VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

Dinesh kumar G(1BM22CS091)

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 **Dinesh kumar G(1BM22CS091)** 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

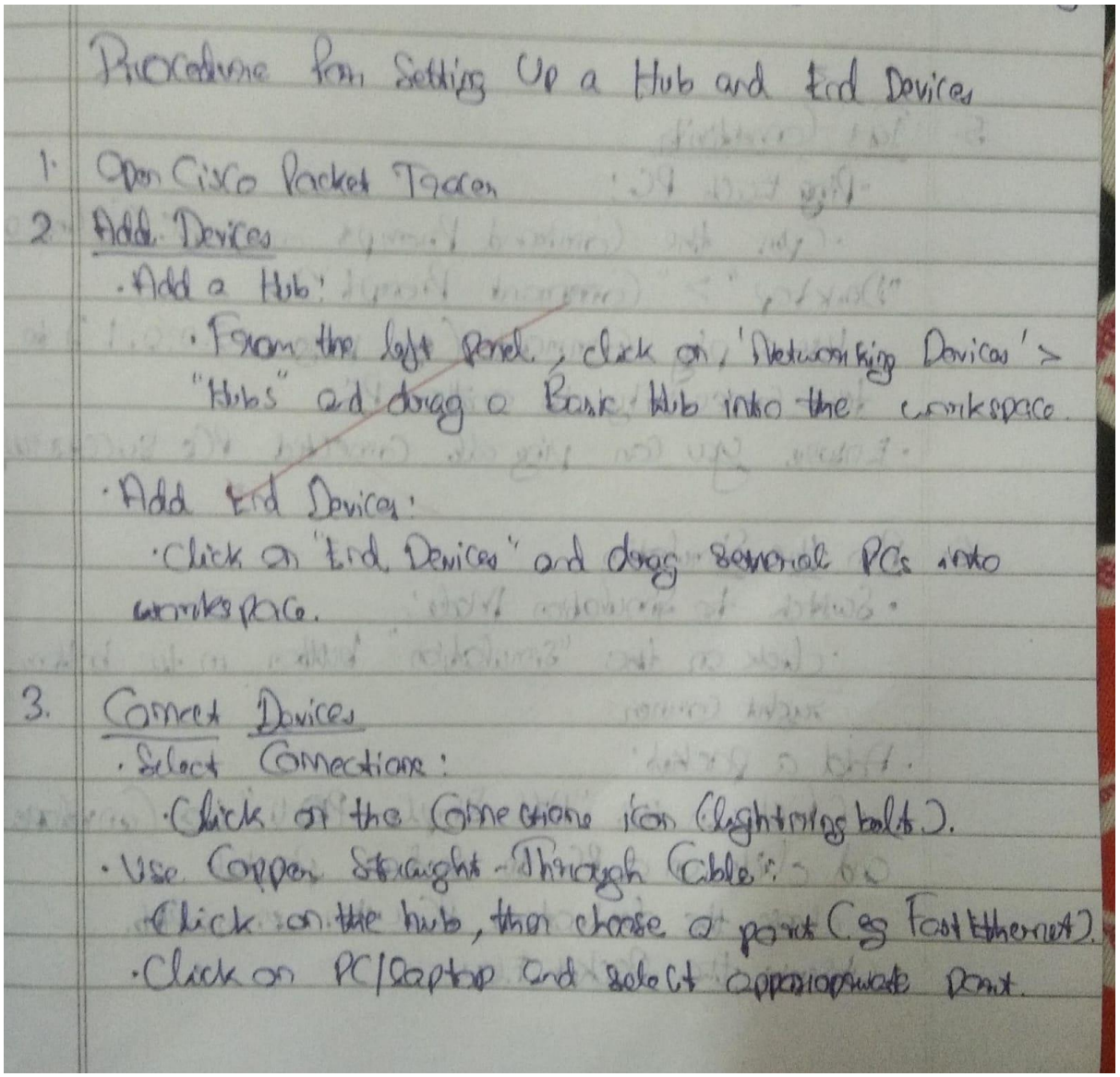
INDEX

INDEX

CYCLE 2

LABORATORY 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 message.



Procedure for Setting Up a Hub and End Devices

1. Open Cisco Packet Tracer
2. Add Devices
   - Add a Hub:
     - From the left panel, click on 'Networking Devices' > "Hubs" and drag a Basic Hub into the workspace.
   - Add End Devices:
     - Click on 'End Devices' and drag several PCs into workspace.
3. Connect Devices
   - Select Connections:
     - Click on the Connections icon (lightning bolt).
   - Use Copper Straight-Through Cable:
     - Click on the hub, then choose a port (eg FastEthernet).
     - Click on PC/Laptop and select appropriate port.

Dinesh

PAGE NO :
DATE :

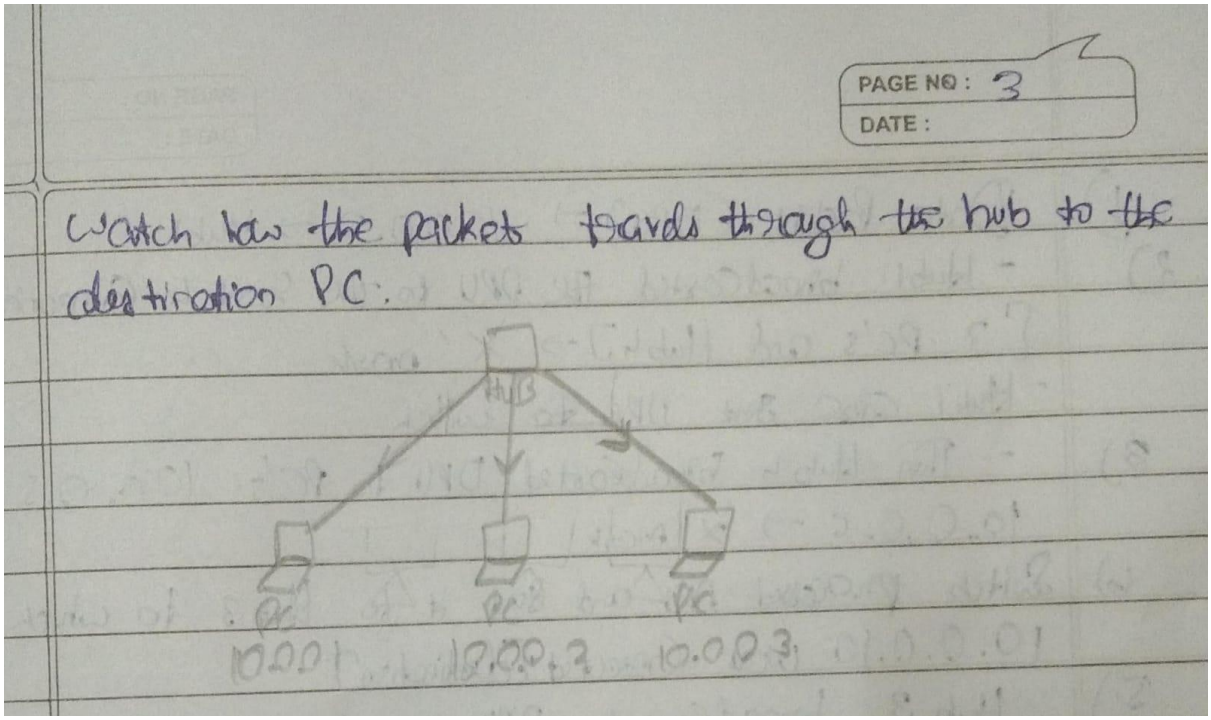
- Repeat for all end devices connected to the hub.

4. Configure End devices:
   - Select each PC:
     - Click on a PC, then click "Desktop" tab.
     - Choose "IP Configuration".
     - Assign an IP address (eg 10.0.0.1 for PC1, 10.0.0.2 for PC2 etc). and a Subnet Mask (eg. 255.255.255.0).
   - Label Devices:
     - Right click on each PC and select "Rename" to give them meaningful names (eg PC1, PC2).

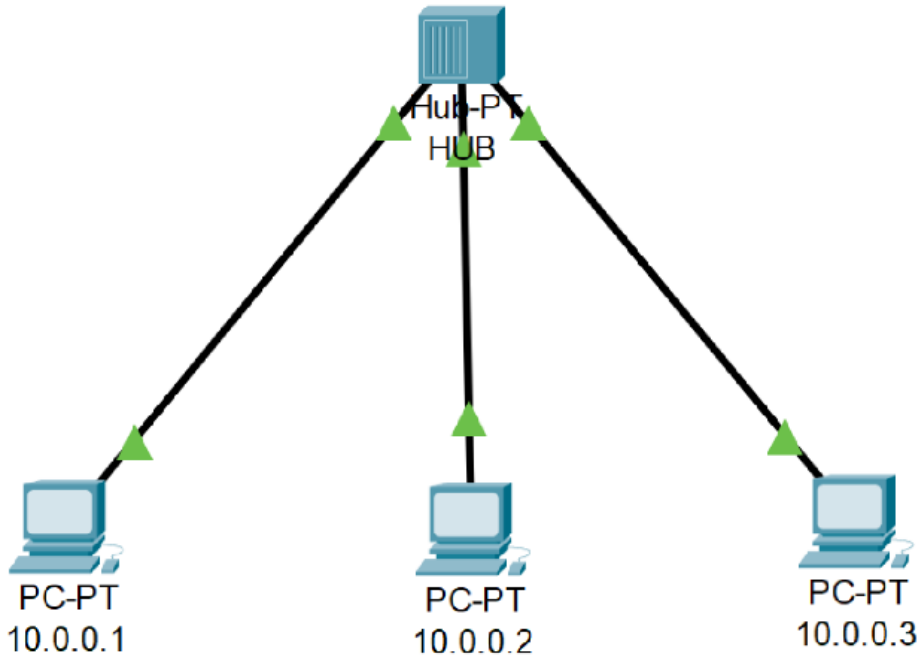
5. Test Connectivity
   - Ping Each PC:
     - Open the Command Prompt on a PC. (Click on "Desktop" > "Command Prompt").
     - Use a ping command (eg ping 10.0.0.1) to test connectivity to another PC.
   - Ensure you can ping all connected PCs successfully

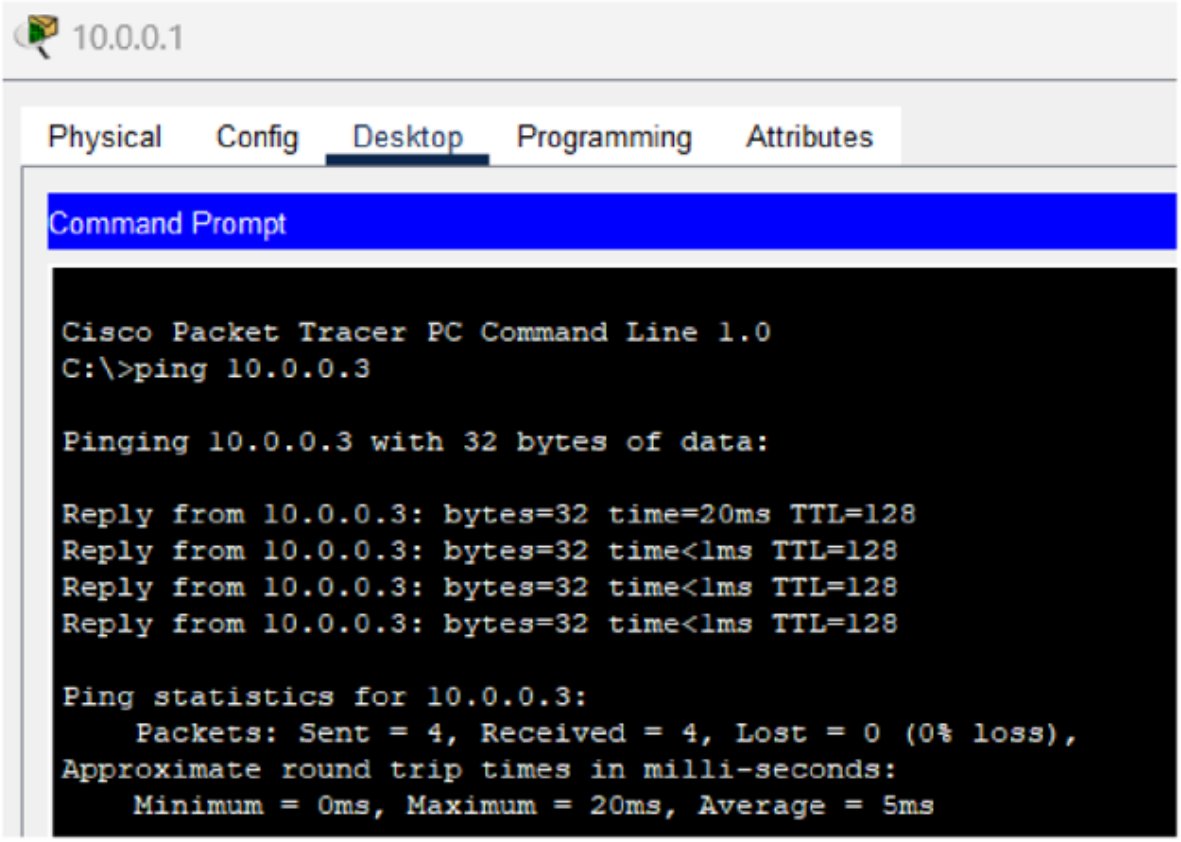
6. Run the Simulation
   - Switch to Simulation Mode:
     - click on the "Simulation" button in the bottom right corner
   - Add a Packet:
     - Click on the "Add Simple PDU" icon (envelope icon) and click on a PC to send a packet.
     - choose the destination PC when prompted.
   - Observe the Packet Flow:

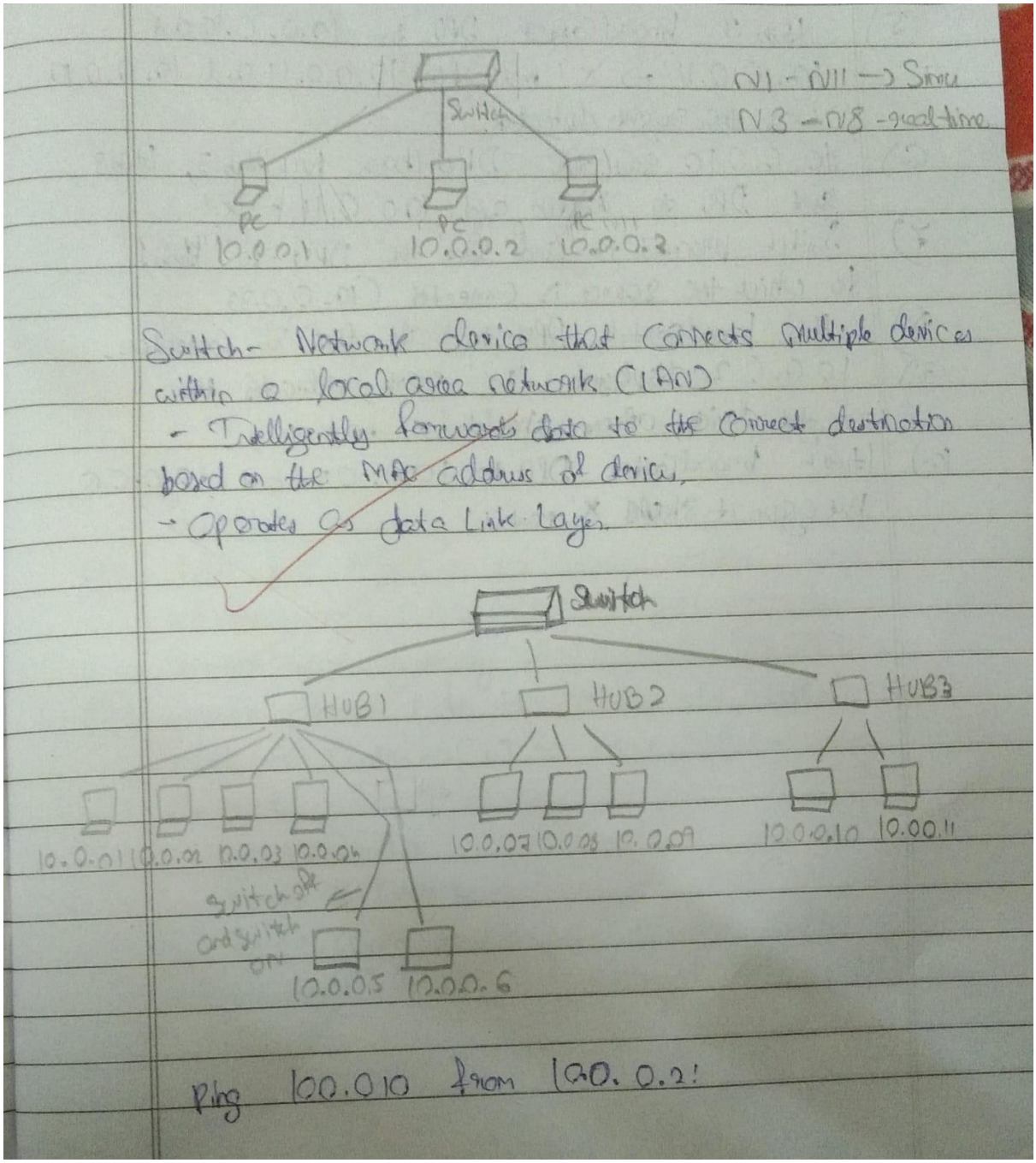
10.


PAGE NO : 3
DATE :
watch how the packets travels through the hub to the destination PC.
HUB
PC
PC
PC
10.0.0.1
10.0.0.2
10.0.0.3

Screenshot:


Hub-PT
HUB
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
10.0.0.3


10.0.0.1
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=20ms 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 = 20ms, Average = 5ms


Switch
PC
PC
PC
10.0.0.1
10.0.0.2
10.0.0.3
NI - NII → Simu
N3 - N8 - real-time
Switch- Network device that connects multiple devices within a local area network (LAN)
- Intelligently forwards data to the correct destination based on the MAC address of device.
- Operates as data Link Layer.
Switch
HUB1
HUB2
HUB3
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.7
10.0.0.8
10.0.0.9
10.0.0.10
10.0.0.11
switch off
and switch on
10.0.0.5
10.0.0.6
Ping 10.0.0.10 from 10.0.0.2!

Procedure for Setting Up a Hub and End Devices

1. Open Cisco Packet Tracer
2. Add Devices
   - Add a Hub:
     - From the left panel, click on 'Networking Devices' > "Hubs" and drag a Basic Hub into the workspace.
   - Add End Devices:
     - Click on "End Devices" and drag several PCs into workspace.
3. Connect Devices
   - Select Connections:
     - Click on the Connections icon (lightning bolt).
   - Use Copper Straight-Through Cable:
     - Click on the hub, then choose a port (eg Fast Ethernet).
     - Click on PC/Laptop and select appropriate port.

Dinesh

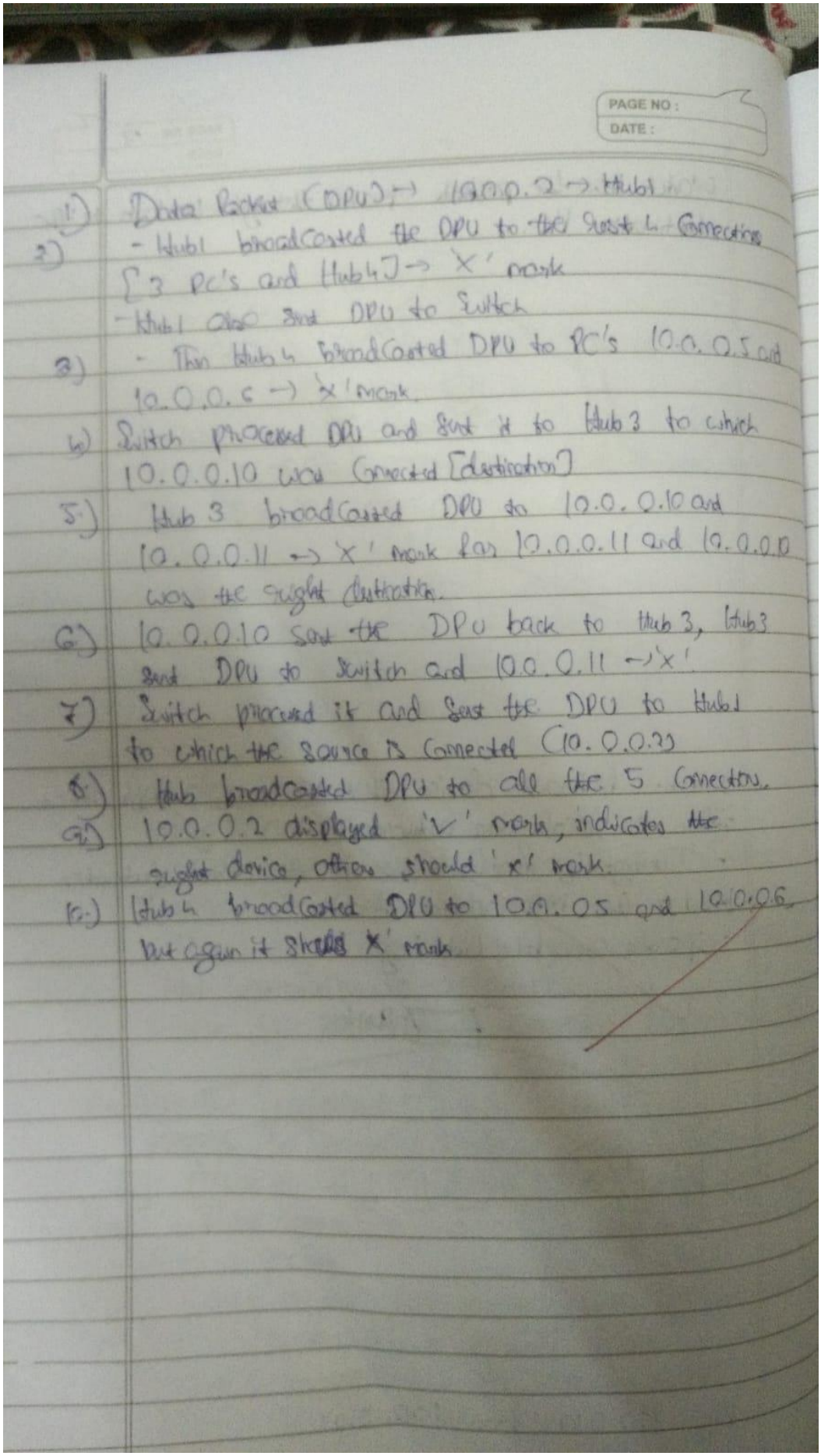
PAGE NO :
DATE :

- Repeat for all end devices connected to the hub.

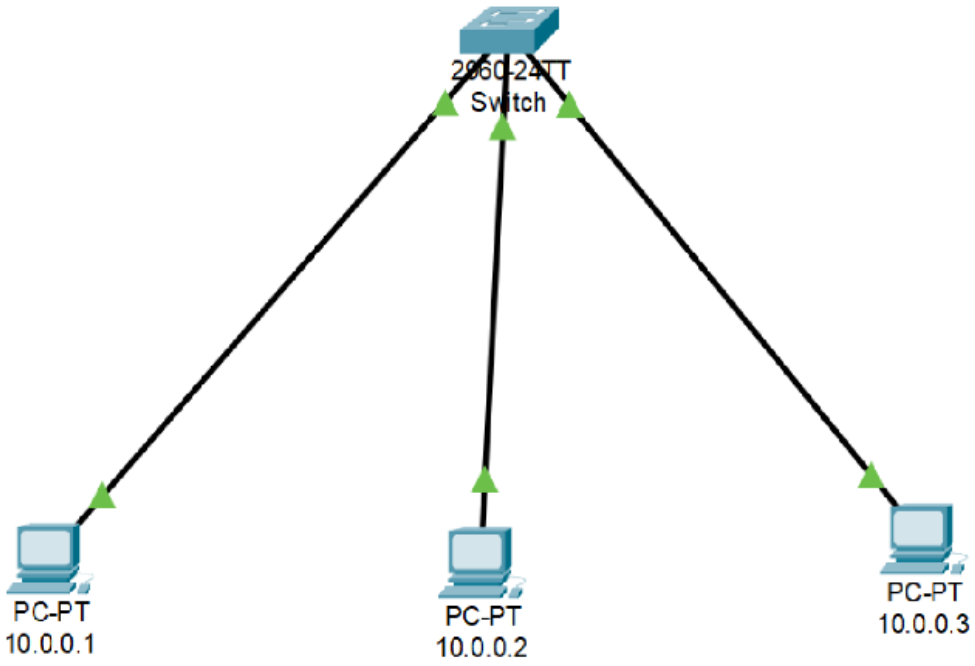
4. Configure End devices:
   - Select each PC:
     - Click on a PC, then click "Desktop" tab.
     - Choose "IP Configuration".
     - Assign an IP address (eg 10.0.0.1 for PC1, 10.0.0.2 for PC2 etc). and a Subnet Mask (eg. 255.255.255.0).
   - Label Devices:
     - Right click on each PC and select "Rename" to give them meaningful names (eg PC1, PC2).

