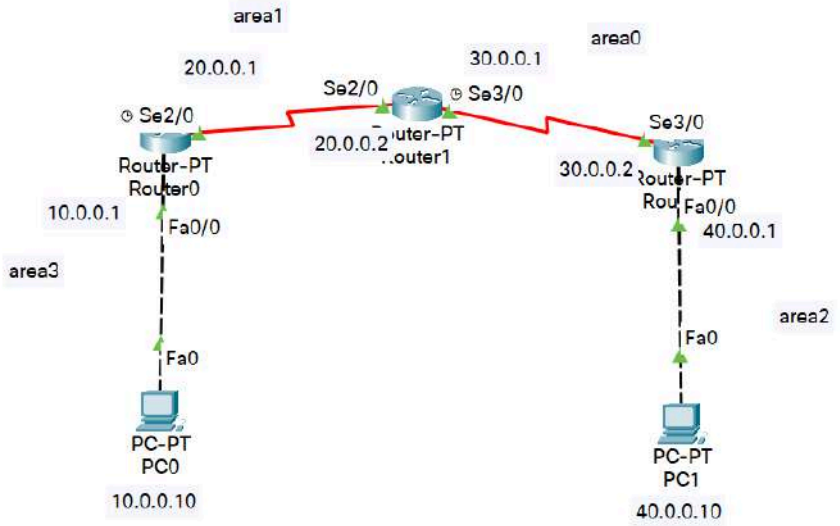
Router 1

interface loopback 0
ip add 172.16.1.253 255.255.0.0
no shutdown

Router 2

interface loopback 0
ip add 172.16.1.254 255.255.0.0
no shutdown

Screen shots/ output



Router0

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

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

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

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

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

Router1

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

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

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

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

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

Router2

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

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

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

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

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

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

OSPF Routing Protocol

Router0

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

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

Gateway of last resort is not set

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

Router1

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

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

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

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

Gateway of last resort is not set

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

Router2

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

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

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

Gateway of last resort is not set

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

Pinging

```
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=24ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=20ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 24ms, Average = 20ms
```

Observation

Date __/__/__
Page

Now check Routing table for Router 3

STEP 6 - Create virtual link b/w R1, R2, by this we create a virtual link to connect area 3 to area 0

Router 0
router ospf 1
area 1 virtual-link 2.2.2.2
fab 10 .....,

Router 1
router ospf 1
area 1 virtual-link 1.1.1.1
exit

STEP 7 - R2 and R3 get update about Area 3

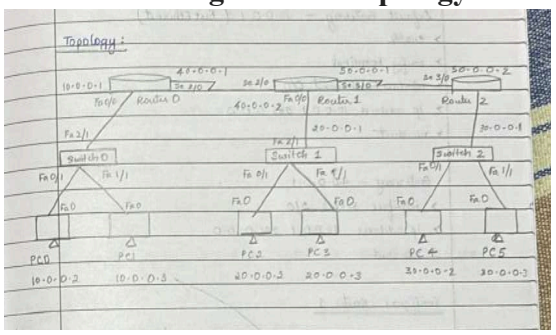
Observation

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

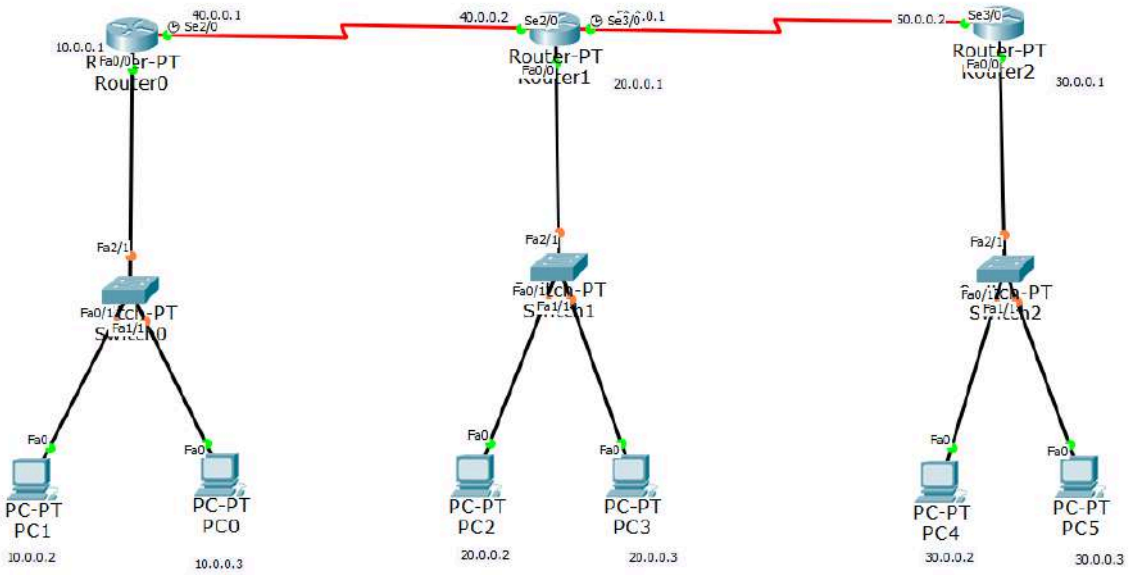
Program 7

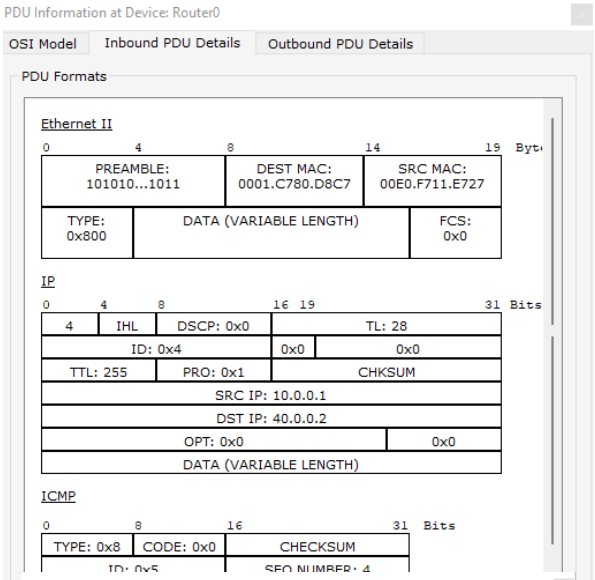
Aim of the program:
Demonstrate the TTL/ Life of a Packet

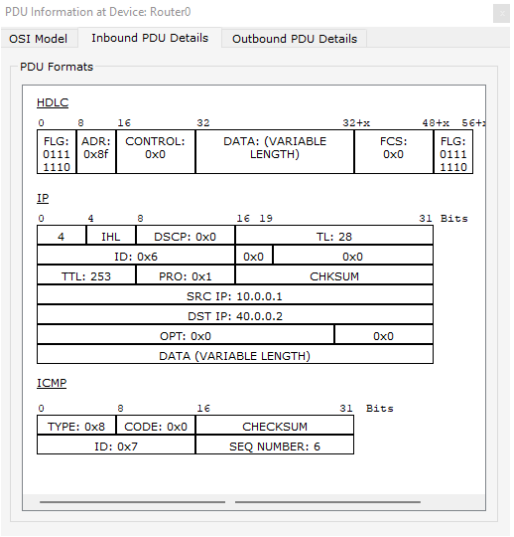
Procedure along with the topology:

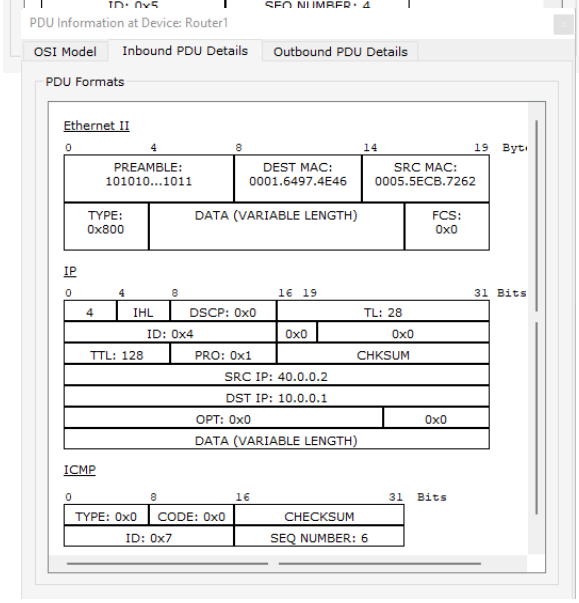


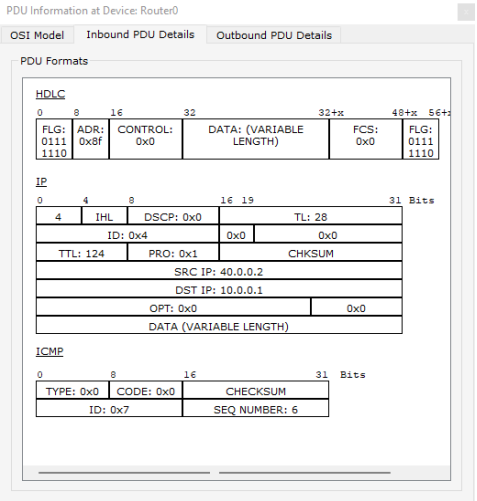
Screen shots/ output:











Observation:

Date __/__/__
Page _____

Procedure:

→ Create the topology as mentioned above
→ Configure all the routers :- Router 0, Router 1, Router 2
→ In simulation layout, we select Simple PDU
  - In that we select a source PC and a destination PC.
  - We then click on Auto/Play Capture.

