VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

GARGI BHARADWAJ (1BM22CS099)

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 **Gargi Bharadwaj (1BM22CS099)**, who is a 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 & HOD
Department of CSE
BMSCE, Bengaluru

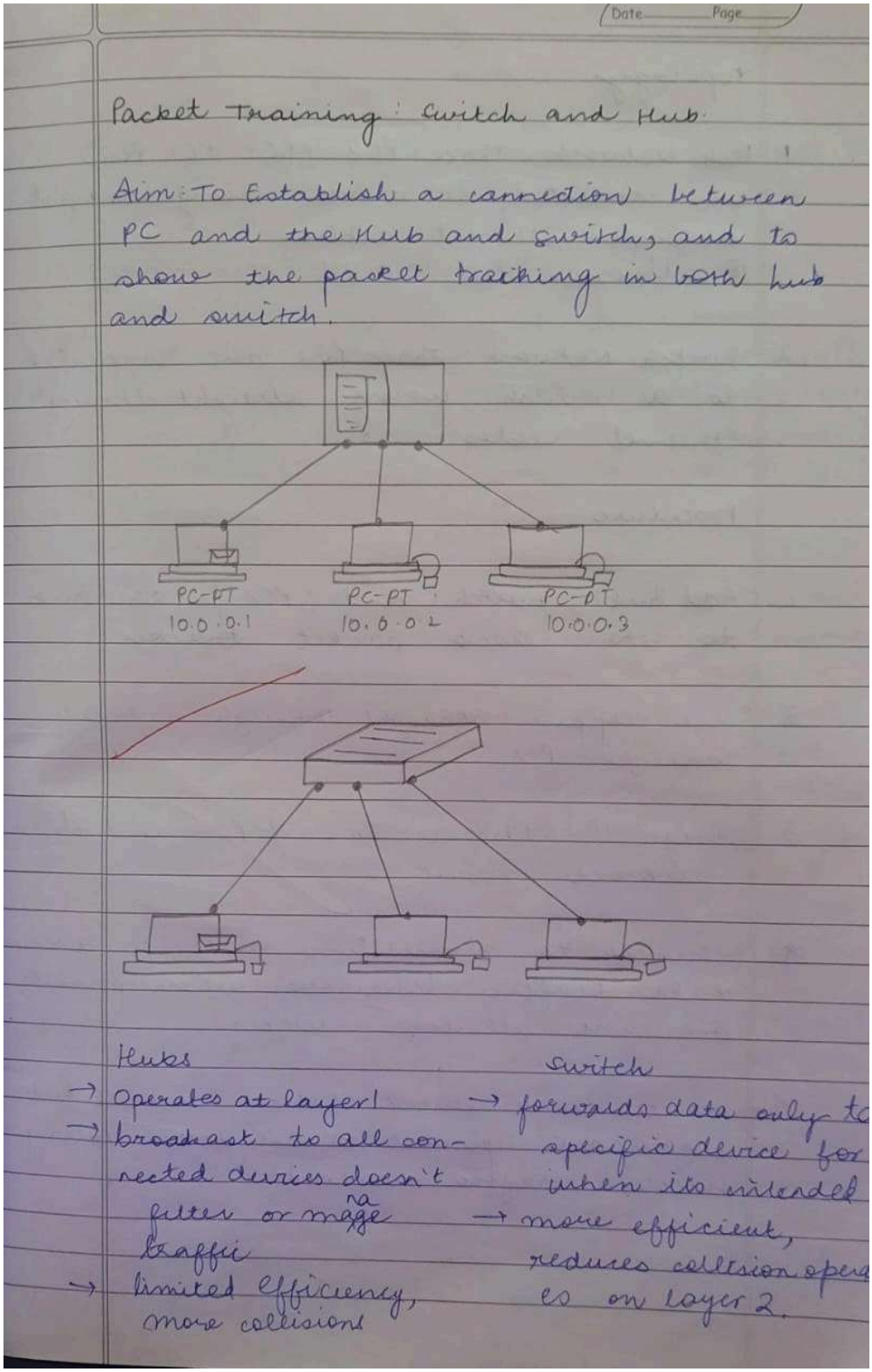
Index

CYCLE 1

10	26-11-24	To demonstrate the working of Address Resolution Protocol (ARP) within a LAN for communication	42-46
11	03-12-24	To create a VLAN on top of the physical LAN and enable communication between physical LAN and virtual LAN	47-52
		CYCLE 2	
12	17-12-24	Write a program for congestion control using Leaky bucket algorithm.	53-55
13	24-12-24	Write a program for error detecting code using CRC-CCITT (8-bits).	56-60
14	24-12-24	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.	61-64
15	24-12-24	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	65-68

PROGRAM – 1

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



SURYA Gold

Date Page

Topology

1. Hub Network: Three PCs (PC0, PC1, PC2) are connected to a hub using straight through Ethernet cables.
   IP addresses: PC0: 10.0.0.1, PC1 = 10.0.0.2..

2. Switch Network: Three PCs are connected to a switch using straight-through ethernet cables.

Procedure

1. Add hub1, switch1 and 6 PCs for the hub to the cisco packet tracker.

2. Use copper straight-through cables to connect PCs

3. Assign IP addresses to each PC and obtain subnet mask.

4. Switch to simulation mode to observe data traffic behaviours when packets are sent between devices

5. In the hub network, notice how all hub broadcasts packets to all devices causing potential traffic overload.

SURYA Gold

Date Page

6. In the switch network, observe how the switch forwards packets to all the devices, inteaded recipients

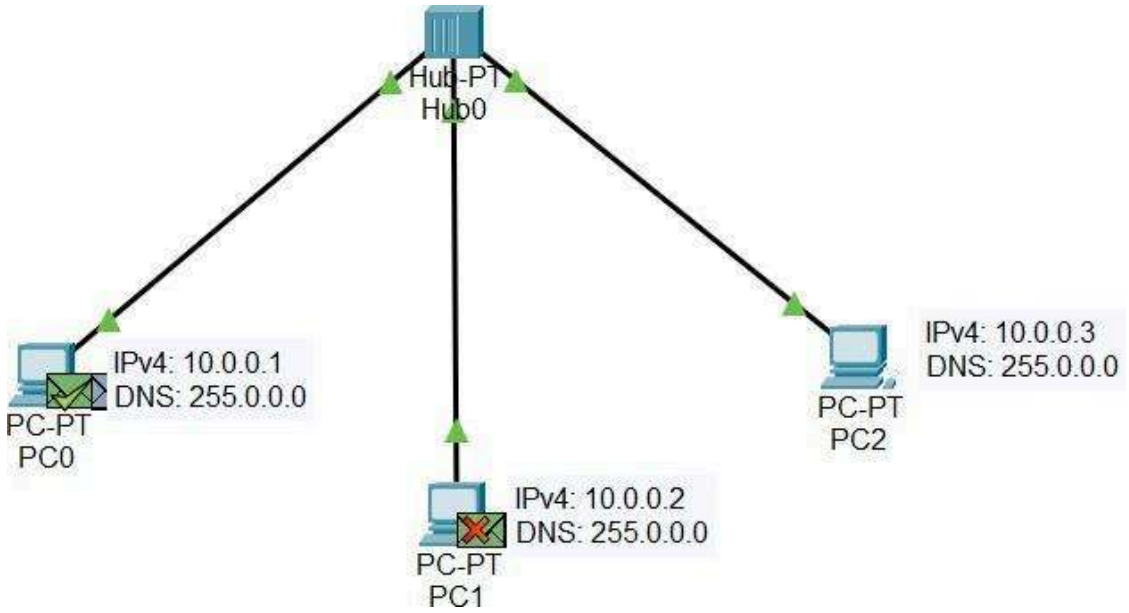
6. The hub broadcasts data to all connected devices leading to more network congestion, while the switch efficiently sends data only to the correct device, optimizing performance.

Observation

1. The hub broadcasts packets to all the devices which may cause unnecessary traffic.

2. The switch only forwards packet to appropriate devices by learning MAC addresses, making it more efficient in reducing traffic.

9/10/24



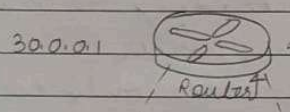

```
C:\>ping 10.0.0.3

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
```

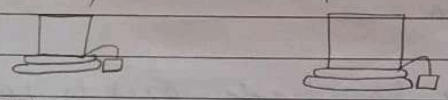
PROGRAM – 2

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

Exp - 2

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





ip 30.0.0.10
gateway 30.0.0.1

ip 40.0.0.10
gateway 40.0.0.1

Topology

Two PCs (PC1 and PC2) are connected to a router.
Star topology

Procedure

1. Add a router, two PCs for star topology
2. use copper crossover cables to connect PCs with router
3. Configure both PCs with IP address, gateway and Net.

SURYA Gold
Date Page

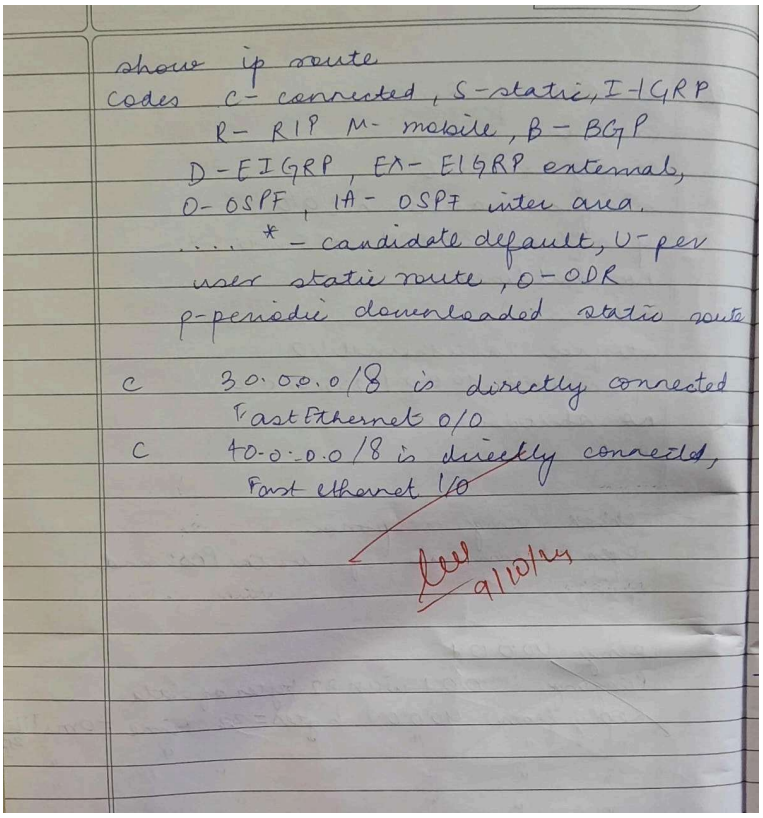
4. In CLI of Router
enable
config terminal
interface fastethernet 0/0
ip address 30.0.0.1 255.0.0.0
no shutdown
exit
interface fastethernet 1/0
ip address 40.0.0.1 255.0.0.0
no shutdown
exit

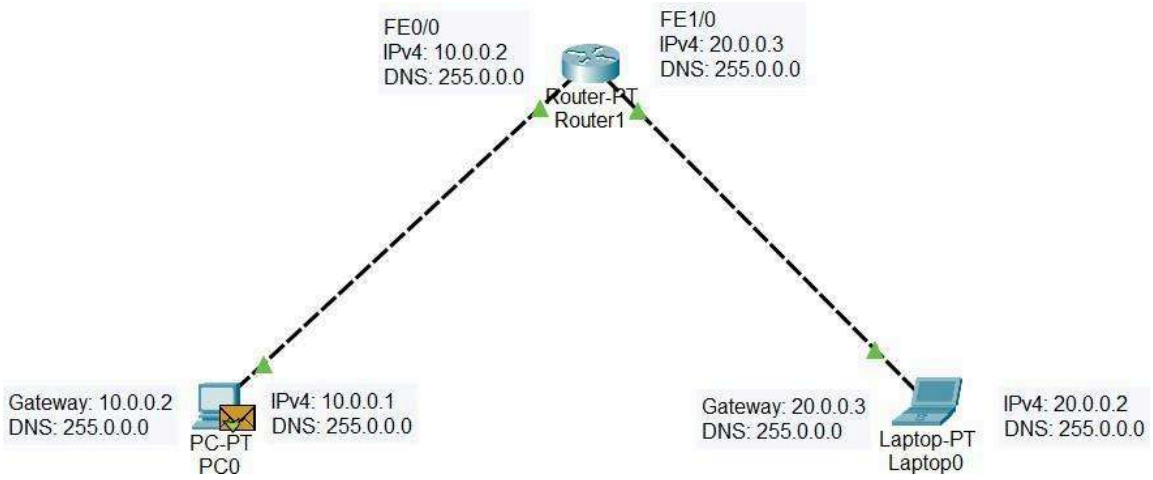
5. check ping response
open terminal of both PCs and ping.

ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data
Reply from 40.0.0.1: bytes=32 time=0ms TTL=
" " " " " " "
" " " " " " "
" " " " " " "

Ping statistics
Packets : sent=4 Recieved=4 Lost=0
(0% loss)

Observations
1. In case of wrong configuration, Request will be timed out
2. A successful ping means routing is set up correctly.


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.
... * - candidate default, U- per
user static route, O- ODR
P- periodic downloaded static route
C 30.00.0/8 is directly connected
Fast Ethernet 0/0
C 40.0.0.0/8 is directly connected,
Fast ethernet 1/0
9/10/24

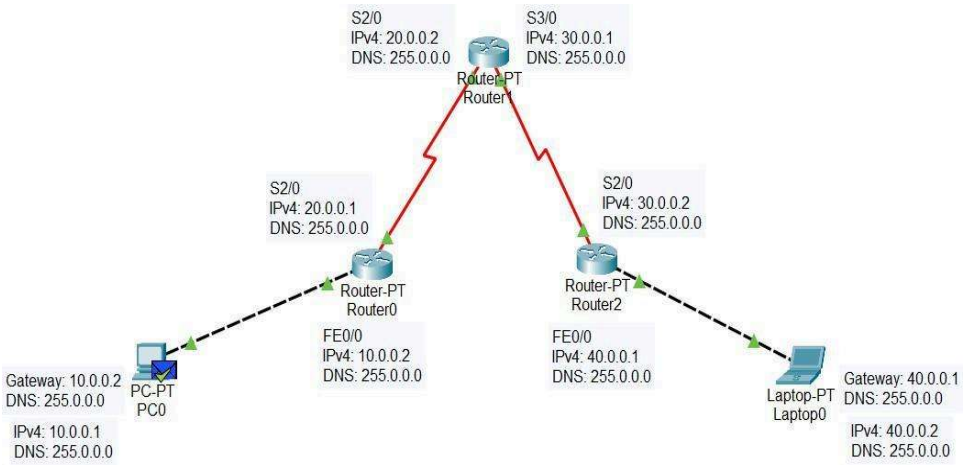

FE0/0
IPv4: 10.0.0.2
DNS: 255.0.0.0
FE1/0
IPv4: 20.0.0.3
DNS: 255.0.0.0
Router-PT
Router1
Gateway: 10.0.0.2
DNS: 255.0.0.0
IPv4: 10.0.0.1
DNS: 255.0.0.0
PC-PT
PC0
Gateway: 20.0.0.3
DNS: 255.0.0.0
Laptop-PT
Laptop0
IPv4: 20.0.0.2
DNS: 255.0.0.0

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
	In Progress	PC0	Laptop0	ICMP		0.000	N	1	(edit)	
	In Progress	PC0	Laptop0	ICMP		0.000	N	2	(edit)	

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PROGRAM – 3

Configure default route to the Router.