5. Test Connectivity
   - Ping Each PC:
     - Open the Command Prompt on a PC. (Click on "Desktop" > "Command Prompt").
     - Use a ping command (eg ping 10.0.0.1) to test connectivity to another PC.
   - Ensure you can ping all connected PCs successfully

6. Run the Simulation
   - Switch to Simulation Mode:
     - Click on the "Simulation" button in the bottom right corner.
   - Add a Packet:
     - Click on the "Add Simple PDU" icon (envelope icon) and click on a PC to send a packet.
     - Choose the destination PC when prompted.
   - Observe the Packet Flow:



1) Data Packet (DPU) → 10.0.0.2 → Hub1
2) - Hub1 broadcasted the DPU to the rest 4 Connections [3 PC's and Hub4] → 'x' mark
- Hub1 also sent DPU to Switch
3) - Then Hub4 broadCasted DPU to PC's 10.0.0.5 and 10.0.0.6 → 'x' mark.
4) Switch processed DPU and sent it to Hub3 to which 10.0.0.10 was Connected [destination]
5) Hub3 broadCasted DPU to 10.0.0.10 and 10.0.0.11 → 'x' mark for 10.0.0.11 and 10.0.0.0 was the right destination.
6) 10.0.0.10 sent the DPU back to Hub3, Hub3 sent DPU to Switch and 10.0.0.11 → 'x'
7) Switch processed it and sent the DPU to Hub1 to which the source is Connected (10.0.0.2)
8) Hub broadCasted DPU to all the 5 Connections.
9) 10.0.0.2 displayed 'V' mark, indicates the right device, others should 'x' mark.
10) Hub4 broadCasted DPU to 10.0.0.5 and 10.0.0.6 but again it shows 'x' mark.



10.0.0.1

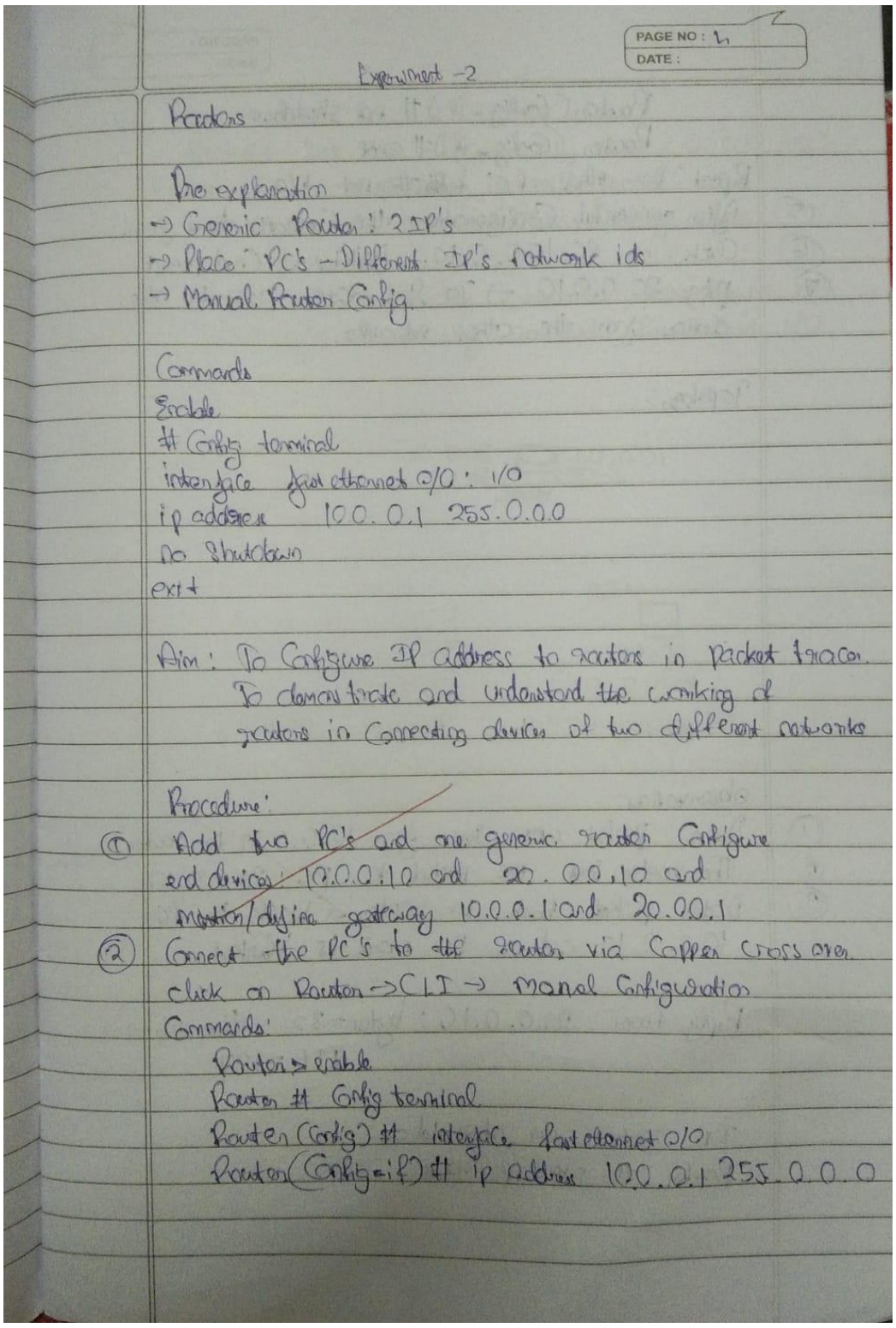
Physical | Config | Desktop | Programming | Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time<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
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 = 0ms, Average = 0ms
```

LABORATORY PROGRAM – 2

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



Experiment -2

PAGE NO : 1
DATE :

Routers

Pre explanation
→ Generic Router : 2 IP's
→ Place : PC's - Different IP's network ids
→ Manual Router Config

Commands
Enable
Config terminal
interface fast ethernet 0/0 : 1/0
ip address 100.0.1 255.0.0.0
no shutdown
exit

Aim : To Configure IP address to routers in packet tracer.
To demonstrate and understand the working of routers in Connecting devices of two different networks

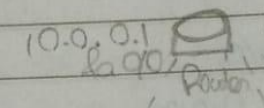
Procedure :
① Add two PC's and one generic router Configure end devices : 10.0.0.10 and 20.00.10 and mention/define gateway 10.0.0.1 and 20.00.1
② Connect the PC's to the router via Copper Cross over
click on Router → CLI → Manual Configuration
Commands :
Router > enable
Router # Config terminal
Router (Config) # interface fastethernet 0/0
Router (Config-if) # ip address 100.0.1 255.0.0.0

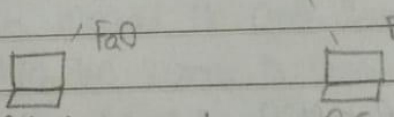
PAGE NO :
DATE :

Router(Config-if)# no shutdown
Router(Config-if)#exit
Repeat for other PC: fastethernet 1/0

(5) After successful Configuration, the Connection turns green.
(6) Click on PC 10.0.0.10 → Desktop → Cmd prompt
(7) ping 20.0.0.10 → To Send data packet to other device from the other networks.

Topology:





PC-PT
10.0.0.10
Def gateway 10.0.0.1

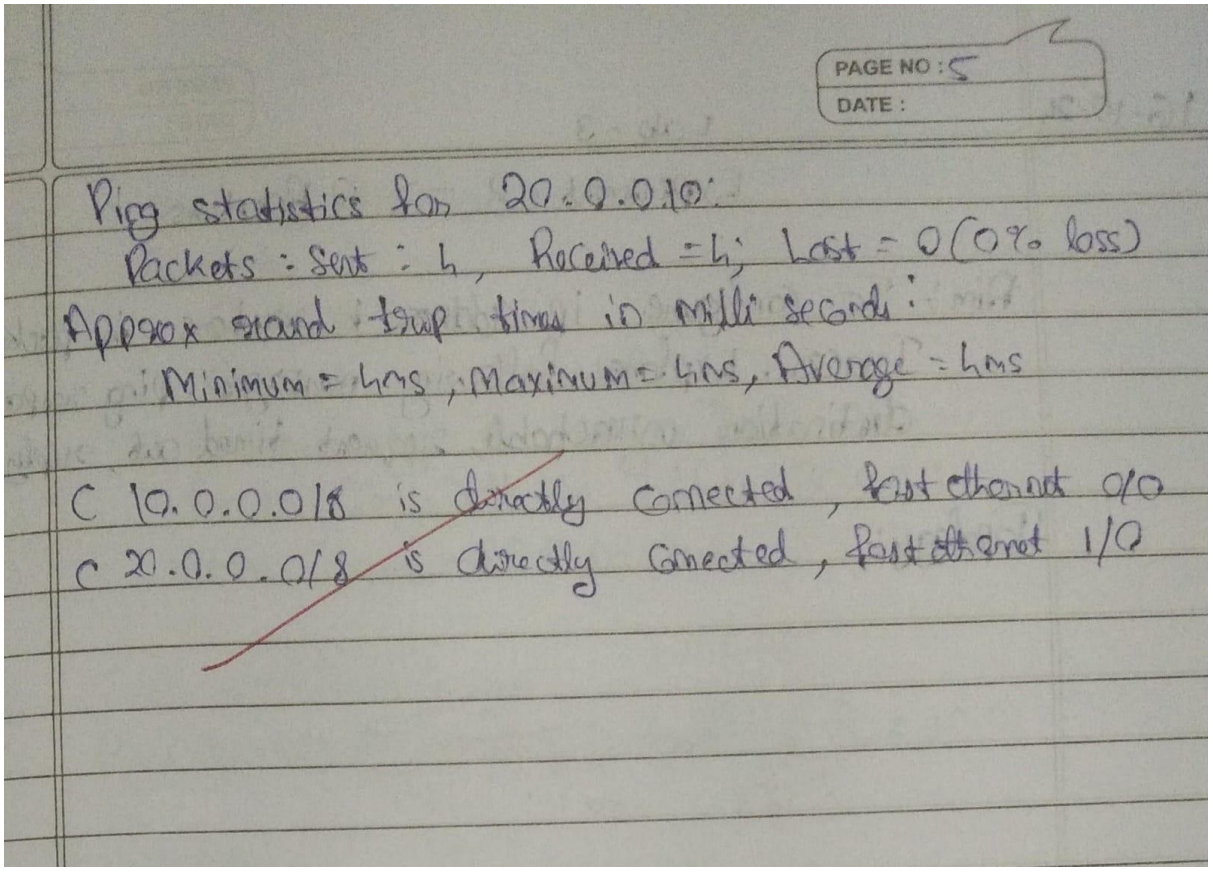
PC-PT
20.0.0.10
Def gateway 20.0.0.1

Observation:

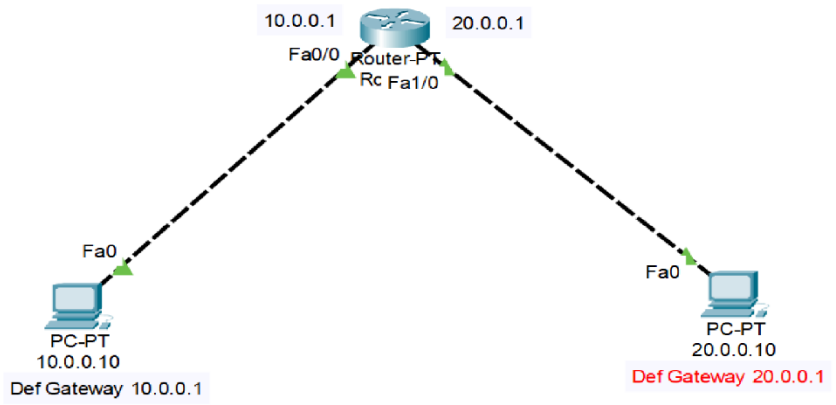
(1) Data packet was sent from 10.0.0.10 to router.
(2) The router sent the packet to PC 20.0.0.10
(3) Data packet back to router → back to PC 10.0.0.10 and a tick mark is blinked.

Reply from 20.0.0.10: bytes=32 time=4ms TTL=127

IP route was observed as:
Router# show ip route


PAGE NO : 5
DATE :
Ping statistics for 20.0.0.10:
Packets : Sent : 4, Received = 4; Lost = 0 (0% loss)
Approx round trip times in milli seconds:
Minimum = 4ms, Maximum = 4ms, Average = 4ms
C 10.0.0.0/8 is directly connected, fast ethernet 0/0
C 20.0.0.0/8 is directly connected, fast ethernet 1/0

Screenshots:



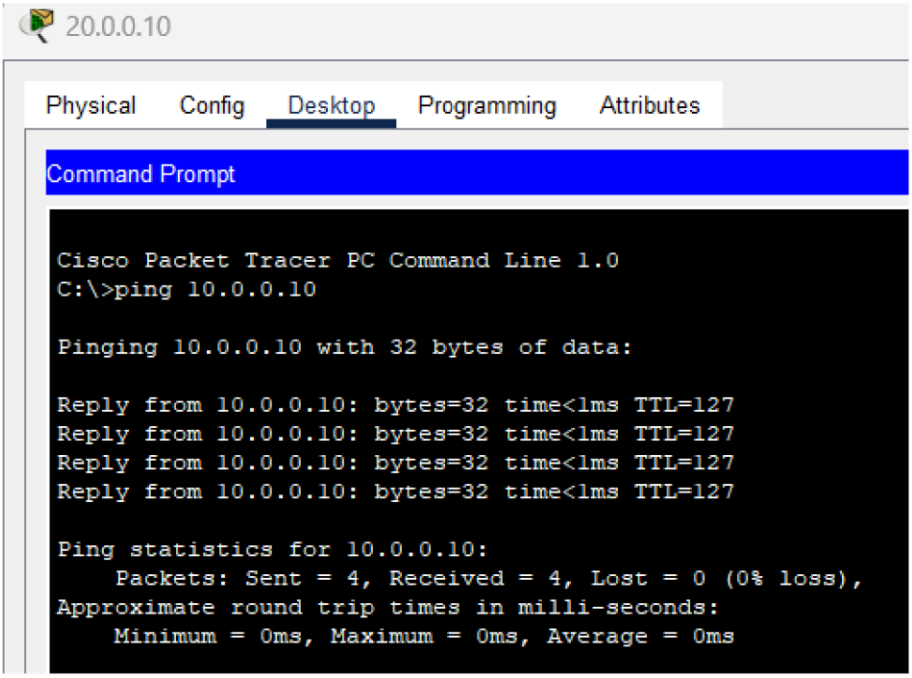
Router0

Physical Config CLI Attributes

IOS Command Line Interface

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


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

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

Router(config-if)#exit
Router(config)#interface fastethernet1/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up

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

Router#
```

```
C:\>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127

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

C:\>
```



PAGE NO :
DATE :

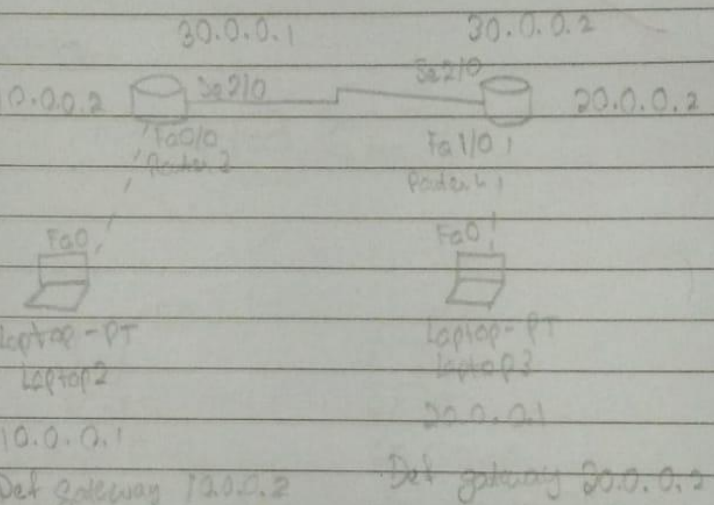
16-10-24

Lab - 3

Experiment - 3 → 2 Routers

Aim:- To configure ip address to routers in packet Tracer. Explore following messages: ping responses, destination unreachable, request timed out, reply

Topology:-



Procedure:

1. Add two PC's and two generic routers. Configure end devices: 10.0.0.1 and 20.0.0.1 and mention/define gateway 10.0.0.2 and 20.0.0.2.
2. Connect the PC's to the routers via copper cross over. Connect the routers to each other using Serial DCE.
3. Configure the routers:
   Click on Route → CLI
   - Configure to end devices similar to last experiment.

DATE :

4. → Connecting the routers:

Commands:

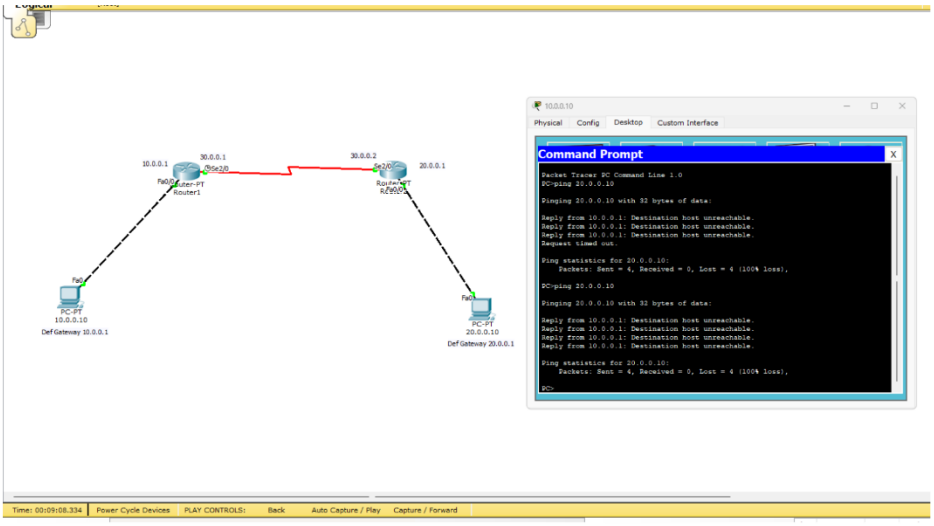
Router > enable
Router # Config terminal
Router (Config) # interface Serial 2/0
Router (Config) # ip address 30.0.0.1
Router (Config-if) # no shut

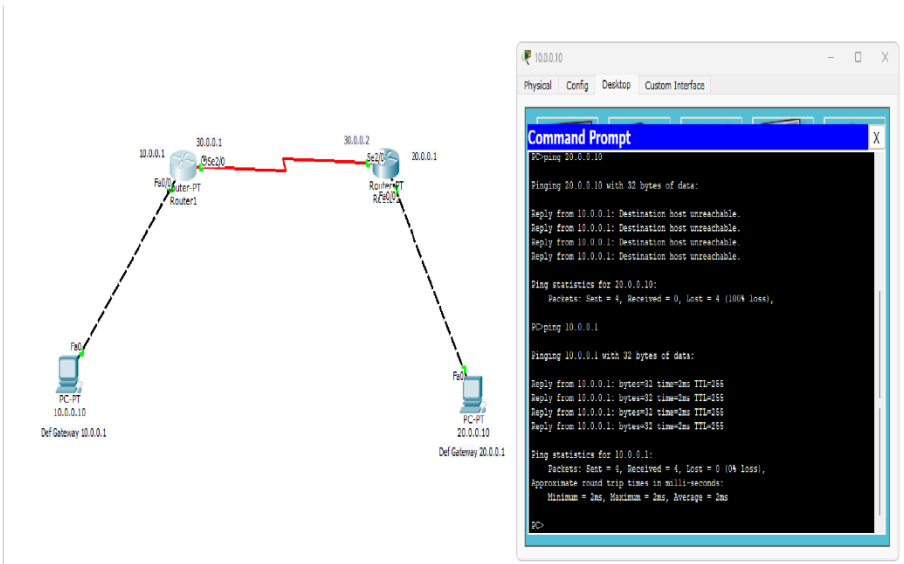
5. Give appropriate naming
6. Ping the device 20.0.0.1 from 10.0.0.1

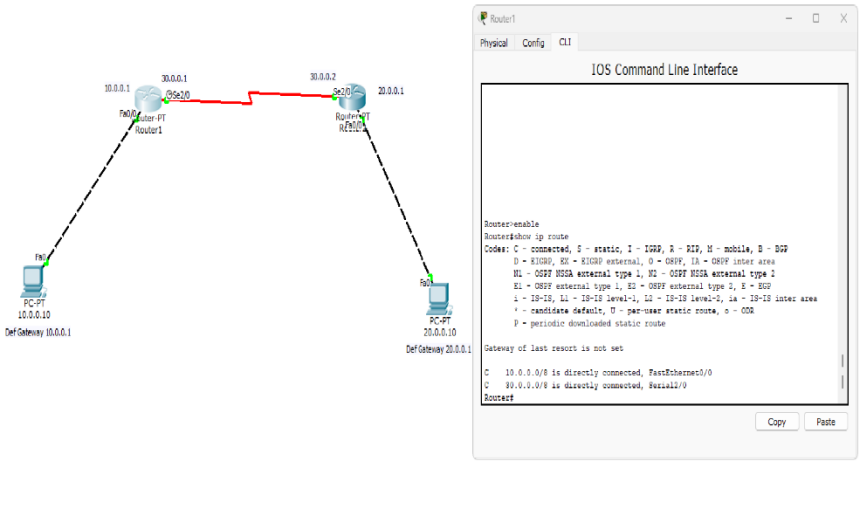
Observations:

1. The connections were done properly and green lights were displayed.
2. But when the device 10.0.0.1 pinged to 20.0.0.1 the output showed that the host was unreachable.
3. Since the networks are not direct neighbours / directly connected, the data sent was failed.
4. Pinging the same networks was a success since they can directly connected, Pinging 30.0.0.1 for 10.0.0.1 was a success.

Screenshots:

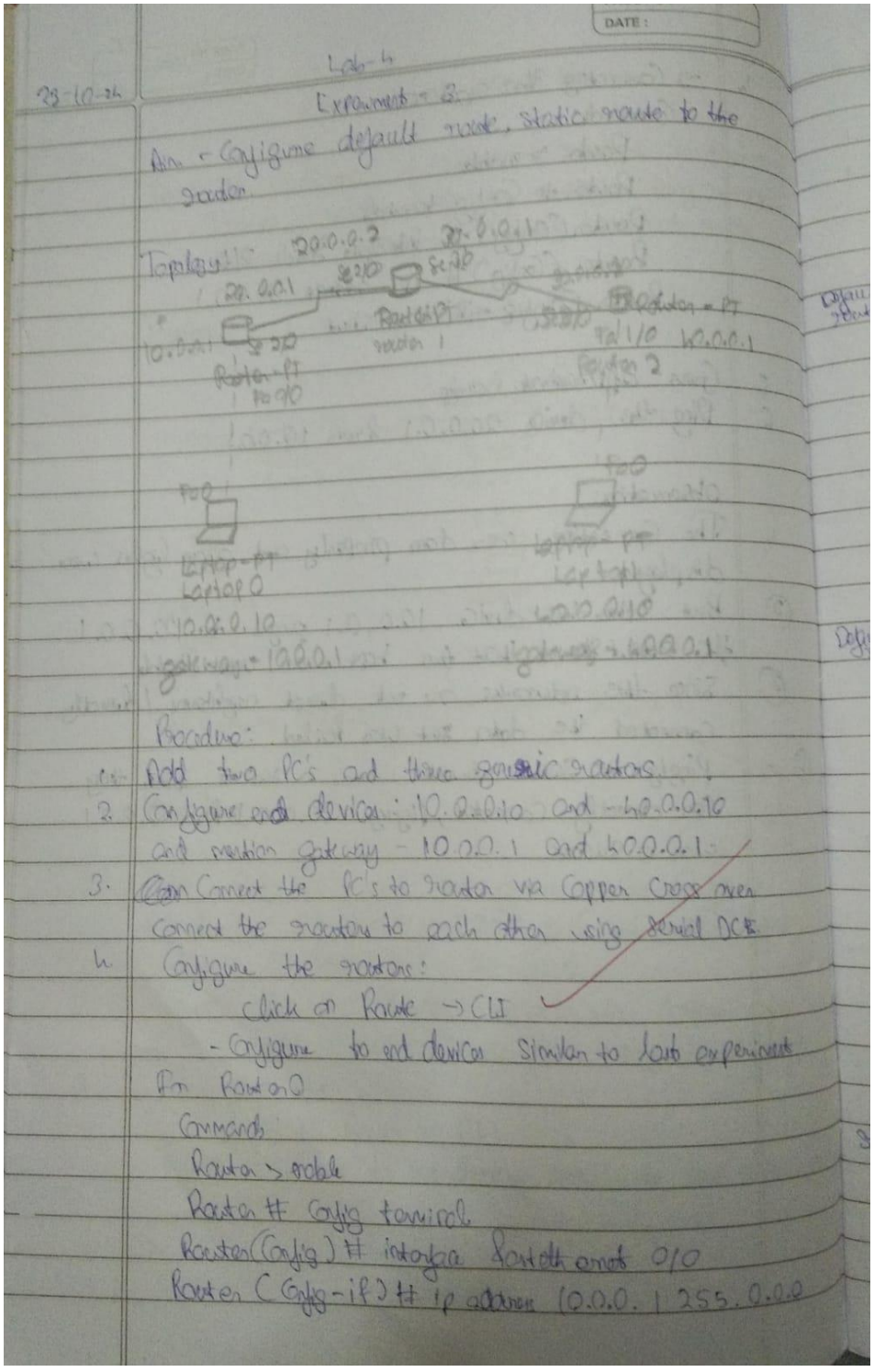





10.0.0.1
30.0.0.1
Se2/0
30.0.0.2
20.0.0.1
Router-PT
Router1
PC-PT
10.0.0.10
Def Gateway 10.0.0.1
PC-PT
20.0.0.10
Def Gateway 20.0.0.1
Router1
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 30.0.0.0/8 is directly connected, Serial2/0
Router#
Copy
Paste

LABORATORY PROGRAM – 3

Configure default route, static route to the Router.