Observation:

→ When the packet arrives at Router 0, the TTL = 255

→ When the packet arrives at Router 1, the TTL = 254

→ When the packet arrives at Router 2, the TTL = 253

So from each router, the TTL reduces by 1.

20/11/24

Program 8

Aim of the program:
Configure Web Server, DNS within a LAN.

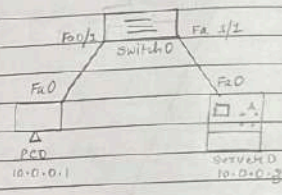
Procedure along with the topology:

Date __/__/__
Page ____

Experiment - 8

Aim - To configure DNS server to demonstrate the mapping of IP addresses and domain names.

Topology



Connect a PC and a server to a switch, assign ip address as 10.0.0.1 and 10.0.0.3

PC0 → 10.0.0.1
Server 0 → 10.0.0.3

Connect PC0 and Server via a switch - PT

For Server 0,

Go to Server → Services → DNS → Enable on
In the text fields add,
name : abc
address : 10.0.0.3
click on add
Click edit for under HTML [change if needed]
click save

Date __/__/__
Page ____

Procedure

1. Go to PC0 → Desktop → Web Browser
2. Search 'abc' in url box (or)
3. Search 10.0.0.3 in url box

Output -

Cisco Packet Tracer
Welcome to Cisco packet Tracer - - - - - -

Quick links:
A small page
Copyrights
Image Page
Image

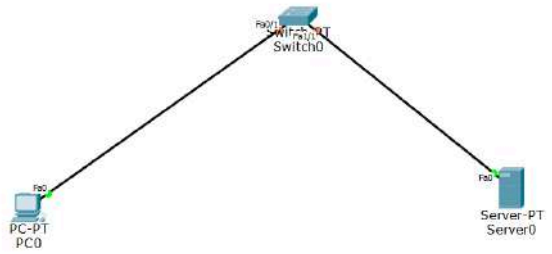
Observations :

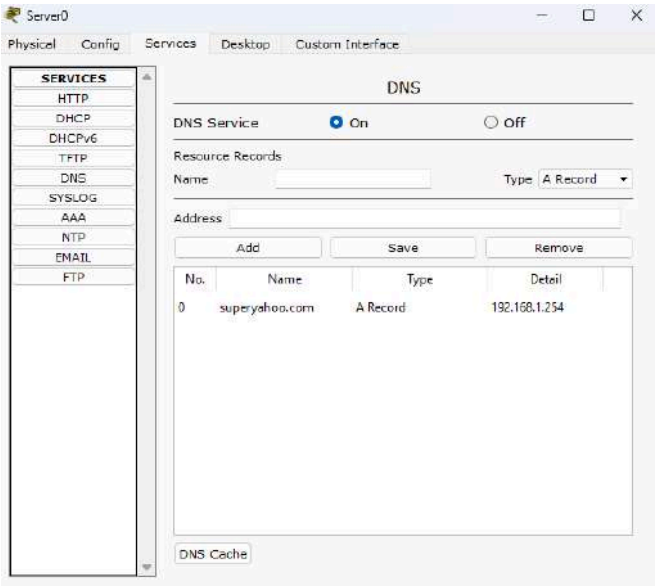
→ DNS translates Domain names to ip addresses
It simplifies accessing websites by using human readable names

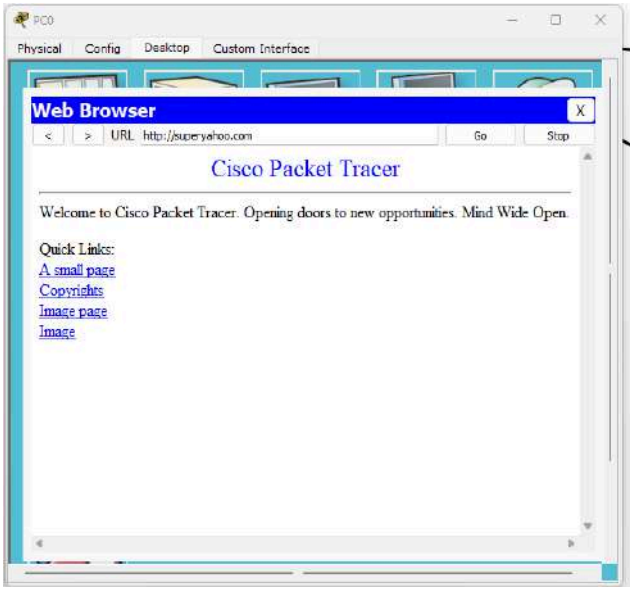
→ In this experiment, a web service and DNS were configured within a LAN to map domain names to ip addresses.

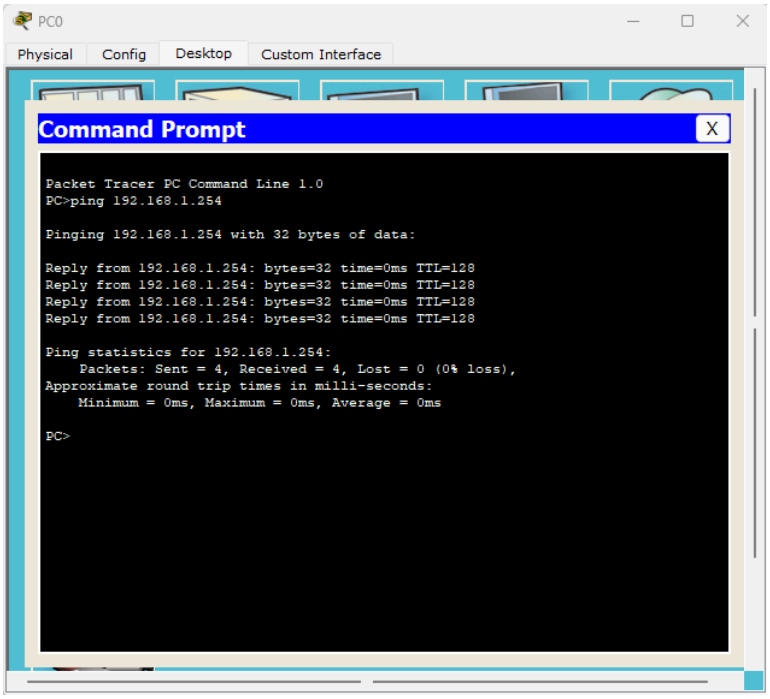
→ The configuration was successful allowing the web page to be accessed via both methods.

Screen shots/ output:

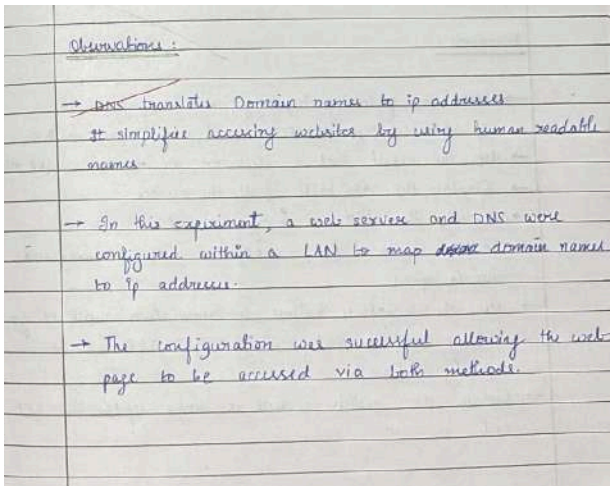








Observation:



Observations :

→ DNS translates Domain names to ip addresses & simplifies accessing websites by using human readable names.

→ In this experiment, a web service and DNS were configured within a LAN to map domain names to ip addresses.

→ The configuration was successful allowing the web page to be accessed via both methods.

Program 9

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

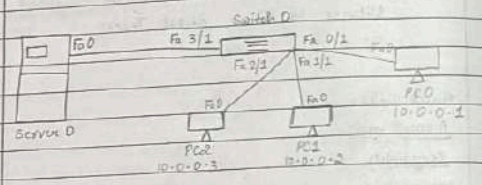
Procedure along with the topology:

Date __/__/__
Page

Experiment - 9

Aim - To construct a simple LAN and understand the concept and operation of Address Resolution Protocol