SHOW IP ROUTE

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

Figure 3.1: Router0

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

Figure 3.2: Router1

```
S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 30.0.0.1
C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
```

Figure 3.3: Router3.3

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

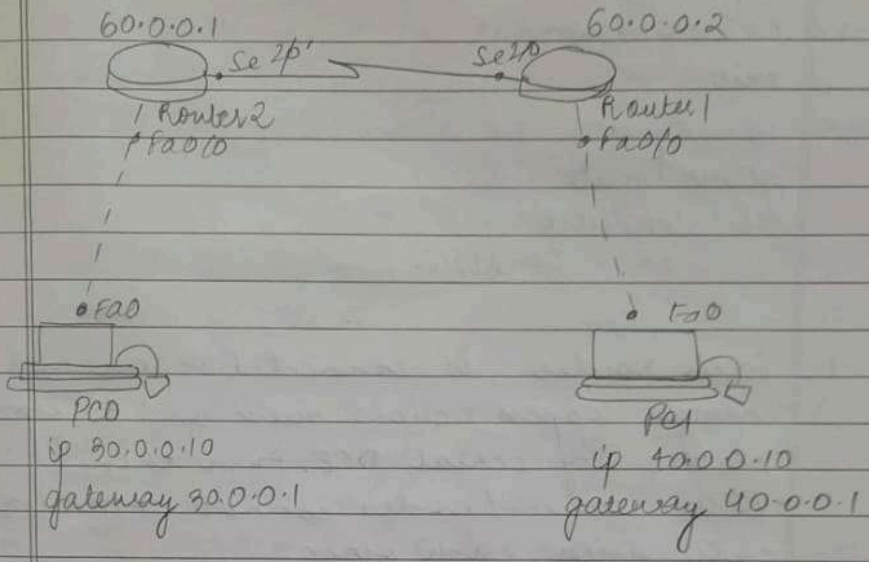
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=36ms TTL=125
Reply from 40.0.0.2: bytes=32 time=34ms TTL=125
Reply from 40.0.0.2: bytes=32 time=30ms TTL=125
Reply from 40.0.0.2: bytes=32 time=26ms 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 = 26ms, Maximum = 36ms, Average = 31ms
```

SURYA Gold
Date Page

16/10/24 Lab - 3.

Exp - 02

Aim Configure default route, static route to the router.



Topology Two PCs are connected of two routers respectively by means of a copper cross.
The routers are connected by Serial wiring (DCE).

Procedure
enable
config terminal
interface serial 2/0
ip address 60.0.0.1 255.0.0.0
no shutdown
exit

SURYA Gold

Date Page

Router2

enable

config terminal

interface serial 2/0

ip address 60.0.0.2 255.0.0.0

no shutdown

exit

show ip route

C - connected . . .

. . . serial2/0.

1. Each router is connected to one each through copper cross over and to other router by serial DCE. Even then both values/nodes cannot communicate with each other.

2. Check ping response from PC0

ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data

Destination host unreachable

Request timed out.

3. Check ping response from 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

Reply from 40.0.0.1 bytes=32 time=0ms

TTL=215

SURYA Gold

Date Page

Ping ~~stated~~ statistics for 400.0.1
Packet: Sent: 4, Recieved: 4 Lost: 0 (0% lost)

Observations
Each router is connected to node and to each other router. But one node cannot communicate with another as it is unaware of another's respond

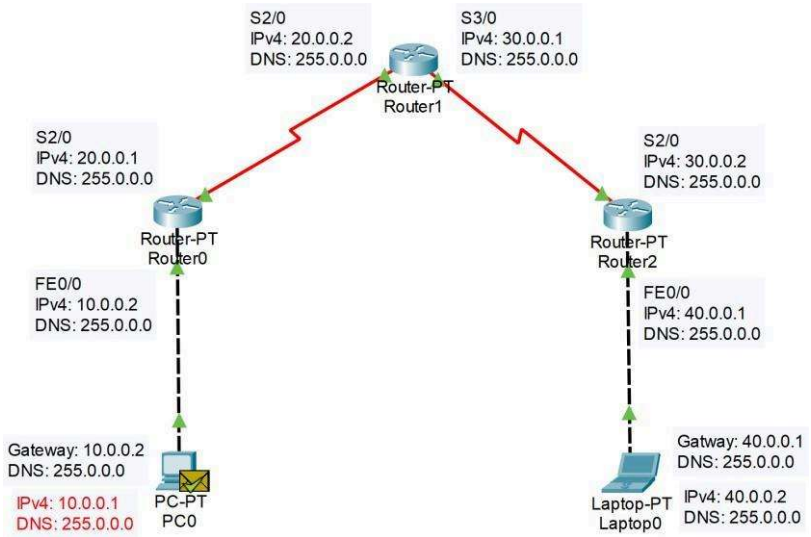
Serial Routing
show ip route

C 10.0.0.0/8 directly connected, FastEthernet 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.

Lab 16/10/24

PROGRAM-4

Configure default route, static route to the Router.



SHOW IP ROUTE

```
Gateway of last resort is 20.0.0.2 to network 0.0.0.0

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

Figure 4.1: Router0

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

Figure4.2: Router1

```
Gateway of last resort is 30.0.0.1 to network 0.0.0.0

C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 30.0.0.1
```

Figure 4.3: Router2

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

```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=34ms TTL=125
Reply from 40.0.0.2: bytes=32 time=33ms TTL=125
Reply from 40.0.0.2: bytes=32 time=30ms TTL=125
Reply from 40.0.0.2: bytes=32 time=33ms 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 = 30ms, Maximum = 34ms, Average = 32ms
```

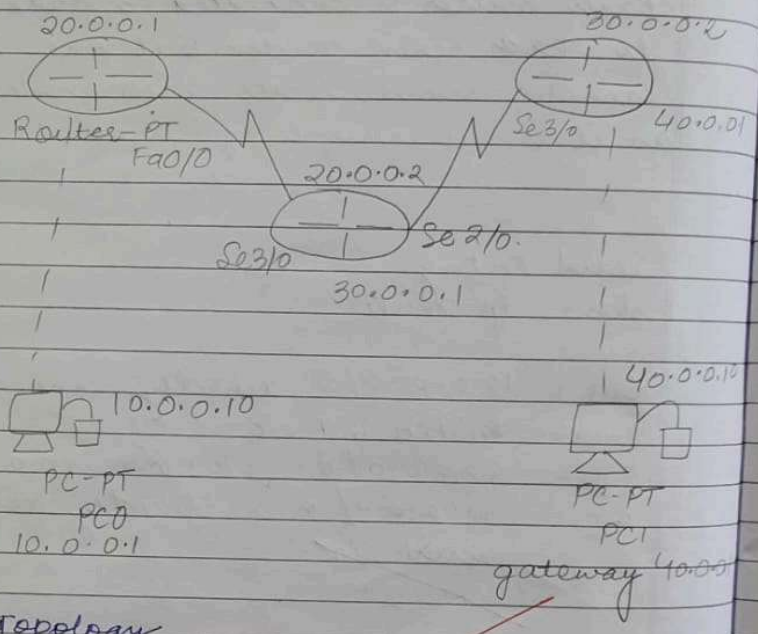
SURYA Gold

Date Page

23/10/24

LAB-4

Static Routing, Default Routing



Topology

PCs. are two in number, PC0 and PC1 as end systems.
configured to 10.0.0.10 (IP), 40.0.0.10 (IP) respectively.

There are three routes connected ones a serial DCE.

Procedure
The end systems are connected

SURYA Gold
Date Page

with a copper cross-over connected with routers.

* Router to system
enable
config terminal
interface fastethernet0/0
ip address 10.0.0.1 255.000
no shut
exit

Repeated for Router 3

* Router to Router

Router0
enable
~~config terminal~~
interface serial4/0
ip address 20.0.0.1 255.0.0.0
no shut

Router1
enable
config terminal
interface serial3/0
ip address 20.0.0.2 255.0.0.0
no shut

SURYA Gold
Date Page

* Default Routing

Router config# ip route 0.0.0.0.0000
20.0.0.2

Router2 config# ip route 0.0.0.0.0.0.0.0
20.0.0.1

show ip route

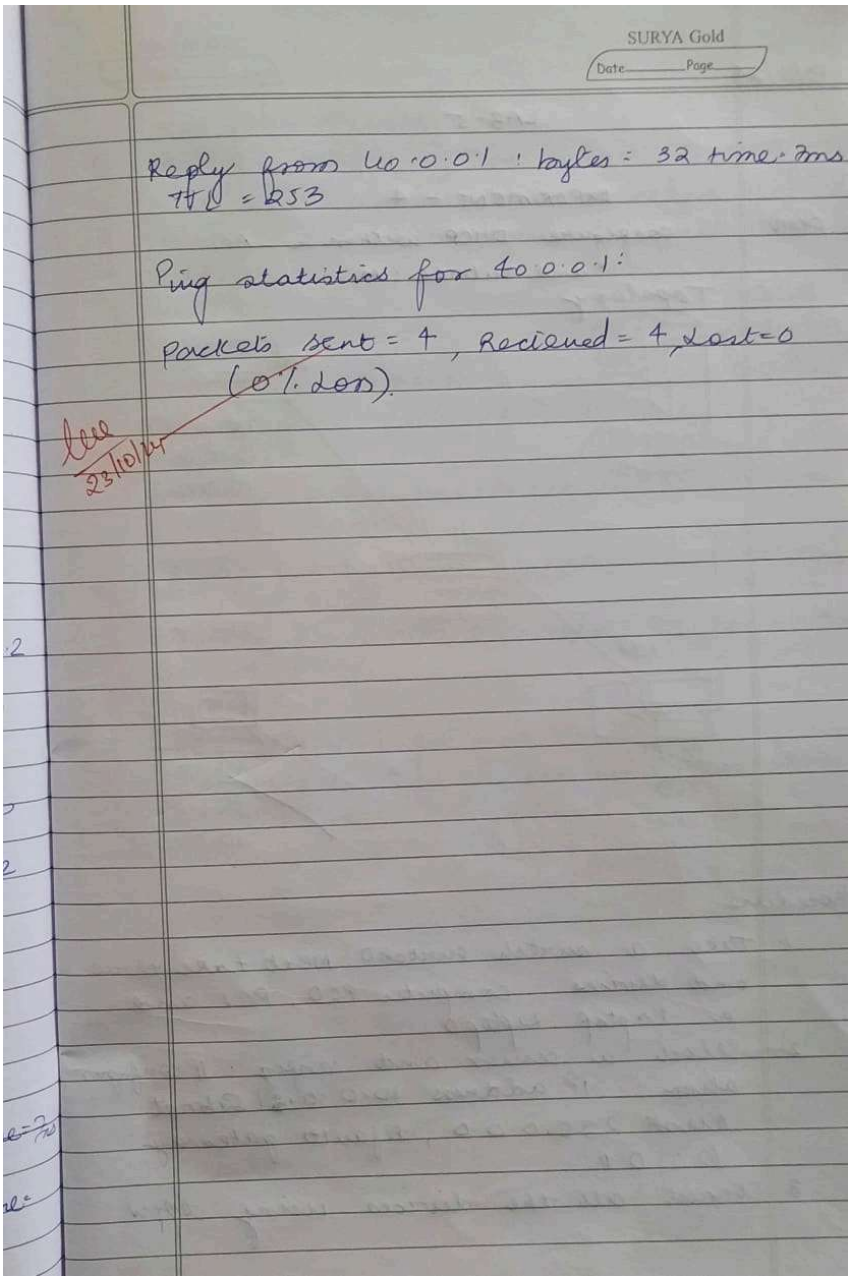
C-connected S-static I- IGRP
R-RIP M-mobile B-BGP
p-periodic downloaded static route

Gateway of last resort is 20.0.0.2 to network 0.0.0.0
C 10.0.0.0/8 is directly connected Fastethernet 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

Output

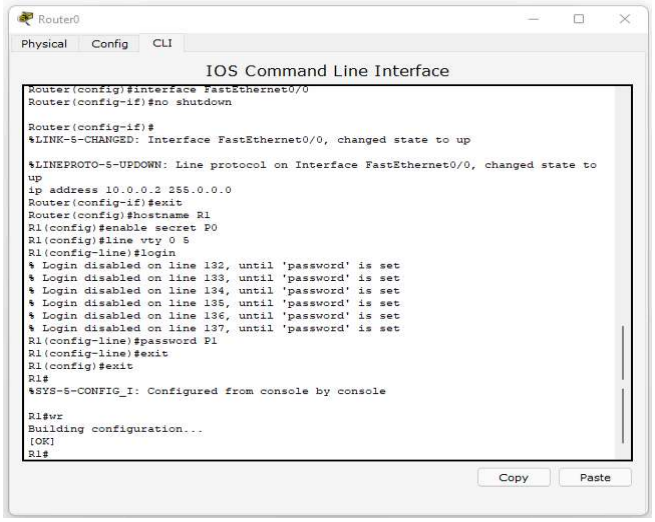
On PC 20.0.0.1
PC> ping 40.0.0.1

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

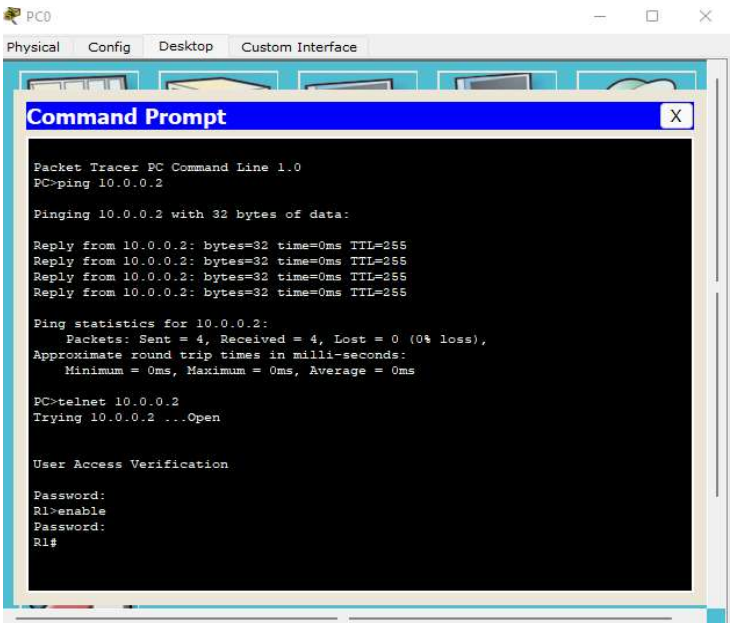

SURYA Gold
Date Page
Reply from 40.0.0.1 : bytes = 32 time=3ms TTL = 253
Ping statistics for 40.0.0.1:
Packets sent = 4, Recieved = 4, Lost = 0 (0% loss).
23/10/24

PROGRAM-5

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



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Router0	ICMP		0.000	N	0	(edit)	



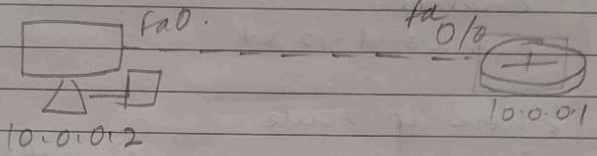
8/12/2024

SURYA Gold
Date Page

LAB-11
TELNET

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

Topology.



Procedure:

1) create the topology and configure the devices.

2) ~~commands~~ commands in router:

```
enable
configt
hostname R1
enable secret 1234
ip address 10.0.0.2 255.0.0.0
no shut
line vty 0 3
login
password ~~12~~ 4321
exit
```

SURYA Gold
Date Page

3) In PC : Command Prompt
- First try pinging.

PC7 telnet 10.0.0.2

User Access Verification
Password: 4321
password : 4321
enable
password : 1234

show ip route
C 10.0.0.0/8 is directly connected
FastEthernet 0/0.

