VISVESVARAYATECHNOLOGICALUNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Dheemanth M (1BM22CS087)

in partial fulfillment for the award of the degree of

BACHELOROFENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Academic Year 2024-25 (odd)

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 Network (23CS5PCCON)" carried out by **Dheemanth M (1BM22CS087),** who is a bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Sandhya A Kulkarni Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	09-10-24	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.	4-8
2	09-10-24	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.	9-11
3	16-10-24	Configure default route, static route to the Router (Part 1).	12-14
4	23-10-24	Configure default route, static route to the Router (Part 2).	15-17
5	13-11-24	Configure DHCP within a LAN and outside LAN.	18-22
6	20-11-24	Configure RIP routing Protocol in Routers .	23-25
7	20-11-24	Demonstrate the TTL/ Life of a Packet.	26-28
8	27-11-24	Configure OSPF routing protocol.	29-31
9	18-12-24	Configure Web Server, DNS within a LAN.	32-33
10	18-12-24	To construct a simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).	34-36
11	18-12-24	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	37-39
12	18-12-24	To construct a VLAN and make the PC's communicate among a VLAN.	40-43
13	18-12-24	To construct a WLAN and make the nodes communicate wirelessly.	44-46
14	18-12-24	Write a program for error detecting code using CRC-CCITT (16-bits).	47-48
15	18-12-24	Write a program for congestion control using Leaky bucket algorithm.	49-50

16	18-12-24	Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.	51-52
17	18-12-24	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.	53-54
18	18-12-24	WIRESHARK	60-62

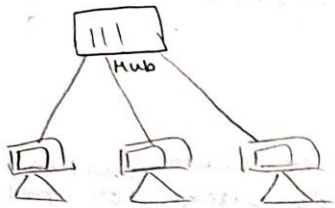
Github Link: LINK

Program 1

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

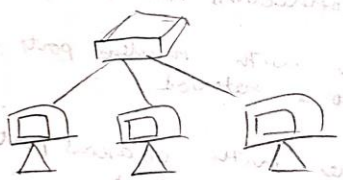
Topology , Procedure and Observation:

Hub:



- Layer: operates at layer1 (physical layer) of OSI model.
- It broadcasts data to all connected devices doesn't filter or manage traffic.
- Limited efficiency, more collisions can occur due to simultaneous data transmission.

Switch:



- operates at data link layer of OSI model.
- Its forwards data only to the specific device for which it is intended, based on MAC address
- More efficient, reduces collisions.

Experiment 1

1. PC to Server

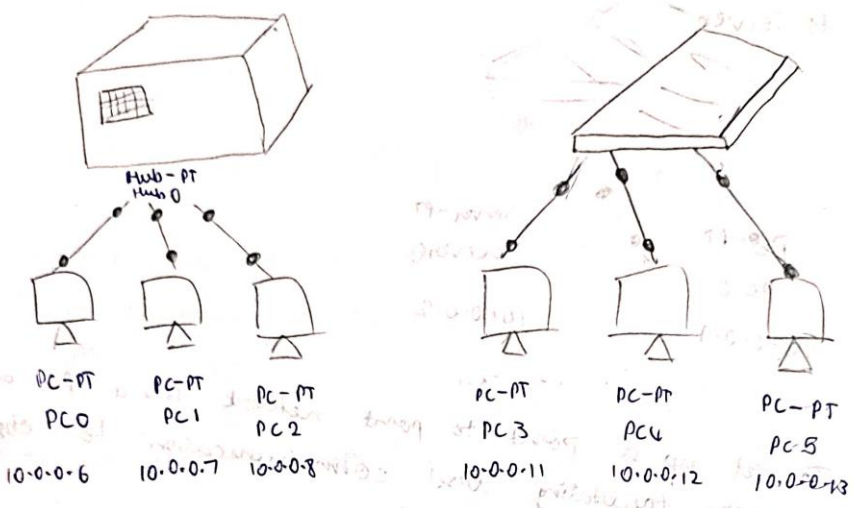
Aim: To set up a point to point network btw a PC and a server, facilitating direct communication to observe data exchange.

Topology: A PC is connected to server using a crossover ethernet cable.

IP address of PC - 10.0.0.1, server - 10.0.0.2.

Observation: Direct communication allows PC to communicate with server, which is typical in small networks for tasks such as file sharing, service request or testing server response to client queries.

2. Hub and Switch



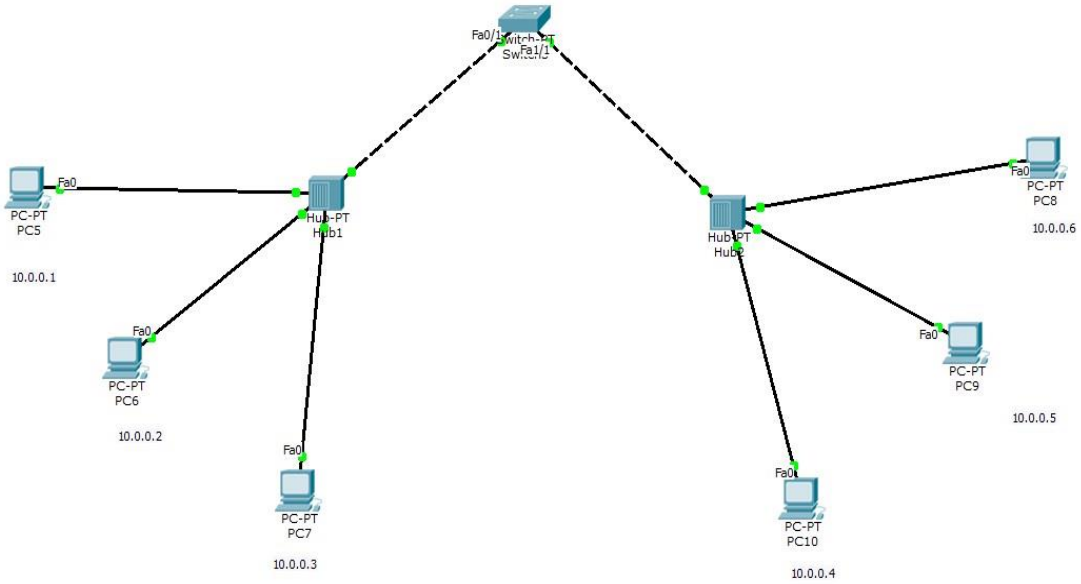
Aim- To create simple network consisting of 3 PCs connected to a central hub and another network with 3 PCs connected to a switch. This connection will help observe the behaviour of data transmission using hub & switch devices.