Topology

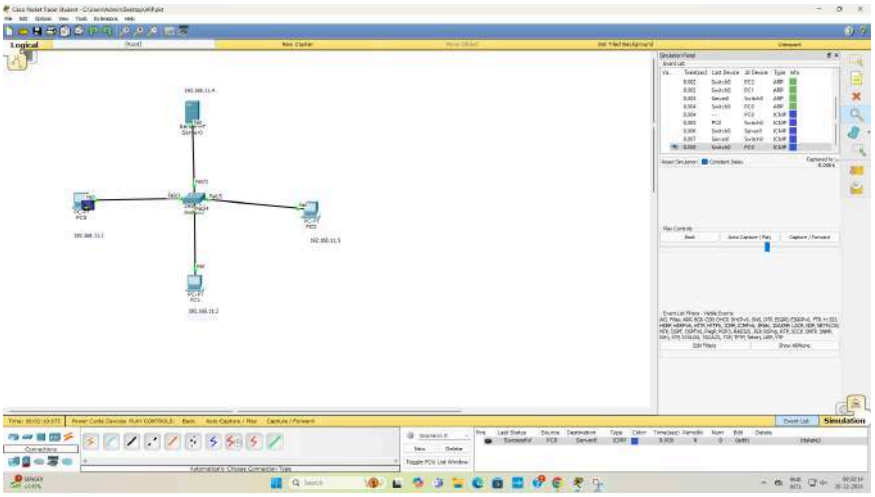


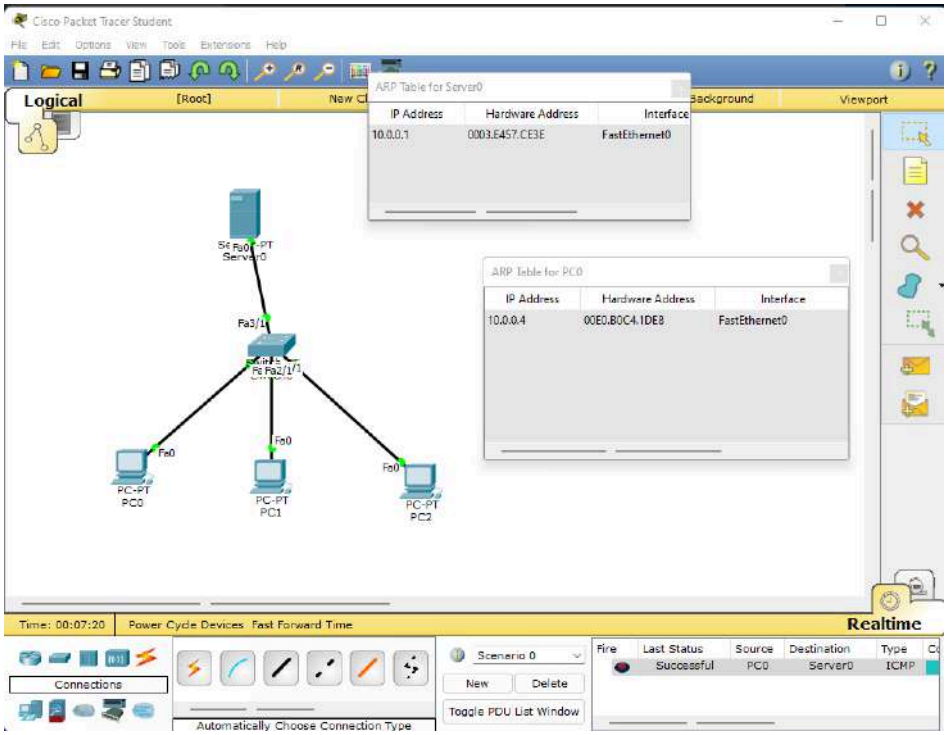
Switch connected to 3 PC's and a server via three fastethernet interfaces and one ethernet interface

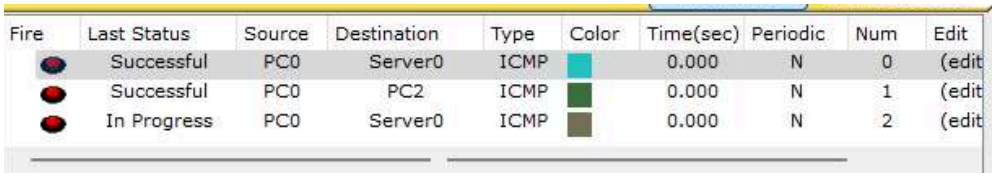
Procedure:

→ Make the following topology and configure the ip addresses for all PC's and also subnet mask's
→ Use the inspect tool, click on PC to view ARP table
→ Display the ARP table of all the devices
→ Initially the ARP is empty for all
→ In the Switch's CLI, show mac address command can be given
→ Use the capture button in simulation panel to go step by step so that changes in ARP can be clearly noted.
→ Observe the switch as well as nodes update the ARP table

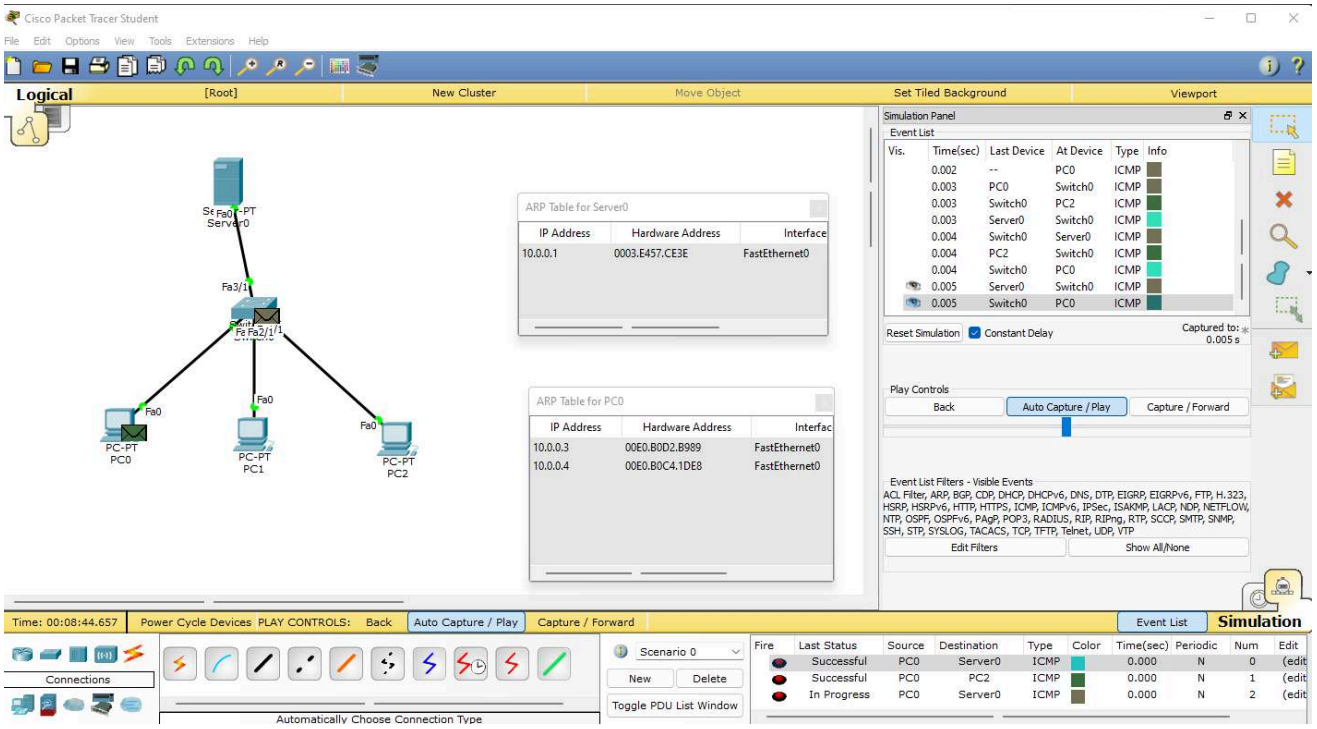
Screen shots/ output:

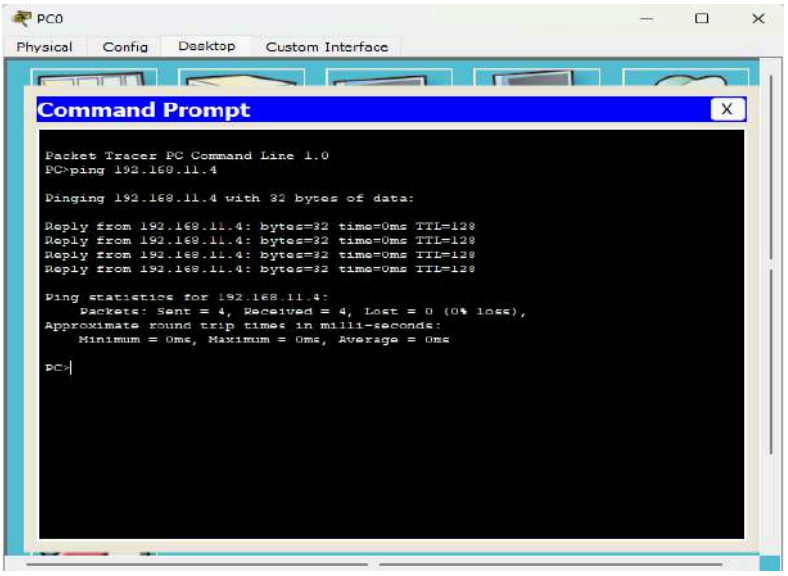


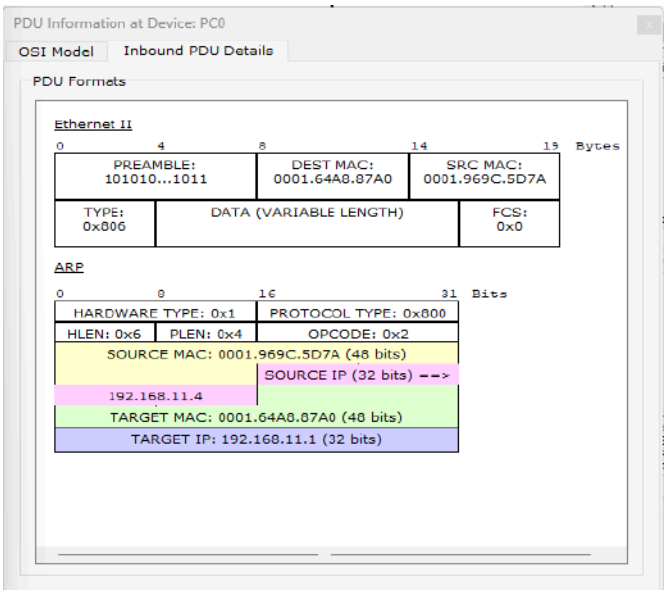




Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC0	Server0	ICMP		0.000	N	0	(edit
	Successful	PC0	PC2	ICMP		0.000	N	1	(edit
	In Progress	PC0	Server0	ICMP		0.000	N	2	(edit