PAGE NO : 7
DATE :

Router (config-if) # no shut
exit
Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
Router (config-if) # no shut

Default routing
Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router 2 →
Router (config) # interface serial 3/0
Router (config-if) # ip address 30.0.0.2 255.0.0.0
Router (config-if) # no shut
exit
Router > enable
Router # config terminal
Default routing
Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1

Router 1 →
Router > enable
Router # config terminal
Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shut

Router (config) # interface serial 3/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if) # no shut

Static routing
Router > enable
Router # 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

DATE :

Router # Config terminal
Router (Config) # ip route 100.0.0.0 255.0.0.0 20.0.0.1
Router (Config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router (Config) # ^Z

Observation:

* All Connections (fast ethernet and Serial) have turned green.
* IP route before Set up:
C 20.0.0.0/8 is directly Connected, Serial 2/0
C 30.0.0.0/8 is directly Connected, Serial 3/0

Show ip route after setup:

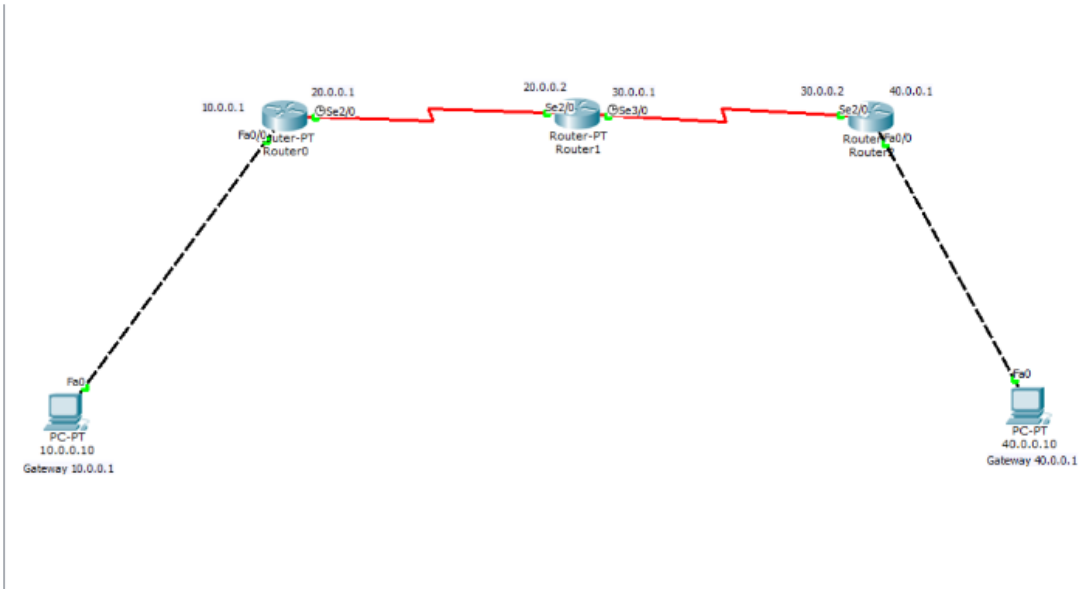
S 100.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial 2/0
C 30.0.0.0/8 is directly Connected, Serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

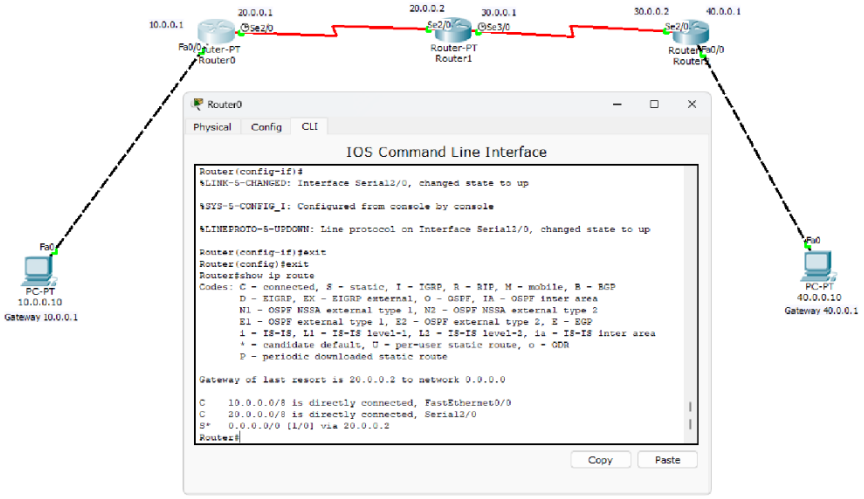
* Ping from one PC to another is Successful.
* So the middle router (Router 1) is setup with 2 next hops.
* Default Router: to transfer when no other route is available
* Static route: Define route with assigned destination.

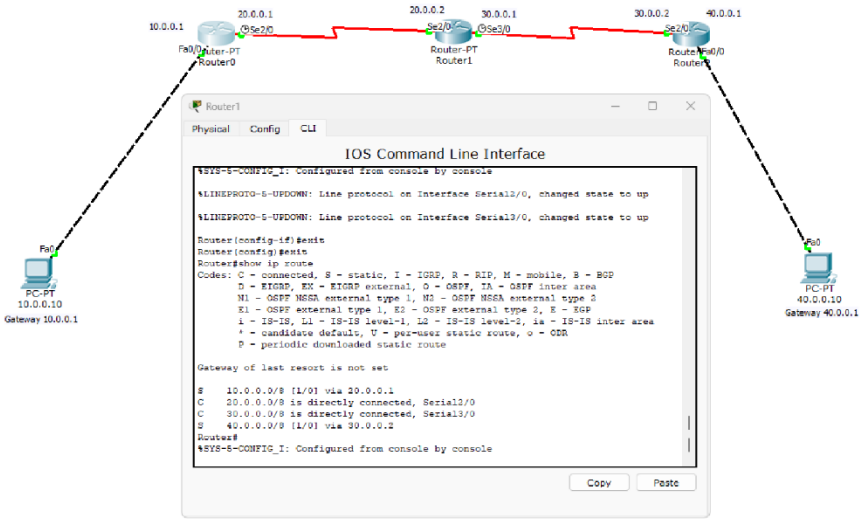
C:\> Ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10 ? bytes = 32 time = 7ms TTL = 125
Reply from 40.0.0.10 bytes = 32 time = 8ms TTL = 125
Reply from 40.0.0.10 bytes = 32 time = 6ms TTL = 125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss)

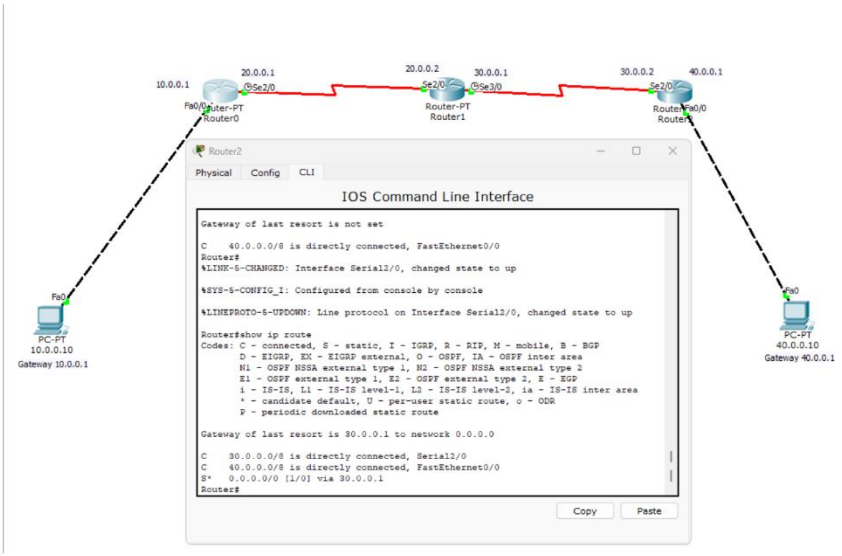
23/10

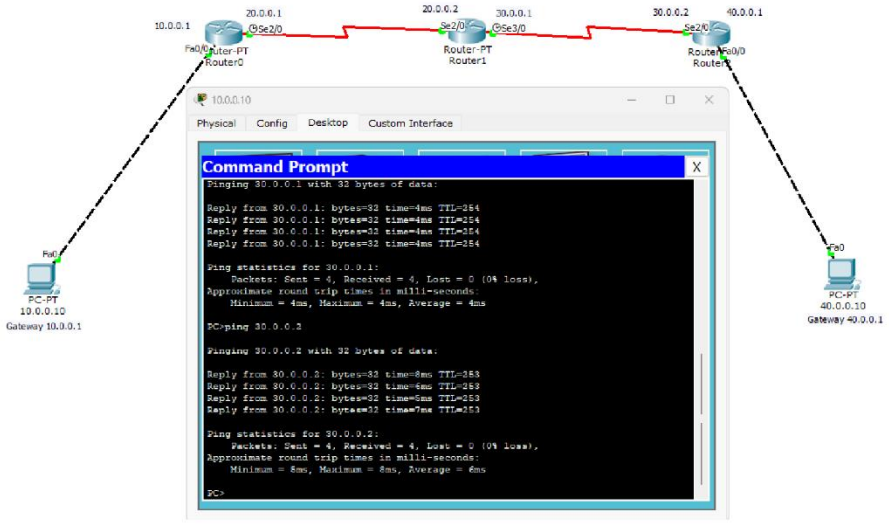
Screenshots:

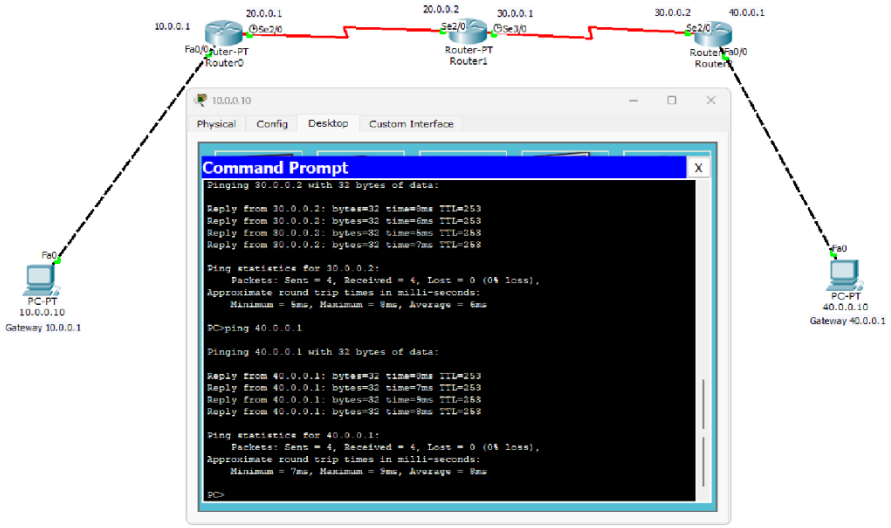


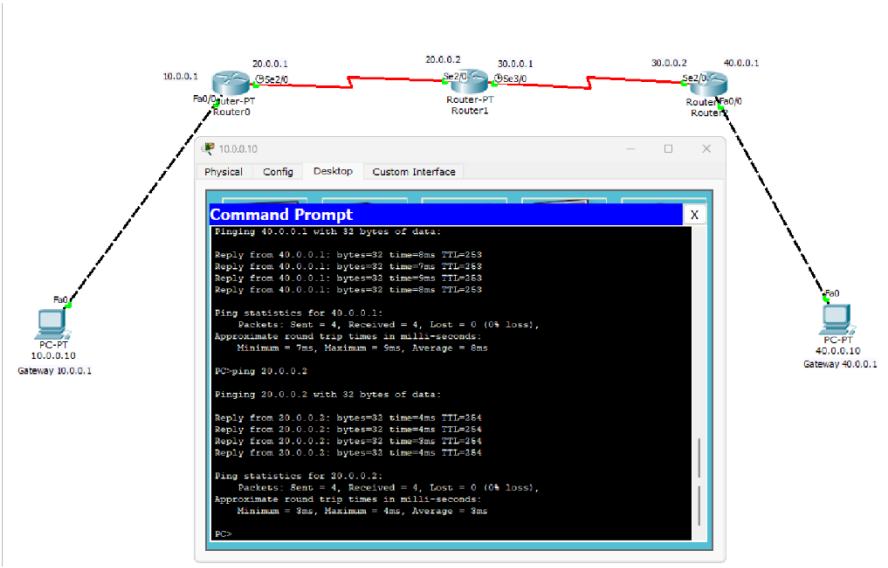

10.0.0.1
20.0.0.1
Se2/0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Router-PT
Router0
Router-PT
Router1
Router-PT
Router2
Router0
Physical
Config
CLI
IOS Command Line Interface
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%SYS-5-CONFIG_I: Configured from console by console
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#exit
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is 20.0.0.2 to network 0.0.0.0
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2
Router#
Copy
Paste
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
PC-PT
40.0.0.10
Gateway 40.0.0.1


10.0.0.1
20.0.0.1
Se2/0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Router-PT
Router0
Router-PT
Router1
Router-PT
Router2
Router1
Physical
Config
CLI
IOS Command Line Interface
%SYS-5-CONFIG_I: Configured from console by console
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
Router(config-if)#exit
Router(config)#exit
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
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
Router#
%SYS-5-CONFIG_I: Configured from console by console
Copy
Paste
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
PC-PT
40.0.0.10
Gateway 40.0.0.1


10.0.0.1
20.0.0.1
20.0.0.2
30.0.0.1
30.0.0.2
40.0.0.1
Router-PT
Router0
Router-PT
Router1
Router2
Router2
Physical
Config
CLI
IOS Command Line Interface
Gateway of last resort is not set
C 40.0.0.0/8 is directly connected, FastEthernet0/0
Router#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%SYS-5-CONFIG_I: Configured from console by console
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is 30.0.0.1 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
Router#
Copy
Paste
PC-PT
10.0.0.10
Gateway 10.0.0.1
PC-PT
40.0.0.10
Gateway 40.0.0.1


10.0.0.1
20.0.0.1
20.0.0.2
30.0.0.1
30.0.0.2
40.0.0.1
Router-PT
Router0
Router-PT
Router1
10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 4ms, Average = 4ms
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=8ms TTL=253
Reply from 30.0.0.2: bytes=32 time=6ms TTL=253
Reply from 30.0.0.2: bytes=32 time=6ms TTL=253
Reply from 30.0.0.2: bytes=32 time=7ms TTL=253
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 8ms, Average = 6ms
PC>
PC-PT
10.0.0.10
Gateway 10.0.0.1
PC-PT
40.0.0.10
Gateway 40.0.0.1


10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Router
10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=8ms TTL=253
Reply from 30.0.0.2: bytes=32 time=6ms TTL=253
Reply from 30.0.0.2: bytes=32 time=8ms TTL=253
Reply from 30.0.0.2: bytes=32 time=7ms TTL=253
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 8ms, Average = 6ms
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=7ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 8ms, Average = 8ms
PC>
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
Fa0
PC-PT
40.0.0.10
Gateway 40.0.0.1


10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Router
10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=7ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 8ms, Average = 8ms
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time=4ms TTL=254
Reply from 20.0.0.2: bytes=32 time=4ms TTL=254
Reply from 20.0.0.2: bytes=32 time=8ms TTL=254
Reply from 20.0.0.2: bytes=32 time=4ms TTL=254
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 8ms, Maximum = 4ms, Average = 8ms
PC>
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
Fa0
PC-PT
40.0.0.10
Gateway 40.0.0.1

LABORATORY PROGRAM – 4

Configure DHCP within a LAN and outside LAN

PAGE NO : 8
DATE :

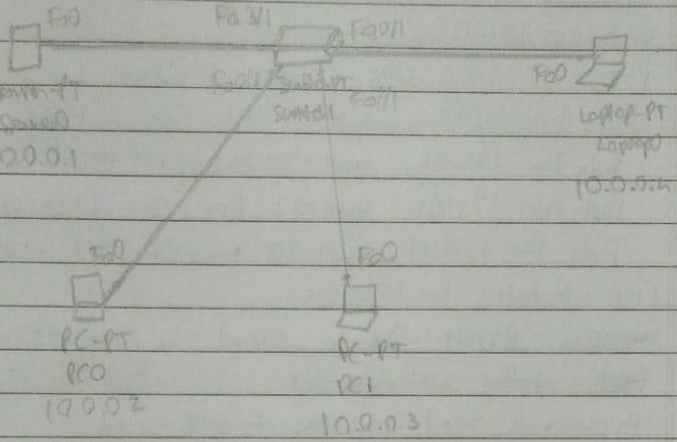
Lab-5

Experiment - 4

Aim - Configure DHCP within a LAN and outside LAN

1) DHCP within a LAN

Topology:



Procedure:

1. Add a generic switch, 2 PC, 1 laptop and One Server
2. Click on Server → Desktop → IP Configuration → It should be Static
3. Give IP address for Server 10.0.0.1 and default Gateway 10.0.0.0
4. Click Config → Services → Click DHCP → Service (ON) change pool name to Switchone (your choice).
5. Change maximum number =100, Start IP Address as 10.0.0.3 and give add
6. For each PC / Laptop → ~~Config~~ Desktop → IP Configuration → Give DHCP automatically generates IP address

Server ~~gives dynamically~~ allocates IP addresses to the devices. ~~those random connection~~

PAGE NO :
DATE :

Observation:

1. IP addresses were allocated dynamically
2. When we ping from one PC to another data was sent successfully.

,10.0.0.2

o/p PC > ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.4:

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

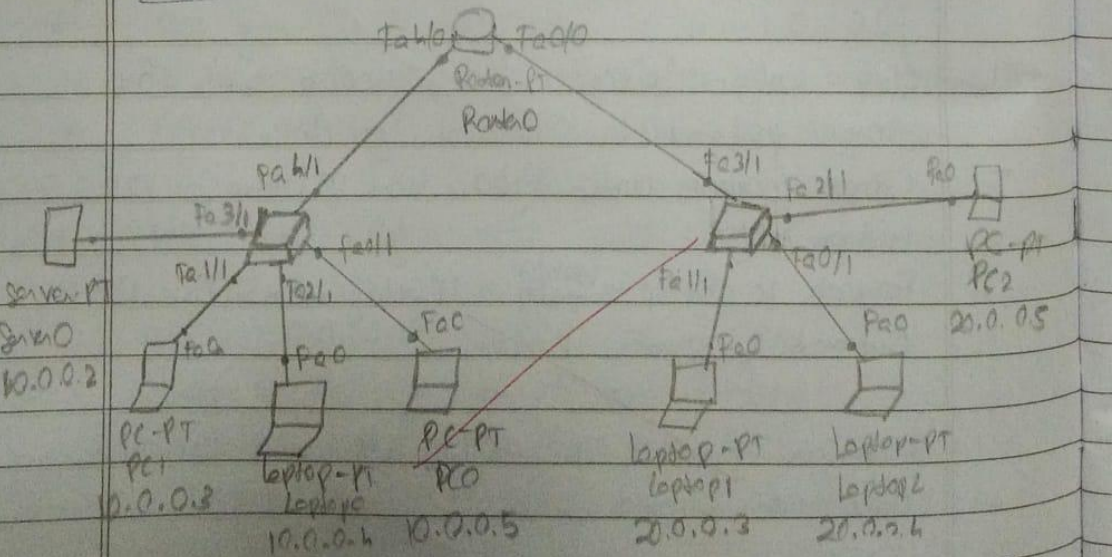
1. ~~Add another serial link that.~~

Approximate round trip times in milli-seconds:

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

2) DHCP outside LAN

Topology:



PAGE NO : 9
DATE :

Procedure:-

1. Add to the same network on another similar networks and a router which connects both the switches
2. Go to Server - IP Configuration → change IP address 10.0.0.2 default Gateway - 10.0.0.1
3. Goto Config → services - DHCP - select Switch 0 etc. change default gateway to 10.0.0.1
4. Do for Switch then → default Gateway as 20.0.0.1, start ip address as 20.0.0.3 and add it.

Router Configuration - CLI

no

Router > enable

Router # config terminal

Router (config) # interface fastethernet 4/0

Router (config-if) # ip address 10.0.0.1 255.0.0.0

Router (config-if) # ip helper-address 10.0.0.2 (to give the other network with ip-address to start with 20.0.0 only)

no shut

exit

For second network Switch

config terminal

interface fastethernet 0/0, ip address 20.0.0.1 255.0.0.0

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

no shut

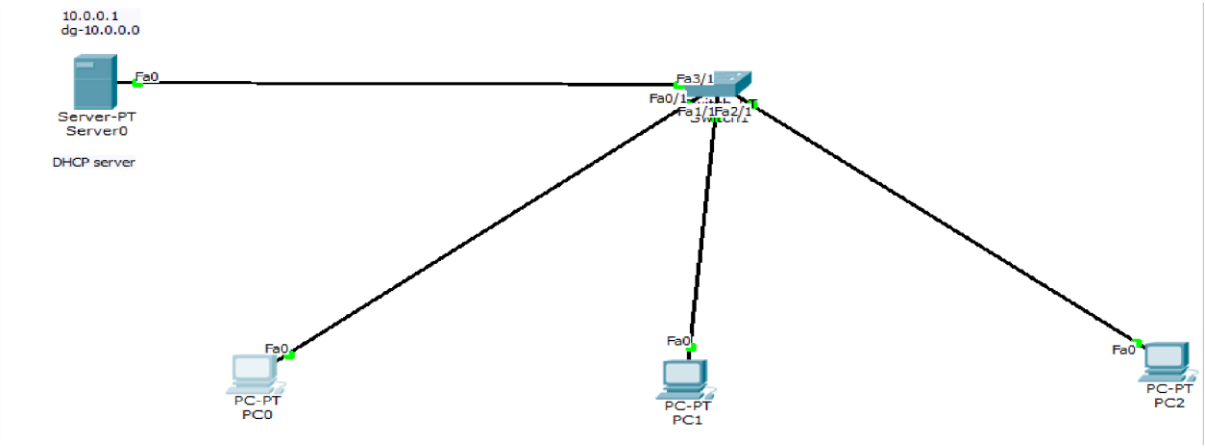
exit

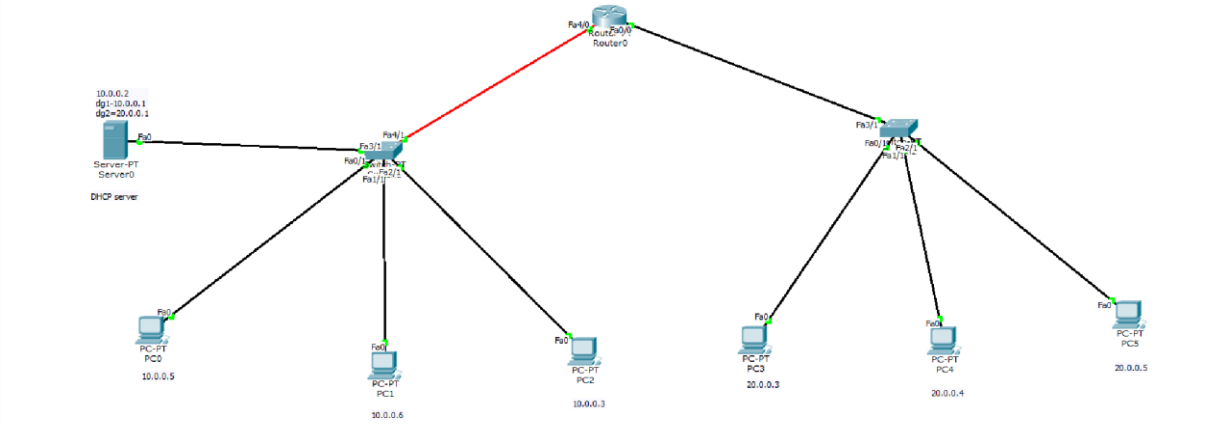
Now goto each PC change to Static and DHCP so the ip address will be allocated dynamically.