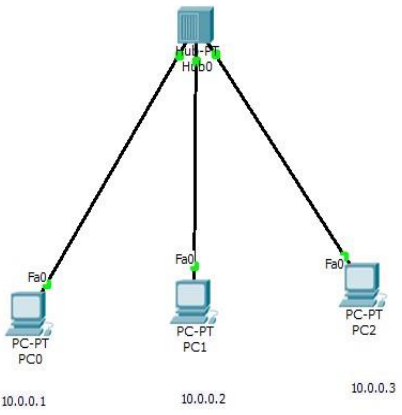
Topology: 3 PC's are connected to a hub & switch using straight through ethernet cables.

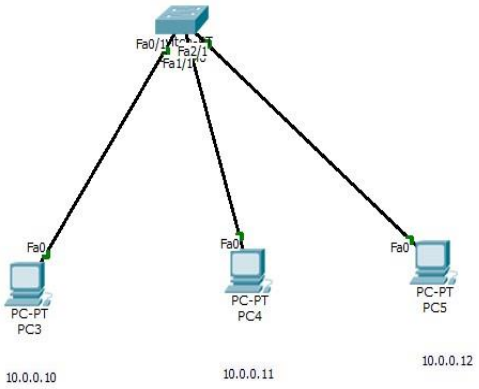
Observation: Hub broadcasts packets to all devices which may cause unnecessary traffic.

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

Screen Shots:







Program 2

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

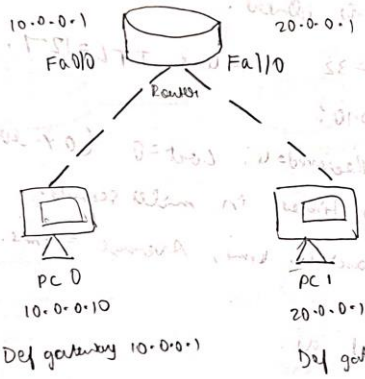
Topology , Procedure and Observation:

LAB 2

Aim:

To connect 2 PC's in two different network using a router

Topology.



Procedure:

1. Place a generic router and 2 PC's.
2. For each PC, set the gateway address and IP address.
   For PC1 set Gateway as 10.0.0.1 and IP address as 10.0.0.10.
   For PC2 set Gateway as 20.0.0.1 and IP address as 20.0.0.10.
3. For the router configuration, config it manually, by clicking the CLI option in router.
   in the command prompt, enter the following commands.

```
>enable
# config terminal
# interface fa0/0 ethernet 0/0
- ip address 10.0.0.1 255.0.0.0
no shutdown
exit
```

4) Similarly ~~for PC1~~ PC2 ~~also~~ configure for the FastEthernet1/0.

5) In the command prompt, enter 'ping' 20.0.0.10 for sending from PC1 to PC2.

Observation:

1. Data packet was sent from PC 10.0.0.10 to router.
2. the router sent the packet to 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.

Ping statistics for 20.0.0.10:

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

Approximate round trip times in milli seconds:

Minimum=4ms, Maximum 4ms, Average = 4ms.

Ip route was observed as.