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Server0
Fa3/1
Fa2/1
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
ARP Table for Server0
IP Address
Hardware Address
Interface
10.0.0.1
0003.E457.CE3E
FastEthernet0
ARP Table for PC0
IP Address
Hardware Address
Interfac
10.0.0.3
00E0.B0D2.B989
FastEthernet0
10.0.0.4
00E0.B0C4.1DE8
FastEthernet0
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.002 -- PC0 ICMP
0.003 PC0 Switch0 ICMP
0.003 Switch0 PC2 ICMP
0.003 Server0 Switch0 ICMP
0.004 Switch0 Server0 ICMP
0.004 PC2 Switch0 ICMP
0.004 Switch0 PC0 ICMP
0.005 Server0 Switch0 ICMP
0.005 Switch0 PC0 ICMP
Reset Simulation
Constant Delay
Captured to: 0.005 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Time: 00:08:44.657
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Successful PC0 Server0 ICMP 0.000 N 0
Successful PC0 PC2 ICMP 0.000 N 1
In Progress PC0 Server0 ICMP 0.000 N 2


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.11.4
Pinging 192.168.11.4 with 32 bytes of data:
Reply from 192.168.11.4: bytes=32 time=0ms TTL=128
Reply from 192.168.11.4: bytes=32 time=0ms TTL=128
Reply from 192.168.11.4: bytes=32 time=0ms TTL=128
Reply from 192.168.11.4: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.11.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>


PDU Information at Device: PC0
OSI Model
Inbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Bytes
PREAMBLE: 101010...1011
DEST MAC: 0001.64A8.87A0
SRC MAC: 0001.969C.5D7A
TYPE: 0x806
DATA (VARIABLE LENGTH)
FCS: 0x0
ARP
0 8 16 31 Bits
HARDWARE TYPE: 0x1
PROTOCOL TYPE: 0x800
HLEN: 0x6
PLEN: 0x4
OPCODE: 0x2
SOURCE MAC: 0001.969C.5D7A (48 bits)
SOURCE IP (32 bits) -->
192.168.11.4
TARGET MAC: 0001.64A8.87A0 (48 bits)
TARGET IP: 192.168.11.1 (32 bits)

Observation:

Date __/__/__
Page

Observation

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	0060.2F99.8C88	FastEthernet 0

ARP table for PC2 (destination)

IP address	Hardware Address	Interface
10.0.0.1	00DD.D302.9608	FastEthernet 0

Program 10

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

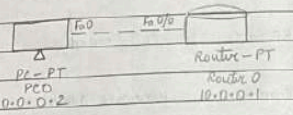
Procedure along with the topology:

Date __/__/__
Page _____

Experiment - 10

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

Topology:



A router connected to a single PC via a fastethernet interface with copper cross over cable

Procedure:

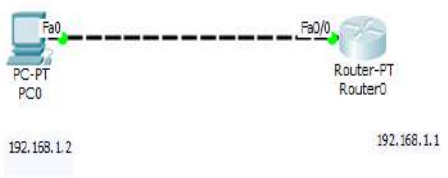
Configure the router

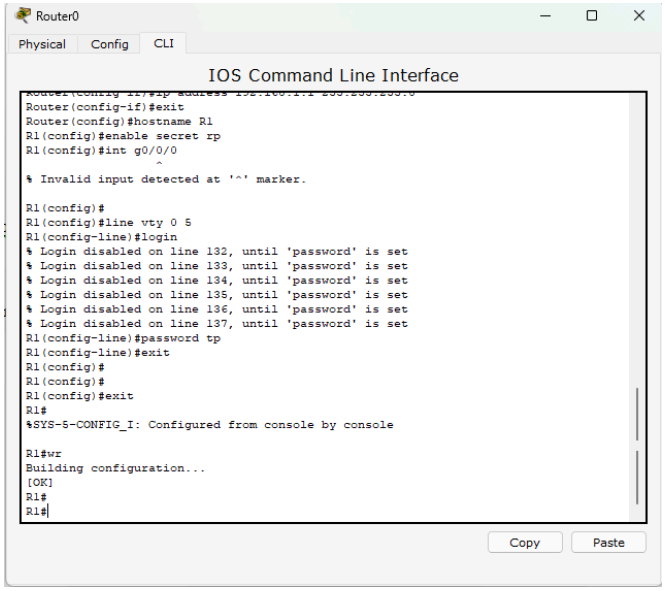
```
enable
config
hostname r1
enable secret p1
interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
no shut

line vty 0 5
login
password p0
exit
exit
wr
```

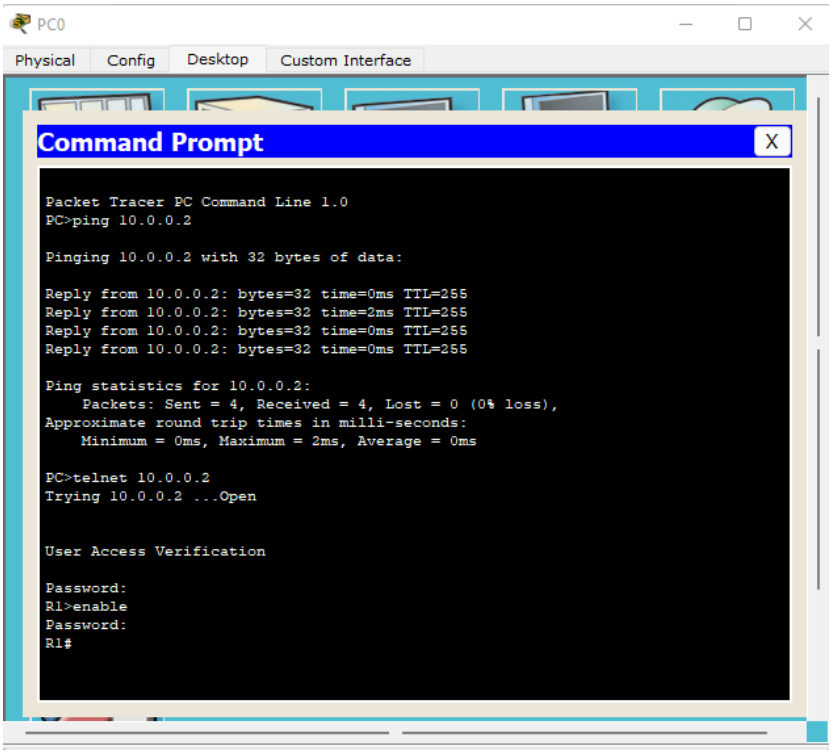
PC - Command Prompt

```
> telnet 10.0.0.1
```

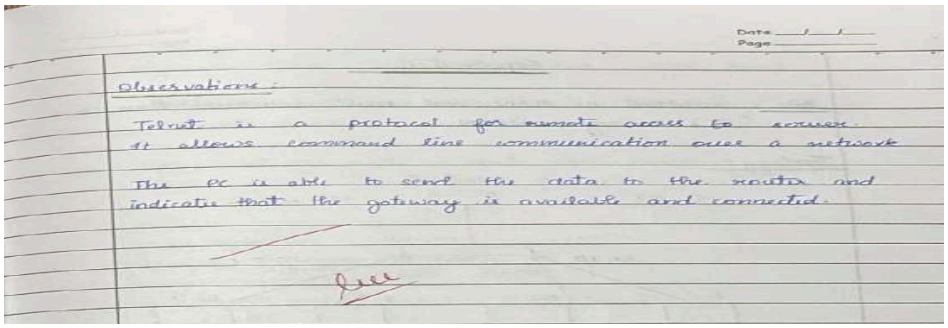
Screen shots/ output:







Observation:



Observations :

Telnet is a protocol for remote access to servers. It allows command line communication over a network.