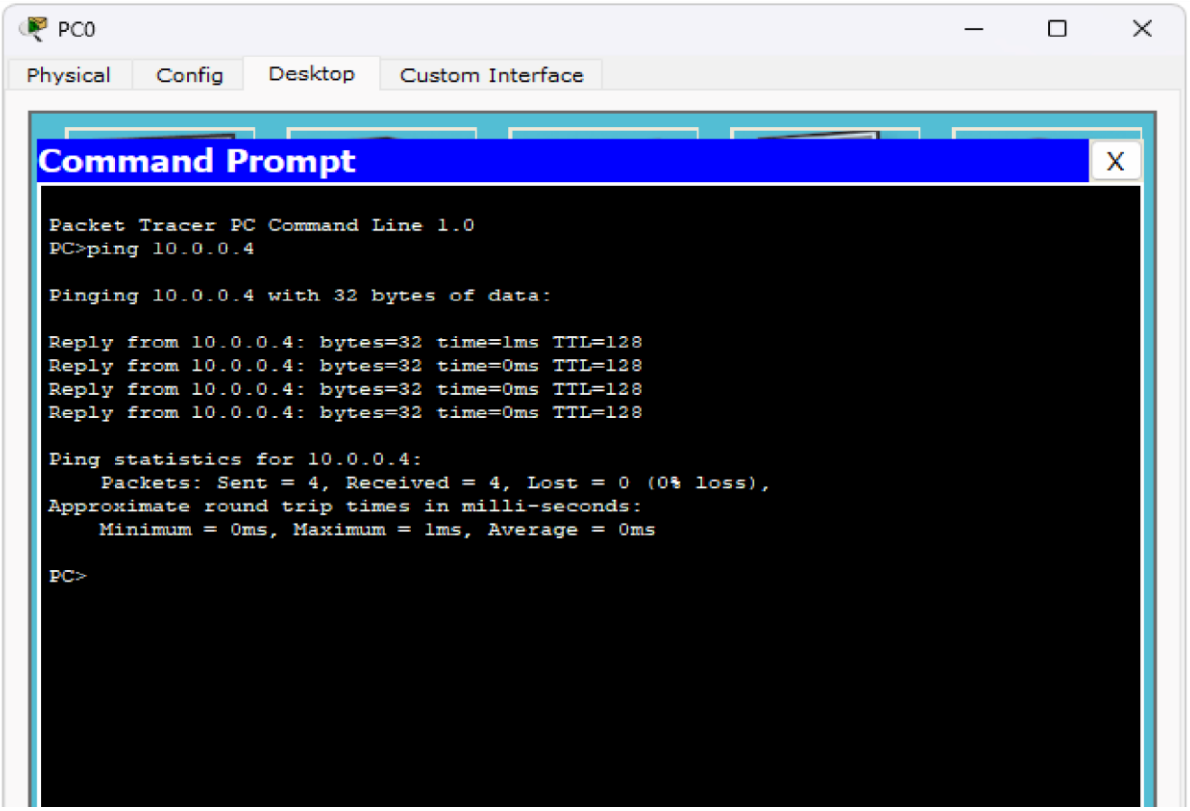
Observation:- 1. IP address are dynamically allocated dynamically

2. ~~Data can't be sent successfully among PC's when pinged.~~

13/11/24

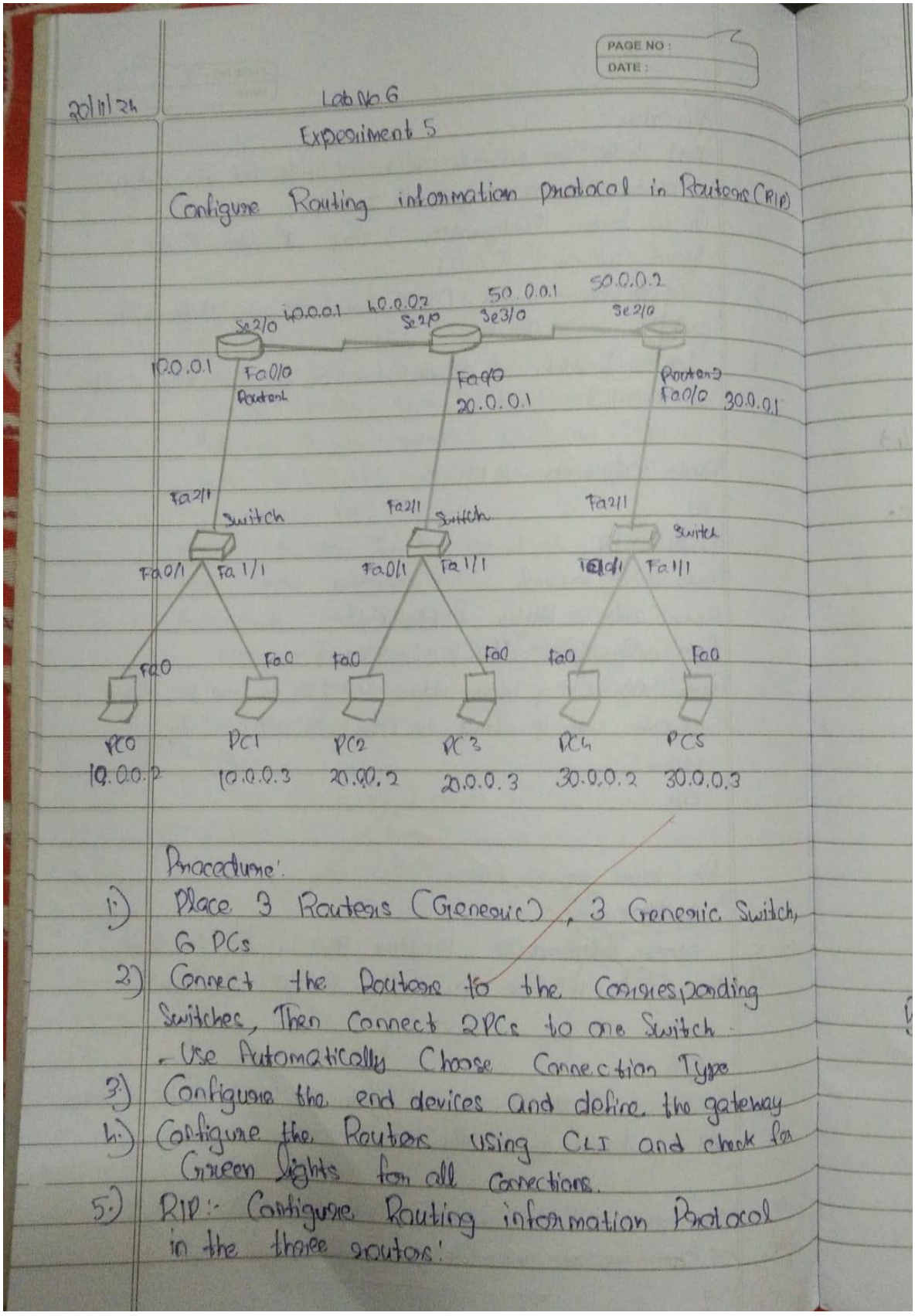

10.0.0.1
dg-10.0.0.0
Fa0
Server-PT
Server0
DHCP server
Fa3/1
Fa0/1
Fa1/1
Fa2/1
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2


Router0
10.0.0.2
dg1-10.0.0.1
dg2-20.0.0.1
Server-PT
Server0
DHCP server
PC-PT
PC0
10.0.0.5
PC-PT
PC1
10.0.0.6
PC-PT
PC2
10.0.0.3
PC-PT
PC3
20.0.0.3
PC-PT
PC4
20.0.0.4
PC-PT
PC5
20.0.0.5


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms 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 = 1ms, Average = 0ms
PC>

LABORATORY PROGRAM – 5

Configure RIP routing Protocol in Routers



PAGE NO : 19
DATE :

In Router 0:

```
Router > enable
Router # config terminal
Router (config) # router rip
Router (config) #
Router (config-router) # network 10.0.0.0
Router (config-router) # network 40.0.0.0
```

In Router 1:

```
Router > enable
Router # config terminal
Router (config) # router rip
Router (config-router) # network 40.0.0.0
Router (config-router) # network 50.0.0.0
Router (config-router) # network 20.0.0.0
```

In Router 2:

```
Router > enable
Router # config terminal
Router (config) # router rip
Router (config-router) # network 50.0.0.0
Router (config-router) # network 30.0.0.0
```

Observations:

1) Before Routing information Protocol:

In Router 2:

```
Router # show ip route
C   30.0.0.0/8 is directly connected
C   50.0.0.0/8 is directly connected
```

PAGE NO :
DATE :

2.) After Routing information protocol:

In Route 2:

Router # show ip route

R 10.0.0.0/8 via 50.0.0.1

R 20.0.0.0/8 via 50.0.0.1

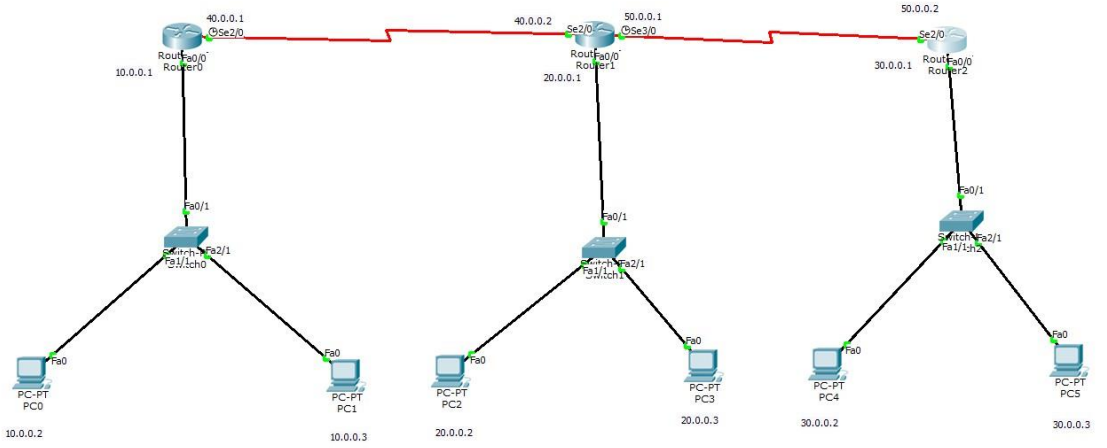
C 30.0.0.0/8 ~~via~~ 50.0 directly Connected

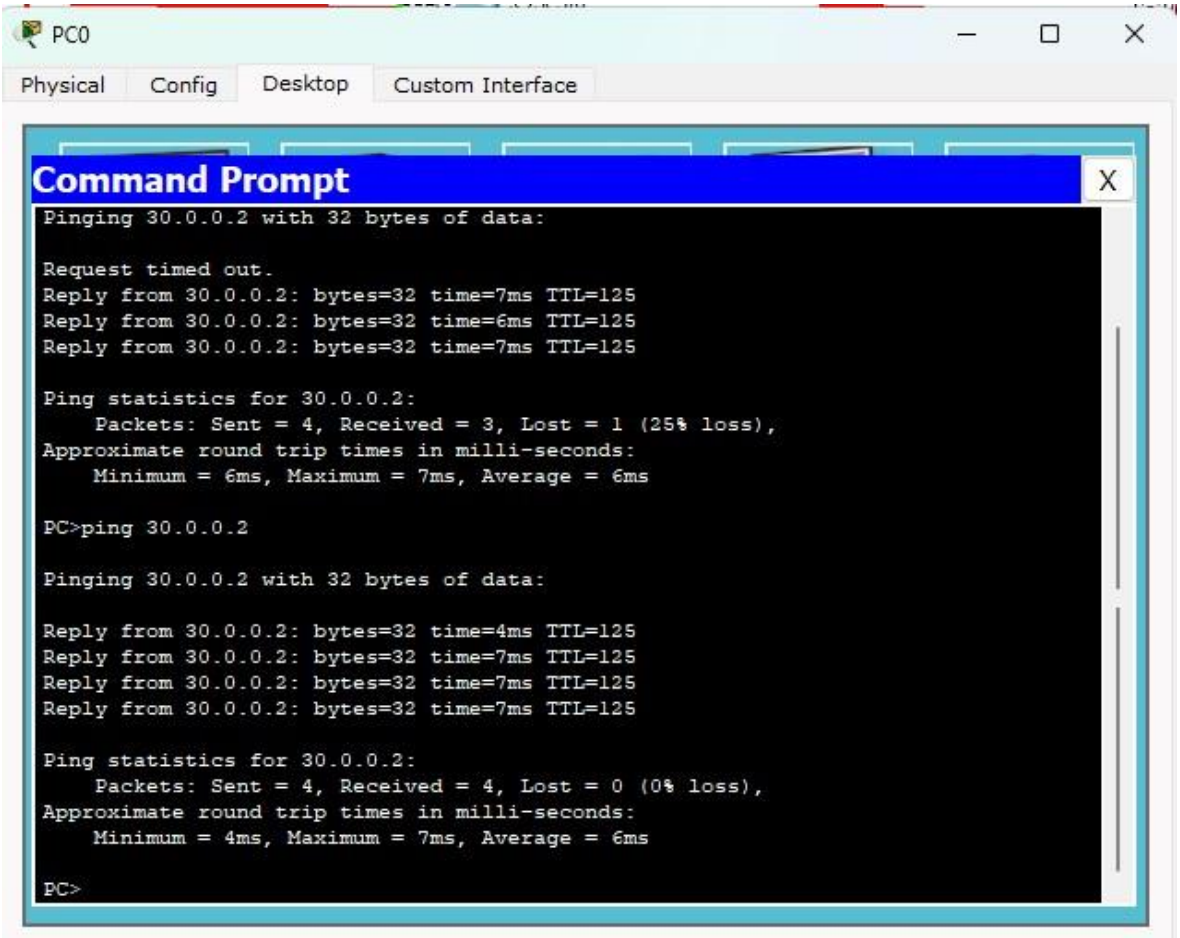
R 40.0.0.0/8 via 50.0.0.1

C 50.0.0.0/8 directly Connected

3.) Before RIP pinging from different network failed.

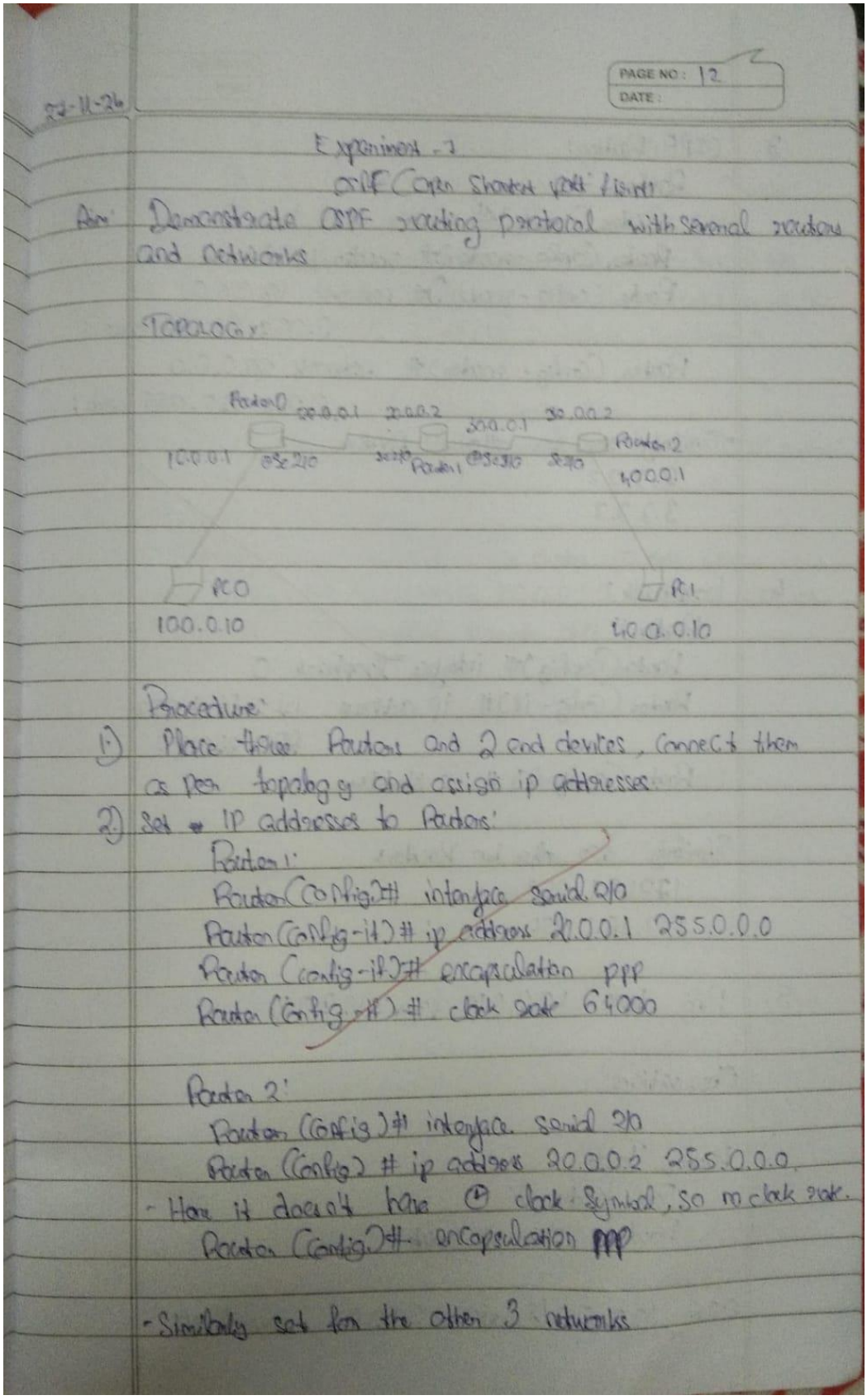
4.) After RIP, we were successfully able to ping across networks as Connections were established through rip


40.0.0.1
Se2/0
40.0.0.2
Se2/0
50.0.0.1
Se3/0
50.0.0.2
Se2/0
Router0
Router1
Router2
Fa0/0
10.0.0.1
20.0.0.1
30.0.0.1
Fa0/1
Switch0
Switch1
Switch2
Fa1/1
Fa2/1
Fa0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
10.0.0.2
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 30.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=6ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 7ms, Average = 6ms
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=4ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 7ms, Average = 6ms
PC>

LABORATORY PROGRAM – 6

Configure OSPF routing protocol



PAGE NO : 12
DATE :

22-11-24

Experiment - 7

OSPF (Open Shortest Path First)

Aim: Demonstrate OSPF routing protocol with several routers and networks

Topology:

Procedure:

1) Place three Routers and 2 end devices, connect them as per topology and assign ip addresses.

2) Set IP addresses to Routers:

Router 1:

Router(Config)# interface serial 2/0
Router(Config-if)# ip address 20.0.0.1 255.0.0.0
Router(Config-if)# encapsulation ppp
Router(Config-if)# clock rate 64000

Router 2:

Router(Config)# interface serial 2/0
Router(Config)# ip address 20.0.0.2 255.0.0.0
- Here it doesn't have (DCE) clock symbol, so no clock rate.
Router(Config)# encapsulation ppp

- Similarly set for the other 3 networks

PAGE NO :
DATE :

3. OSPF Routing:

Router 1:

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

- Similarly set for other 2 Routers

2.2.2.2

3.3.3.3

4. Loopback:

Router 1:

Router (config)# interface loopback 0

Router (config-if)# ip address 172.16.1.252 255.255.0.0

Router (config-if)# no shutdown

Similarly for other two Routers:

172.16.1.253

172.16.1.254

5. Ping device 40.0.0.10 from 10.0.0.10

Observations:

1) Routing Table:

- For Router 3:

Router# show ip route

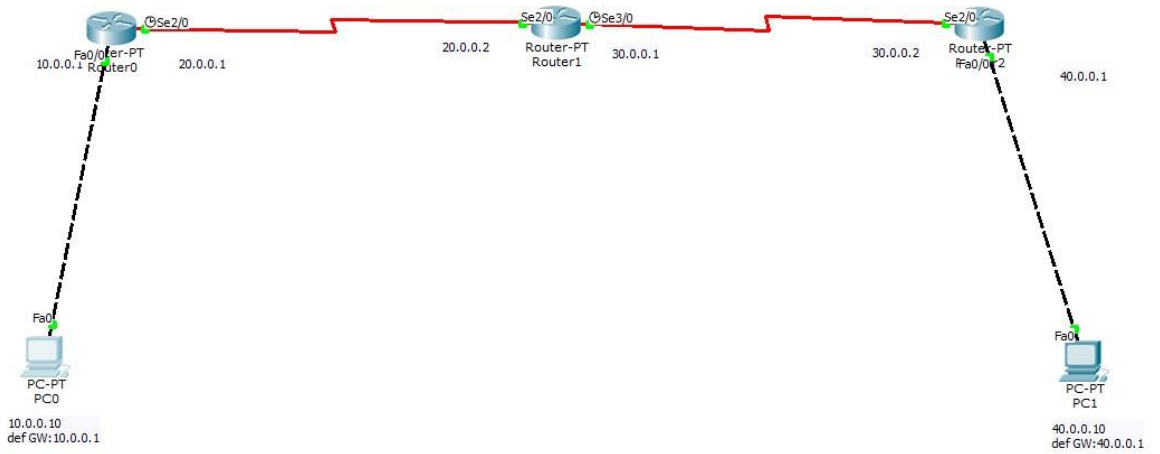
OIA 10.0.0.0/8 via 30.0.0.1, serial 2/0

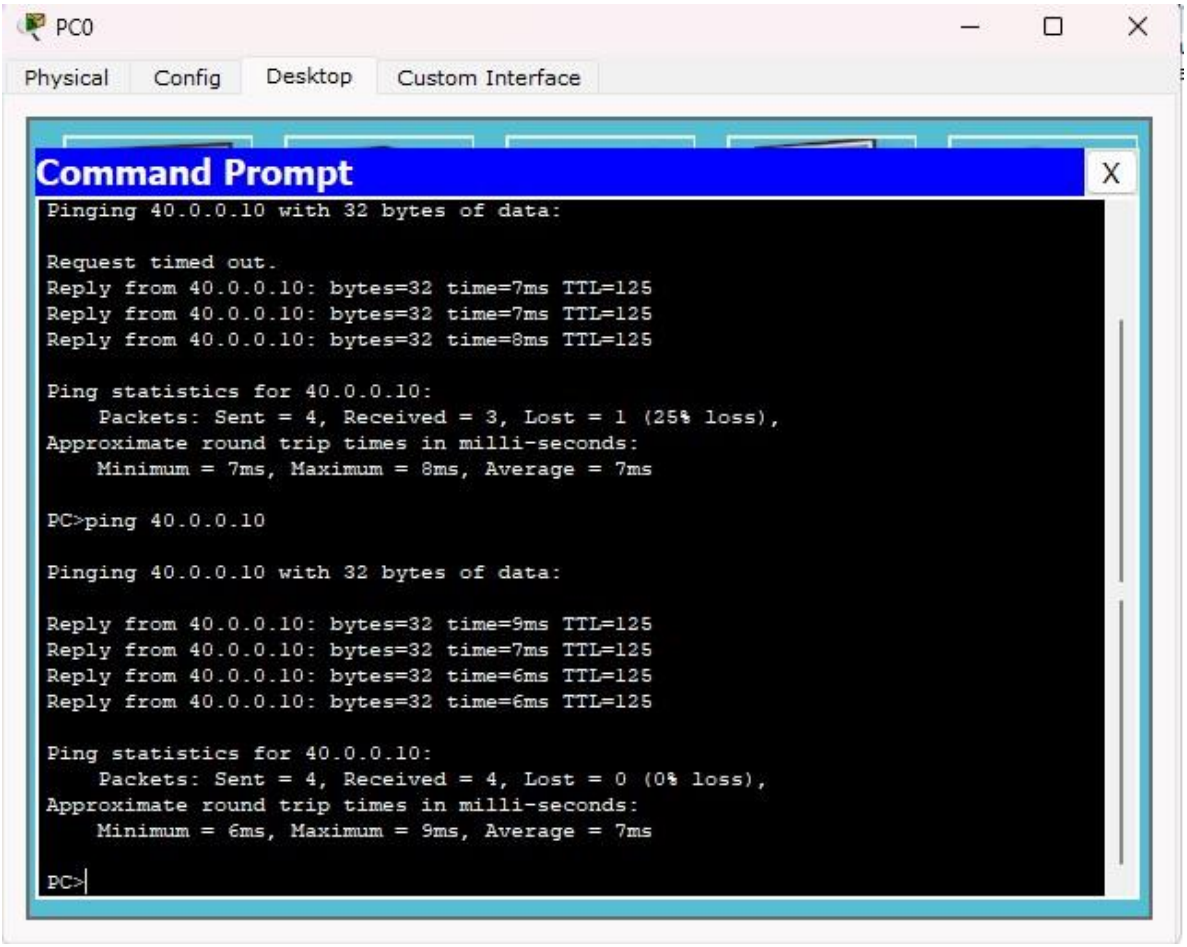
PAGE NO : 13
DATE :