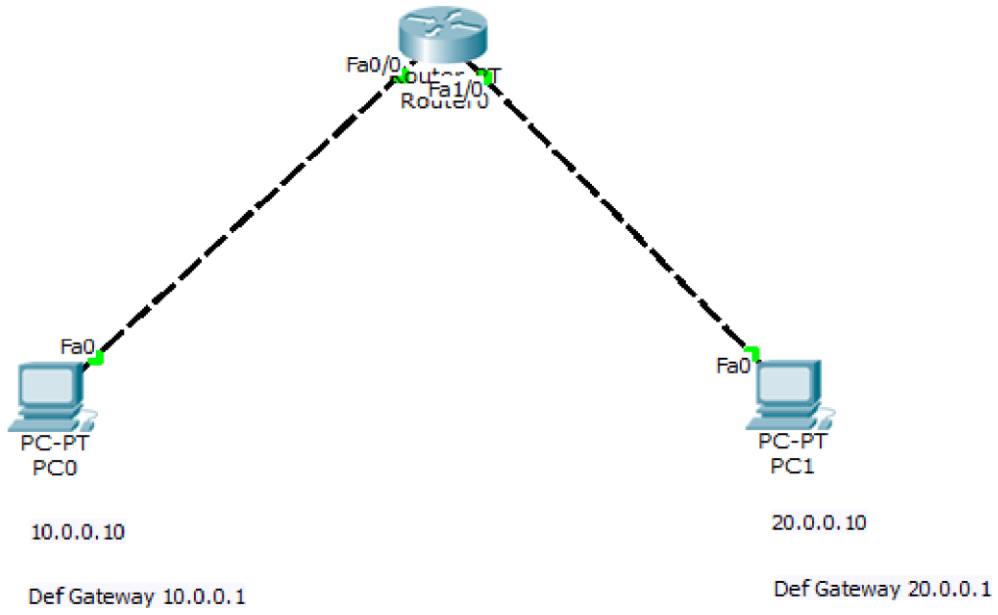
C => connected

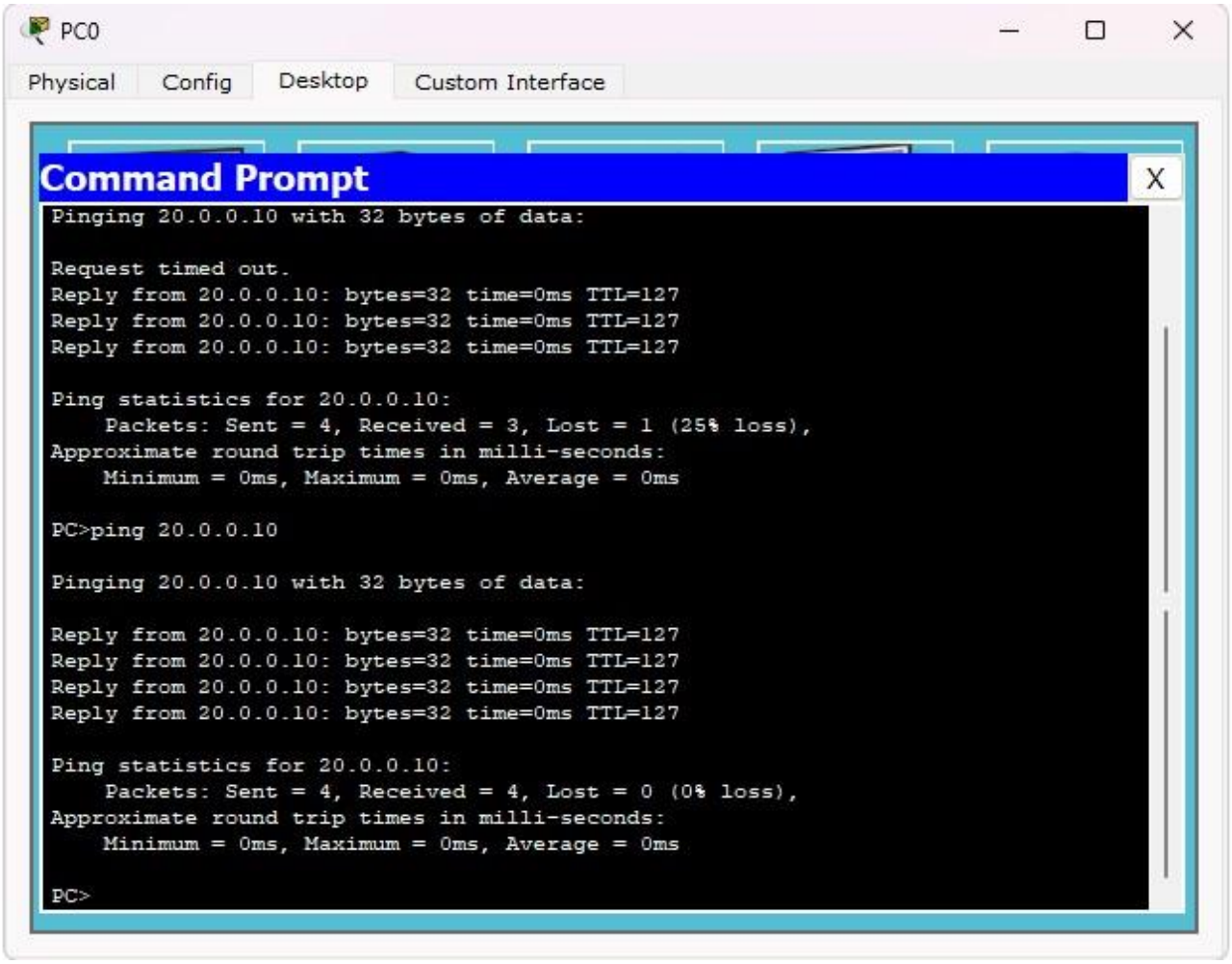
show ip route

C 10.0.0.0/8 is directly connected, FastEthernet0/0.

C 20.0.0.0/8 is directly connected, FastEthernet1/0.

Screen Shots:





Program 3

Aim: Configure default route, static route to the Router(Part 1).

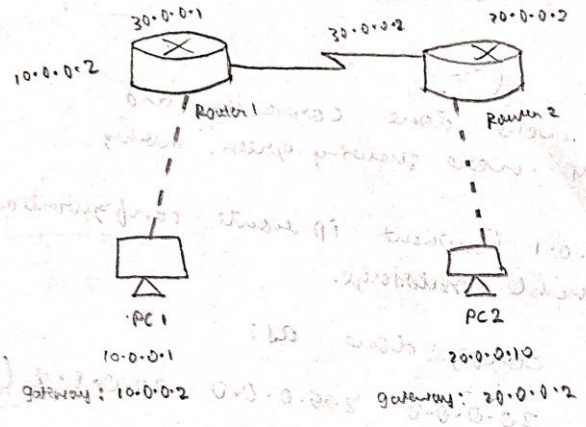
Topology , Procedure and Observation:

16.10.24

LAB-3

AIM: To configure IP address to routers in packet tracer. Explore following messages, ping response destination unreachable, request timed out, reply.

Topology.



Procedure:

1. Place 2 generic router and 2 PC's.
2. For each pc, set gateway address and IP address.
   For each router, set the IP address.
   For PC1 → Gateway as 10.0.0.2 and IP address 10.0.0.10.
   For PC2 → Gateway as 20.0.0.2 and IP address 20.0.0.10.
   For Router 1 → ip address 30.0.0.1
   For Router 2 → ip address 30.0.0.2.
3. For the router configuration, config it manually by clicking CLI option in router.
   in command prompt, enter following command.