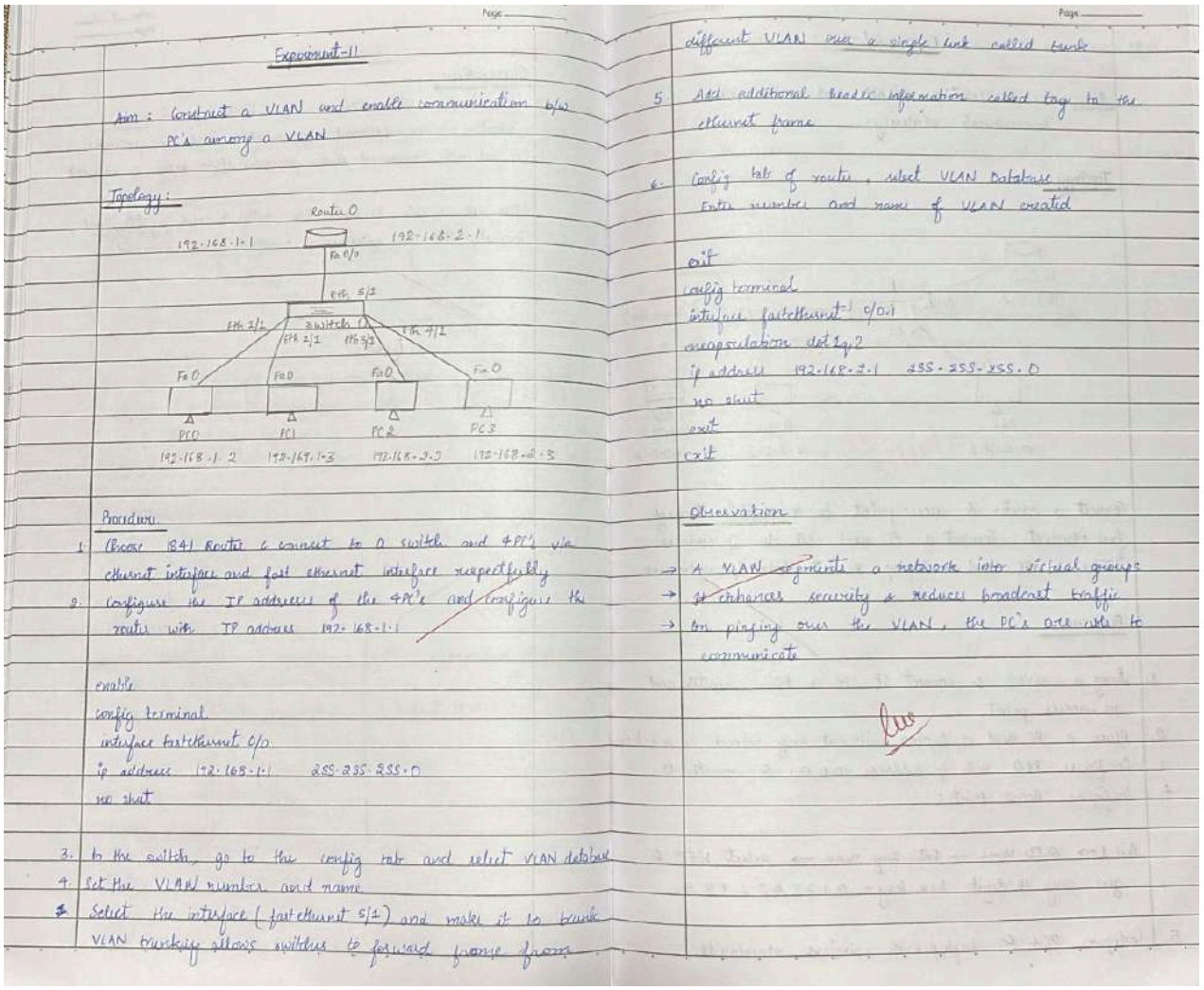
The PC is able to send the data to the router and indicate that the gateway is available and connected.

Program 11

Aim of the program:

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

Procedure along with the topology:



Experiment-11

Aim: Construct a VLAN and enable communication b/w PC's among a VLAN

Topology:

Procedure

1. Choose 1841 Router & connect to a switch and 4 PC's via ethernet interface and fast ethernet interface respectfully
2. Configure the IP addresses of the 4 PC's and configure the router with IP address 192.168.1.1

enable
config terminal
interface fastethernet 0/0
ip address 192.168.1.1 255.255.255.0
no shut

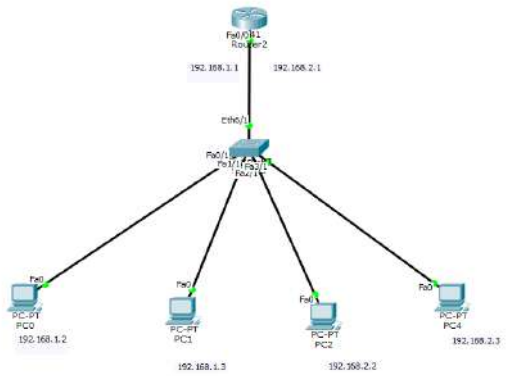
3. In the switch, go to the config tab and select VLAN database
4. Set the VLAN number and name
5. Select the interface (fastethernet 5/1) and make it to trunk
VLAN trunking allows switches to forward frames from different VLAN over a single link called trunk
5. Add additional header information called tag to the ethernet frame
6. Config tab of router, select VLAN database
Enter number and name of VLAN created

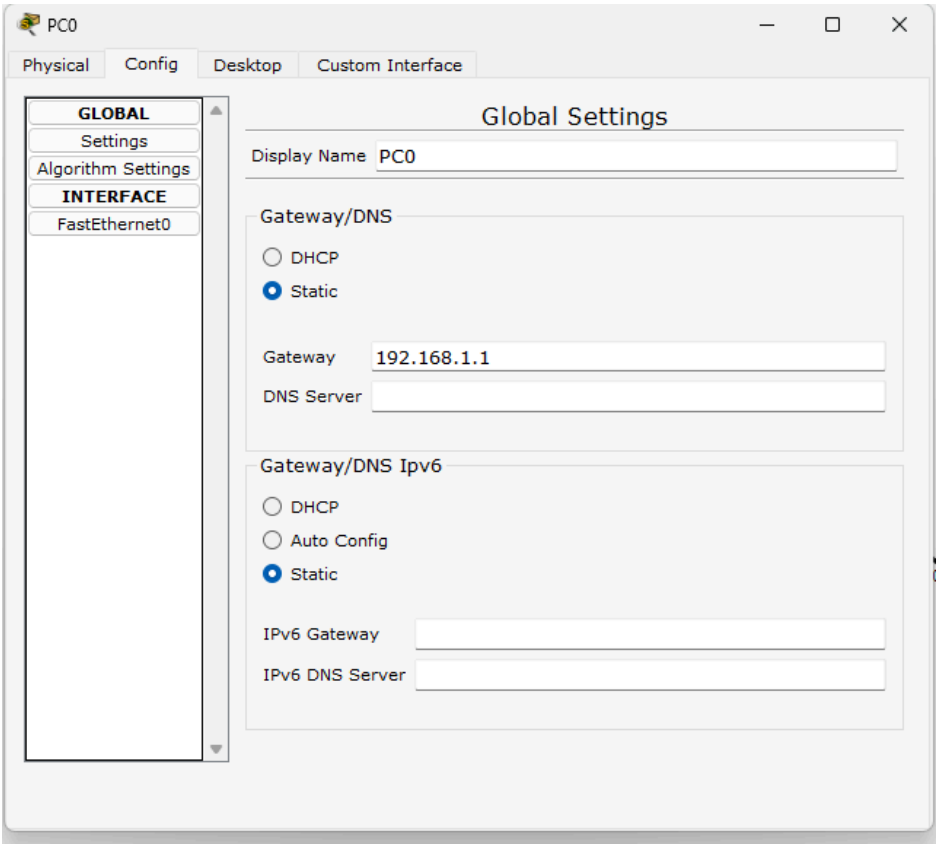
exit
config terminal
interface fastethernet 0/0.1
encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0
no shut
exit
exit