O IA 20.0.0.0/8 via 30.0.0.1, serial 2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial 2/0
C 30.0.0.1/32 is directly connected, Serial 2/0
C 40.0.0.0/8 is directly connected, Fast Ethernet 0/0
C 172.16.0.0/16 is directly connected, Loop back 0

- All the routers know about each other.
- This connection successfully established through OSPF protocol

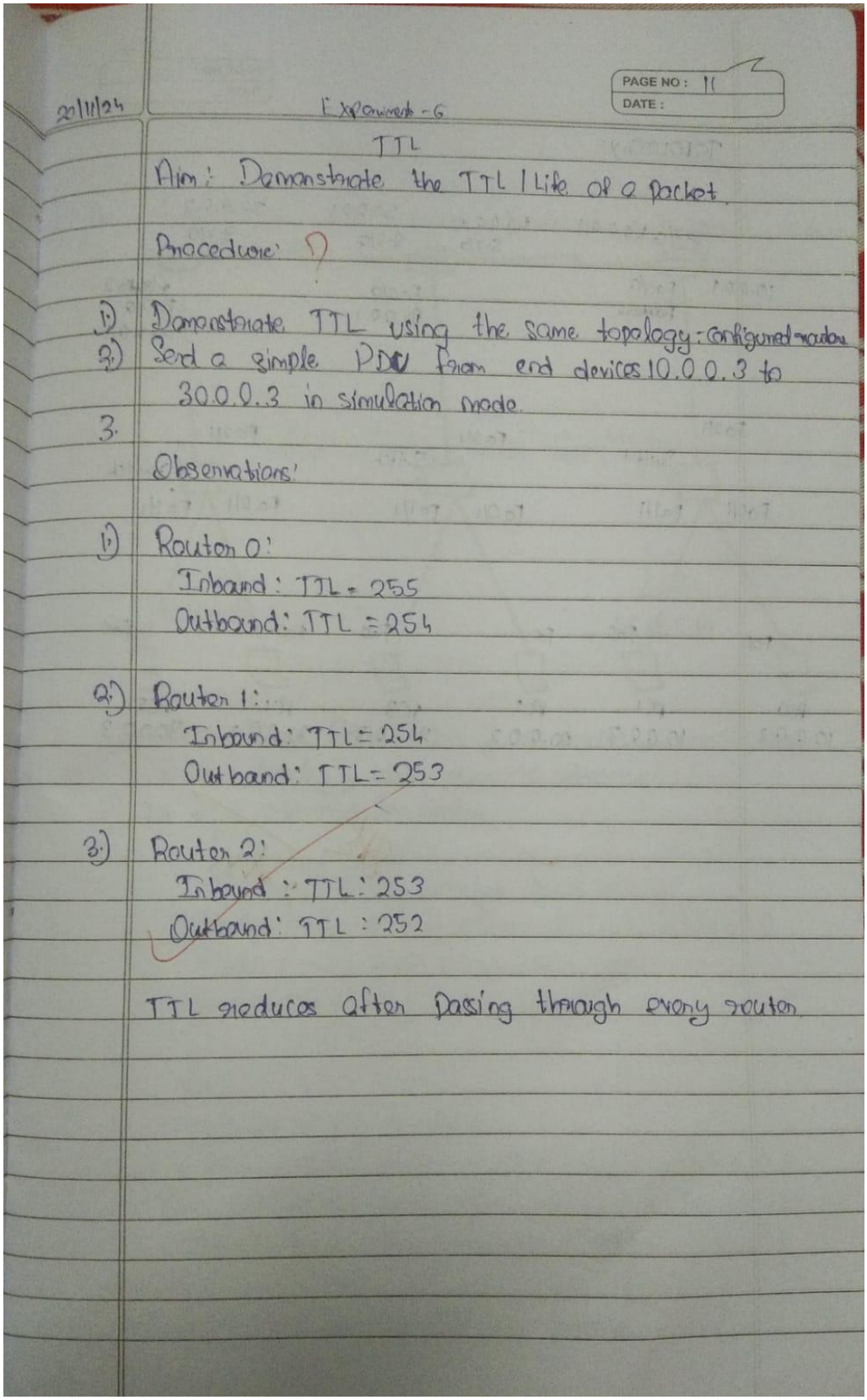
2) Pinging:
- Pinging was successful : The data packets were successfully sent from device 10.0.0.10 to 40.0.0.10 as connection was established through OSPF protocol.


Se2/0
Fa0/0
Router0
10.0.0.1
20.0.0.1
Se2/0
20.0.0.2
Router-PT
Router1
Se3/0
30.0.0.1
Se2/0
30.0.0.2
Router-PT
Fa0/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
def GW:10.0.0.1
Fa0
PC-PT
PC1
40.0.0.10
def GW:40.0.0.1


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 8ms, Average = 7ms
PC>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes=32 time=9ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms 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 = 6ms, Maximum = 9ms, Average = 7ms
PC>

LABORATORY PROGRAM – 7

Demonstrate the TTL/ Life of a Packet



20/11/24 Experiment - 6

PAGE NO : 11
DATE :

TTL

Aim: Demonstrate the TTL / Life of a packet.

Procedure:

1) Demonstrate TTL using the same topology: configured routers
2) Send a simple PDU from end devices 10.0.0.3 to 30.0.0.3 in simulation mode.
3.

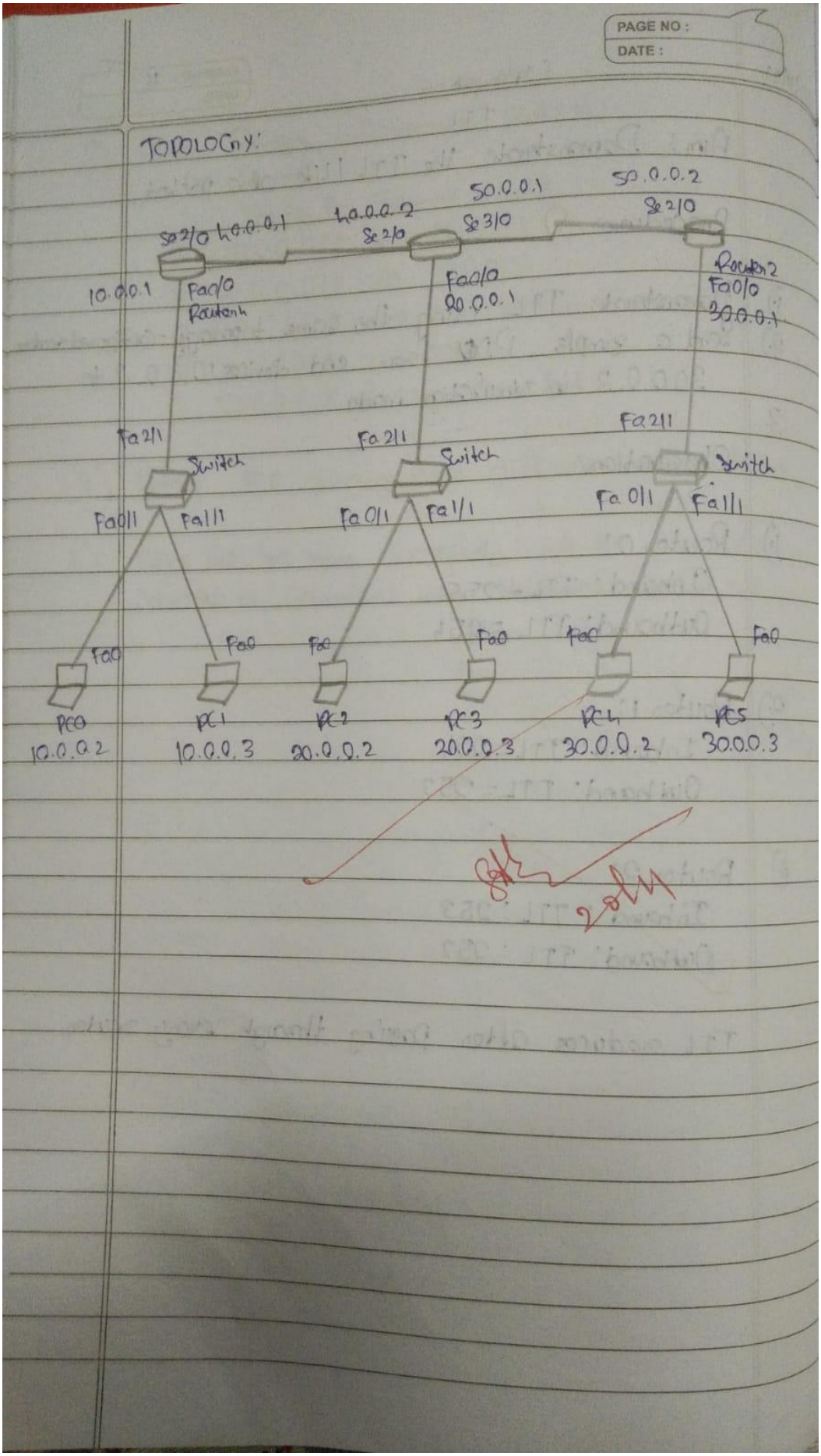
Observations:

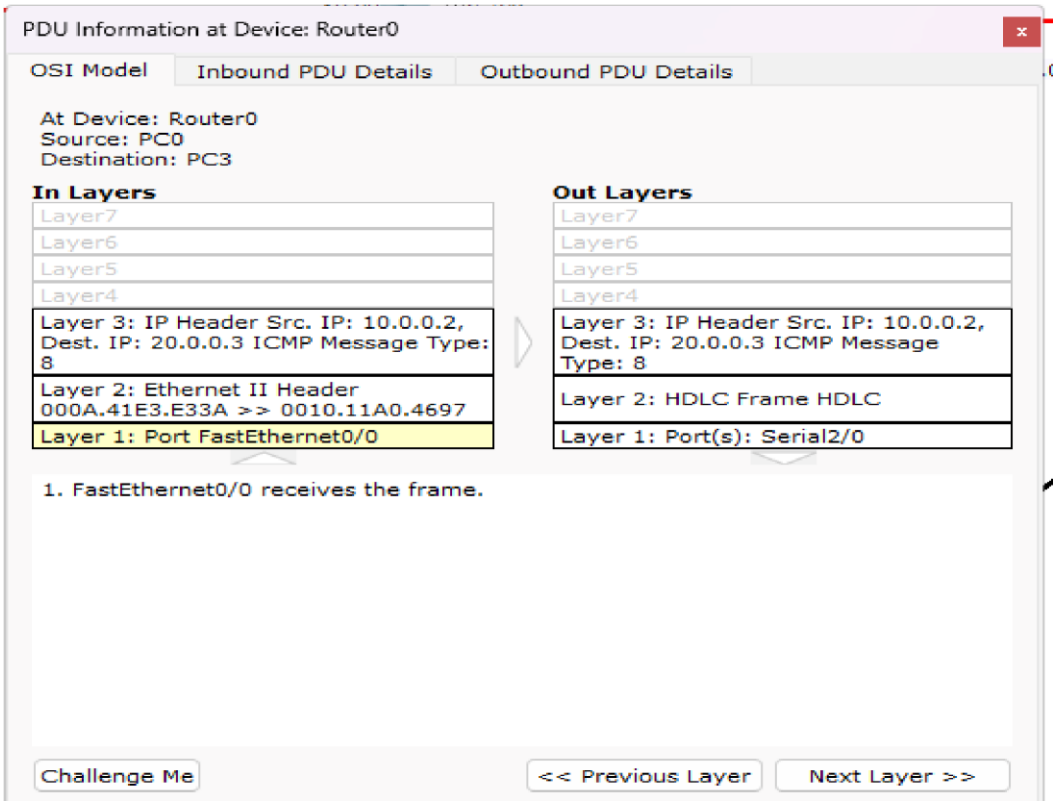
1) Router 0:
   Inbound: TTL = 255
   Outbound: TTL = 254

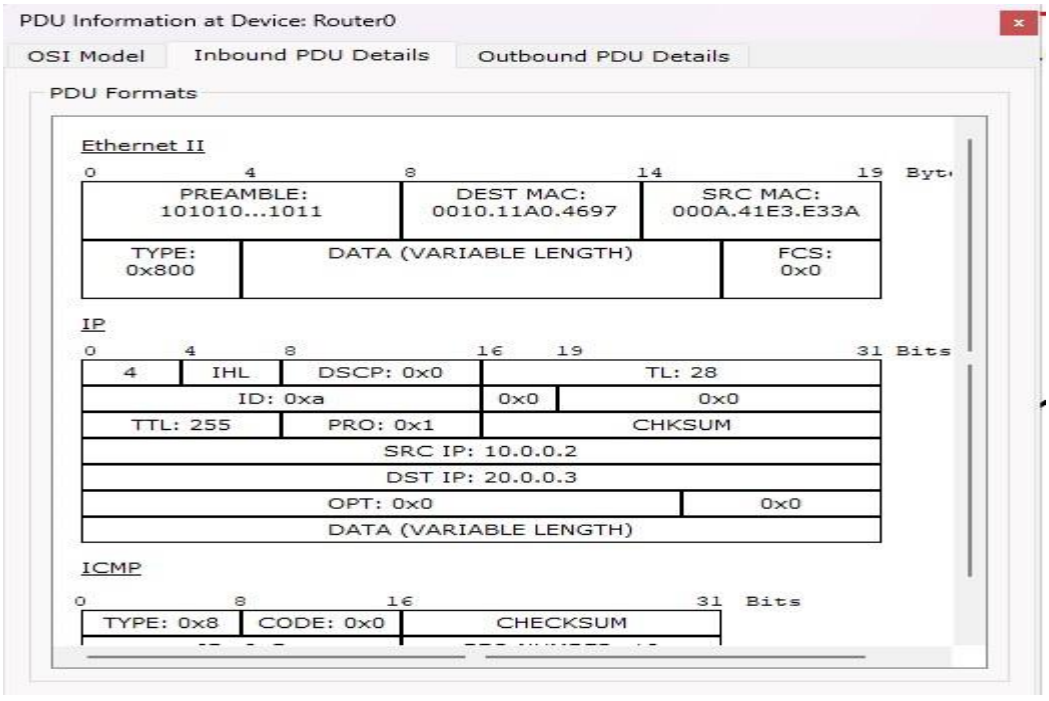
2) Router 1:
   Inbound: TTL = 254
   Outband: TTL = 253

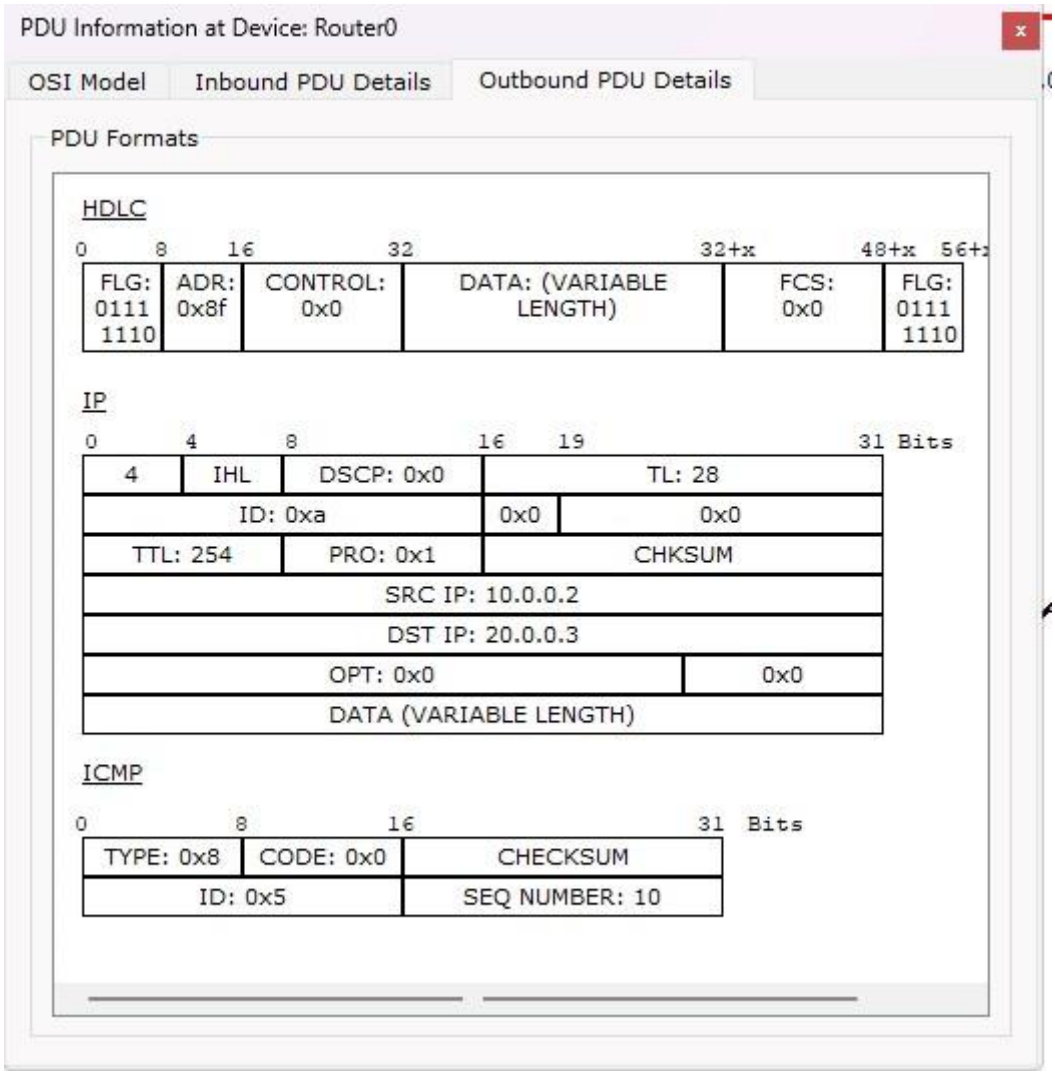
3) Router 2:
   Inband: TTL: 253
   Outband: TTL: 252

TTL reduces after passing through every router.


PAGE NO :
DATE :
TOPOLOGY:
Se2/0 10.0.0.1
10.0.0.2 Se2/0
50.0.0.1 Se3/0
50.0.0.2 Se2/0
10.0.0.1
Fa0/0
Router0
Fa0/0
20.0.0.1
Router2
Fa0/0
30.0.0.1
Fa2/1
Switch
Fa2/1
Switch
Fa2/1
Switch
Fa0/1
Fa1/1
Fa0/1
Fa1/1
Fa0/1
Fa1/1
Fa0
Fa0
Fa0
Fa0
Fa0
Fa0
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

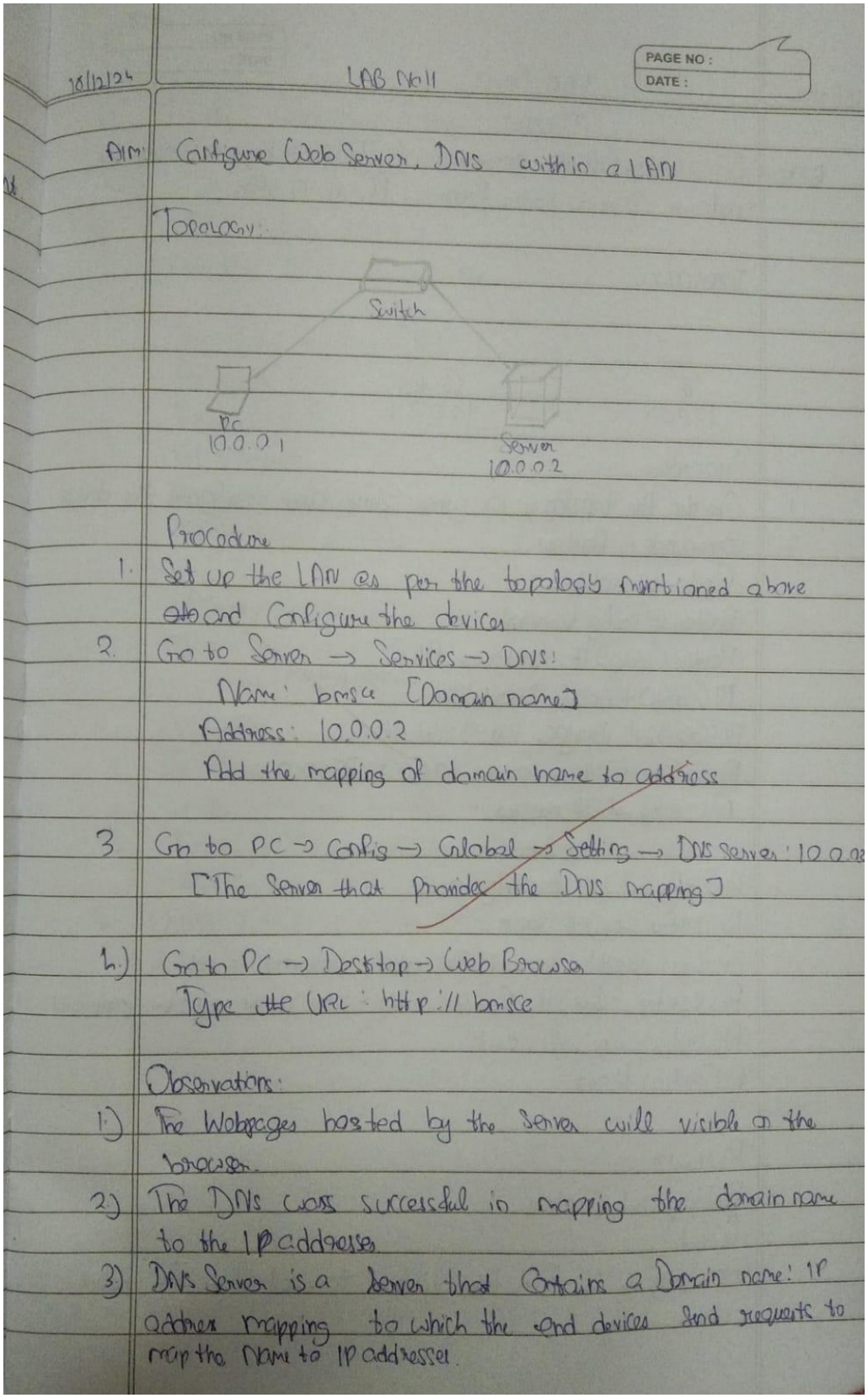

PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Router0
Source: PC0
Destination: PC3
In Layers
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: Ethernet II Header 000A.41E3.E33A >> 0010.11A0.4697
Layer 1: Port FastEthernet0/0
Out Layers
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: HDLC Frame HDLC
Layer 1: Port(s): Serial2/0
1. FastEthernet0/0 receives the frame.
Challenge Me
<< Previous Layer
Next Layer >>


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0010.11A0.4697
SRC MAC: 000A.41E3.E33A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x5
SEQ NUMBER: 10

LABORATORY PROGRAM – 8

Configure Web Server, DNS within a LAN.