>enable

config terminal

interface fastEthernet 0/0

ip address 10.0.0.10 255.0.0.0

no shutdown

exit.

for router, ⇒ interface serial 2/0

ip address 30.0.0.10 255.0.0.0.

Observation:

1. The connections were done correctly and all the lights were showing green, working.

2. when ping 20.0.0.1 [without ip route configuration] we get unreachable message.

3. when ip route config. done as:

ip route 20.0.0.0 255.0.0.0 30.0.0.2 (for R1)

likely for R2,

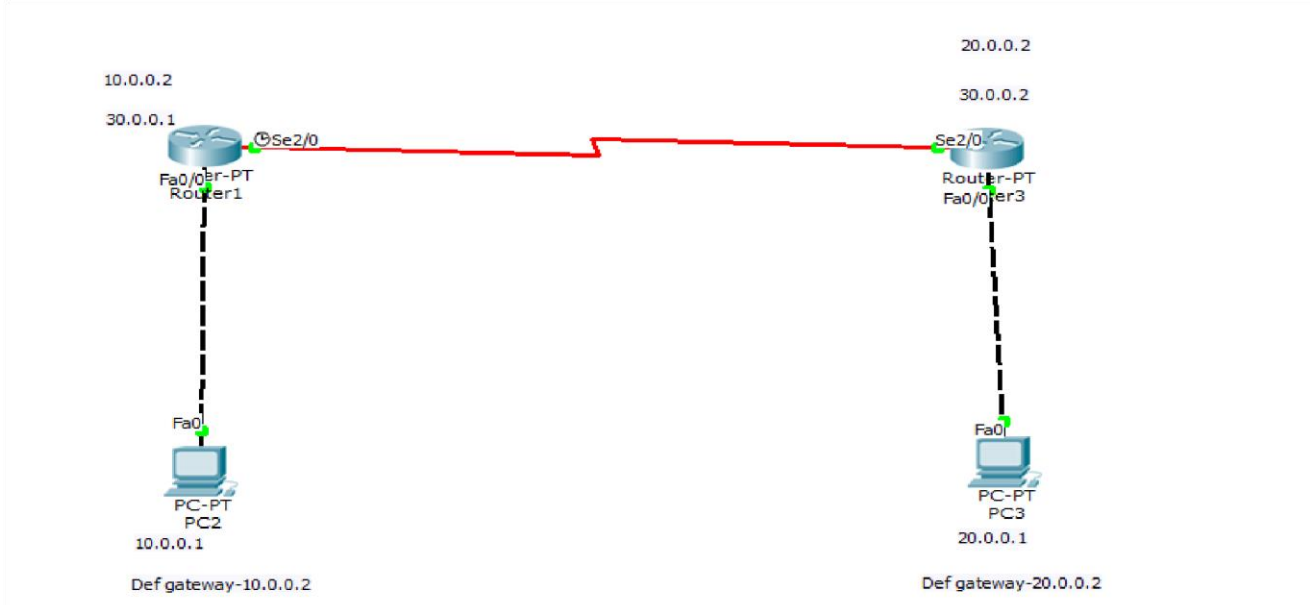
we get successful message.