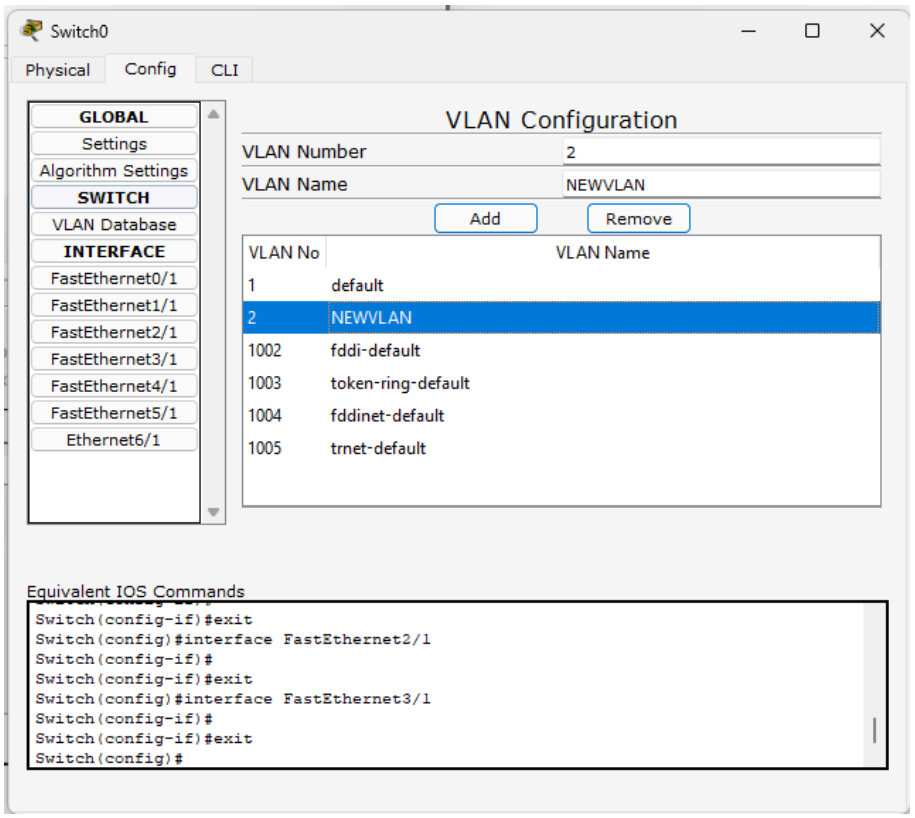
Observation

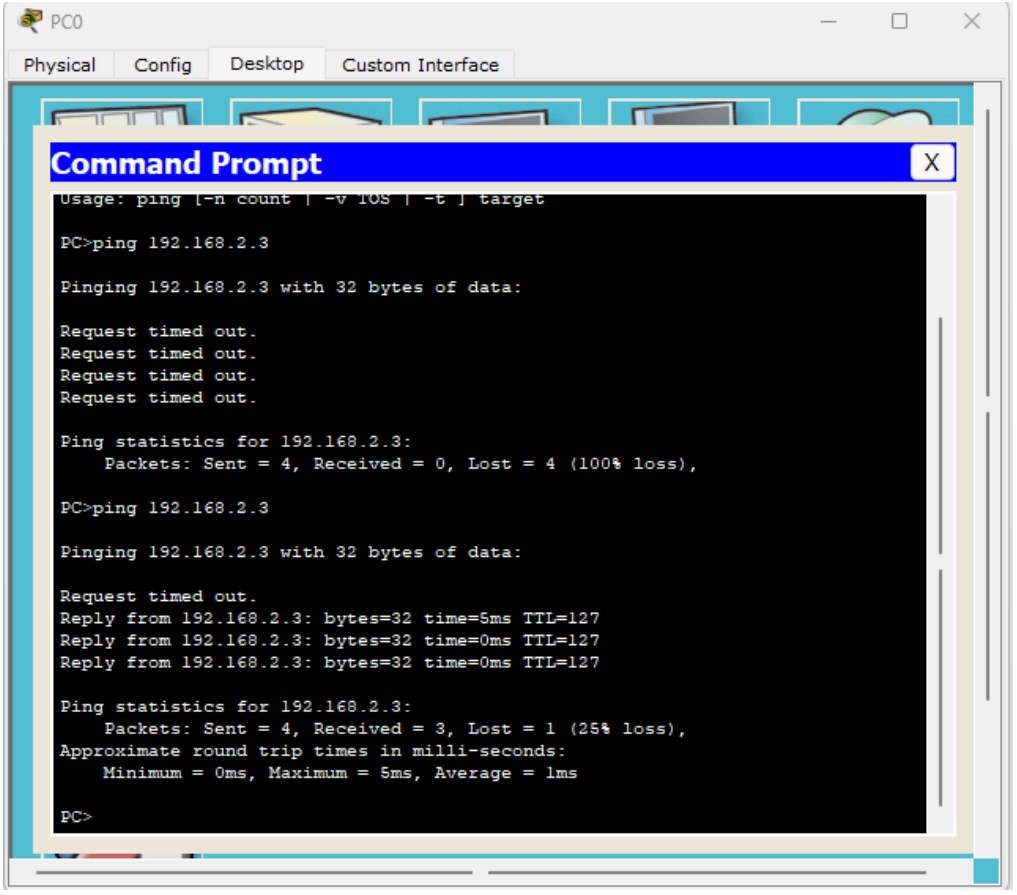
→ A VLAN segments a network into virtual groups
→ It enhances security & reduces broadcast traffic
→ On pinging over the VLAN, the PC's are able to communicate

Screen shots/ output:

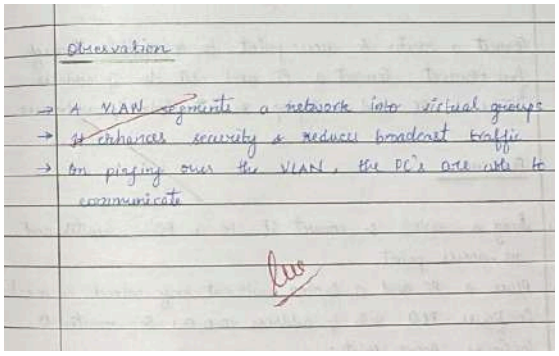



PC0
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Global Settings
Display Name
PC0
Gateway/DNS
DHCP
Static
Gateway
192.168.1.1
DNS Server
Gateway/DNS Ipv6
DHCP
Auto Config
Static
IPv6 Gateway
IPv6 DNS Server


Switch0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet1/1
FastEthernet2/1
FastEthernet3/1
FastEthernet4/1
FastEthernet5/1
Ethernet6/1
VLAN Configuration
VLAN Number
2
VLAN Name
NEWVLAN
Add
Remove
VLAN No
VLAN Name
1
default
2
NEWVLAN
1002
fddi-default
1003
token-ring-default
1004
fddinet-default
1005
trnet-default
Equivalent IOS Commands
Switch(config-if)#exit
Switch(config)#interface FastEthernet2/1
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#interface FastEthernet3/1
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#



Observation:



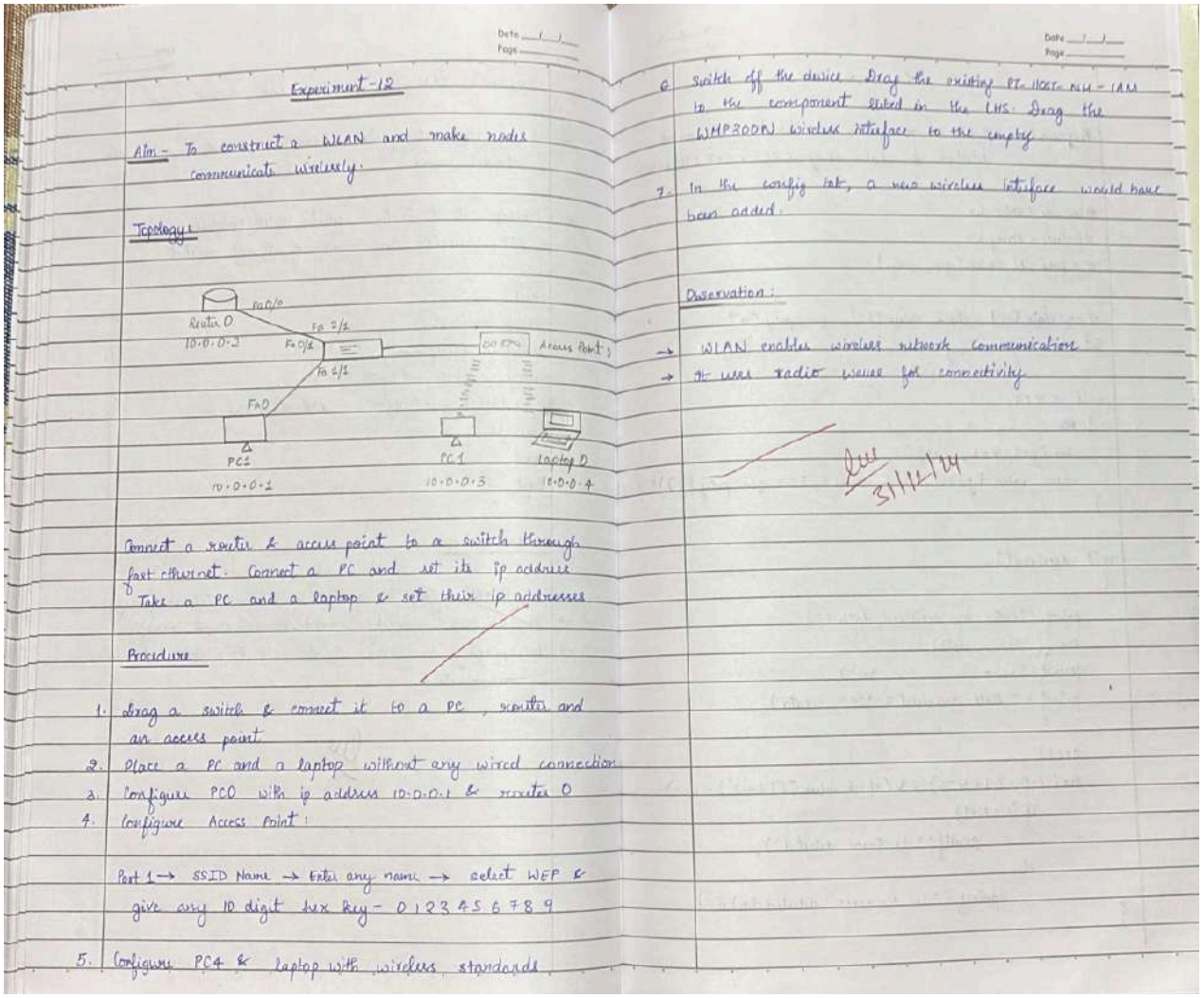
Observation

→ A VLAN segments a network into virtual groups
→ It enhances security & reduces broadcast traffic
→ In pinging over the VLAN, the PC's are able to communicate

Program 12

Aim of the program:
To construct a WLAN and make the nodes communicate wirelessly

Procedure along with the topology:



Experiment-12

Aim:- To construct a WLAN and make nodes communicate wirelessly.