Observations

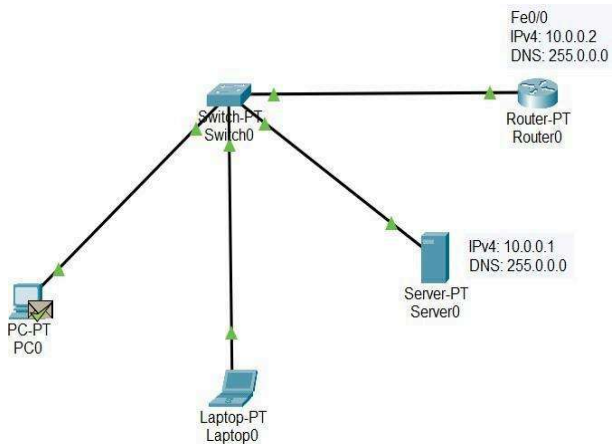
1) The admin in PC is able to run commands as run in router CLI and see results from PC

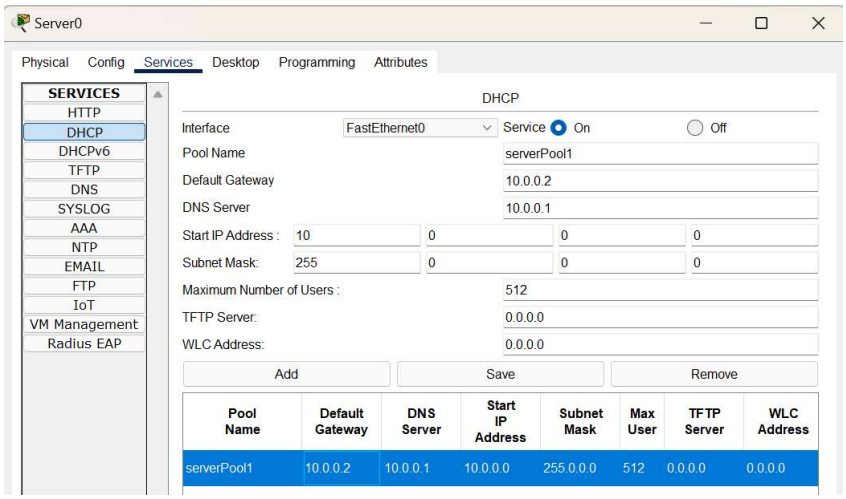
2) Telnet allows user to establish a remote session with another device like router, over a TCP/IP network

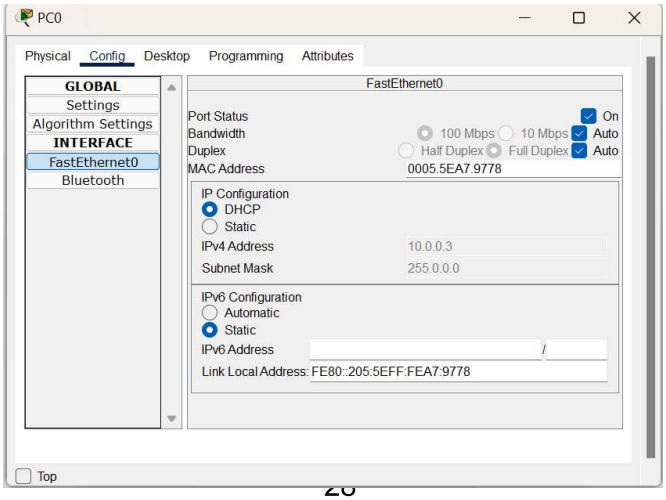
3) Using Telnet, we can access and control the remote device's CLI as if you were physically connected to it

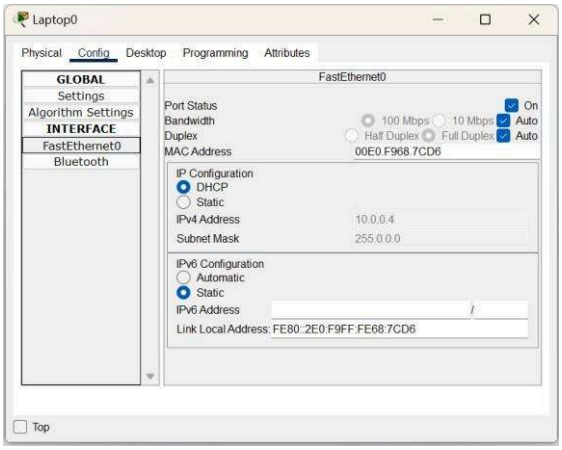
PROGRAM-6(A)

To Configure IP addresses of the host using DHCP server within a LAN.

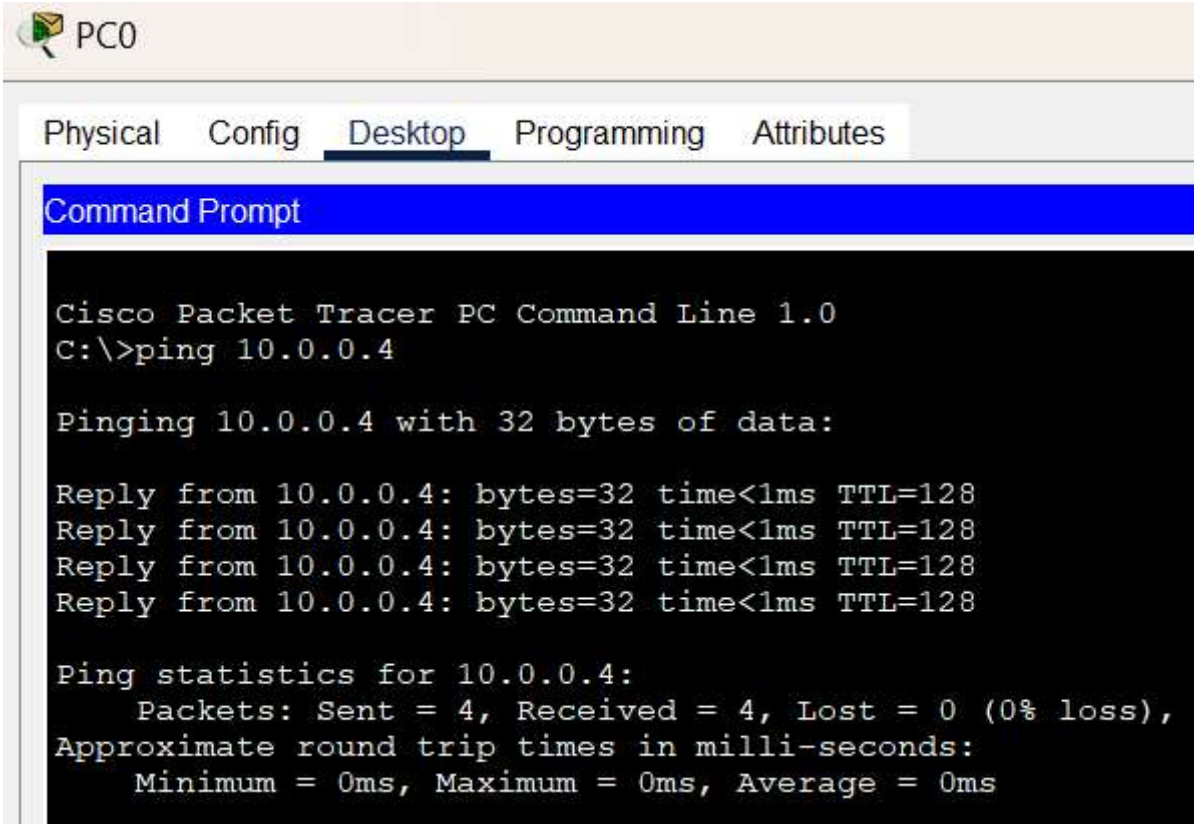






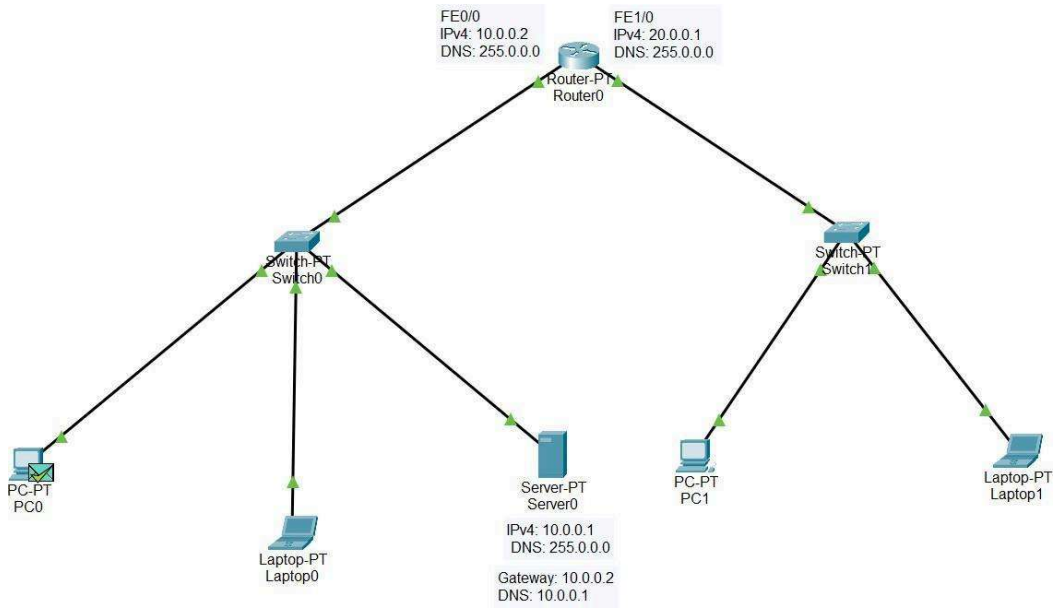


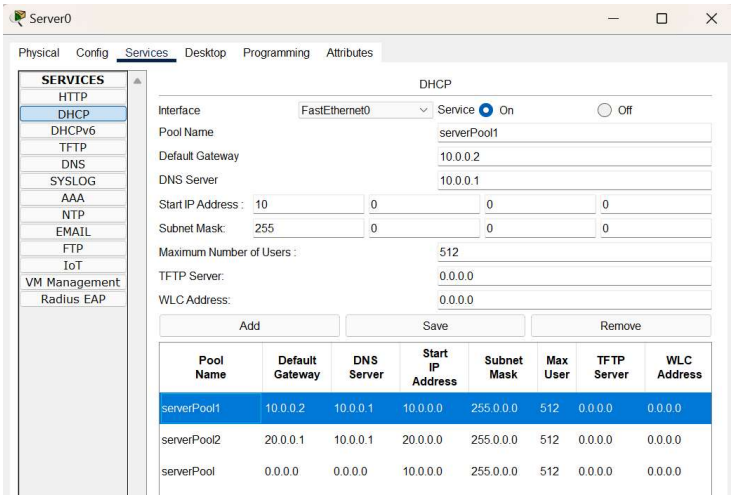
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

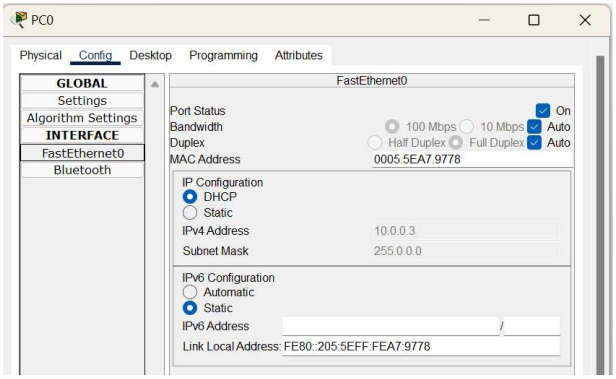


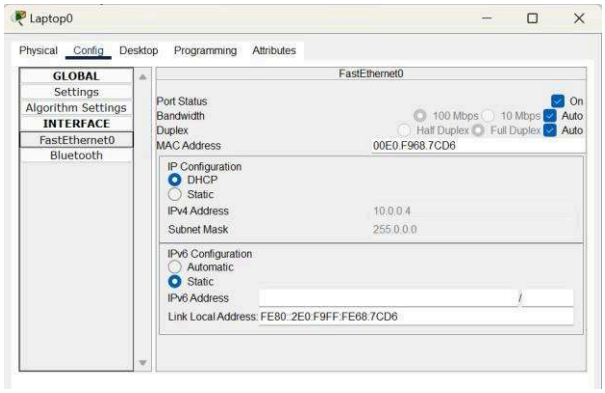
PROGRAM-6(B)

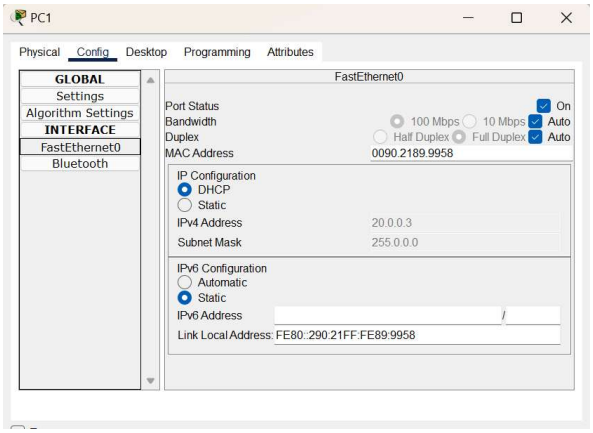
To Configure IP addresses of the host using DHCP server outside a LAN.

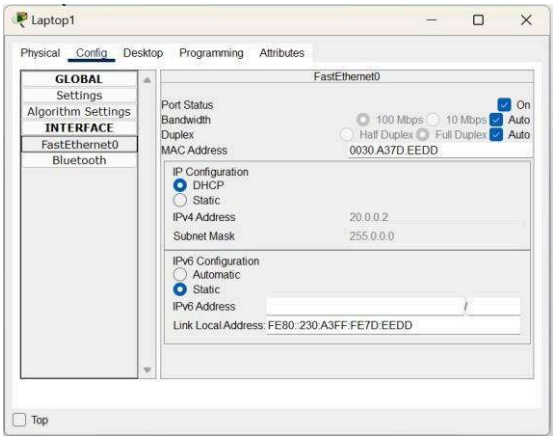












Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
	Successful	PC1	Laptop1	ICMP		0.004	N	1	(edit)	