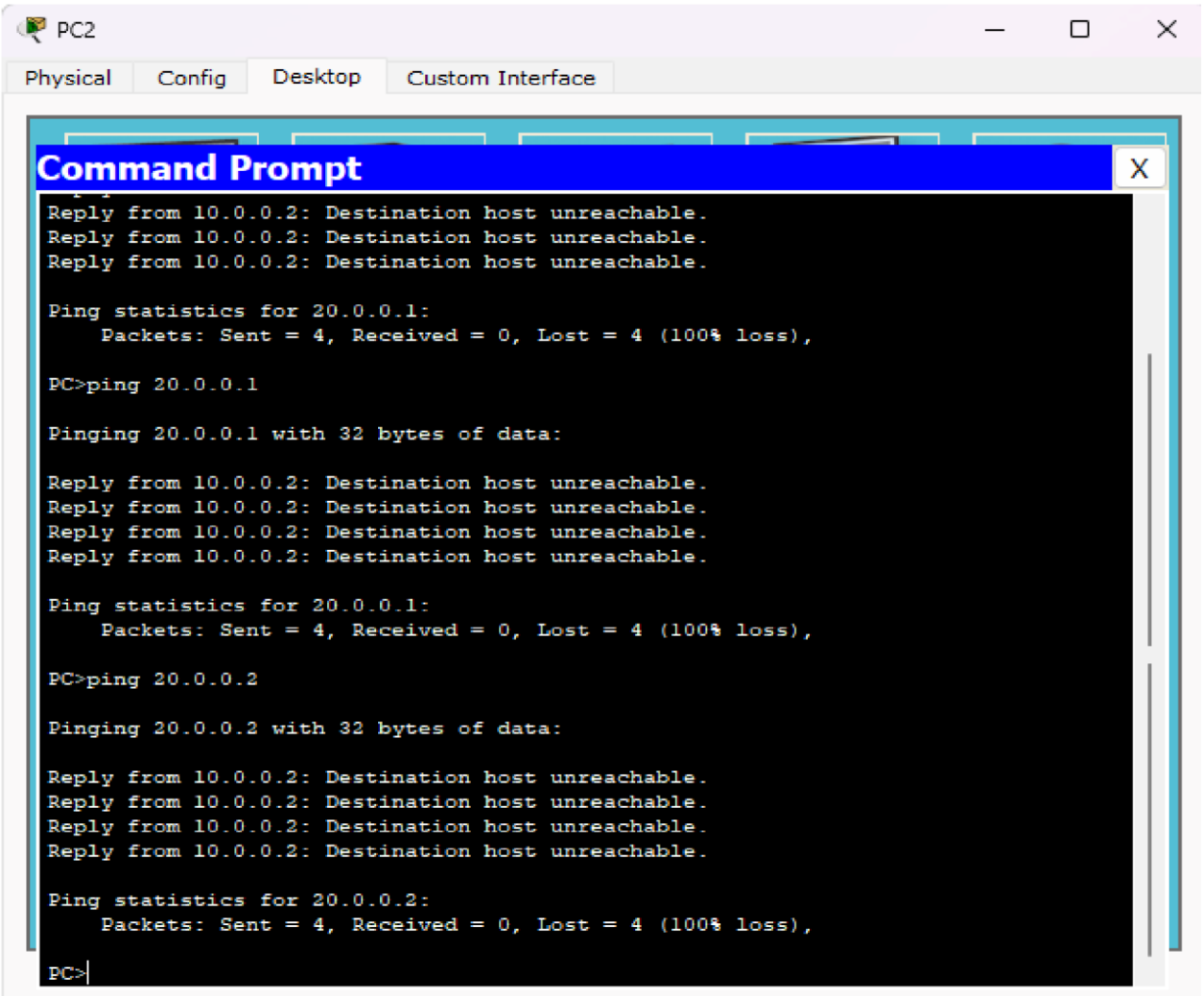
like, ping 30.0.0.1

output. Reply from 30.0.0.1 : byte=32 time=1ms TTL=255

23/10

Screen Shots:





Program 4

Aim: Configure default route, static route to the Router(Part 2).

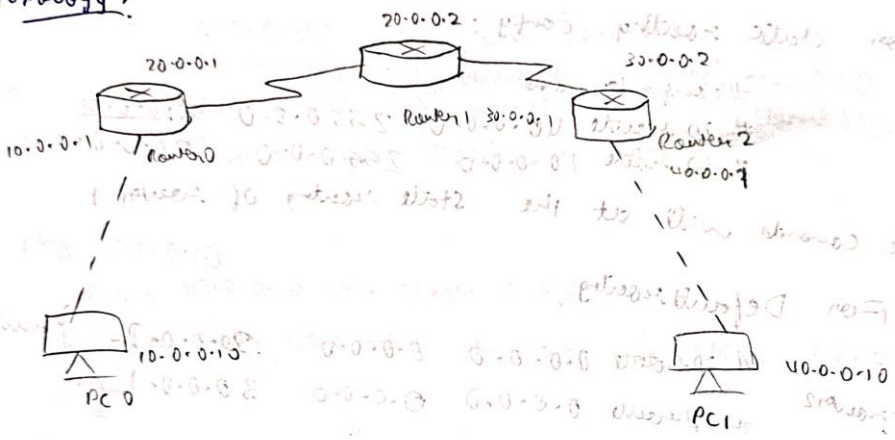
Topology , Procedure and Observation:

23.10.24.

LAB 4

AIM: To configure a default route & static route to a router in a network & verify connectivity.

Topology:



Procedure:

1. Take 3 generic routers and 2 PC's
2. connect the routers through serial DCE and PC to routers through copper crossover
3. For each PC set gateway address and IP address and for each router, set the IP address.

   For PC0 → ip address 10.0.0.10 and gateway 10.0.0.1
   For PC1 → ip address 40.0.0.10 & gateway 40.0.0.1
   For Router0 → ip address 20.0.0.1 (serial) & 10.0.0.1 (fast Ethernet)
   For router 1 → ip address 20.0.0.2 (serial)
   For router 2 → ip address 30.0.0.2 (serial) & 40.0.0.1 (fast ethernet)

4. For router config in CLI,

PTO.

> enable
config terminal
interface fastEthernet 0/0
ip address 10.0.0.10 255.0.0.0
no shut
exit.

This configures connection b/w PC and router

5. For static routing config:

config terminal
ip route 40.0.0.0 255.0.0.0 30.0.0.2
ip route 10.0.0.0 255.0.0.0 20.0.0.1

These cmnds will set the static routing of router 1.

6. For Default routing,

in router2 # ip route 0.0.0.0 0.0.0.0 20.0.0.2 } router0
ip route 0.0.0.0 0.0.0.0 30.0.0.1.

0.0.0.0 means any ip address will be sent to next hop address.

Observation:

After config. the default state, PC0 could communicate with external nw, including PC1.
The static route ensured that packets followed the specified path as given in static routing.

O/P:

Router 0: #show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/2
C 20.0.0.0/8 is directly connected, Serial 2/0
S* 0.0.0.0/8 [1/0] via 20.0.0.2

Router2: # show ip route

C 40.0.0.0/8 is directly connected, FastEthernet 0/0

C 30.0.0.0/8 is directly connected, Serial 3/0

S* 0.0.0.0/8 [1/0] via 30.0.0.1

Router1: # show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

C 20.0.0.0/8 is directly connected, Serial 2/0.

C 30.0.0.0/8 is directly connected, Serial 3/0.

S 40.0.0.0/8 [1/0] via 30.0.0.2.

Ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data.