18/12/24 LAB No 11

PAGE NO :
DATE :

Aim: Configure Web Server, DNS within a LAN

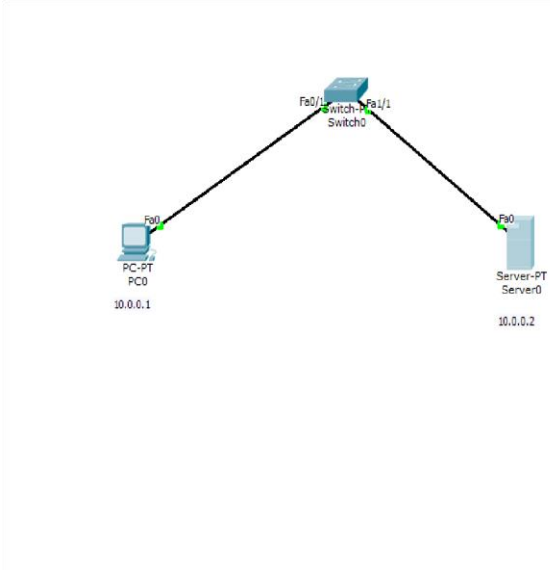
Topology:

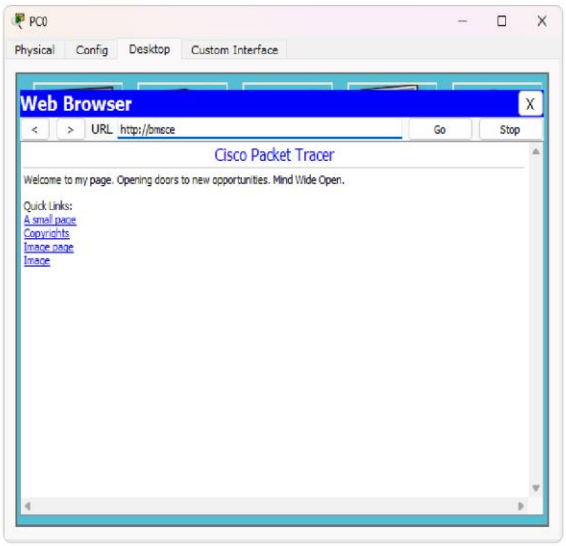
Procedure

1. Set up the LAN as per the topology mentioned above and Configure the devices
2. Go to Server → Services → DNS:
   Name: bmsce [Domain name]
   Address: 10.0.0.2
   Add the mapping of domain name to address
3. Go to PC → Config → Global → Setting → DNS Server: 10.0.0.2
   [The Server that provides the DNS mapping]
4. Go to PC → Desktop → Web Browser
   Type the URL: http://bmsce

Observations:

1) The Webpages hosted by the Server will visible on the browser.
2) The DNS was successful in mapping the domain name to the IP addresses
3) DNS Server is a Server that Contains a Domain name: IP address mapping to which the end devices send requests to map the Name to IP addresses.


Fa0/1
Switch-PT
Switch0
Fa1/1
Fa0
PC-PT
PC0
10.0.0.1
Fa0
Server-PT
Server0
10.0.0.2


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL
http://bmsce
Go
Stop
Cisco Packet Tracer
Welcome to my page. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image

LABORATORY PROGRAM – 9

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

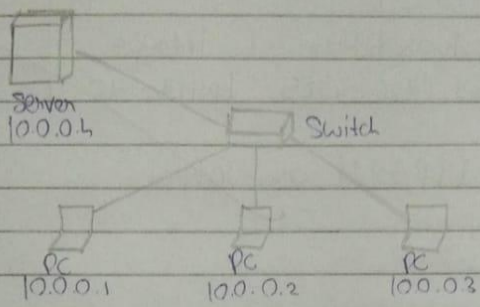
18/12/24 LAB NO 10

PAGE NO :
DATE :

ARP

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

Topology:



Procedure:

1. Create the topology as shown above
2. Configure the PC's and the Server
3. Click on Inspect mode (9), then click on the end devices and open ARP tables
4. Send a data packet from any end device say Server to other end devices say 10.0.0.3 PC
5. Open Simulation mode to capture each step of data transfer

Observations

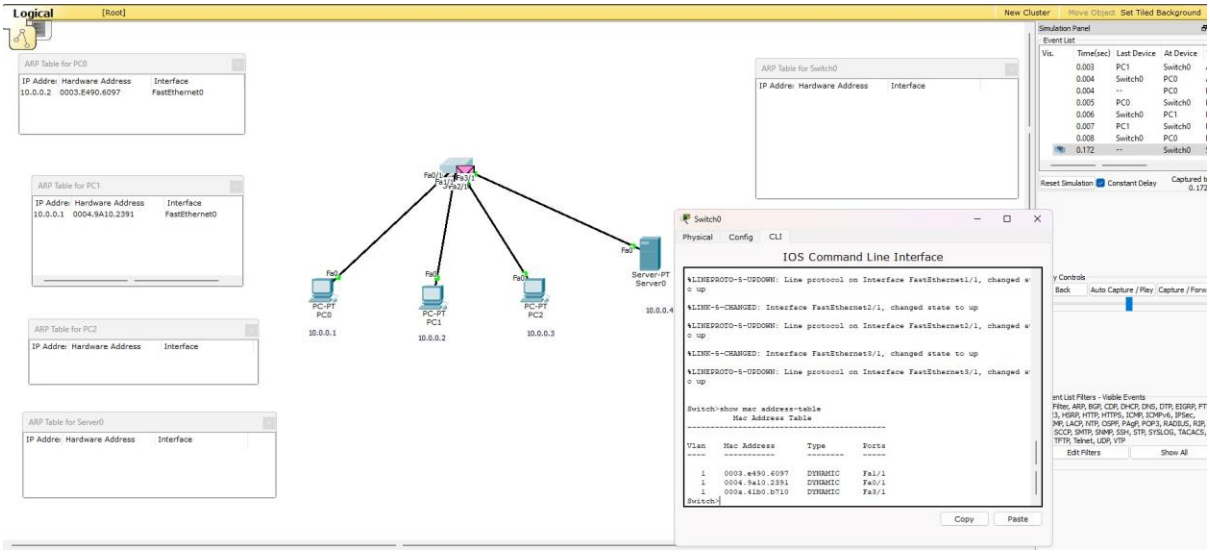
1. The ARP tables of the all end devices are initially empty.
2. When the data packet from device arrives at the Switch, since the source MAC address is unknown, it sends a broadcast message to all devices
3. The device with the IP address present in the destination address of the data packet responds to the message

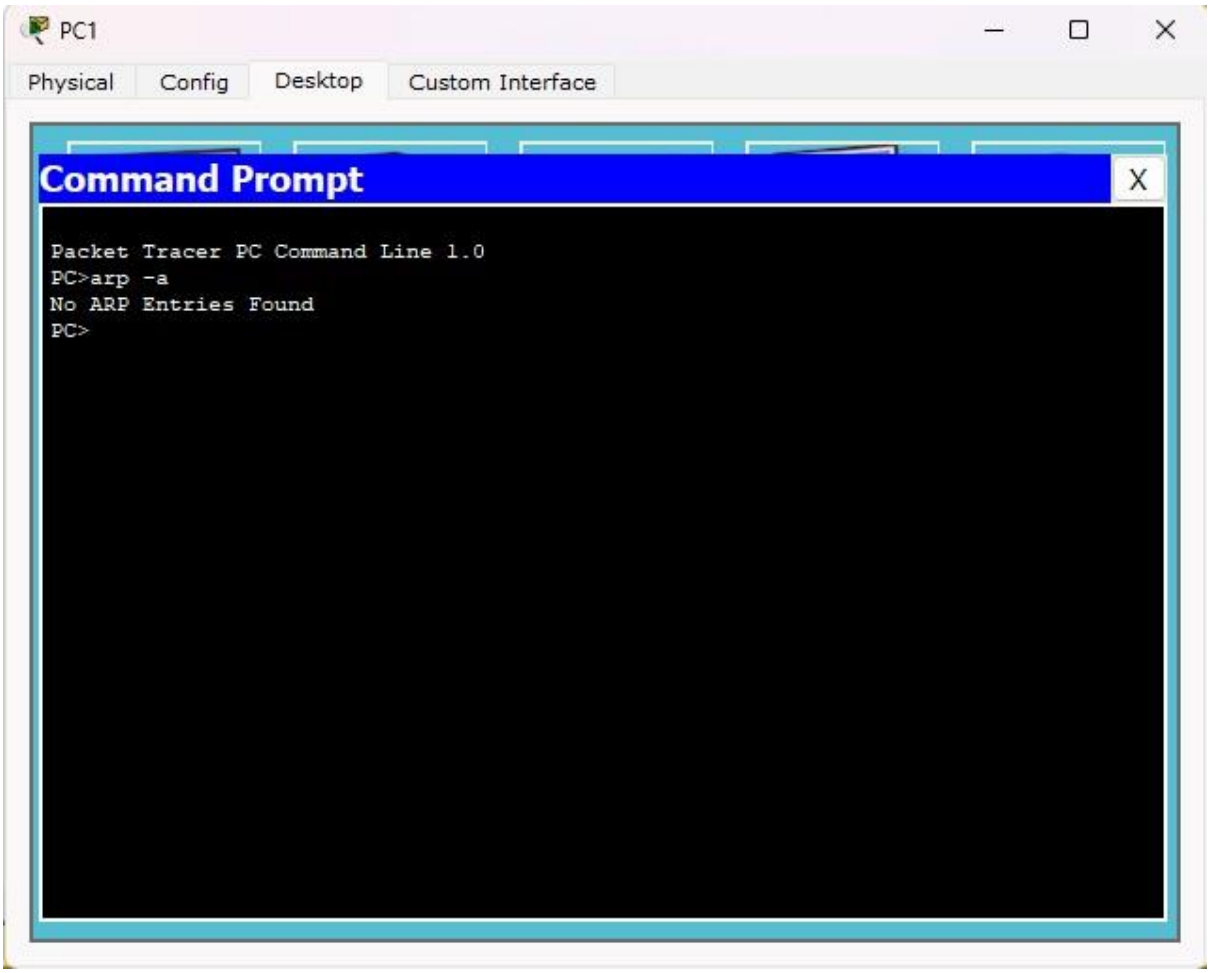
PAGE NO :
DATE :

4. The Server and the PC update their ARP tables matching IP addresses to MAC address.
5. Over Time, the ARP tables grows as data packets are sent
6. The MAC table of the switch which was initially empty updates its MAC table gradually too

ARP table for 10.0.0.4:		
IPaddress	Hardware Address	Interface
10.0.0.3	0001.C726.47E5	FastEthernet0

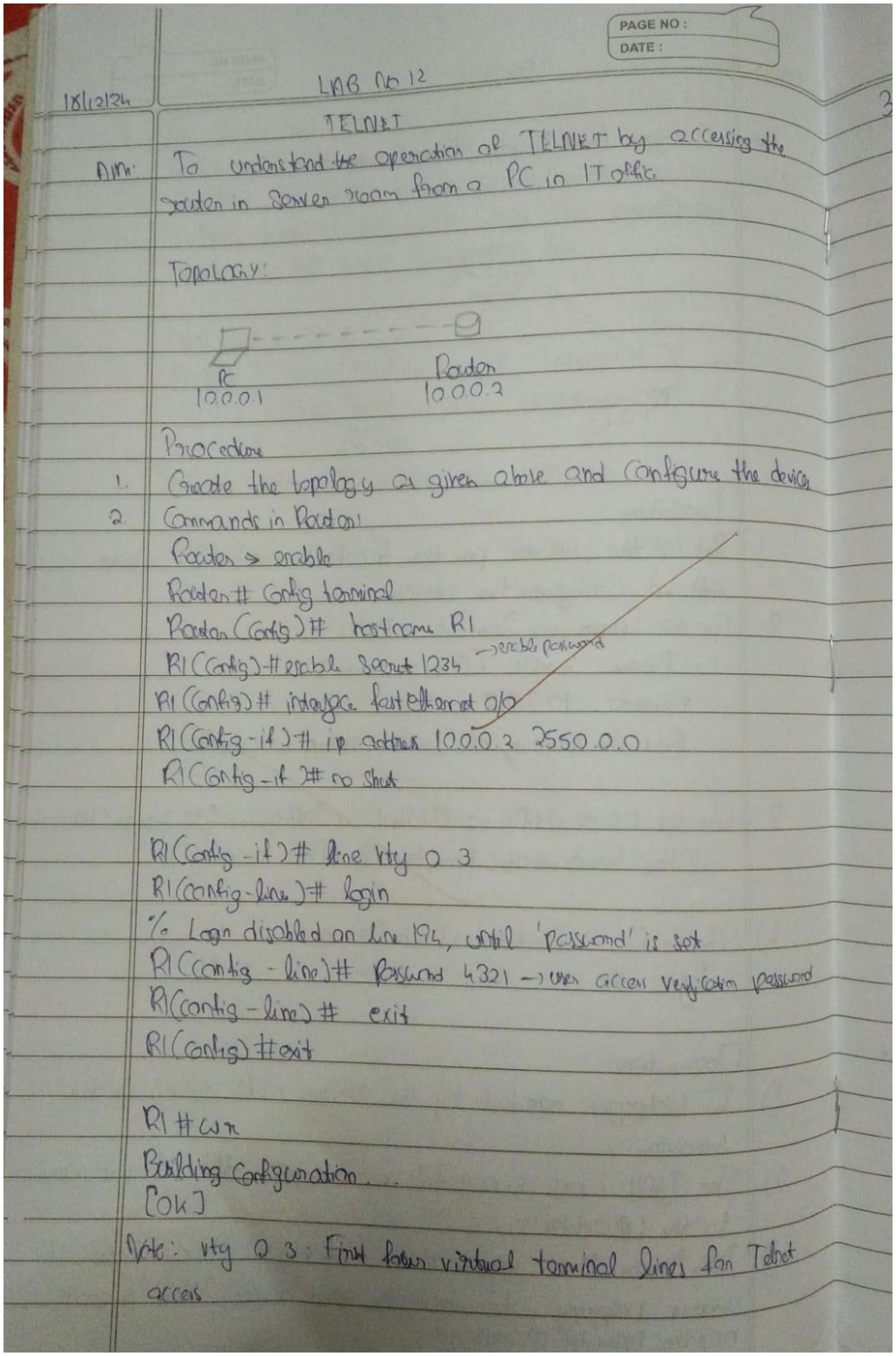
7. Similarly other ARP tables are updated.


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Simulation Panel
Event List
ARP Table for PC0
IP Addre: Hardware Address Interface
10.0.0.2 0003.E490.6097 FastEthernet0
ARP Table for Switch0
IP Addre: Hardware Address Interface
ARP Table for PC1
IP Addre: Hardware Address Interface
10.0.0.1 0004.9A10.2391 FastEthernet0
ARP Table for PC2
IP Addre: Hardware Address Interface
ARP Table for Server0
IP Addre: Hardware Address Interface
PC-PT PC0
PC-PT PC1
PC-PT PC2
Server-PT Server0
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
Switch0
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet2/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state to up
Switch>show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0003.e490.6097 DYNAMIC Fa1/1
1 0004.9a10.2391 DYNAMIC Fa2/1
1 000a.41b0.b710 DYNAMIC Fa3/1
Switch>
Copy
Paste


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>

LABORATORY PROGRAM – 10

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



LAB No 12

18/12/24

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 as given above and configure the device
2. Commands in Router:

Router > enable
Router# Config terminal
Router(Config)# hostname R1
R1(Config)# enable secret 1234 → enable password
R1(Config)# interface fastEthernet 0/0
R1(Config-if)# ip address 10.0.0.2 255.0.0.0
R1(Config-if)# no shut

R1(Config-if)# line vty 0 3
R1(config-line)# login
% Login disabled on line 194, until 'password' is set
R1(Config-line)# Password 4321 → user access verification password
R1(Config-line)# exit
R1(Config)# exit

R1# wr
Building Configuration...
[OK]

Note: vty 0 3: First four virtual terminal lines for Telnet access

PAGE NO :
DATE :

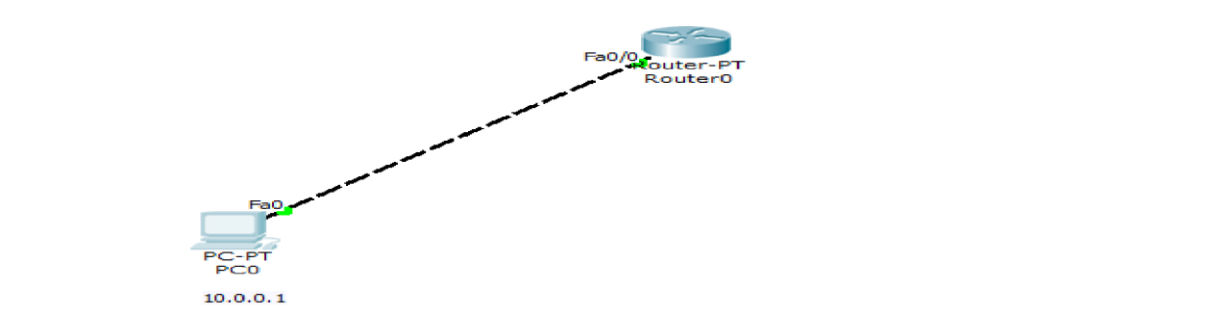
3. In PC: Command Prompt:
- First try pinging to see if devices are connected.

PC > telnet 10.0.0.2
Trying 10.0.0.2 ... Open

User Access Verification
Password: 4321
Password: 4321
R1 > enable
Password: 1234
R1 # show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0
R1 #

Observations:
1. The admin in PC is able to run commands as run in router CLI and see the results from PC
2. Telnet allows users 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.

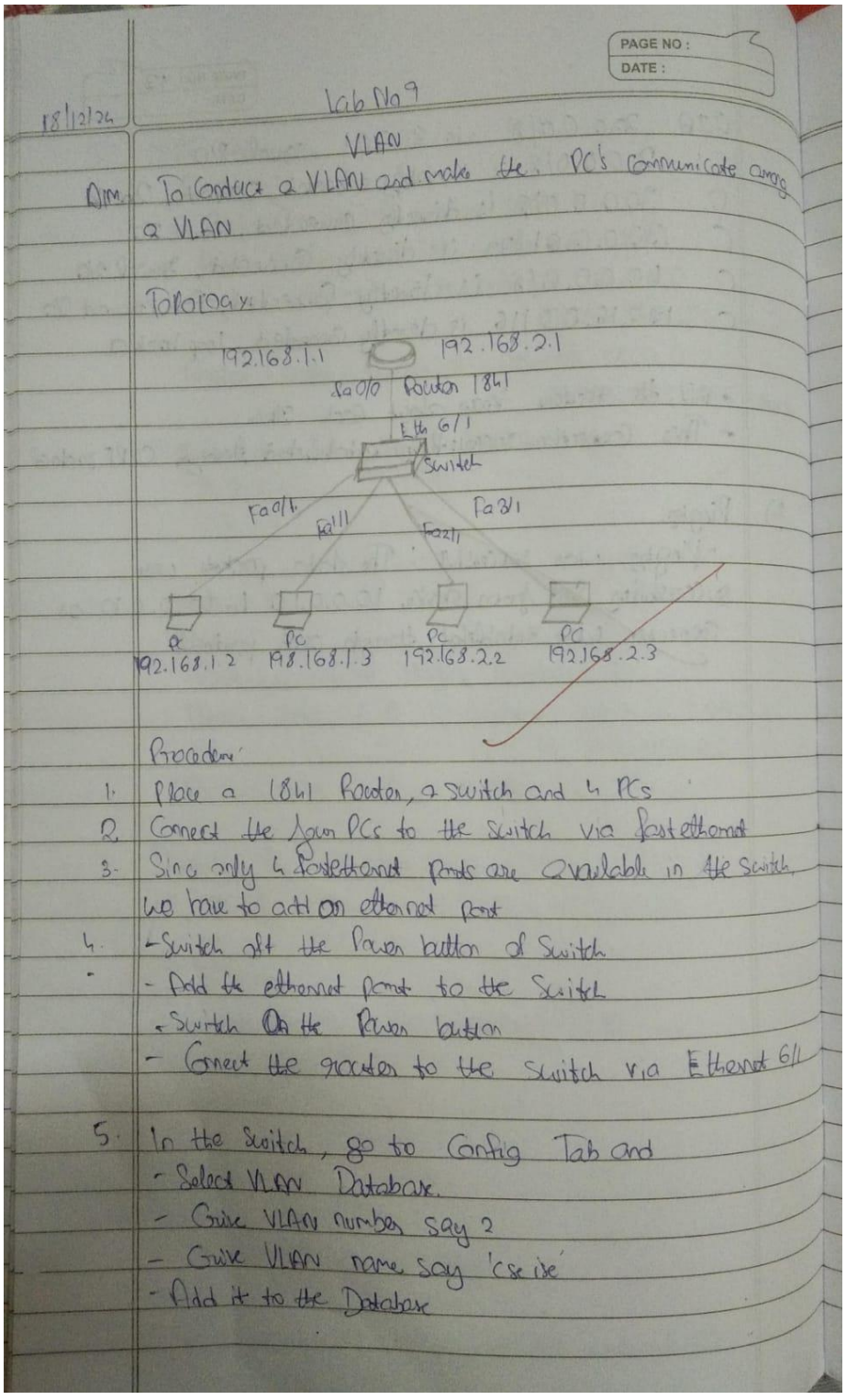


Command Prompt

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

Pinging 10.0.0.2 with 32 bytes of data:

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

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

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

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

LABORATORY PROGRAM – 11

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



PAGE NO :
DATE :

6. Select the Switch
- Go to Config
- Go to Ethernet 6/1 ie Connected to Router
- Make it the trunk

7. Configure the PCs as shown in the topology

8. Select Switch
- Go to Config
- Go to FastEthernet 2/1
- Set VLAN number as 2 ie 'Close'
- Similarly set VLAN 2 for FastEthernet 3/1 interface

9. Configure the Router
Router (Config) # interface fastethernet 0/0
Router (Config-if)# ip address 192.168.1.1 255.255.255.0
Router (Config-if)# no shut
Router (Config-if)# exit
Now, to Configure the router's VLAN interface
Router (Config)# interface fastethernet 0/0.1
Router (Config-subif)# encapsulation dot1q 2
Router (Config-subif)# ip address 192.168.2.1 255.255.255.0
Router (Config-subif)# no Shut
Router (Config-subif)# exit

10. Ping devices within the same VLAN and to devices of different VLAN

Observations:
1. When devices are pinged within same VLAN:
- Pinging 192.168.1.3 from 192.168.1.2
- The data packet doesn't go to the router
- The Switch forwards the packet without the need of the router

PAGE NO :
DATE :

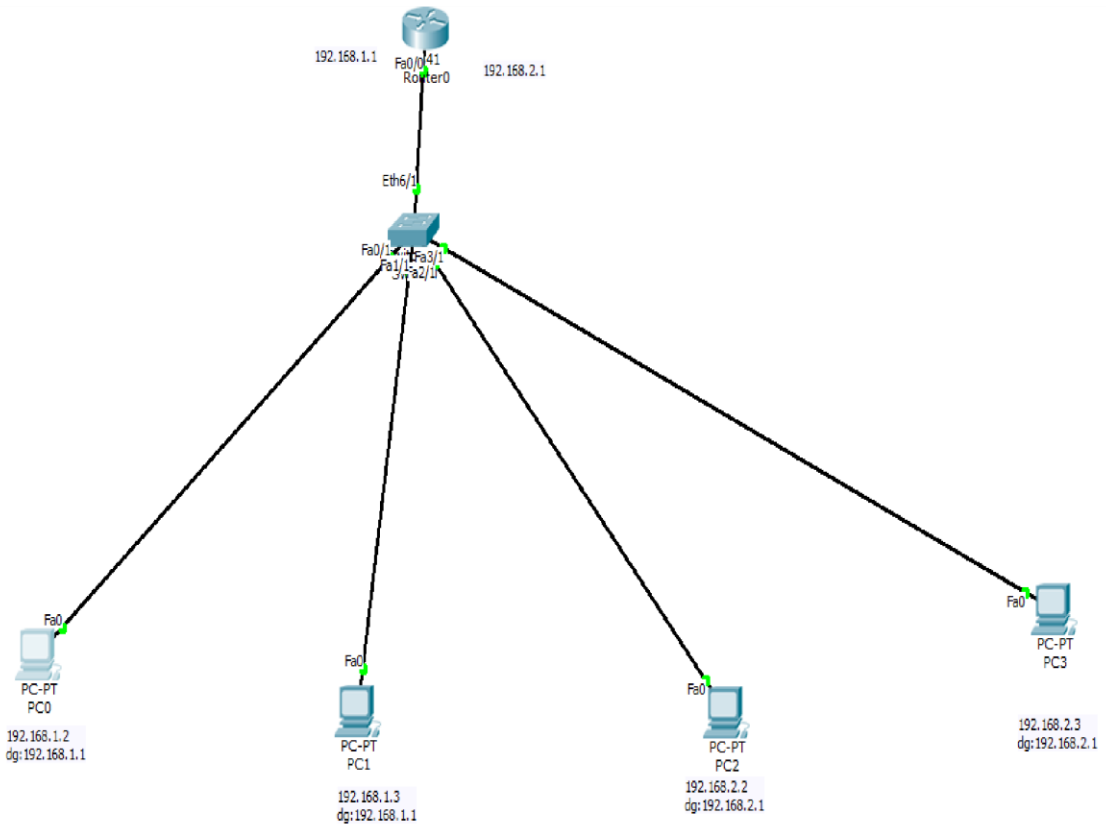
2. When a device pings a device of another VLAN
- Pinging 192.168.2.3 from 192.168.1.2
- The data packet journey is as follows:
192.168.1.2 → Switch → Router
↓
192.168.2.3 ← Switch

3. VLAN's divide a single switch into multiple logical switches
- Devices in one VLAN cannot directly communicate with devices in another VLAN without a router.