SURYA Gold

Date Page

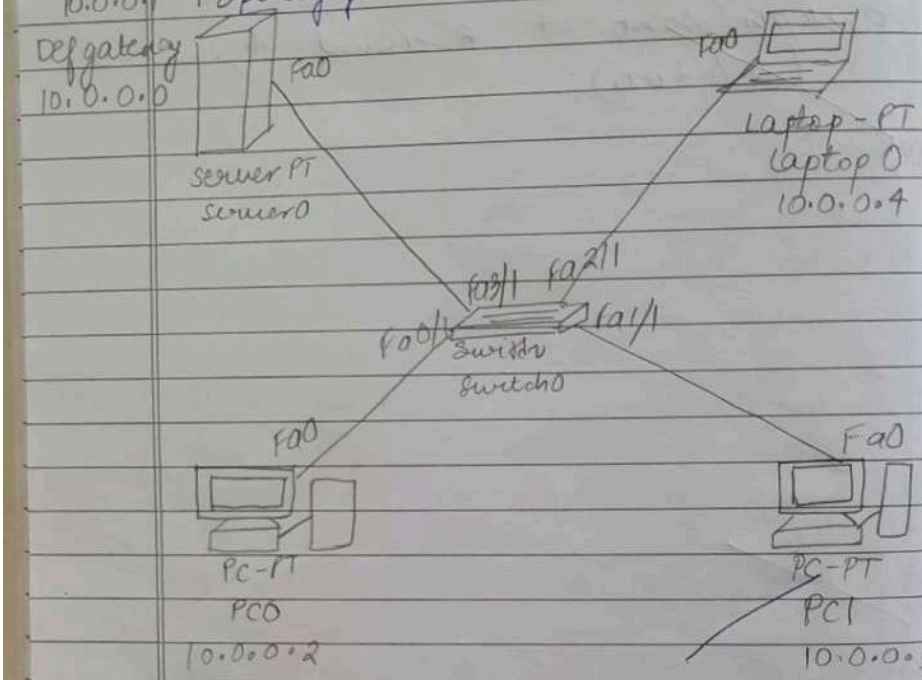
13/11/24

LAB-5

EXPERIMENT - 4

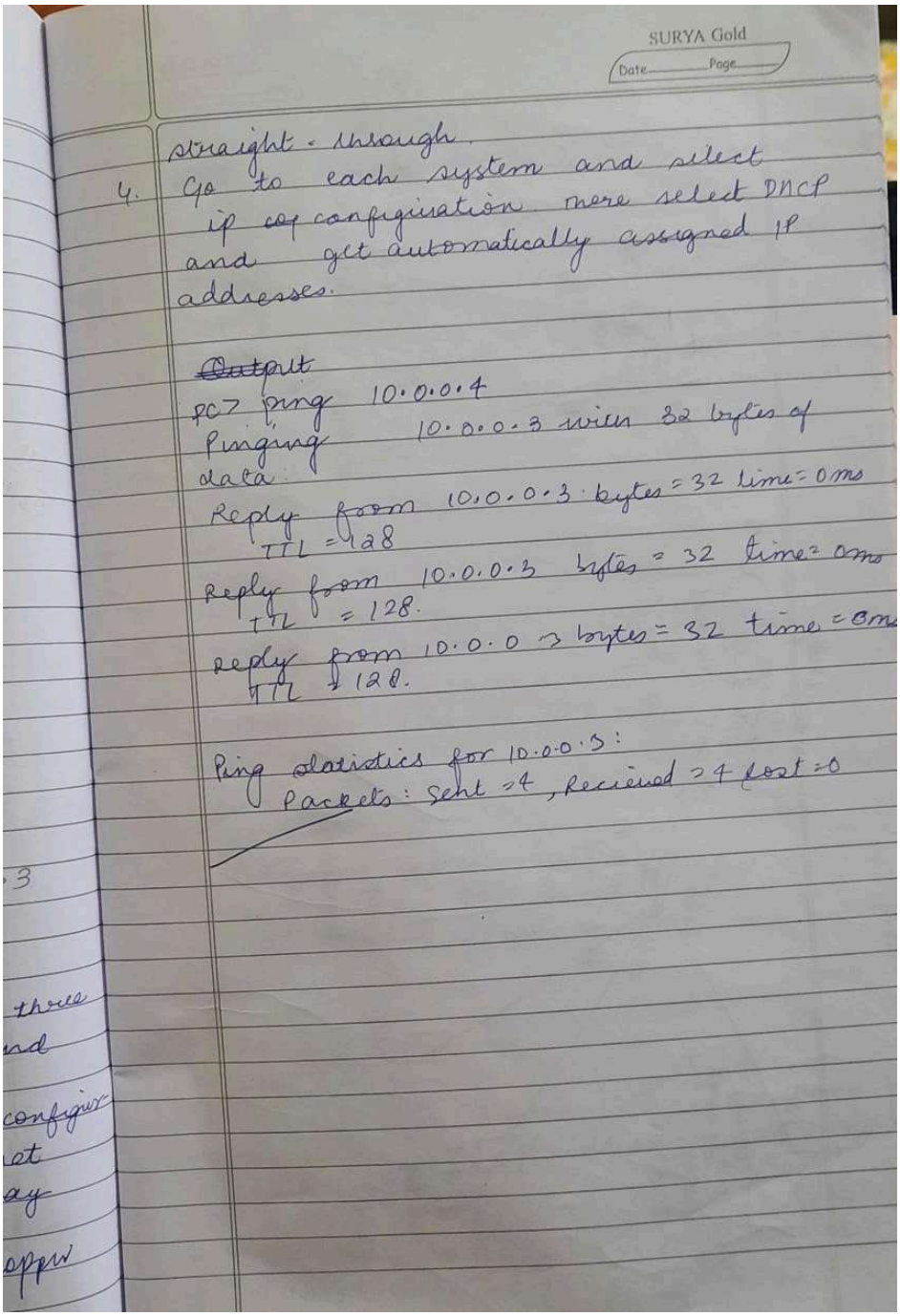
Aim configure DHCP within a LAN and outside LAN.

10.0.0.1 Topology



Procedure.

1. Take a switch switch0. Next take three end devices computer PC0, PC1 and a laptop laptop0.
2. Select a server and config : IP configuration : IP address 10.0.0.1 Subnet Mask 255.0.0.0, default gateway 10.0.0.1.
3. Connect all the devices using copper



straight - through.

4. Go to each system and select ip configuration there select DHCP and get automatically assigned IP addresses.

~~Output~~

PC> ping 10.0.0.4

Pinging 10.0.0.3 with 32 bytes of data:

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

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

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

Ping statistics for 10.0.0.3:

Packets: Sent = 4, Received = 4 Lost = 0

SURYA Gold
Date
Page
Router1
Fa0/0
Fa0/1
Fa0/2
Fa3/1
100Mbps
Fa0/1
Fa0/1
Fa3/1
Fa0/0
10.0.0.2
10.0.0.1
10.0.0.5
10.0.0.6
10.0.0.3
80.0.0.3
80.0.0.4
PC-PT
PC1
PC-PT
PC
PC-PT
Laptop0

SURYA Gold
Date Page

Procedure:

1. Add a router between two LAN environments.
2. Connect 2 PCs and 1 Laptop to each switch using copper straight through.
3. Connect switches to router using Automatically choose connection type
4. Server should be connected to one switch.

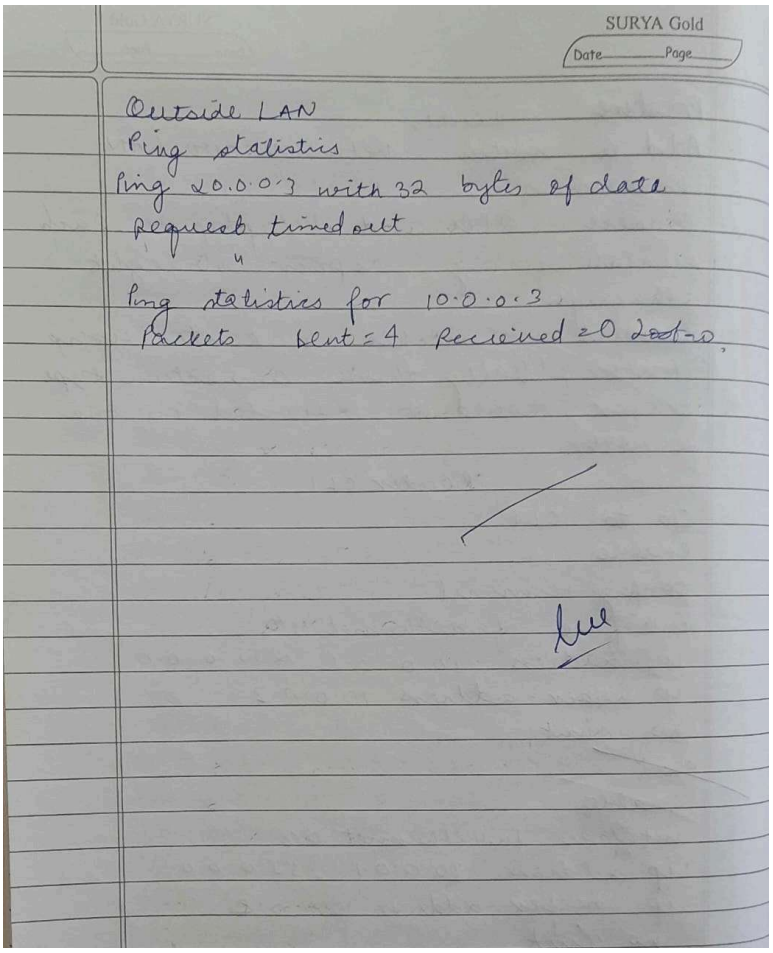
Router CLI

Go to CLI,
enable
config terminal
interface fastethernet 4/0
ip address 10.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut
exit
enable
interface fastethernet 0/0
ip address 20.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut
exit

Observation.

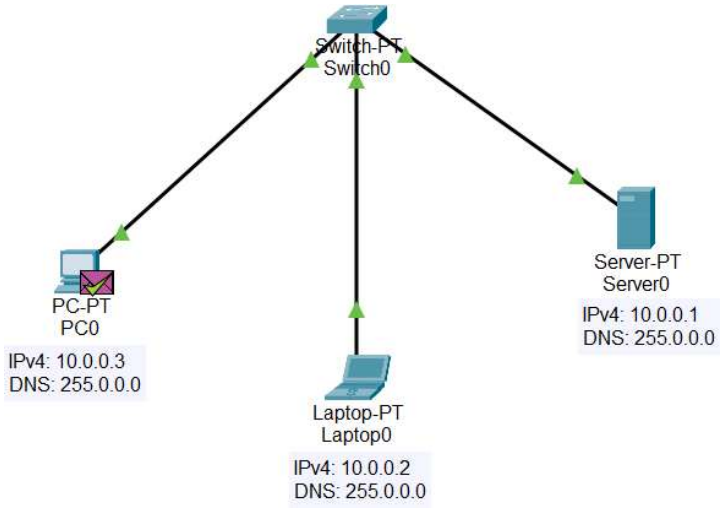
- For every system, we select DHCP. Automatically IP addresses are assigned.

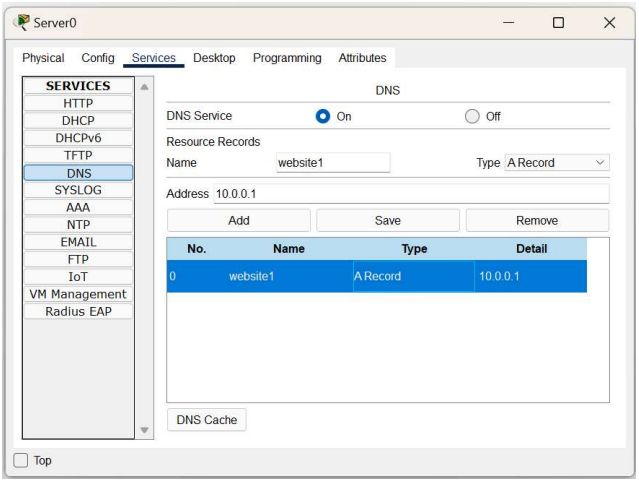
switch0		switch1	
PC0	10.0.0.5	PC2	20.0.0.3
PC1	10.0.0.6	PC3	20.0.0.4
Laptop0	10.0.0.3	Laptop1	20.0.0.5

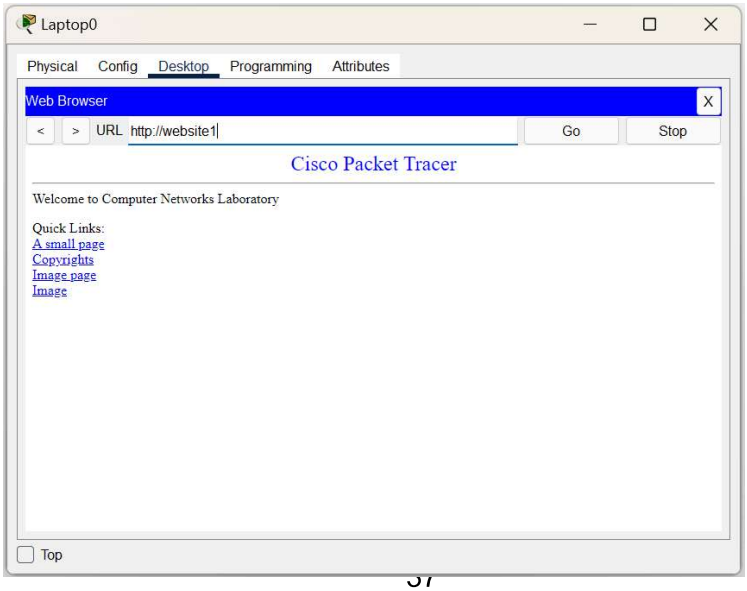

SURYA Gold
Date Page
Outside LAN
Ping statistics
Ping 10.0.0.3 with 32 bytes of data
Request timed out
Ping statistics for 10.0.0.3
Packets sent = 4 Received = 0

PROGRAM-7

To Configure DNS server to demonstrate the mapping of IP addresses and Domain names.







SURYA Gold

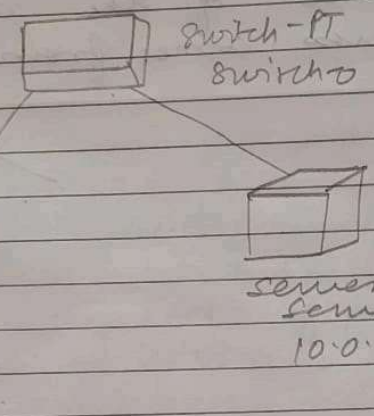
Date Page

8/12/2024

LAB - 13

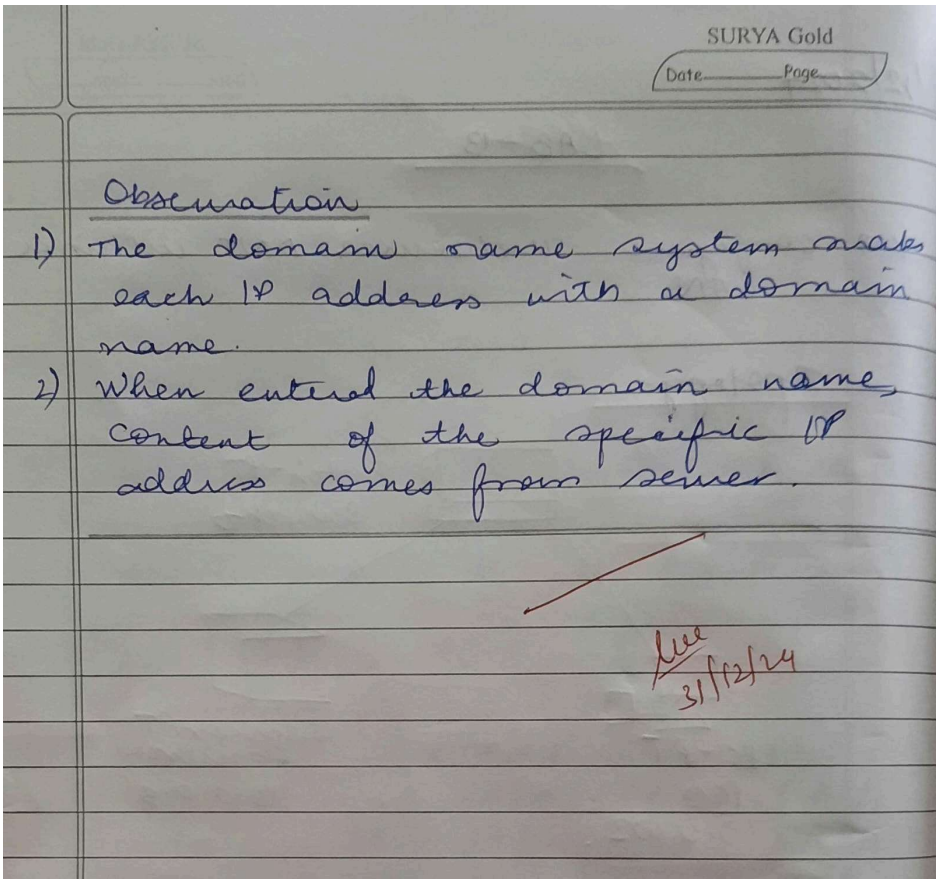
Aim configure webserver, DNS within a LAN

Topology



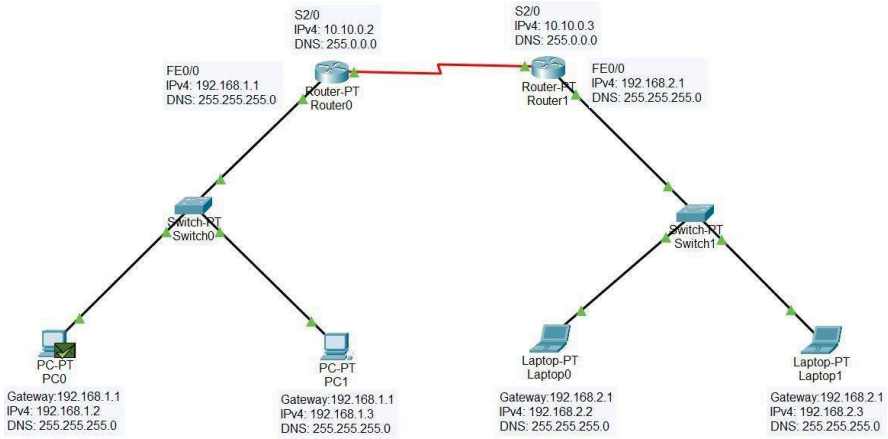
Procedure

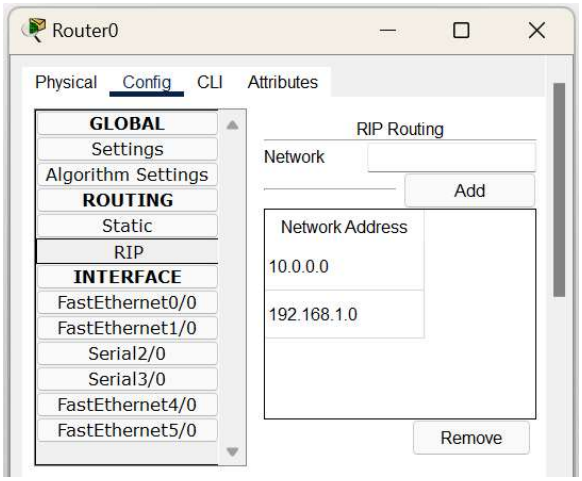
1. Select the PC, switch and server and connect them using cable
2. Assign IP addresses to PC and server
3. In server, go to services, DNS make it on write the name and address and press add
4. In server select HTTP or edit the file.
5. Then in PC, go to desktop, in web browser write the domain name to get the result.