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

"

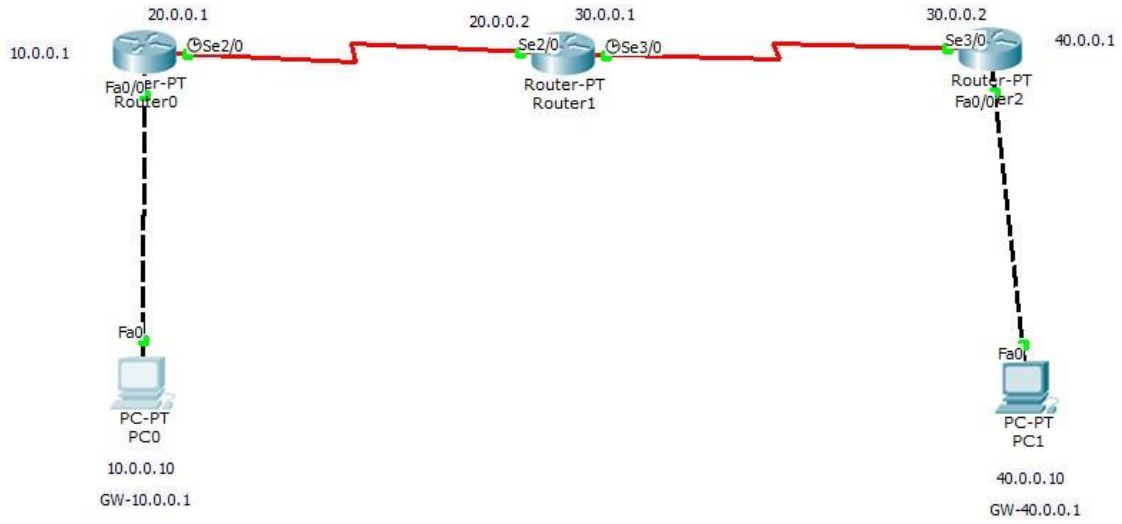
"

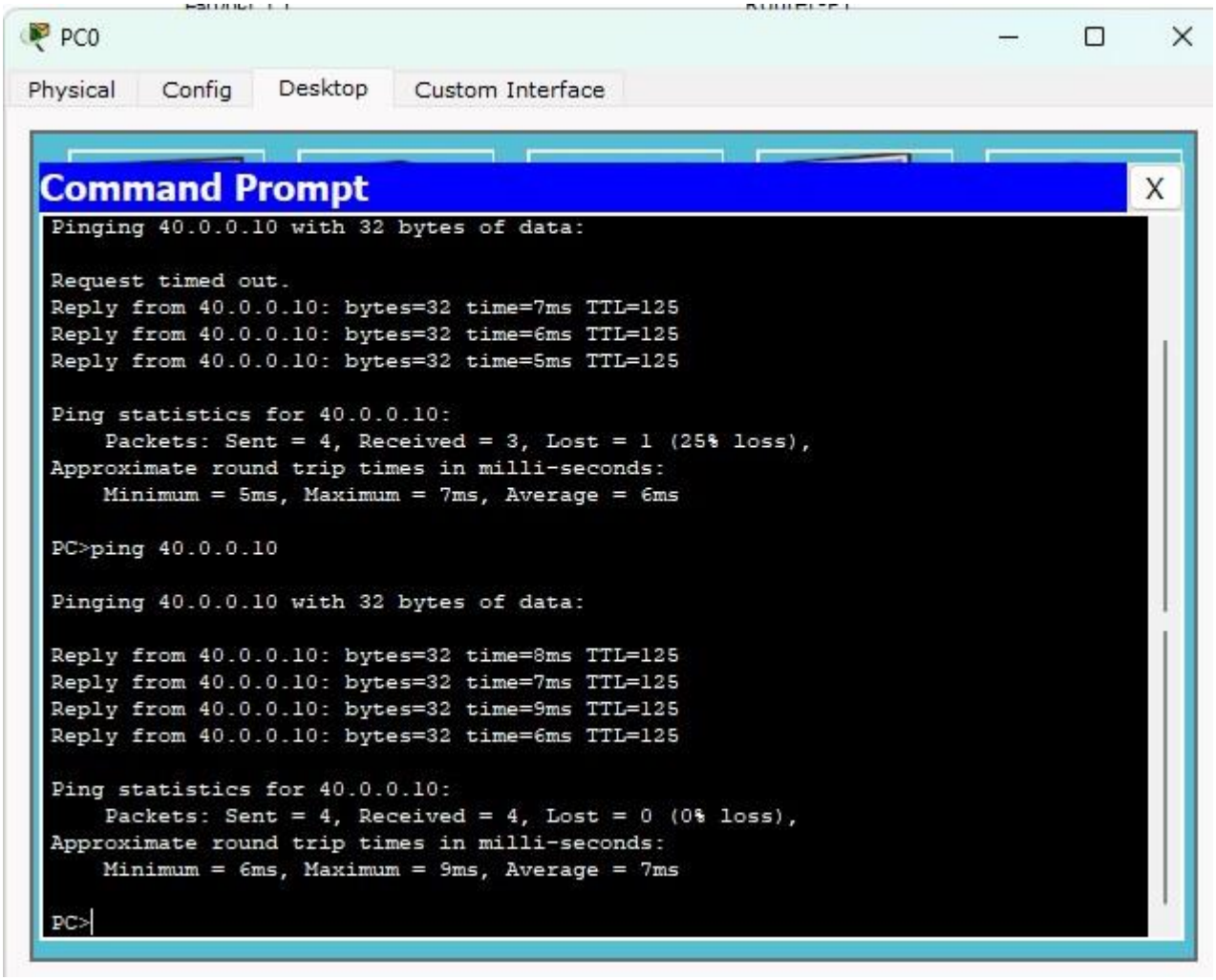
"

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

23/10

Screen Shots:





Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:

13.11.24

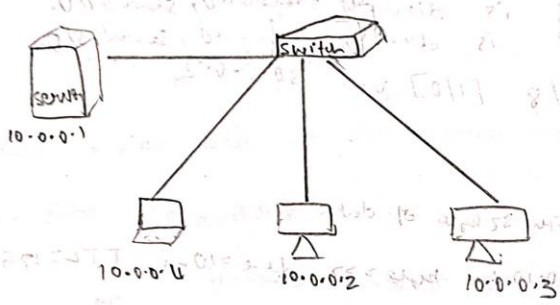
Lab 5

DHCP

Aim: Configure DHCP within a LAN & outside LAN.