Topology:

Connect a router & access point to a switch through fast ethernet. Connect a PC and set its ip address. Take a PC and a laptop & set their ip addresses

Procedure

1. Drag a switch & connect it to a PC, router and an access point.
2. Place a PC and a laptop without any wired connection
3. Configure PC0 with ip address 10.0.0.1 & router 0
4. Configure Access Point:

Port 1 → SSID Name → Enter any name → select WEP & give any 10 digit hex key - 0123456789

5. Configure PC4 & laptop with wireless standards.

6. Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag the WMP300N wireless interface to the empty

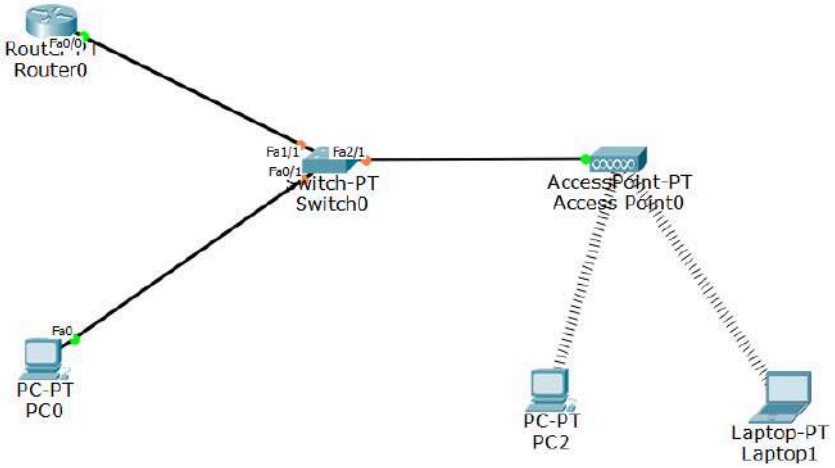
7. In the config tab, a new wireless interface would have been added.

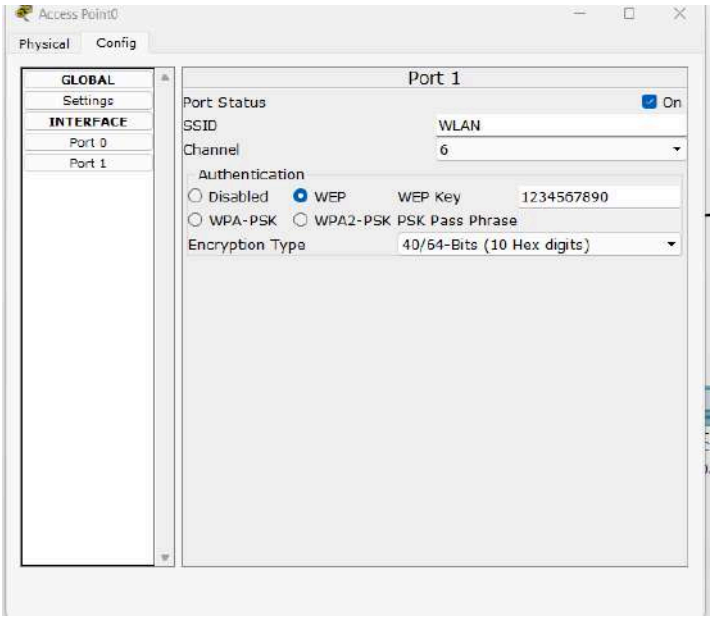
Observation:

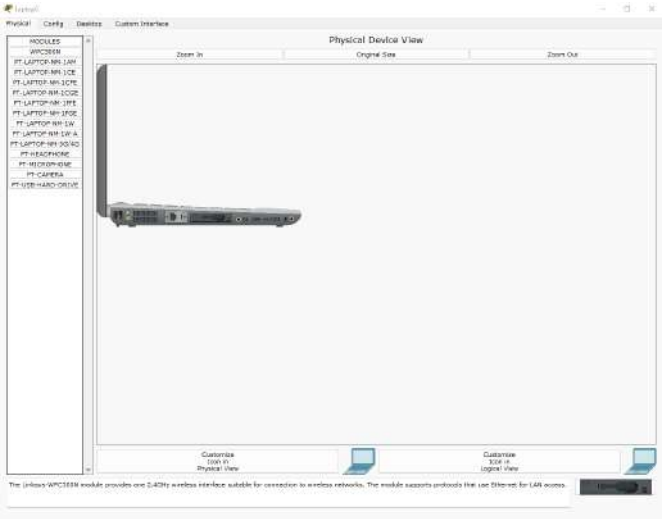
→ WLAN enables wireless network communication
→ It uses radio waves for connectivity

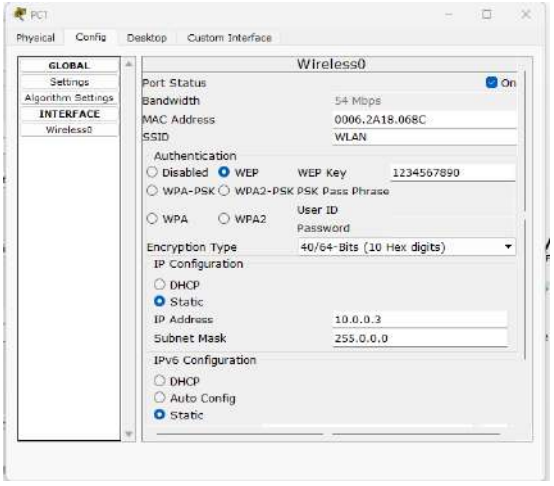
31/12/24

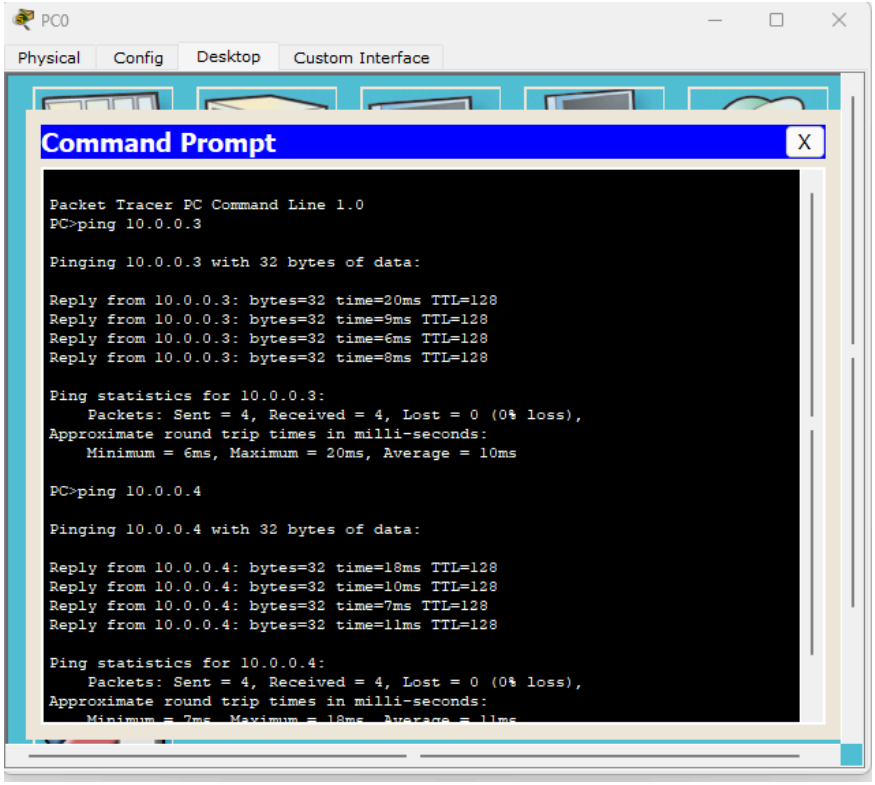
Screen shots/ output :



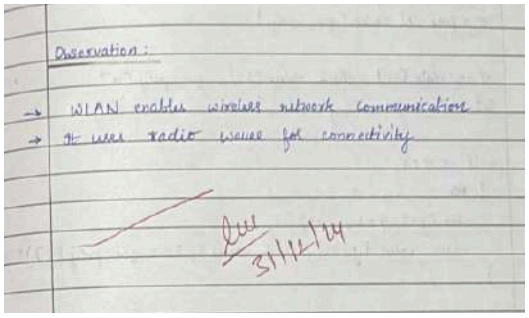

Access Point0
Physical
Config
GLOBAL
Settings
INTERFACE
Port 0
Port 1
Port 1
Port Status
On
SSID
WLAN
Channel
6
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Encryption Type
40/64-Bits (10 Hex digits)


Physical Device View


PC1
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
Wireless0
Port Status
On
Bandwidth
54 Mbps
MAC Address
0006.2A18.068C
SSID
WLAN
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
WPA
WPA2
User ID
Password
Encryption Type
40/64-Bits (10 Hex digits)
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static



Observation:



Observation:

→ WLAN enables wireless network communication

→ It uses radio waves for connectivity

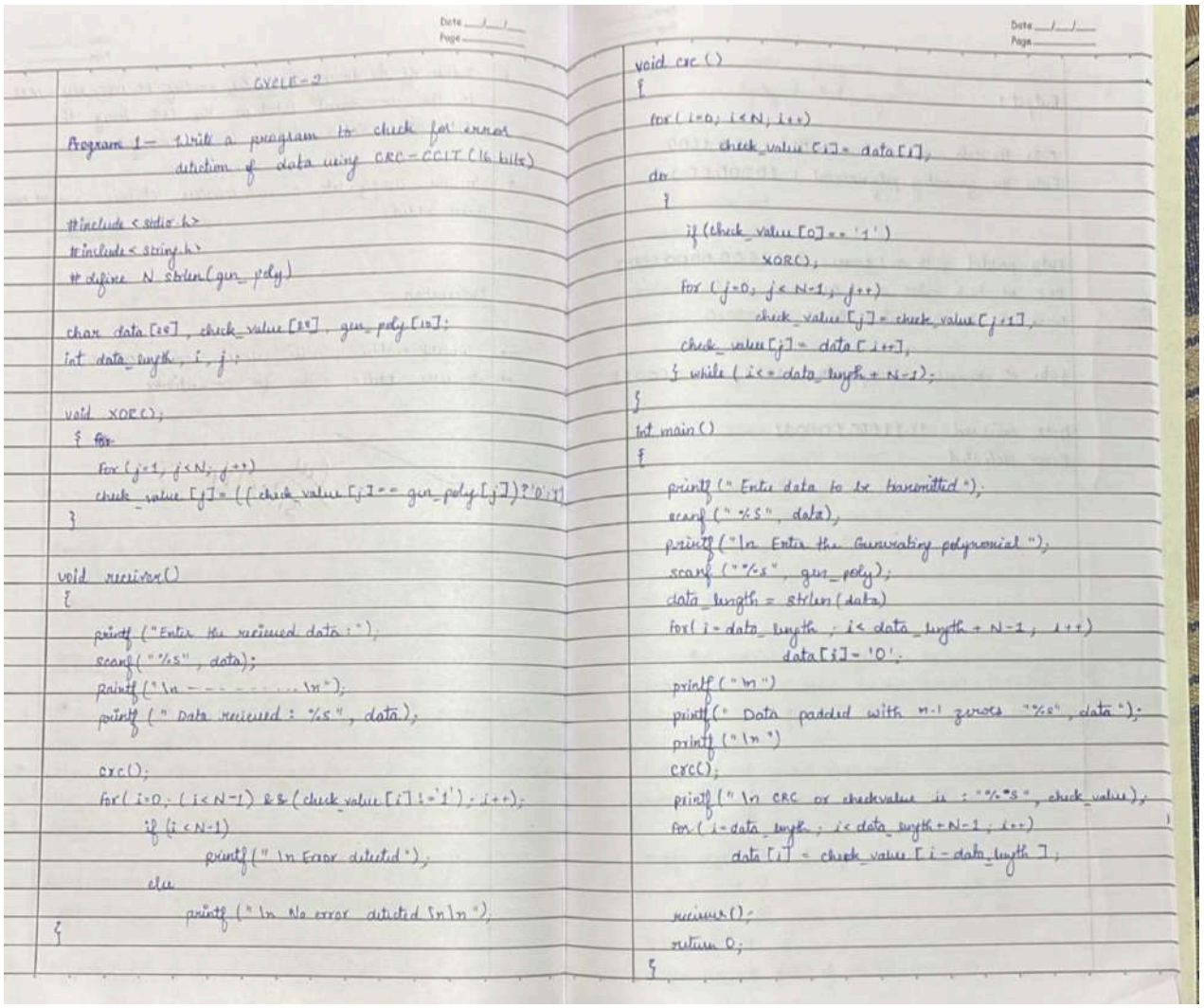
31/12/24

CYCLE-2

Program 1

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

Observation Book:



```
CYCLE-2

Program 1 - Write a program to check for error
            detection of data using CRC-CCIT (16 bits)

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

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

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

void receiver()
 {
    printf ("Enter the recieved data :");
    scanf("%s", data);
    Printf("\n - - - - - - - - - \n");
    printf (" Data recieved : %s", data);

    crc();
    for( i=0; (i<N-1) && (check_value[i]=='1'); i++);
        if (i<N-1)
            printf(" \n Error detected ");
        else
            printf (" \n No 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 (" Enter 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")
    printf(" Data padded with n-1 zeroes %s", data");
    printf ("\n")
    crc();
    printf(" \n CRC or checkvalue is : %s", check_value);
    for ( i=data_length ; i< data_length+N-1 ; i++)
        data [i] = check_value [ i - data_length ];

    reciever();
    return 0;
}
```

Code:

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

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

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

```
void receiver(){

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

    crc();

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


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

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

OUTPUT:

Output :

Enter the data to be transmitted : 1001100
Enter the generating polynomial : 100001011

Data padded with n-1 zeroes : 1001100 0000 0000
CRC or check value is : 0100010
Final data to be sent : 10011000100010

Enter the recieved value/data : 10011 0001 00011

Data recieved : 10110001 00011
Error detected

```
"C:\Users\ADMIN\Desktop\CI

Enter data to be transmitted: 1001100

 Enter the Generating polynomial: 100001011

----------------------------------------
 Data padded with n-1 zeros : 100110000000000
----------------------------------------
CRC or Check value is : 10100010
----------------------------------------
 Final data to be sent : 100110010100010
----------------------------------------
Enter the received data: 10011000100011

-----------------------------
Data received: 10011000100011
Error detected


Process returned 0 (0x0)   execution time : 21.865 s
Press any key to continue.
```

Program 2

Aim of the program:
Write a program for congestion control using Leaky bucket algorithm.

Observation Book:

Date __/__/__
Page _____

Program 2 — Write a program for congestion control using leaky bucket algorithm

```
storage = 0
no_of_queries = 4
bucket_size = 10
input_pkt_size = 4
output_pkt_size = 1
for i in range (0, no_of_queries):
        size_left = bucket_size - storage
        if input_pkt_size <= size_left:
                storage += input_pkt_size
        else:
                print("Packet loss =", input_pkt_size)
        print(f"Buffer size = {storage} out of bucket size
            = {bucket_size}")
        storage -= output_pkt_size
```

OUTPUT:

```
Enter the initial packets in the bucket : 0
Enter total no. of times bucket content is checked : 4
Enter total no. of packets that can be accomodated
 in the bucket : 10
Enter no. of packets that enter the bucket at a time : 4
Enter no. of packets that exits the bucket at a time : 1

Buffer size = 4 out of bucket size = 10
      "     = 7      "      "      = 10
      = 10                        = 10

Packet loss = 4
Buffer size = 9 out of bucket size = 10
```

Code:

```
# Getting user inputs
storage = int(input("Enter initial packets in the bucket: "))
no_of_queries = int(input("Enter total no. of times bucket content is checked: "))
bucket_size = int(input("Enter total no. of packets that can be accommodated in the bucket: "))
input_pkt_size = int(input("Enter no. of packets that enters the bucket at a time: "))
output_pkt_size = int(input("Enter no. of packets that exits the bucket at a time: "))

for i in range(no_of_queries):  # space left
    size_left = bucket_size - storage
    if input_pkt_size <= size_left:
        # update storage
        storage += input_pkt_size
    else:
        print("Packet loss =", input_pkt_size)

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

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

OUTPUT:

```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
    Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64
    AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    ========== RESTART: C:\Users\ADMIN\Desktop\CN\Cycle 2\Leaky Bucket.py ====
    Enter initial packets in the bucket: 0
    Enter total no. of times bucket content is checked: 4
    Enter total no. of packets that can be accommodated in the bucket: 10
    Enter no. of packets that enters the bucket at a time: 4
    Enter no. of packets that exits the bucket at a time: 1
    Buffer size = 4 out of bucket size = 10
    Buffer size = 7 out of bucket size = 10
    Buffer size = 10 out of bucket size = 10
    Packet loss = 4
    Buffer size = 9 out of bucket size = 10
>>>
```

Program 3

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

Observation Book:

Program 3 – using TCP/IP write a client server program to make the client sending the file name and server to send back the contents of the requested file if present.

Client TCP

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\n Enter file name")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print('\n From Server: \n')
print(file contents)
clientSocket.close()
```

Server TCP

```
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('\n Sent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Client UDP

Server UDP

Code:

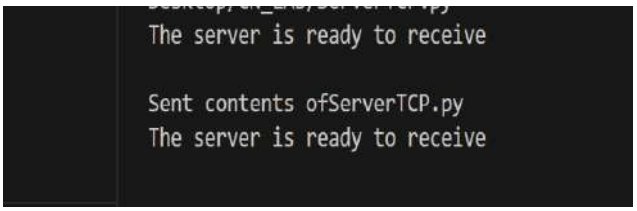
ClientTCP.py

```
from socket import *

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

ServerTCP.py

```
from socket import *

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

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

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

OUTPUT:



Program 4

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

Observation Book:

Date __/__/__
Page _____

Program 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.

client UDP

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

server UDP

```
from socket import *
ServerPort = 12000
ServerSocket = socket(AF_INET, SOCK_DGRAM)
ServerSocket.bind(("127.0.0.1", ServerPort))
print(" The server is ready to recieve")
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('\n Send contents of', end = ' ')
        print(sentence)
        file.close()
```

Code:

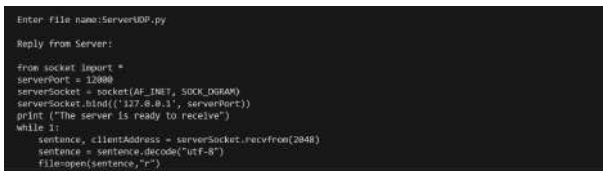
ClientUDP.py

```
from socket import *

serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)

print('\nReply from Server:\n')
print(filecontents.decode("utf-8"))
clientSocket.close()
```

ServerUDP.py

```
from socket import *

serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))

print("The server is ready to receive")

while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")

    file = open(sentence, "r")
    con = file.read(2048)
    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)

    print('\nSent contents of ', end=' ')
    print(sentence)

    file.close()
```

OUTPUT:



Tool Exploration - Wireshark

Date __/__/__
Page _____

Tool Exploration - Wireshark

Wireshark is a powerful tool widely used network protocol analysis. It allows you to capture and inspect data packets travelling and a network in real-time, making it a crucial tool for studying computer networks, troubleshooting network issues.

Key features:

1) Packet Capture : Captures live network traffic

2) Protocol Analysis : supports hundreds of protocols

3) Filtering : offers powerful filters to isolate specific packets or traffic types.

4) Visualization : Displays packets details with hierarchial layers.

Use Cases

1) Network troubleshooting

2) Security Analysis

3) Protocol Study