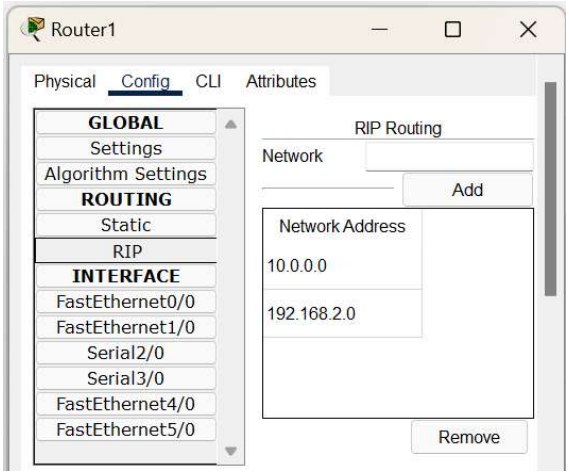

SURYA Gold
Date Page
Observation
1) The domain name system maps each IP address with a domain name.
2) When entred the domain name, content of the specific IP address comes from server.
31/12/24

PROGRAM-8

To Configure RIP routing protocol in Routers.







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

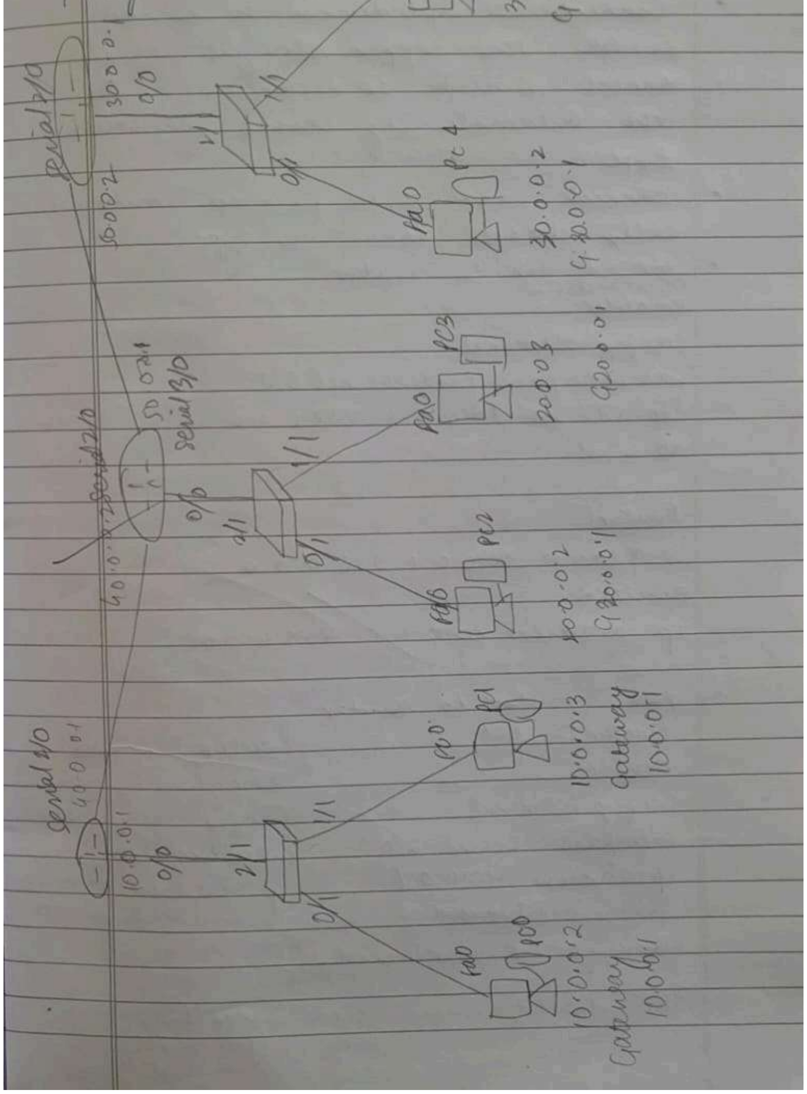
```
C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=18ms TTL=126
Reply from 192.168.2.3: bytes=32 time=14ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 18ms, Average = 8ms
```

SURYA Gold
Date ______ Page ______

20/11/2024 LAB-6

Aim: configure routing information protocol.

Topology.



Date Page

Procedure

- Connect 2 end devices to each switch via copper straight through.
- connect switch to respective router via automatically choose connection type.
- Connect each router pair via automatically choose connection type.
- For switch to router

router0 CLI
enable
config terminal
~~fas~~ interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
no shut.

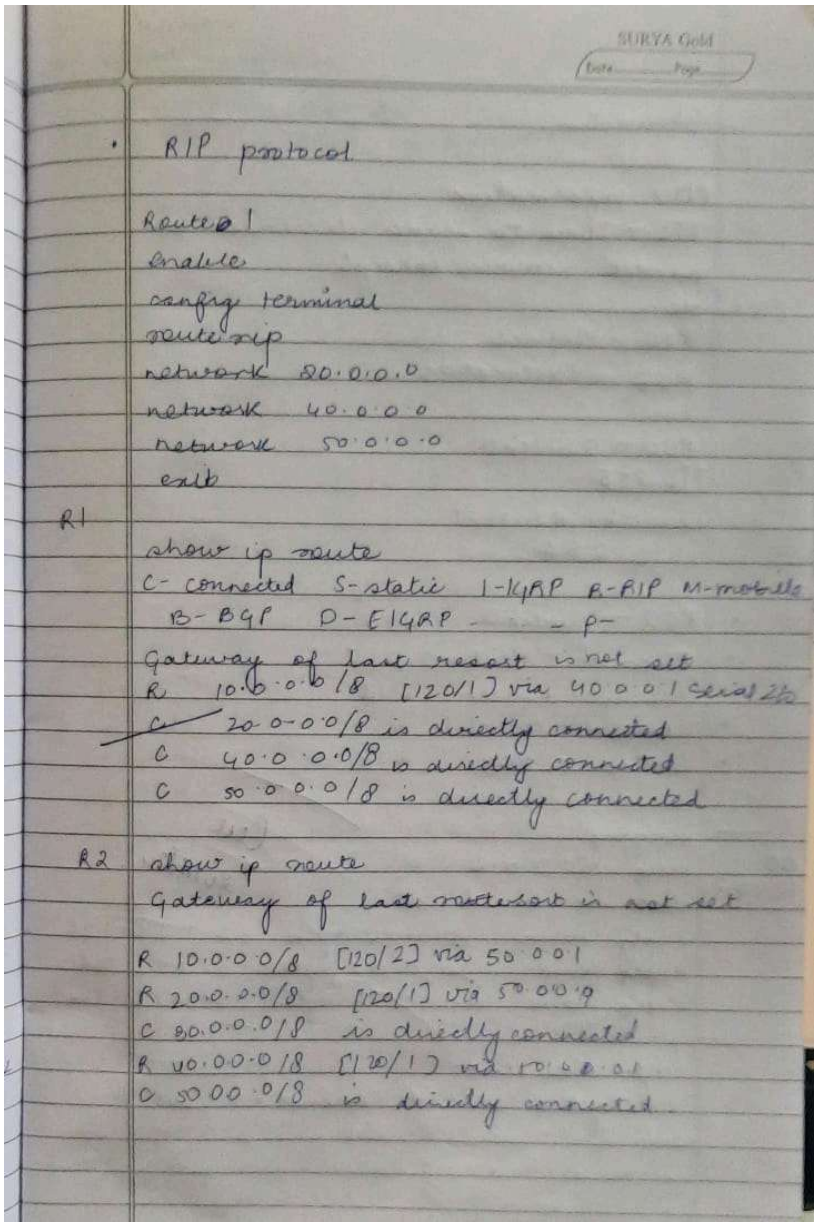
Router1
ip address 20.0.0.1 255.0.0.0
Router2
ip address 30.0.0.1 255.0.0.0

- for router to router

Router 0 CLI	Router 1
enable	enable
Config terminal	config terminal
interface serial 2/0	interface serial 2/0
ip address 40.0.0.1 255.0.0.0	ip address 40.0.0.2 255.0.0.0

same for Router1 and 2

This should ensure green signalling.



SURYA GoM

Date Page

- RIP protocol

Router 1
enable
config terminal
router rip
network 80.0.0.0
network 40.0.0.0
network 50.0.0.0
exit

R1

show ip route
C - connected S - static I-IGRP R-RIP M-mobile
B - BGP D - EIGRP - - P -
Gateway of last resort is not set
R 10.0.0.0/8 [120/1] via 40.0.0.1 serial 2/0
C 20.0.0.0/8 is directly connected
C 40.0.0.0/8 is directly connected
C 50.0.0.0/8 is directly connected

R2 show ip route
Gateway of last resort is not set
R 10.0.0.0/8 [120/2] via 50.0.0.1
R 20.0.0.0/8 [120/1] via 50.0.0.9
C 80.0.0.0/8 is directly connected
R 40.0.0.0/8 [120/1] via 50.0.0.1
C 50.0.0.0/8 is directly connected

Date Page

Observation

PDU information
TTL - Time to Leave decreases as packet moves through.

Example:
PC0 - Source
PC3 - Destination

PC0 - Switch0
TTL 255
Switch - Router1
TTL 254.

		TTL
•	PC0 - Switch0	255
•	PC Switch - Router0	255
•	Router0 - Router1	254
•	Router1 - Switch1	254
•	Switch1 - PC3	253.

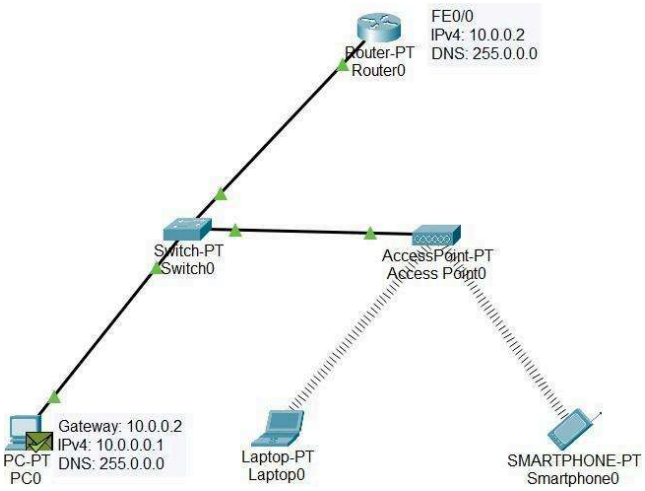
→ pinging
At 10.0.0.2
ping 20.2.0.2
Request timed out
Reply from 20.0.0.2 bytes = 32 TTL = 126
Reply from 20.0.0.2 bytes = 32 TTL 126
Reply from 20.0.0.02 bytes = 32 TTL = 126

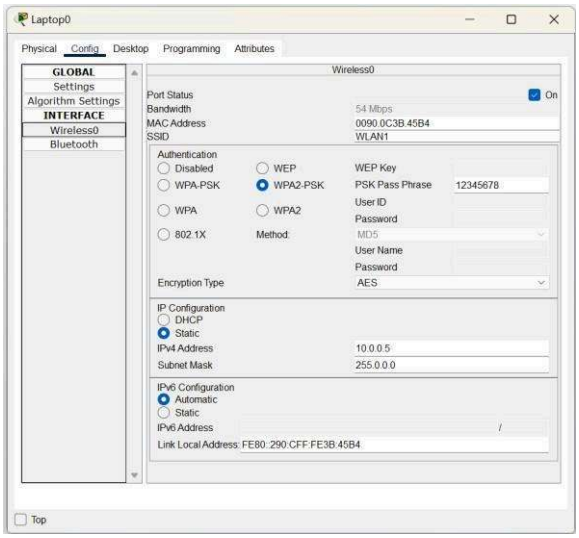
Ping statistics
Packets sent = 4, Received = 3 Lost %

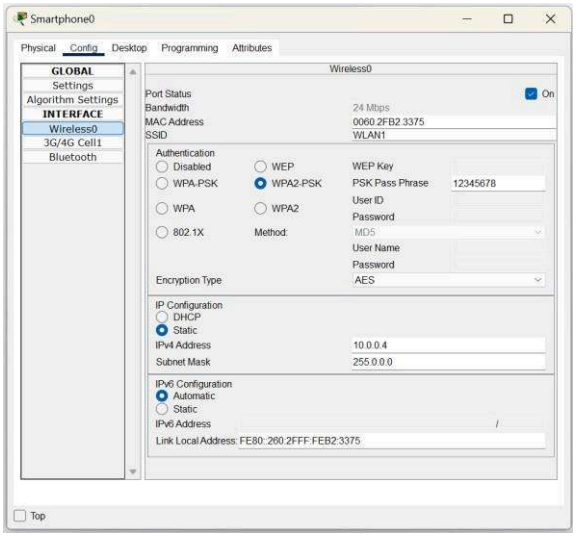
Lee
20/11/24

PROGRAM-9

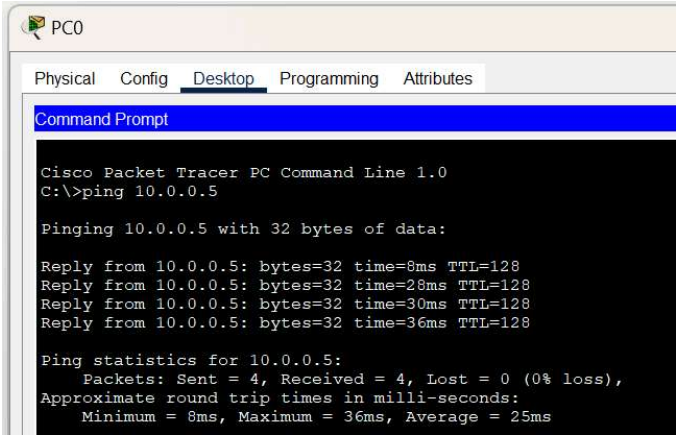
To demonstrate communication between two devices using a wireless LAN.







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



18/12/24

SURYA Gold

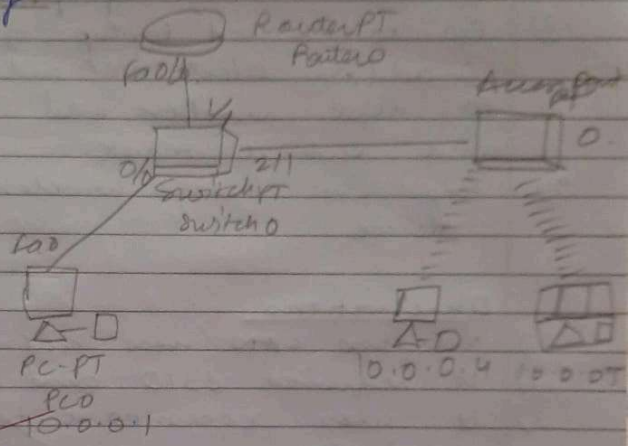
Date Page

LAB- 12

To construct a WLAN and make nodes communicate wirelessly

Aim: * Setting up virtual LAN configuring them to communicate with each other

Topology:



Procedure:

1) Set up a router, connect it to a switch and connect to PCs.
2) configure IP addresses, PCs and router
3) Select acupoint P1 and P0 from wire and laptop.
4) Configure acupoint ~~cot~~ port1 SSID name select WEP and give 10 digit key ~~tay~~ in port1
5) configure PC1 and laptop with the wireless WLAN.

SURYA Gold
Date Page

5) Go to PC4, switch off the device, drag PT-host-NM-LAN to LNS component.

6) Drag WMP300N interface to the empty port, switch on.

7) In both PC and laptop.

8) Go to PC and laptop, go to wireless 0 add SSID select WEP and add key 10 digit number. Add IP address and subnet mask.