⇒ within LAN

Topology:



Procedure:

1. Choose a generic server, a switch, 2 pc and a laptop, connect them to switch using auto cable.

2. Click on server → Desktop → IP config → static, Set IP address as 10.0.0.1 & def gateway as 10.0.0.0

3. Again configure the server PT by config → services → DHCP, service → ON, pool name: switchnet, def gateway 10.0.0.0, max no. of user = 100. Start IP → 10.0.0.3 click Add.

4- For each PC → go to config → ip config ~~DHCP~~, change static to DHCP.

Observation:

1. ip address was allocated dynamically.
2. Data was sent successfully among pc's when pinged.

Output

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

" — "

" — "

" — "

Ping statistics for 10.0.0.4:

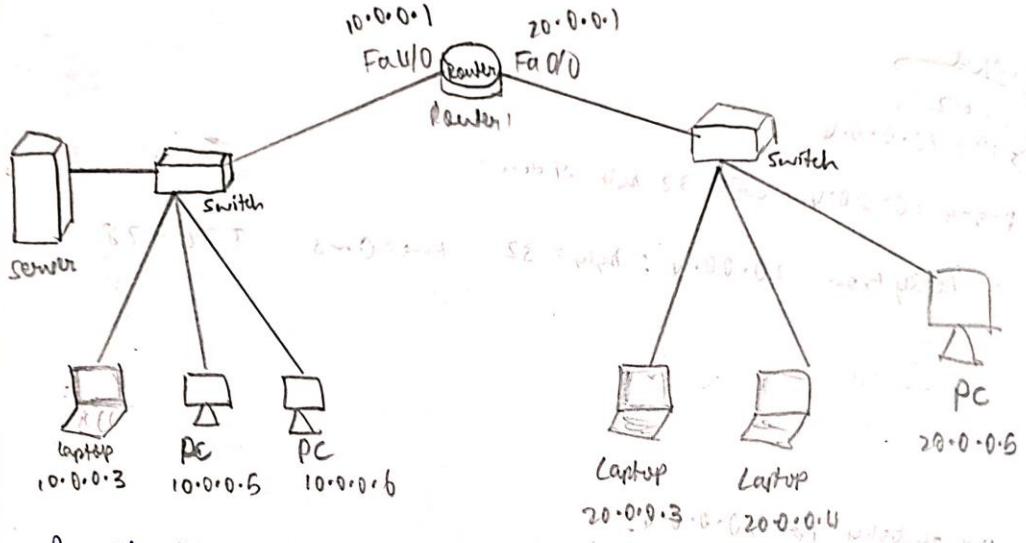
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss).

Approximate round trip times in milli seconds:

Minimum = 0ms, Maximum = 0ms, Avg = 0ms.

II) Outside LAN

Topology



Procedure:

1) for the existing sysm with switch, server, a laptop & 2 PC, add a router & add second network switch. & connect 2 laptops & PC to 2nd switch, as shown.

2) In server. IP config → static → ipaddress → 10.0.0.2
def gtw → 10.0.0.1

3) In server, config → Services → DHCP → modify the existing switch one → defgtw → 10.0.0.1
ip will same 10.0.0.3

4) in server → DHCP → for switch 2,
Pool name: Switch1pool
def gtw → 20.0.0.1
Start ip → 20.0.0.3
add.

5). Do the router configuration.

enable,
config terminal
interface FastEthernet 1/0
ip address 10.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut
exit.

6) Now same for 2nd network as,

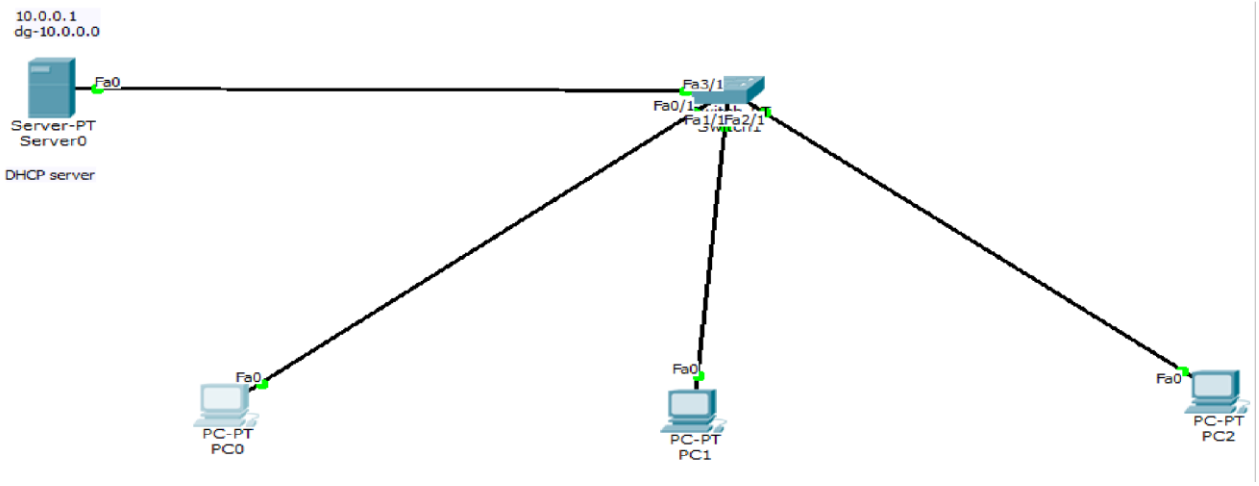
interface FastEthernet 0/0
ip addres 20.0.0.1 255.0.0.0
ip helper-addres 10.0.0.2
no shut
exit.