4. Traffic Isolation:
- Each VLAN maintains its own broadcast domain.
- Broadcasts sent by devices in one VLAN do not reach devices in another VLAN.

5. VLAN trunking allows switches to forward frames from different VLAN's over a single link called trunk

- This is done by adding an additional header information called tag to the ethernet frame - VLAN tagging.

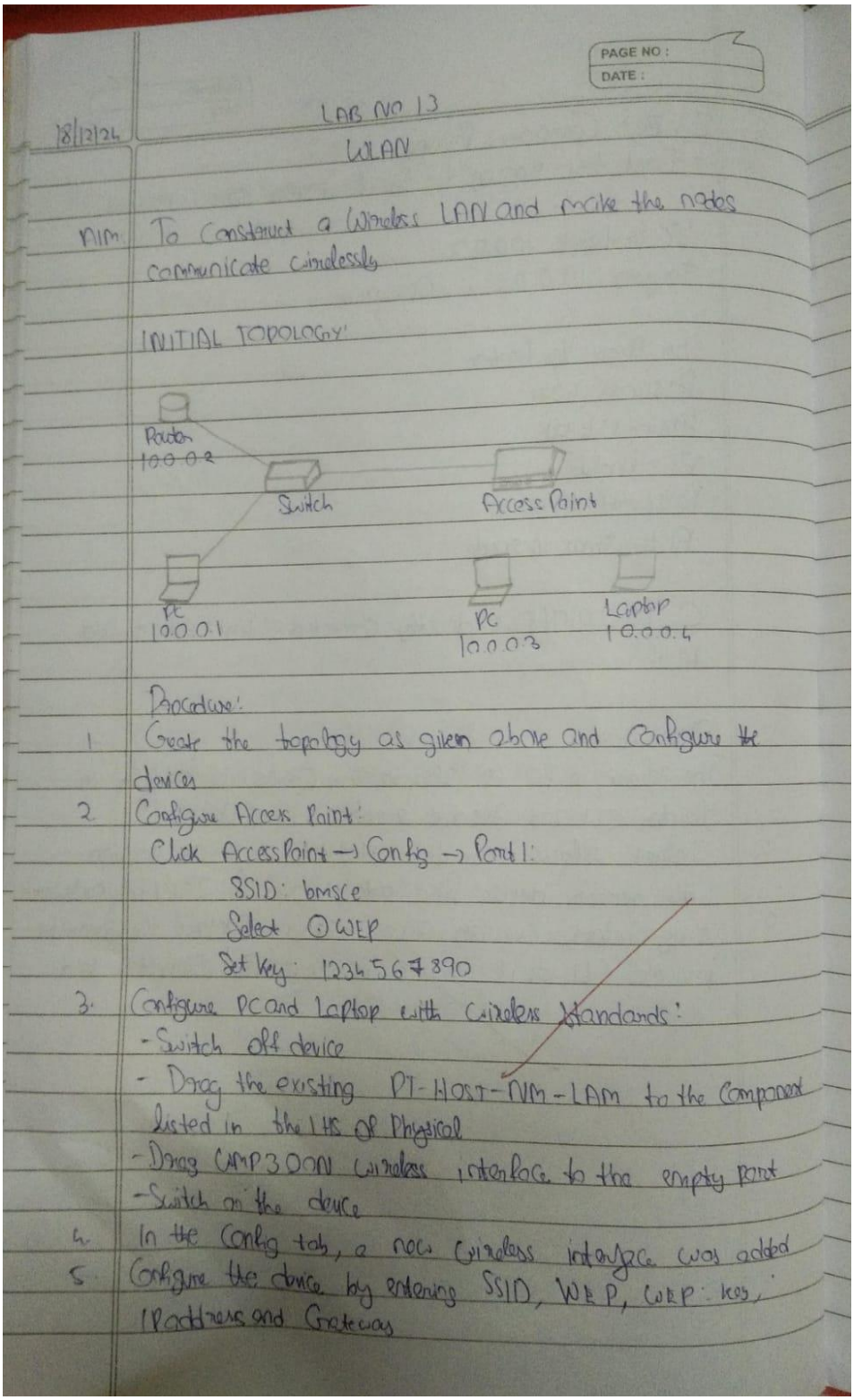

192.168.1.1
Fa0/0
Router0
192.168.2.1
Eth6/1
Fa0/1
Fa1/1
Fa3/1
Fa2/1
Fa0
PC-PT
PC0
192.168.1.2
dg:192.168.1.1
Fa0
PC-PT
PC1
192.168.1.3
dg:192.168.1.1
Fa0
PC-PT
PC2
192.168.2.2
dg:192.168.2.1
Fa0
PC-PT
PC3
192.168.2.3
dg:192.168.2.1

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

LABORATORY PROGRAM – 12

To construct a WLAN and make the nodes communicate wirelessly



PAGE NO :
DATE :

18/12/24

LAB NO 13

WLAN

Aim: To Construct a Wireless LAN and make the nodes communicate wirelessly

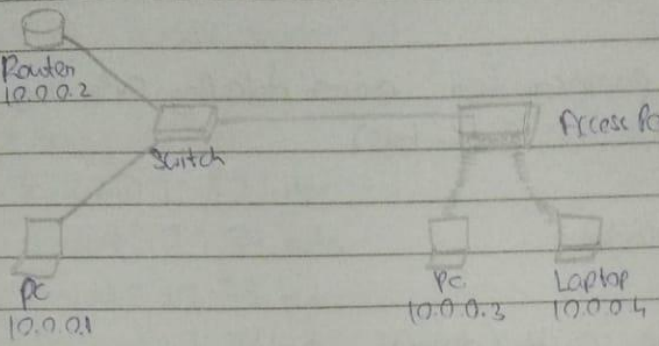
INITIAL TOPOLOGY:

Procedure:

1. Create the topology as given above and Configure the devices
2. Configure Access Point:
   Click AccessPoint → Config → Port1:
   SSID: bmsce
   Select ⊙ WEP
   Set key: 1234567890
3. Configure PC and Laptop with Wireless Standards:
   - Switch off device
   - Drag the existing PT-HOST-NM-1AM to the Component listed in the LHS of Physical
   - Drag CMP300N Wireless interface to the empty port
   - Switch on the device
4. In the Config tab, a new Wireless interface was added
5. Configure the device by entering SSID, WEP, WEP key, IPaddress and Gateway

PAGE NO :
DATE :

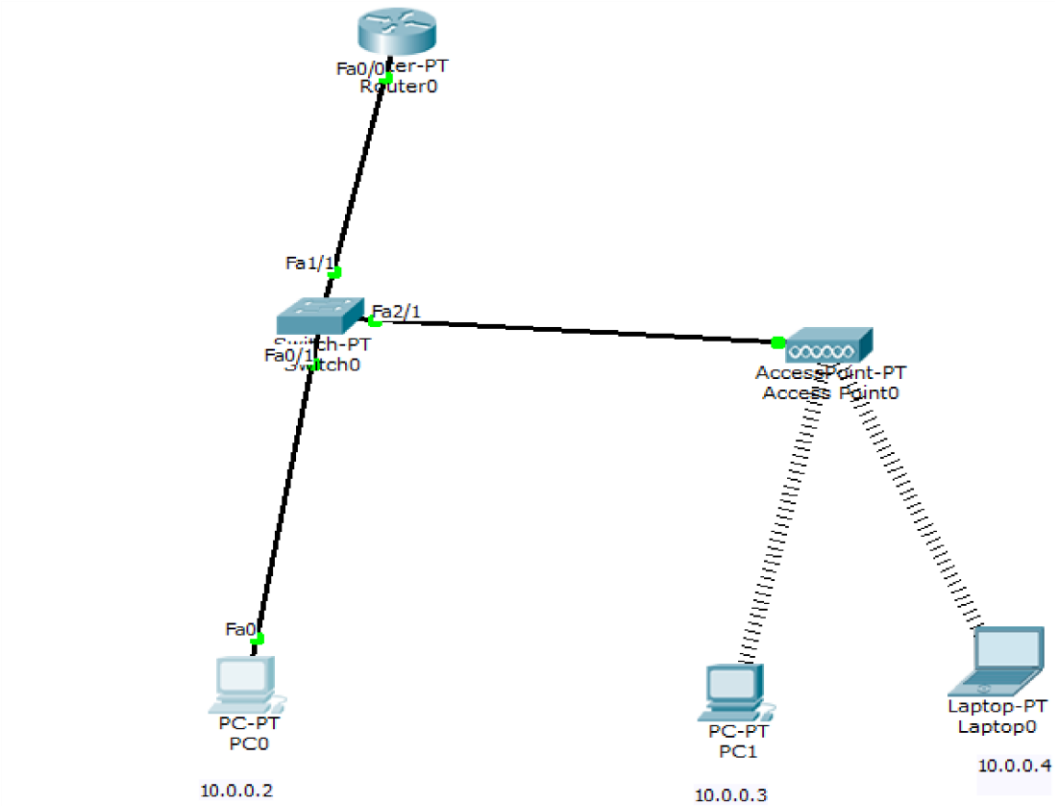
Topology after Wireless Configuration:

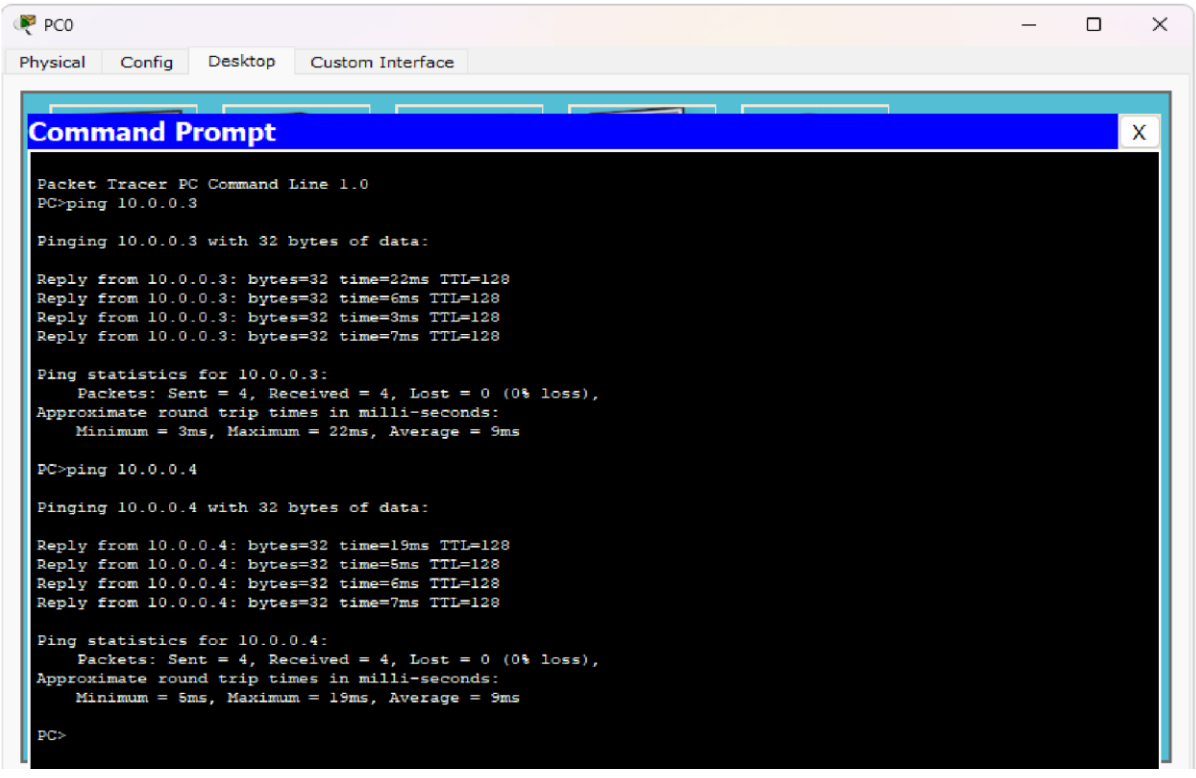


6) Ping from every device to every other device to check for Connection

Observations:

1. We were able to ping from every device to every other device
2. Access Point: Creates bridge between wired and wireless devices
   -SSID Broadcasting: announces the wireless network's name (SSID) to allow devices to connect using WEP, WPA or WPA2
3. WMP 300N wireless interface:
   - Wireless network adapter that enables devices to communicate with access point using wireless signals
4. Pinging: 10.0.0.1 to 10.0.0.3:
   10.0.0.1 → Switch → Access Point → 10.0.0.3
   - This is after the ARP tables are updated after broadcasting
5. Pinging: 10.0.0.3 to 10.0.0.1:
   10.0.0.3 → Access Point → Switch → 10.0.0.1
6. Pinging: 10.0.0.3 to 10.0.0.4:
   10.0.0.3 → Access Point → 10.0.0.4
7. Every device is now connected to every other device in the WLAN.

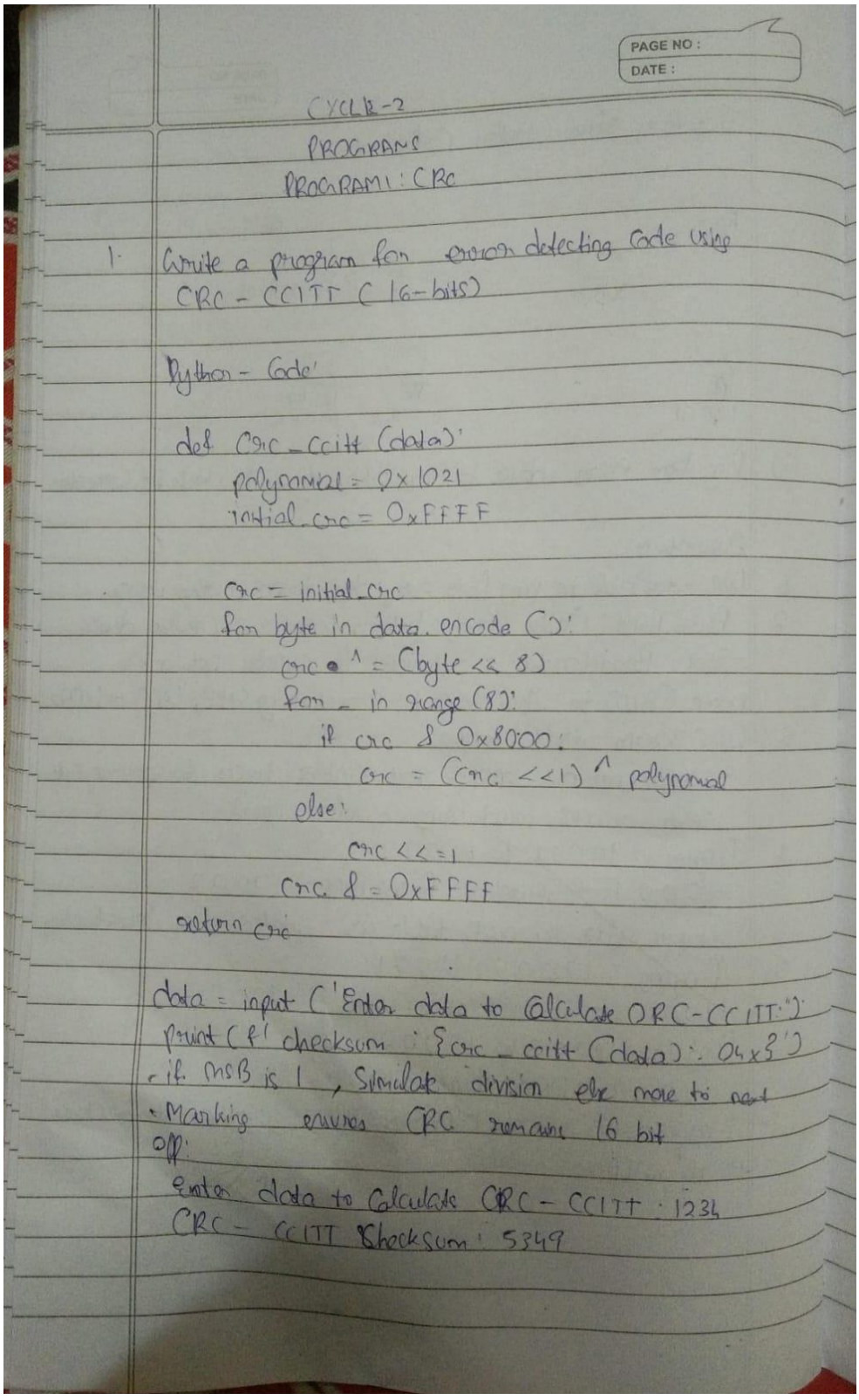

Router0
Fa1/1
Fa2/1
Fa0/1
Switch0
Access Point0
Fa0
PC-PT
PC0
10.0.0.2
PC-PT
PC1
10.0.0.3
Laptop-PT
Laptop0
10.0.0.4


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=22ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms 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 = 3ms, Maximum = 22ms, Average = 9ms
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=19ms TTL=128
Reply from 10.0.0.4: bytes=32 time=5ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms 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 = 5ms, Maximum = 19ms, Average = 9ms
PC>

\

LABORATORY PROGRAM – 1

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



CYCLE-2

PROGRAMS

PROGRAM1: CRC

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

Python - Code:

```
def crc_ccitt(data):
    polynomial = 0x1021
    initial_crc = 0xFFFF

    crc = initial_crc
    for byte in data.encode():
        crc ^= (byte << 8)
        for _ in range(8):
            if crc & 0x8000:
                crc = (crc << 1) ^ polynomial
            else:
                crc <<= 1
            crc &= 0xFFFF
    return crc

data = input("Enter data to calculate CRC-CCITT:")
print(f"checksum : {crc_ccitt(data):04x}")
```

- if MSB is 1, Simulate division else move to next
- Masking ensures CRC remains 16 bit

op:

Enter data to Calculate CRC - CCITT : 1234
CRC - CCITT Checksum: 5349

PAGE NO :
DATE :

Output :
Enter output rate : 30
Enter bucket size : 70

Incoming Packets :
Packet [0] : 82 bytes
Packet [1] : 39 bytes
Packet [2] : 43 bytes
Packet [3] : 74 bytes
Packet [4] : 67 bytes

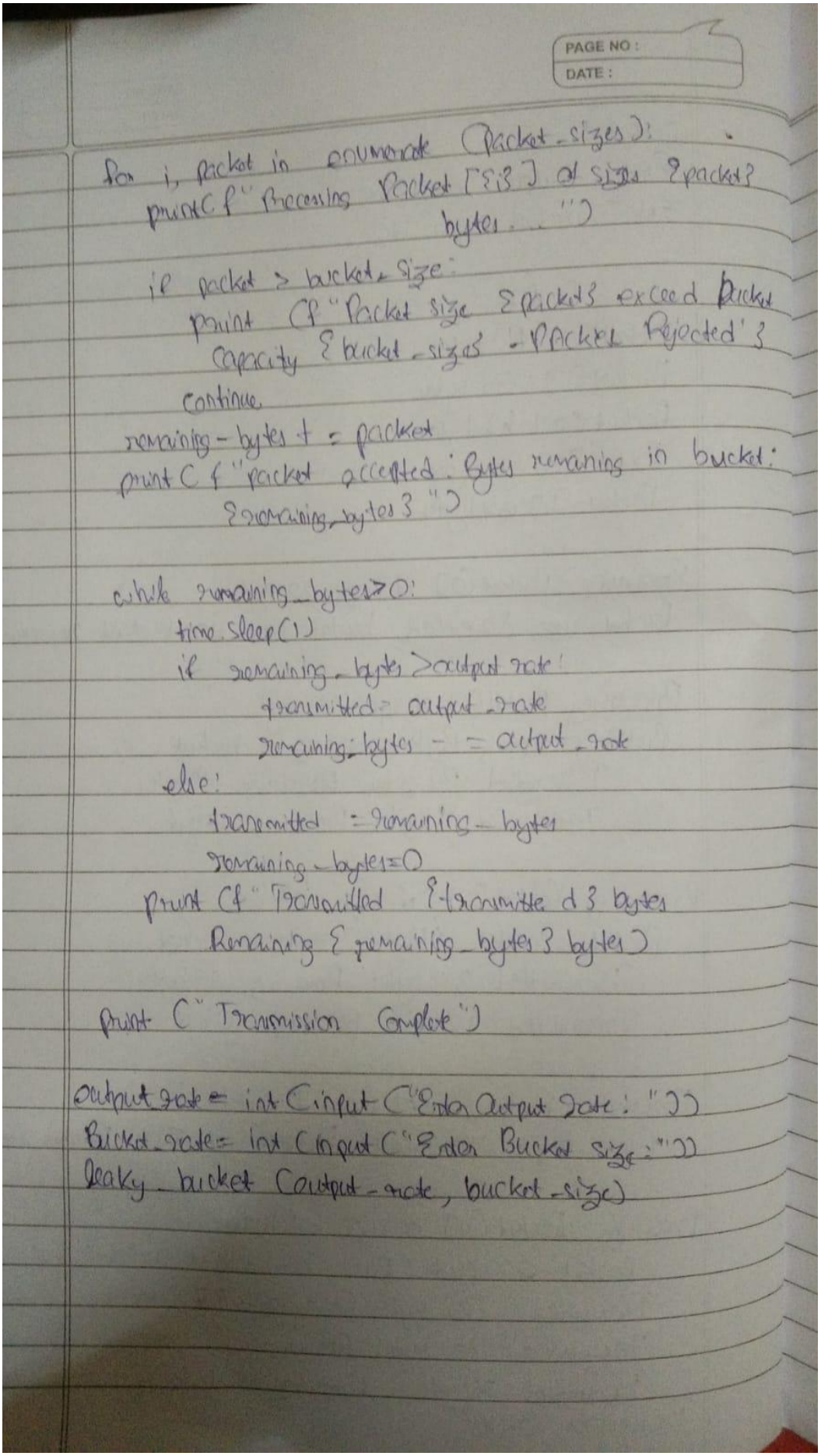
Processing Packet [0] of size 82 bytes
Packet size 82 exceed bucket size 70 : Packet rejected

Processing Packet [1] of size 39 bytes
Packet accepted : Bytes remaining in bucket : 39
Transmitted 30 bytes Remaining : 9 bytes
Transmitted 9 bytes : Remaining : 0 bytes

Processing Packet [2] of size 43 bytes
Packet accepted : Bytes remaining in bucket : 43
Transmitted 30 bytes Remaining : 13 bytes
Transmitted 13 bytes Remaining : 0 bytes

Processing Packet [3] of size 74 bytes
Packet size 74 exceed bucket size 70 : Packet rejected

Processing Packet [4] of size 67 bytes
Packet accepted : Bytes remaining in bucket : 67
Transmitted 30 bytes Remaining : 37 bytes
Transmitted 30 bytes Remaining : 7 bytes
Transmitted 7 bytes Remaining : 0 bytes
Transmission Complete



```
for i, packet in enumerate(packet_sizes):
    printf("Processing Packet [{i}] of size {packet} bytes...")
    if packet > bucket_size:
        print(f"Packet size {packet} exceed bucket capacity {bucket_size} - Packet Rejected")
        continue
    remaining_bytes += packet
    print(f"Packet accepted: Bytes remaining in bucket: {remaining_bytes}")

    while remaining_bytes > 0:
        time.sleep(1)
        if remaining_bytes > output_rate:
            transmitted = output_rate
            remaining_bytes -= output_rate
        else:
            transmitted = remaining_bytes
            remaining_bytes = 0
        print(f"Transmitted {transmitted} bytes. Remaining {remaining_bytes} bytes")

print("Transmission Complete")

output_rate = int(input("Enter Output rate: "))
Bucket_rate = int(input("Enter Bucket size: "))
leaky_bucket(output_rate, bucket_size)
```