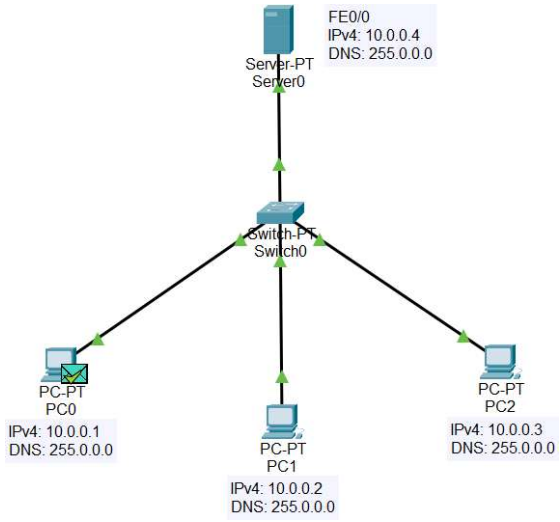
9) Ping for heavy wireless to wired device.

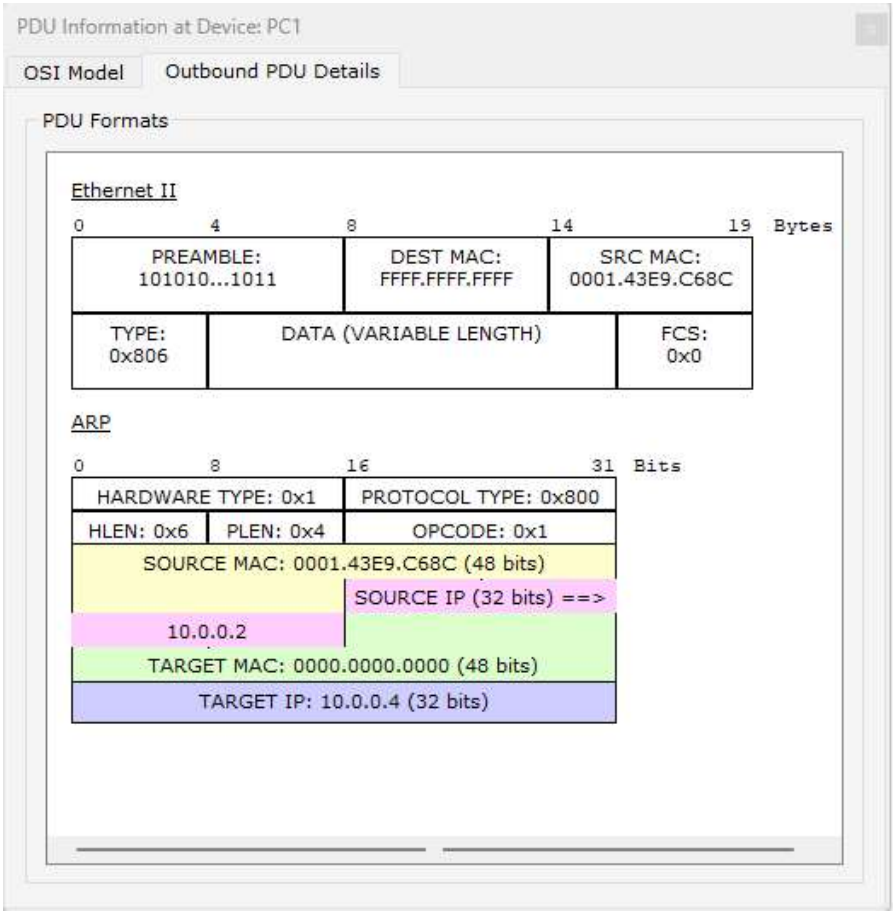
Observations

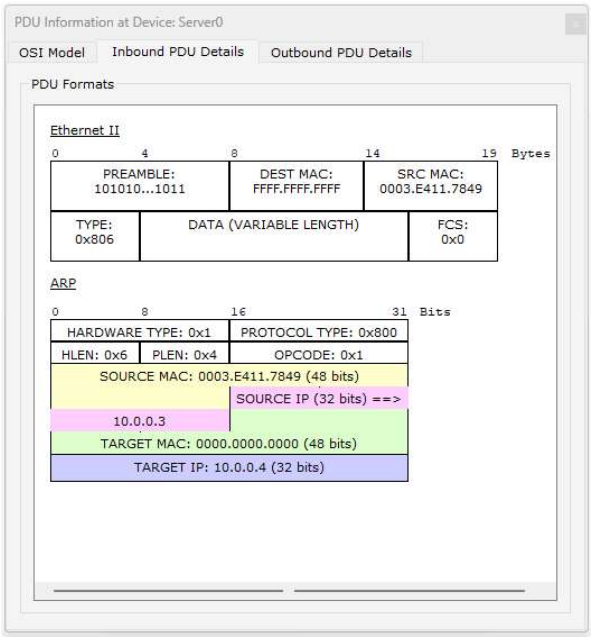
1) Wireless LAN uses WEP protocol
2) It requires SSID and key present
3) It uses access point to establish wireless connection
4) The wireless device and wired device can communicate with each other.

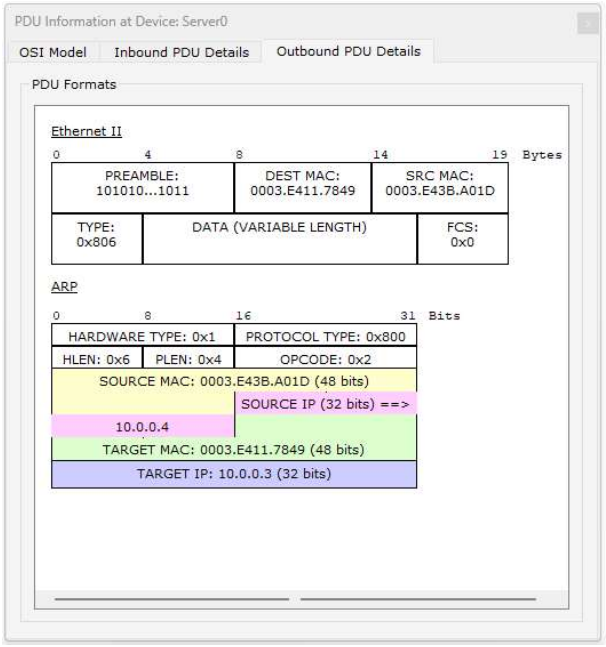
PROGRAM-10

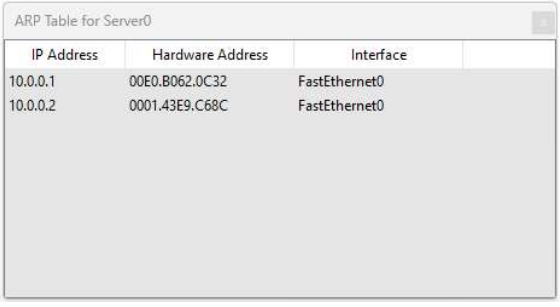
To demonstrate the working of Address Resolution Protocol (ARP) within a LAN for communication.






PDU Information at Device: Server0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0
4
8
14
19
Bytes
PREAMBLE:
101010...1011
DEST MAC:
FFFF.FFFF.FFFF
SRC MAC:
0003.E411.7849
TYPE:
0x806
DATA (VARIABLE LENGTH)
FCS:
0x0
ARP
0
8
16
31
Bits
HARDWARE TYPE: 0x1
PROTOCOL TYPE: 0x800
HLEN: 0x6
PLEN: 0x4
OPCODE: 0x1
SOURCE MAC: 0003.E411.7849 (48 bits)
SOURCE IP (32 bits) ==>
10.0.0.3
TARGET MAC: 0000.0000.0000 (48 bits)
TARGET IP: 10.0.0.4 (32 bits)


PDU Information at Device: Server0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0
4
8
14
19
Bytes
PREAMBLE:
101010...1011
DEST MAC:
0003.E411.7849
SRC MAC:
0003.E43B.A01D
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: 0003.E43B.A01D (48 bits)
SOURCE IP (32 bits) ==>
10.0.0.4
TARGET MAC: 0003.E411.7849 (48 bits)
TARGET IP: 10.0.0.3 (32 bits)


ARP Table for Server0
IP Address
Hardware Address
Interface
10.0.0.1
00E0.B062.0C32
FastEthernet0
10.0.0.2
0001.43E9.C68C
FastEthernet0

ARP Table for PC1

IP Address	Hardware Address	Interface
10.0.0.4	0003.E43B.A01D	FastEthernet0

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Server0	ICMP		0.000	N	0	(edit)	

PC0

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=23ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 23ms, Average = 5ms
```

SURYA Gold
Date Page
20/11/2024 LAB-6
Aim configure routing information protocal.
Topology.
PC5
30.0.0.3
Gateway 30.0.0.0
PC4
30.0.0.2
Gateway 30.0.0.1
PC3
20.0.0.3
Gateway 20.0.0.1
PC2
40.0.0.2
Gateway 20.0.0.1
PC1
10.0.0.3
Gateway 10.0.0.1
PC0
10.0.0.2
Gateway 10.0.0.1
Serial2/0
30.0.0.1
Serial3/0
50.0.0.1
Serial2/0
40.0.0.1
Serial4/0
10.0.0.1

SURYA Gold
Date Page

- RIP protocol

Router 1
enable
config terminal
router rip
network 20.0.0.0
network 40.0.0.0
network 50.0.0.0
exit

R1

show ip route
C - connected S - static I-IGRP R-RIP M-mobile
B - BGP D - EIGRP - - P -
Gateway of last resort is not set
R 10.0.0.0/8 [120/1] via 40.0.0.1 serial 2/
C 20.0.0.0/8 is directly connected
C 40.0.0.0/8 is directly connected
C 50.0.0.0/8 is directly connected.

R2 show ip route
Gateway of last resort is not set

R 10.0.0.0/8 [120/2] via 50.0.0.1
R 20.0.0.0/8 [120/1] via 50.0.0.9
C 30.0.0.0/8 is directly connected
R 40.0.0.0/8 [120/1] via 50.0.0.1
C 50.0.0.0/8 is directly connected.

SURYA Gold

Date Page

Observation

PDU information

TTL - Time to Leave decreases as packet moves through

Example:

PC0 - Source

PC3 - Destination

PC0 - Switch0

TTL 255

Switch - Router1

TTL 254.

		TTL
•	PC0 - Switch0	255
•	PC Switch - Router0	255
•	Router0 - Router1	254
•	Router1 - Switch1	254
•	Switch1 - PC3	253.

→ pinging

t ping 20.2.0.2

.0.0.2 Request timed out

Reply from 20.0.0.2 bytes = 32 TTL = 126

Reply from 20.0.0.2 bytes = 32 TTL 126

Reply from 20.0.0.02 bytes = 32 TTL = 126

Ping statistics

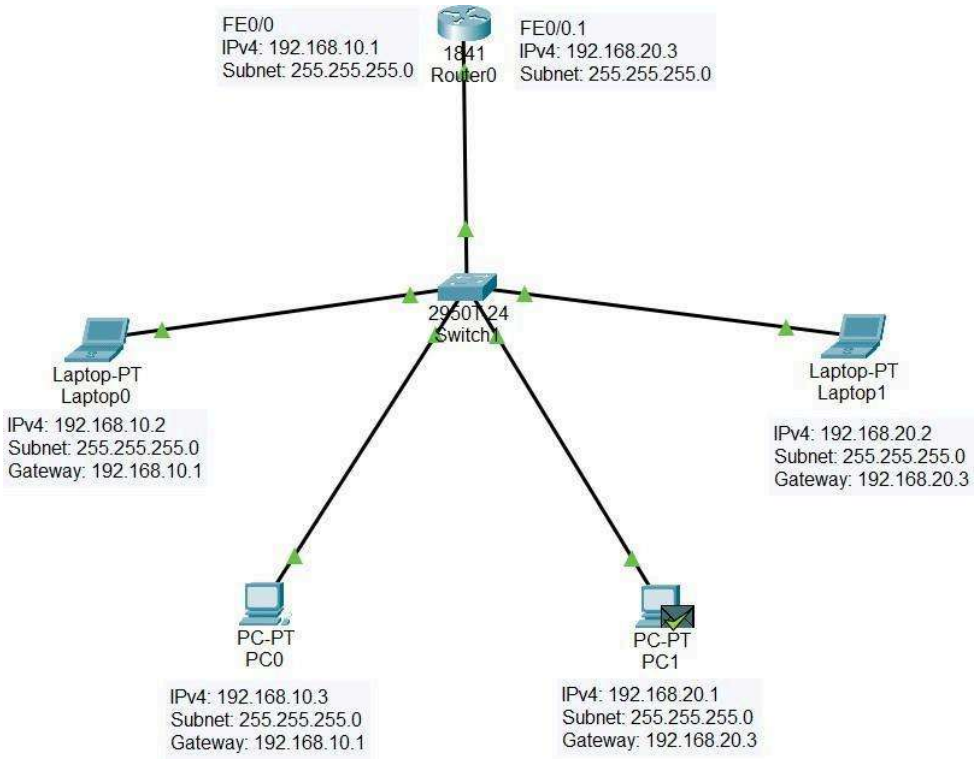
Packets sent = 4, Received = 3 Lost = 1

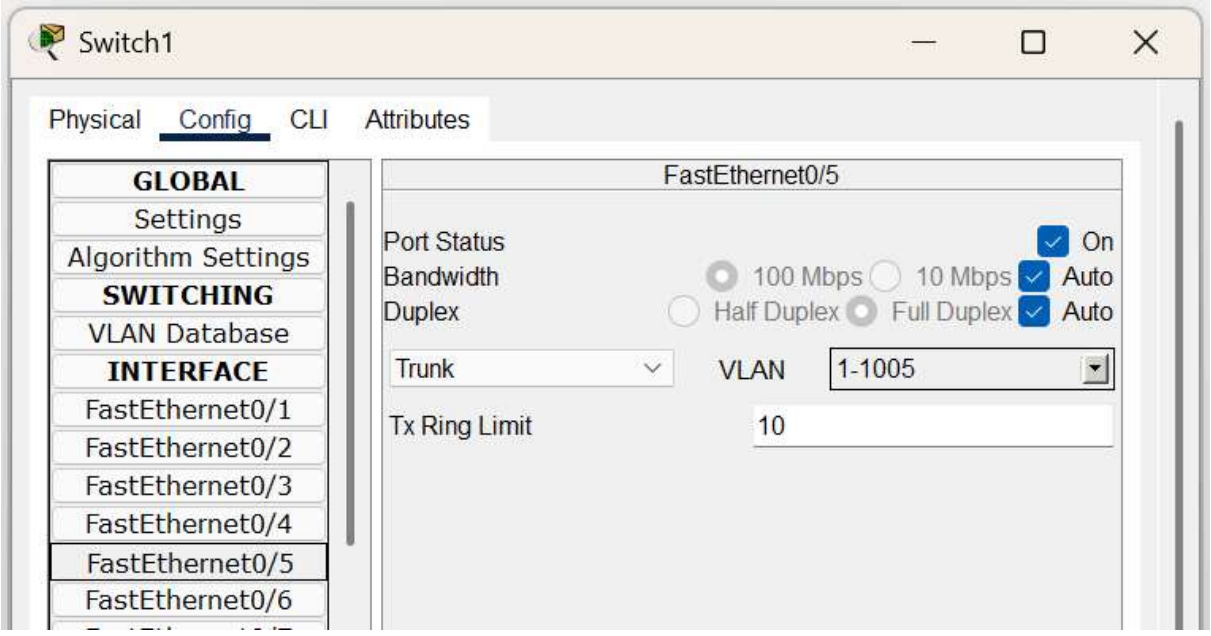
Lee

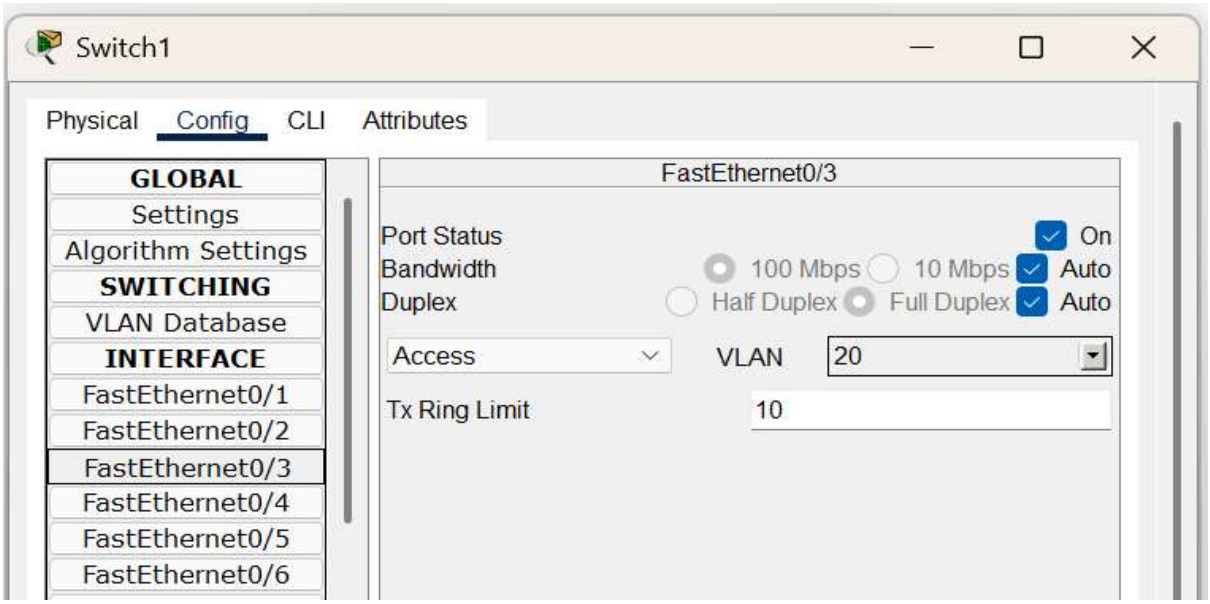
20/11/24

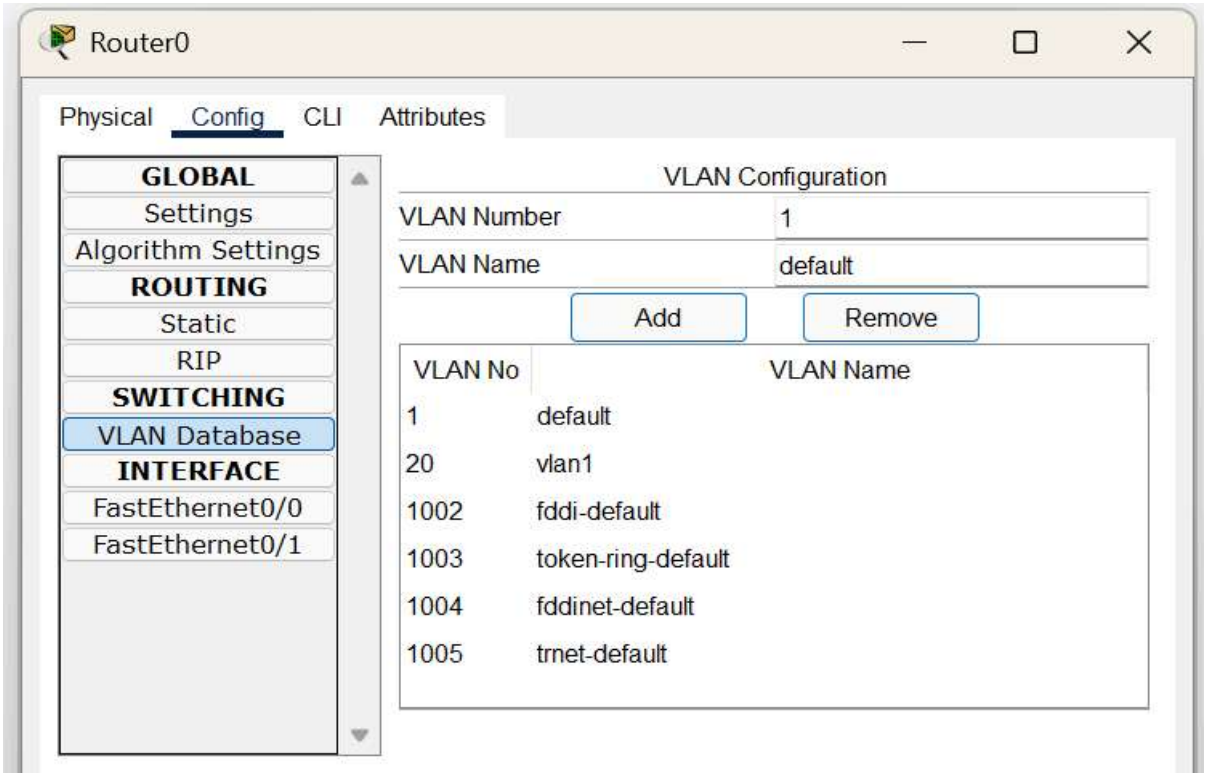
PROGRAM-11

To create a VLAN on top of the physical LAN and enable communication between physical LAN and virtual LAN.










```
Router(config)#interface FastEthernet0/0.1
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.20.3 255.255.255.0
Router(config-subif)#no shutdown
```

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	Router0	ICMP		0.000	N	0	(edit)	

```
C:\>ping 192.168.20.3

Pinging 192.168.20.3 with 32 bytes of data:

Reply from 192.168.20.3: bytes=32 time=2ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.20.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

Mantai
Date
Page 25.

Date: 03/12/2024:

#). Aim: To create a virtual LAN on top of the physical LAN and enable communication b/w physical LAN and virtual LAN.

→ Trunk: Interface that helps to pass messages b/w multiple LAN's

→ Switch:- 2950T-24

→ Router: 1841/Router 0

Router configuration:

```
enable
config t
#interface fastethernet 0/0
ip address 192.168.10.1  255.255.255.0
no shut down
```

Split interface into physical & virtual lan:

```
enable
config t
interface fa 0/0.1
ip address 192.168.20.3  255.255.255.0
exit
```

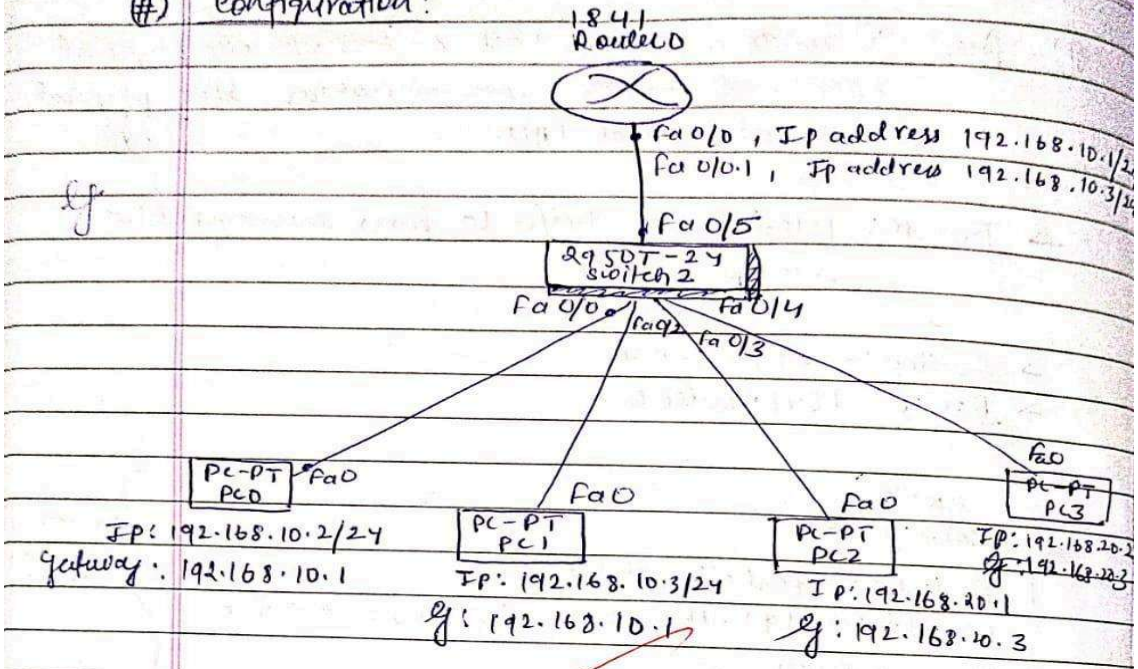
Router CLI:

```
config t
interface fastethernet0/0.1
encapsulation dot1q 20   {20 is vlan id}
```

CRC - $x^8+x^5+x^1+1$

Manjal
Date
Page 26.

(#) Configuration:



(#) Configuration steps:

(i) Configure the above configuration using 1841 router, 2950T switch & Generic PC's.
Connect the devices using automatically chosen connection type.

(ii) Setup ip address of PC0 & PC1 correct network 192.168.10.X.

(iii) Setup ip address of router for Fa 0/0 as 192.168.10.1 & add the same for gateways of PC0 & PC1.

(iv) In switch's VLAN database add a VLAN with VLAN number 20, & VLAN name as VLAN1.

(v) Now set VLAN as 20 for Fa 0/4, Fa 0/3, i.e connection of PC2 & PC3.

PROGRAM-12

Write a program for congestion control using Leaky bucket algorithm.

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

```
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
```

SURYA Gold
Date Page

cycle: 2.

1. Leaky Bucket Algorithm.

Write a program for congestion control using leaky bucket algorithm.

code:

```
storage = int(input("Enter initial packets in the bucket:"))
nog = int(input("Enter total number of times bucket content is checked"))
input_pkt = int(input("No. of packets that enter a bucket at a time:"))
output_pkt = int(input("No. of packets that exit a bucket at a time"))
bucket_size = int(input("
for i in range(nog):
    sizel = bucket_size - storage
    if input_pkt <= sizel:
        storage += input_pkt
    else
        print("Packet loss:", input_pkt)
    print(f"Buffer size = {storage} out of bucket size = {bucket_size}")
    storage -= output_pkt

    if storage < 0:
        storage = 0
```

Date Page

Output

Enter initial packets in the bucket = 0
Enter number of queries = 4
Enter total no. of packets bucket can accomodate: 10
Enter no. of packets that enter the bucket at a time: 4
Enter the no of packets that exit 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-13

Write a program for error detecting code using CRC-CCITT (8-bits).

Code

```
def xor(a, b):
    # XOR operation between two binary strings
    result = []
    for i in range(1, len(b)):
        result.append('0' if a[i] == b[i] else '1') return
    ''.join(result)

def mod2div(dividend, divisor): #
    Performs Modulo-2 division
    pick = len(divisor)
    tmp = dividend[:pick]

    while pick < len(dividend): if
        tmp[0] == '1':
            tmp = xor(divisor, tmp) + dividend[pick]
        else:
            tmp = xor('0' * pick, tmp) + dividend[pick]
        pick += 1

    # For the last set of bits if
    tmp[0] == '1':
        tmp = xor(divisor, tmp)
    else:
        tmp = xor('0' * pick, tmp)

    return tmp

def  encode_data(data,  key):
    #  Encode  data  with  CRC
    l_key = len(key)
    padded_data = data + '0' * (l_key - 1)
    remainder = mod2div(padded_data, key)
    codeword = data + remainder
    return codeword, remainder

def check_data(received_data, key): #
    Check received data for errors
    remainder = mod2div(received_data, key)
    return '0' * (len(key) - 1) == remainder

# Main program
if _name___== "_main_":
```

```
    print("Error Detection using CRC-CCITT (8-bits)")

# Transmitter
    data = input("Enter data to be transmitted: ").strip()
    key = input("Enter the Generating polynomial: ").strip()

    print("\n                                                ")
    padded_data = data + '0' * (len(key) - 1) print("Data
    padded with n-1 zeros:", padded_data)

    encoded_data, crc = encode_data(data, key)
    print("CRC or Check value is:", crc)
    print("Final data to be sent:", encoded_data)
    print("                                                ")

    # Receiver
    received_data = input("\nEnter the received data: ").strip()
    print("\n                                    ")
    print("Data received:", received_data)

    if check_data(received_data, key): print("No
       error detected")
    else:
       print("Error detected")
    print("                                    ")
```

Output

```
Enter data to be transmitted: 1001100
Enter the Generating polynomial: 100001011

----------------------------------------
Data padded with n-1 zeros: 100110000000000
CRC or Check value is: 0100010
Final data to be sent: 10011000100010
----------------------------------------

Enter the received data: 10011000100011

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

```
Error Detection using CRC-CCITT (8-bits)
Enter data to be transmitted: 1001100
cell output actions rating 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: 100110010100010

------------------------------
Data received: 100110010100010
No error detected
------------------------------
```

SURYA Gold
Date Page

2. Write a program for error detecting code using CRC-CCITT(8-bit)

```
def xor(a,b):
    result = []
    for i in range(1, len(b)):
        result.append('0' if a[i]==b[i] else '
    return result

def mod2div( dividend , divisor):
    pick = len(dividend)
    tmp = dividend[:pick]
    while pick < len(dividend):
        if tmp[0] = '1':
            tmp = xor(divisor, tmp) +
                dividend[pick]
        else:
            tmp = xor('0'*pick, tmp) + dividend[pi
        pick+=1
    if tmp[0] == '1':
        tmp = xor(divisor, tmp):
    else:
        tmp = xor('0'*pick, tmp).
    return tmp.

def encode_data(data, key):
    lkey = len(key)
    padded_data = data + '0'*(lkey-1)
    remainder = mod2div(padded_data, key)
    codeword = data+remainder
    return codeword, remainder.
```

```
def check_data(received_data, key):
    remainder = mod2div(received_data,key)
    return 0*(len(key)-1) == remainder

if __name__ == "__main__":
    data = input("Enter data").strip()
    key = input("Enter generating polynomial").strip()
    encoded data,crc = encode_data(data,
        key)
    print(crc)
    received_data = input("Enter ... data").
        strip()
    if checkdata(recieved data 3,key):
        print("No error")
    else:
        print("Error")
```

Outputs:

Enter data : 1001100

Enter generating polynomial : ~~1001100~~ 1000101

Data padded with n-1 zeroes:

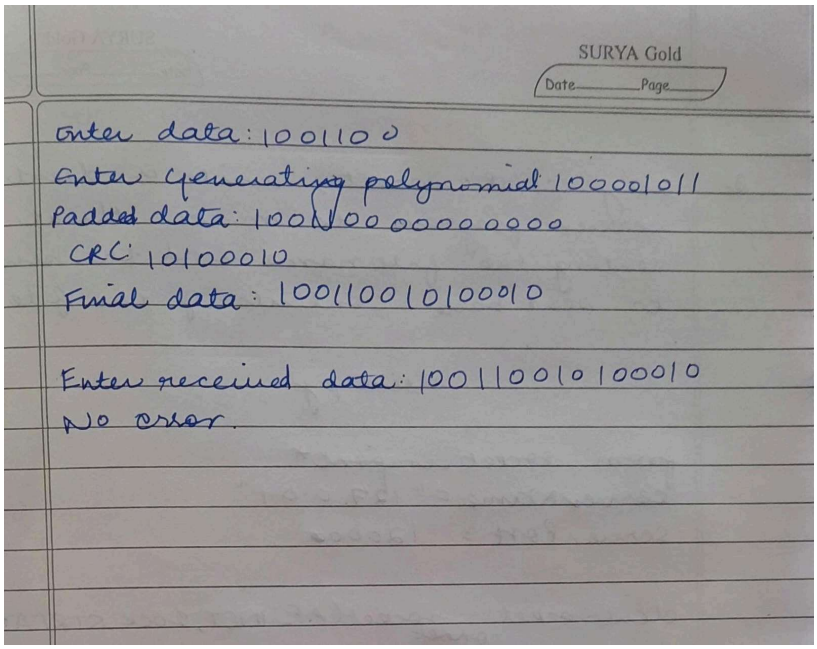
1001100 000000

CRC: 0100010

Final data: 1001100100010

Enter the received data: 1001100100011

Error detected.


SURYA Gold
Date Page
Enter data: 1001100
Enter Generating polynomial 100001011
Padded data: 1001100000000000
CRC: 10100010
Final data: 10011001010010
Enter received data: 10011001010010
No error.

PROGRAM-14(A)

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.

Code: Client.py

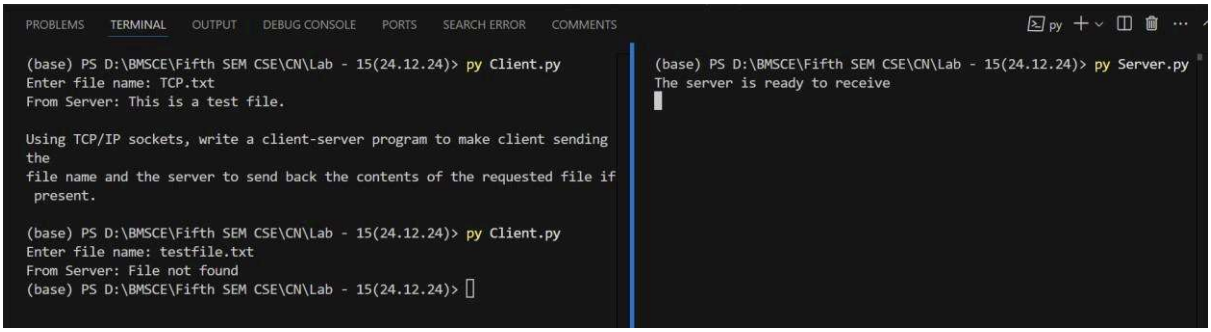
```
from socket import *
serverName = "127.0.0.1" # Server address (localhost)
serverPort = 12000 # Port number where the server listens