Code

```
def xor(dividend, divisor):
    """Perform XOR operation between dividend and divisor."""
    result = ''
    for i in range(1, len(divisor)):
        result += '0' if dividend[i] == divisor[i] else '1'
    return result

def crc(data, gen_poly):
    """Compute the CRC check value using CRC-CCITT (8-bit)."""
    data_length = len(data)
    gen_length = len(gen_poly)

    # Append n-1 zeros to the data
    padded_data = data + '0' * (gen_length - 1)
    check_value = padded_data[:gen_length]

    for i in range(data_length):
        if check_value[0] == '1':
            # XOR operation if the first bit is 1
            check_value = xor(check_value, gen_poly)
        else:
            # Retain original check value if first bit is 0
            check_value = check_value[1:]

        # Shift left and add the next data bit
        if i + gen_length < len(padded_data):
            check_value += padded_data[i + gen_length]

    return check_value[1:]  # Remove the leading bit

def receiver(data, gen_poly):
    """Simulate the receiver side to check for errors."""
    print("\n-----------------------------")
    print("Data received:", data)

    # Perform CRC computation on received data
    remainder = crc(data, gen_poly)

    # Check if the remainder is all zeros
    if '1' in remainder:
        print("Error detected")
    else:
        print("No error detected")

if __name__ == "__main__":
    # Input data and generator polynomial
    data = input("Enter data to be transmitted: ")
    gen_poly = input("Enter the Generating polynomial: ")
```

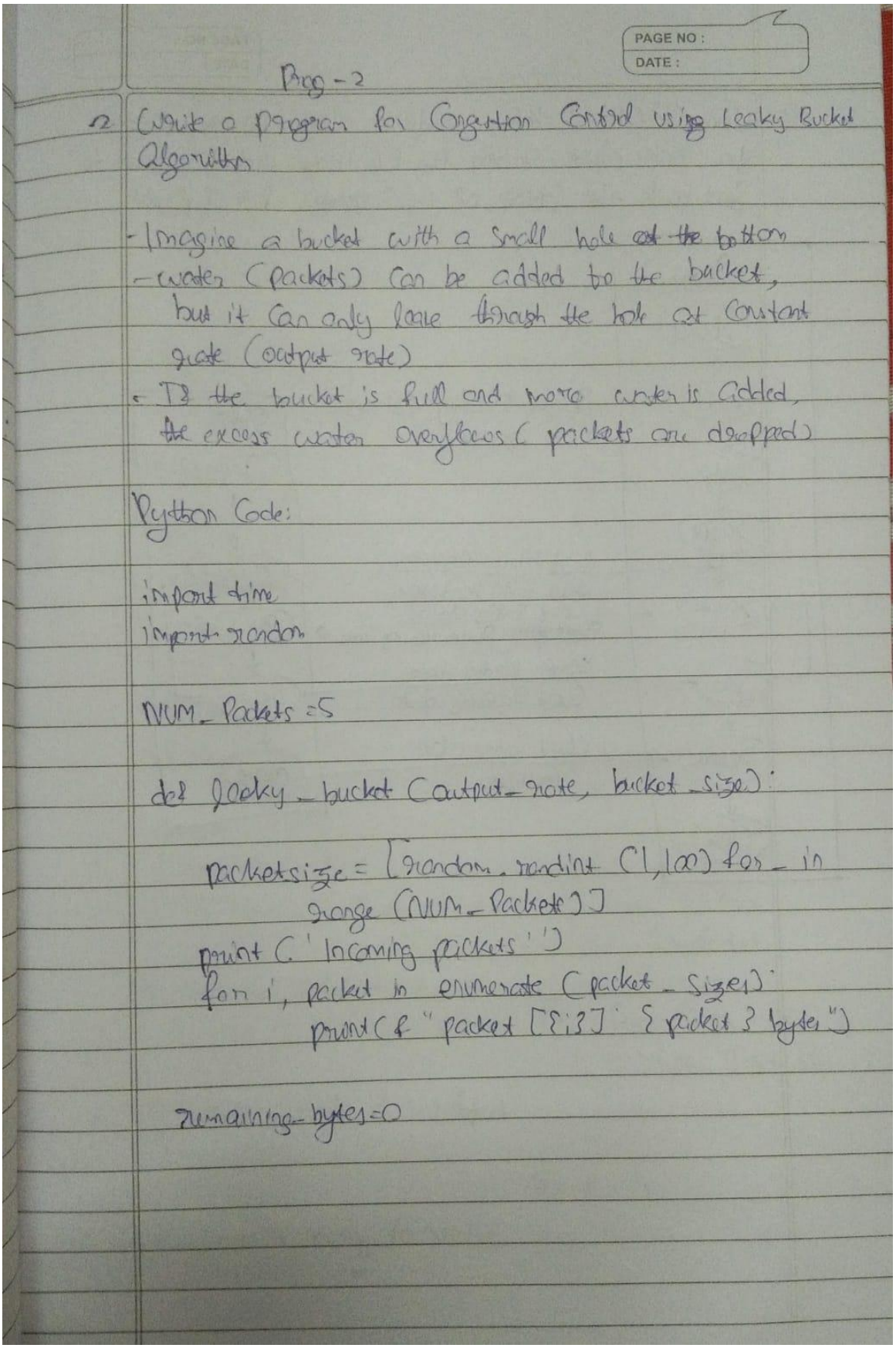
```
# Compute CRC check value
check_value = crc(data, gen_poly)
print("\n----------------------------------------")
print("Data padded with n-1 zeros:", data + '0' * (len(gen_poly) - 1))
print("CRC or Check value is:", check_value)

# Append check value to data for transmission
transmitted_data = data + check_value
print("Final data to be sent:", transmitted_data)
print("----------------------------------------\n")

# Simulate the receiver side
received_data = input("Enter the received data: ")
receiver(received_data, gen_poly)
```

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
```

LABORATORY PROGRAM – 2

Write a program for congestion control using Leaky bucket algorithm.



Prog - 2

2 Write a program for Congestion Control using Leaky Bucket Algorithm

- Imagine a bucket with a small hole at the bottom
- water (packets) can be added to the bucket, but it can only leave through the hole at constant rate (output rate)
- If the bucket is full and more water is added, the excess water overflows (packets are dropped)

Python Code:

```
import time
import random

NUM_Packets = 5

def leaky_bucket (output_rate, bucket_size):

    packetsize = [random.randint (1,100) for _ in
                  range (NUM_Packets)]
    print ("Incoming packets")
    for i, packet in enumerate (packet_size):
        print (f"packet [{i}]: {packet} bytes")

remaining_bytes = 0
```

Code

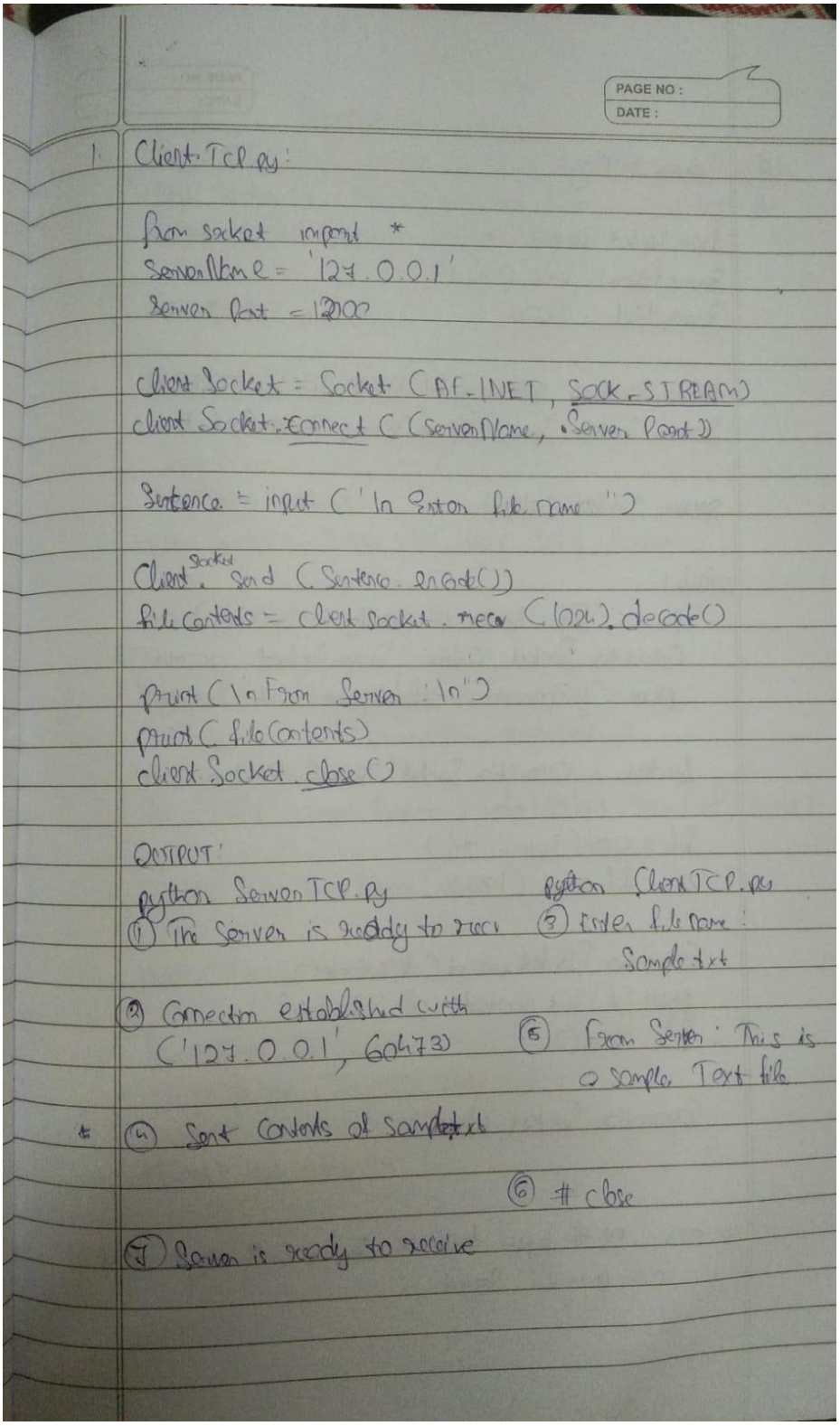
```
# Getting user inputs
storage = int(input("Enter initial packets in the bucket: "))
no_of_queries = int(input("Enter total no. of times bucket content is checked: "))
bucket_size = int(input("Enter total no. of packets that can be accommodated in the bucket:
"))
input_pkt_size = int(input("Enter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

LABORATORY PROGRAM – 3

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



1. Client TCP.py:

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000

clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))

sentence = input('Input file name ')

clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()

print('From Server :\n')
print(fileContents)
clientSocket.close()
```

OUTPUT:

python ServerTCP.py

① The server is ready to receive

② Connection established with ('127.0.0.1', 60473)

④ Sent contents of sample.txt

⑦ Server is ready to receive

python ClientTCP.py

③ Enter file name: sample.txt

⑤ From Server: This is a sample Text file

⑥ # close

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()
```

DATE :

2) Server TCP

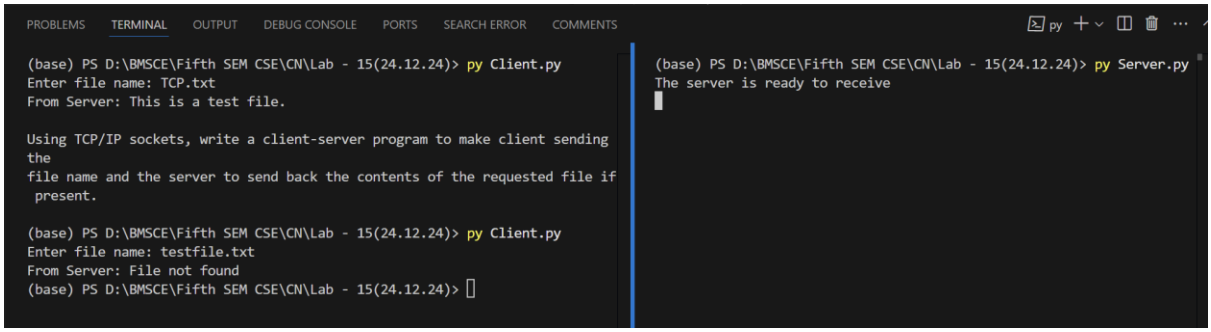
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000

serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))

serverSocket.listen()

while 1:
    print('The server is ready to receive')
    connectionSocket, addr = serverSocket.accept()
    print('Connection established with {addr}')

    sentence = connectionSocket.recv(1024).decode()

    file = open(sentence, 'r')
    l = file.read(1024)

    connectionSocket.send(l.encode())
    print('Sent contents of {sentence}')
    file.close()

    connectionSocket.close()
```

-1024 - no of bytes that the server will try to send at once from the socket

Code: Server.py

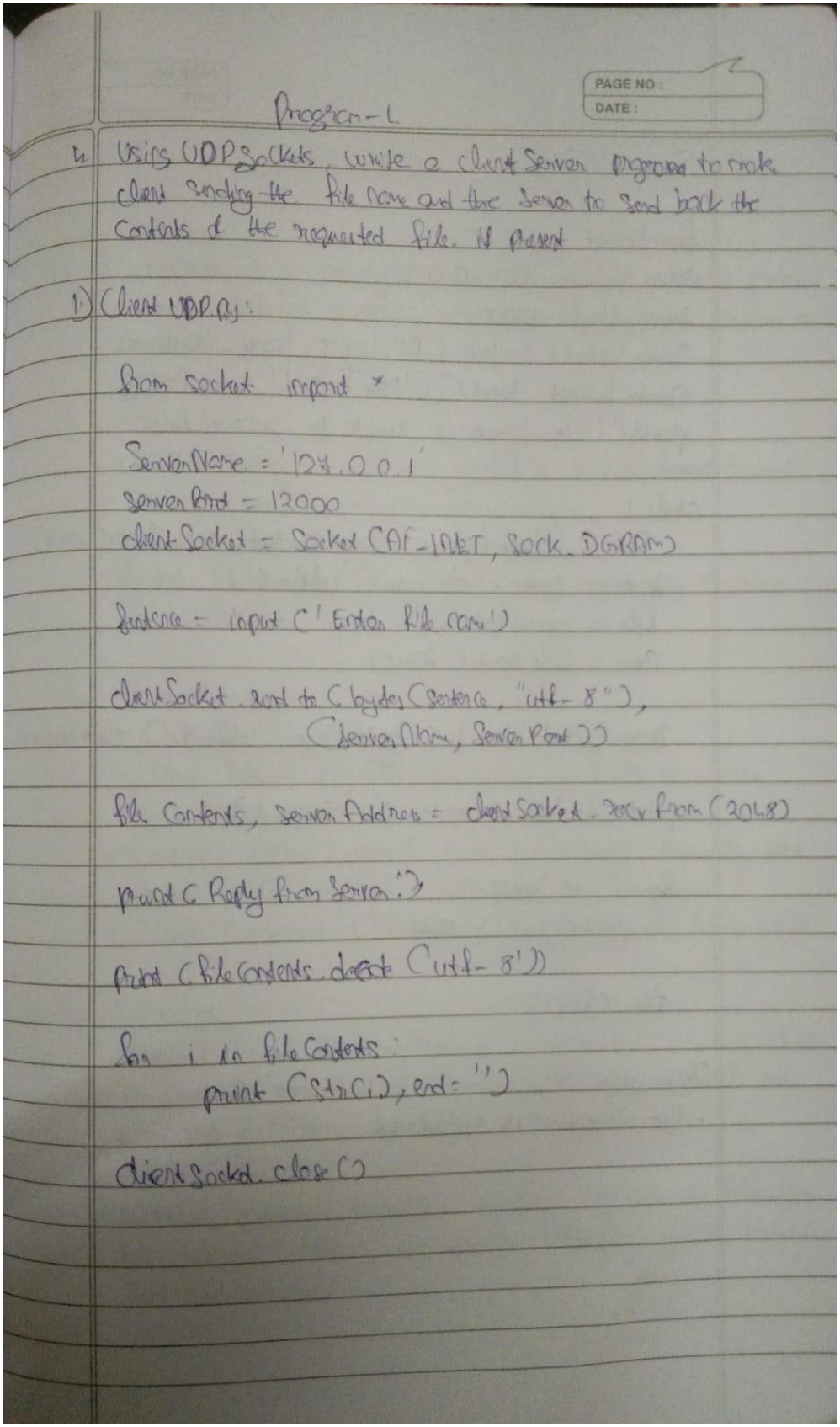
```
from socket import *
serverName = "127.0.0.1"  # Server address (localhost)
serverPort = 12000  # Port number to listen on

# 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)
        connectionSocket.send(fileContents.encode())  # Send file contents to client
        file.close()
    except FileNotFoundError:
        # Send error message if file not found
        connectionSocket.send("File not found".encode())

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

Output



LABORATORY 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.



Program-4

PAGE NO :
DATE :

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

1) Client UDP.py:

```
from socket import *

ServerName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input('Enter file name')

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

fileContents, serverAddress = clientSocket.recvfrom(2048)

print('Reply from Server:')

print(fileContents.decode('utf-8'))

for i in fileContents:
    print(str(i), end='')

clientSocket.close()
```

Code: ClientUDP.py

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

# Create UDP socket
clientSocket = socket(AF_INET, SOCK_DGRAM)

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

# Send the file name to the server using UDP
clientSocket.sendto(sentence.encode("utf-8"), (serverName, serverPort))

# Receive file contents from the server
fileContents, serverAddress = clientSocket.recvfrom(2048)

# Print the file contents received from the server
print("From Server:", fileContents.decode())

# Close the UDP socket
clientSocket.close()
```

DATE :

2. Server UDP.py:

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('127.0.0.1', serverPort))
print('The server is ready to receive')

while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode('utf-8')
    file = open(sentence, 'r')
    con = file.read(2048)

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

    print('Sent contents of ' + sentence)

    for i in sentence:
        print(str(i), end='')

    file.close()
```

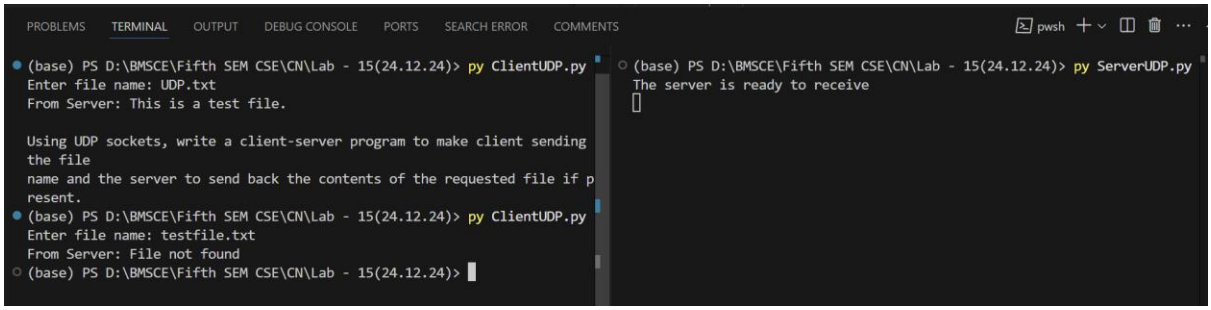
Observation:

- No connections is established

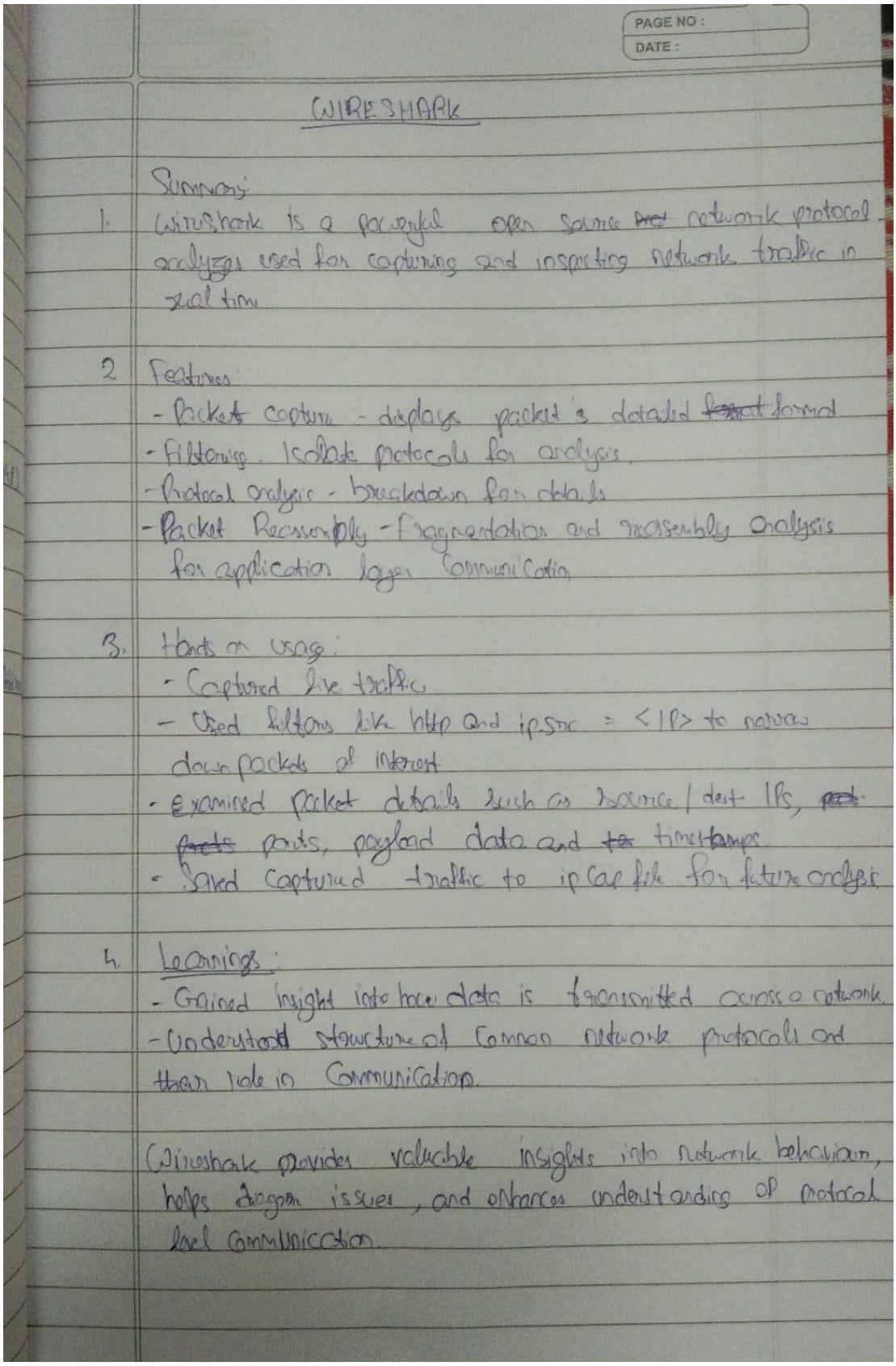
Code: ServerUDP.py

```
from socket import *
serverPort = 12000  # Port number to listen on

# Create UDP socket
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))  # Bind the socket to the server address and port

print("The server is ready to receive")

while True:
    # Receive file name from the client
    sentence, clientAddress = serverSocket.recvfrom(2048)

    # Try opening the file
    try:
        file = open(sentence.decode(), "r")  # Open file in read mode
        fileContents = file.read(2048)  # Read file content (up to 2048 bytes)
        serverSocket.sendto(fileContents.encode("utf-8"), clientAddress)  # Send file contents to
client
        file.close()
    except FileNotFoundError:
        # Send error message if file not found
        serverSocket.sendto("File not found".encode("utf-8"), clientAddress)
```

Output



Wireshark



PAGE NO :
DATE :

WIRESHARK

Summary:

1. Wireshark is a powerful open source ~~free~~ network protocol analyzer used for capturing and inspecting network traffic in real time.

2. Features:
- Packet capture - displays packet's detailed ~~format~~ format
- Filtering - Isolate protocols for analysis.
- Protocol analysis - breakdown for details
- Packet Reassembly - Fragmentation and reassembly analysis for application layer communication

3. Hands on usage:
- Captured live traffic
- Used filters like http and ip.src = <IP> to narrow down packets of interest
- Examined packet details such as source / dest IPs, ~~ports~~ ports, payload data and ~~to~~ timestamps.
- Saved captured traffic to .pcap file for future analysis

4. Learnings:
- Gained insight into how data is transmitted across a network
- Understood structure of common network protocols and their role in communication.

Wireshark provides valuable insights into network behaviour, helps diagnose issues, and enhances understanding of protocol level communication.