# Create TCP socket
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort)) # Connect to server

# Ask user for file name to request
sentence = input("Enter file name: ")

# Send file name to server
clientSocket.send(sentence.encode())

# Receive file contents from server
filecontents = clientSocket.recv(1024).decode()
print('From Server:', filecontents)

# Close the connection
clientSocket.close()
```

Code: Server.py

```
from socket import *
serverName = "127.0.0.1" # Server address (localhost)
serverPort = 12000 # Port number to listen on

# Create TCP socket
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort)) # Bind socket to the address and port
serverSocket.listen(1) # Listen for 1 connection
print("The server is ready to receive")

while True:
    # Accept a connection
    connectionSocket, addr = serverSocket.accept()

    # Receive the file name from the client
    sentence = connectionSocket.recv(1024).decode()

    # Try opening the file try:
```

```
file = open(sentence, "r") # Open file in read mode
            fileContents = file.read(1024) # Read file content (up to 1024 bytes)
            file.close()
        except FileNotFoundError:
            # Send error message if file not found
            connectionSocket.send("File not found".encode())

        # Close the connection
        connectionSocket.close()
```

Output



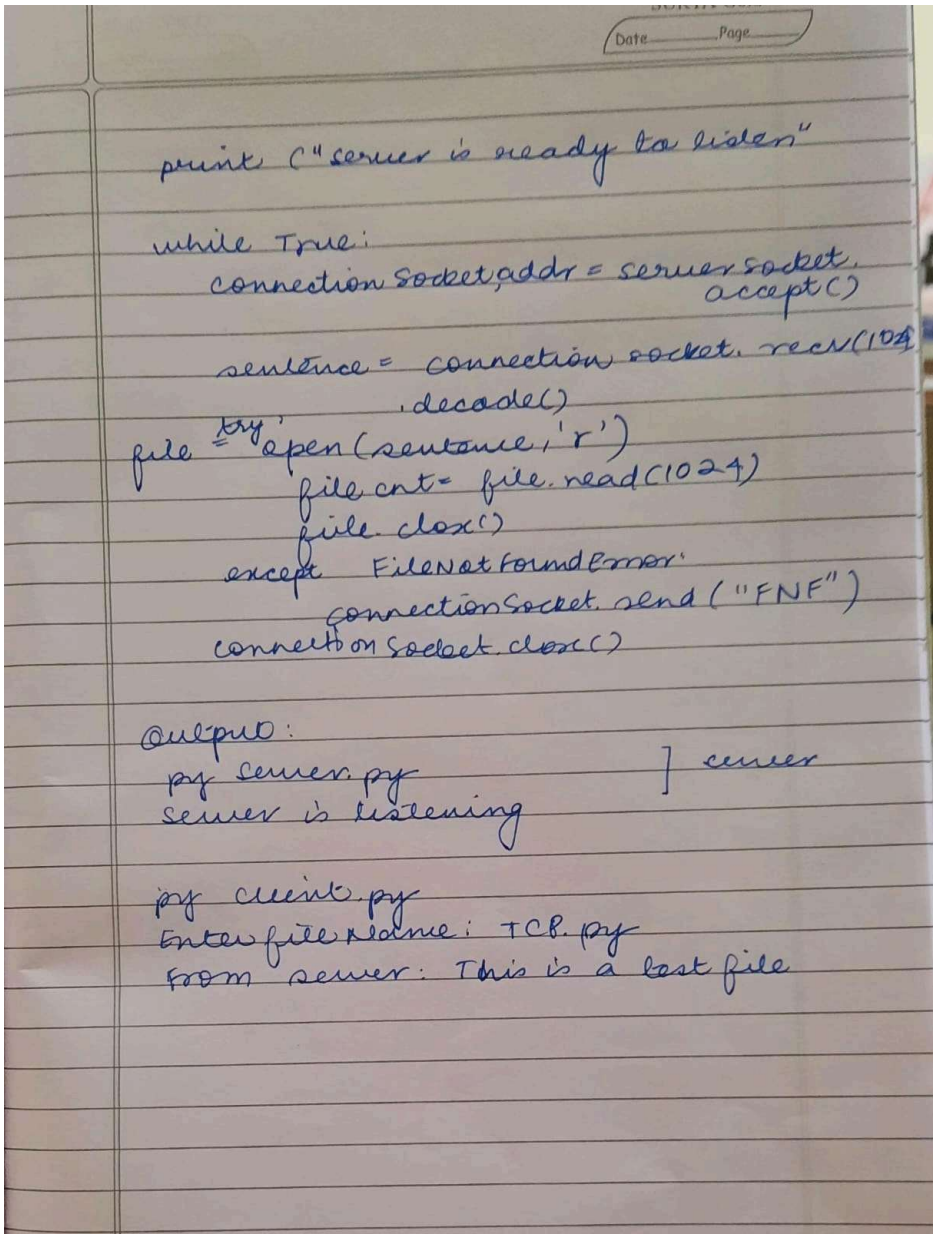
3. Using TCP/IP sockets, write a client server program to make client sending the file name and the server to send back contents of the file.

client.py

```
from socket import*
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
filename = input("Enter file name")
clientSocket.send(filename.encode())
filecats = clientSocket.recv(1024).decode()
print(filecnts)
clientSocket.close()
```

server.py

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



```
print ("server is ready to listen"

while True:
    connectionSocket,addr = serverSocket.
                                accept()
    sentence = connectionSocket.recv(1024
                .decode()
    file = try:open(sentence,'r')
           fileCnt = file.read(1024)
           file.close()
    except FileNotFoundError:
           connectionSocket.send("FNF")
    connectionSocket.close()
```

@output:

```
py server.py                          ] server
server is listening

py client.py
Enter filename: TCP.py
From server: This is a test file
```

PROGRAM-14(B)

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.

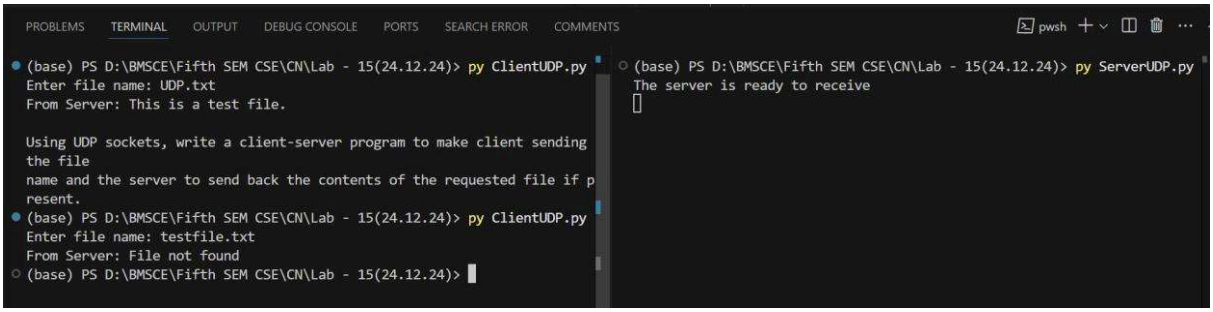
Code: ClientUdp.py

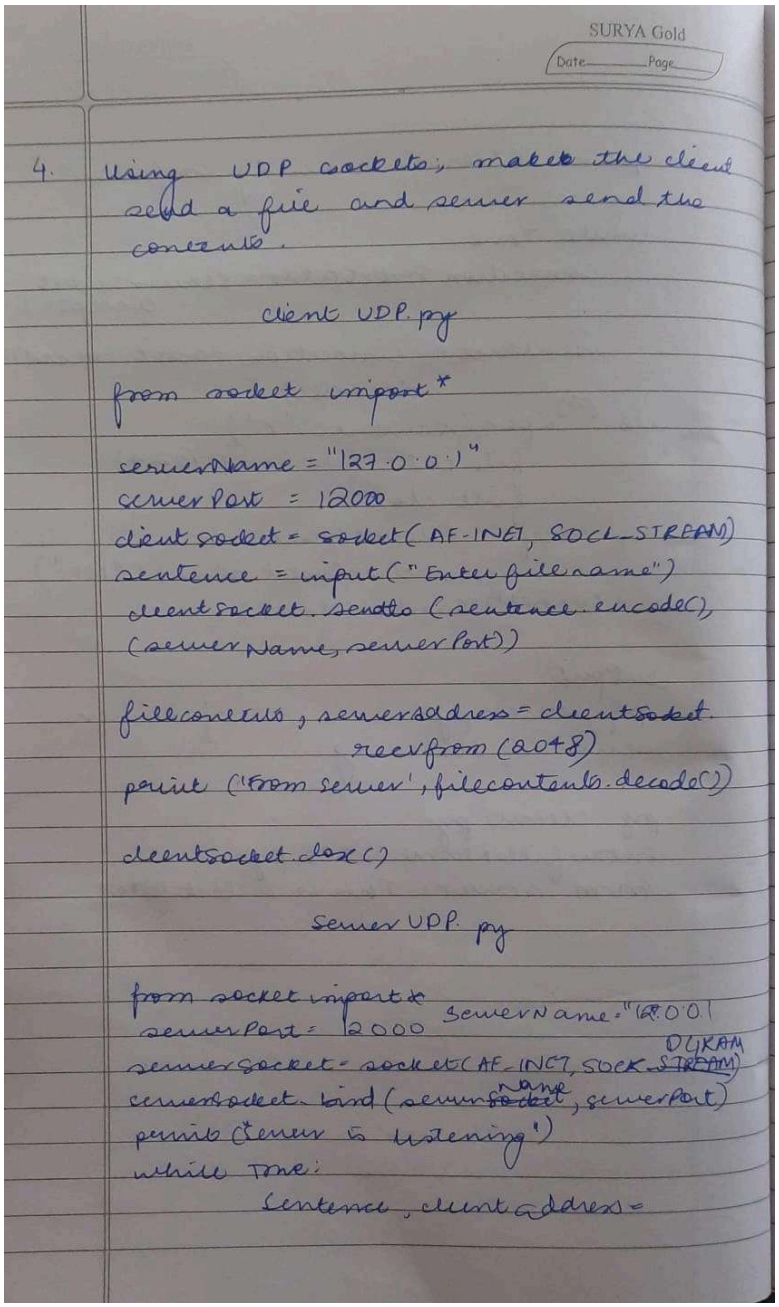
```
from socket import *

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

sentence = input("Enter file name: ")
clientSocket.sendto(sentence.encode(), (serverName, serverPort))

filecontents, serverAddress = clientSocket.recvfrom(2048)
print('From Server:', filecontents.decode())

clientSocket.close()
```

Code: 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 True:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    try:
        with open(sentence.decode(), "r") as file:
            l = file.read(2048)
            serverSocket.sendto(l.encode(), clientAddress)
            print(f"Sent back to client: {l}")
    except FileNotFoundError:
        serverSocket.sendto("File not found.".encode(), clientAddress)
```

Output





SURYA Gold
Date Page

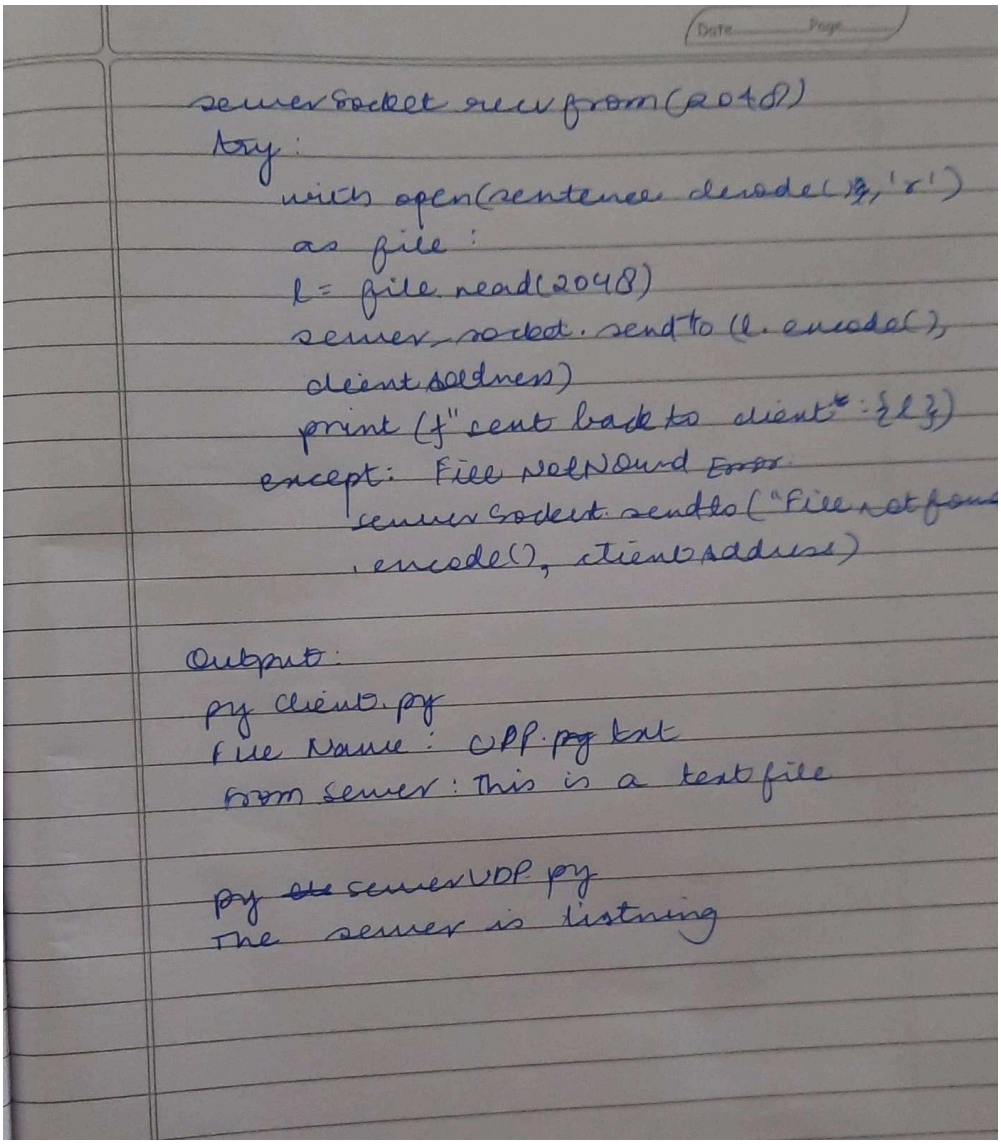
4. Using UDP sockets, make the client send a file and server send the contents.

client UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
sentence = input("Enter filename")
clientSocket.sendto(sentence.encode(),
(serverName, serverPort))

fileContents, serverAddress = clientSocket.
                    recvfrom(2048)
print('From Server', fileContents.decode())

clientSocket.close()
```

Server UDP.py

```
from socket import *
serverPort = 12000   serverName = "127.0.0.1"
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
print('Server is listening')
while True:
        sentence, clientAddress =
```



```
serverSocket.recvfrom(2048)
try:
    with open(sentence.decode(),'r')
    as file:
    l = file.read(2048)
    server_socket.sendto(l.encode(),
    clientAddress)
    print(f"sent back to client:{l}")
except: FileNotFound Error
    serverSocket.sendto("File not found
    ,encode(), clientAddress)
```

Output:

py client.py
File Name: UDP.py txt
from Server: This is a test file

py ~~ste~~ serverUDP.py
The server is listening