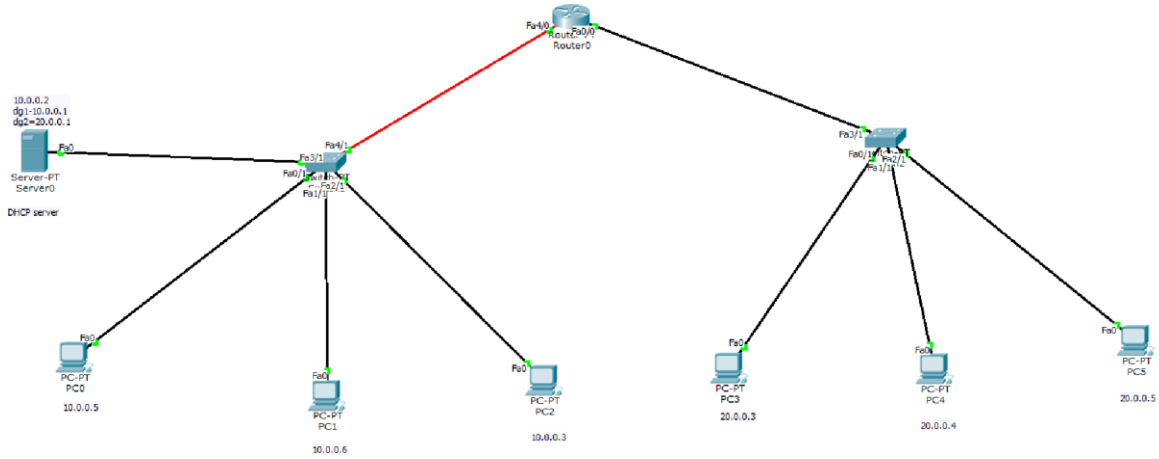
Observation:

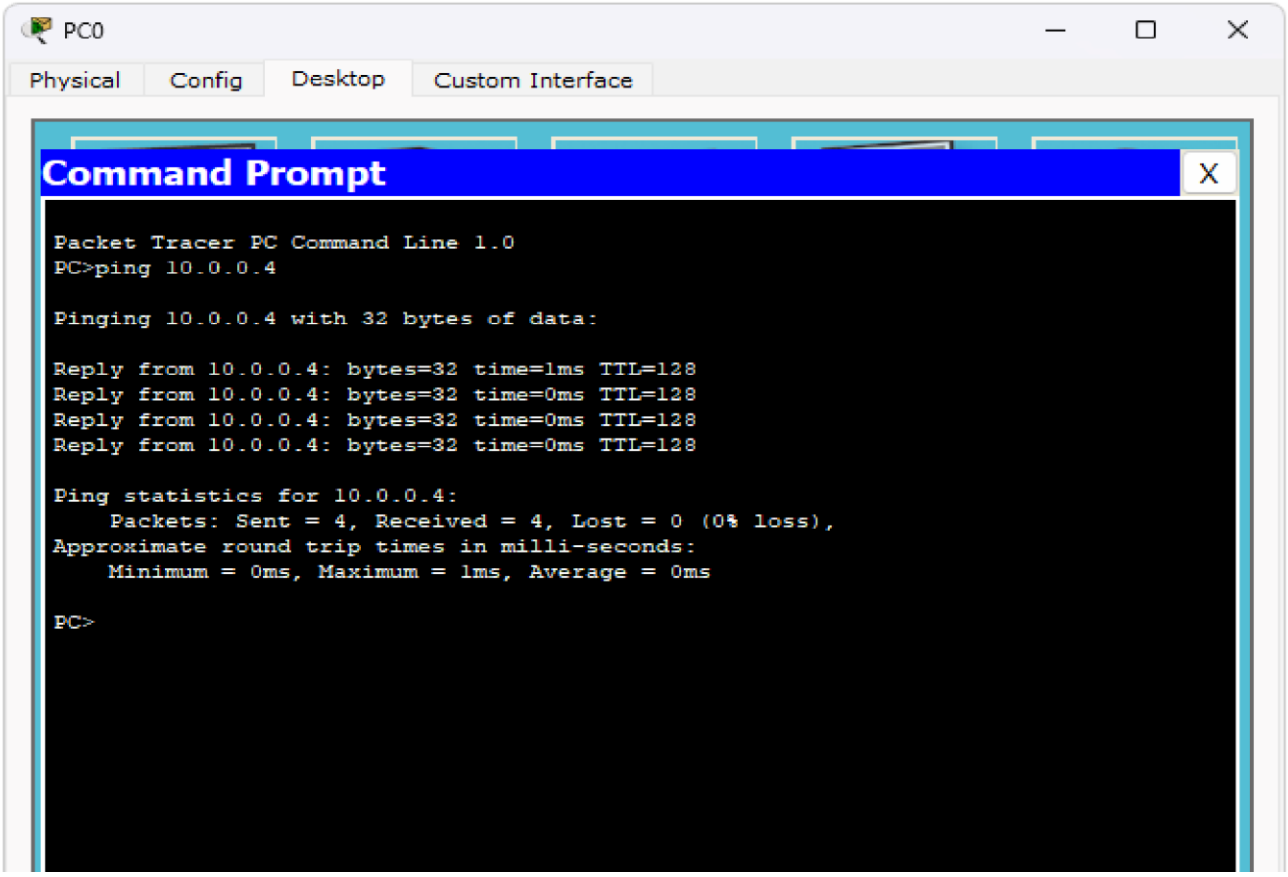
1. IP addresses are allocated dynamically.
2. Data was sent successfully among PC's when pinged

13/11

Screen Shots:



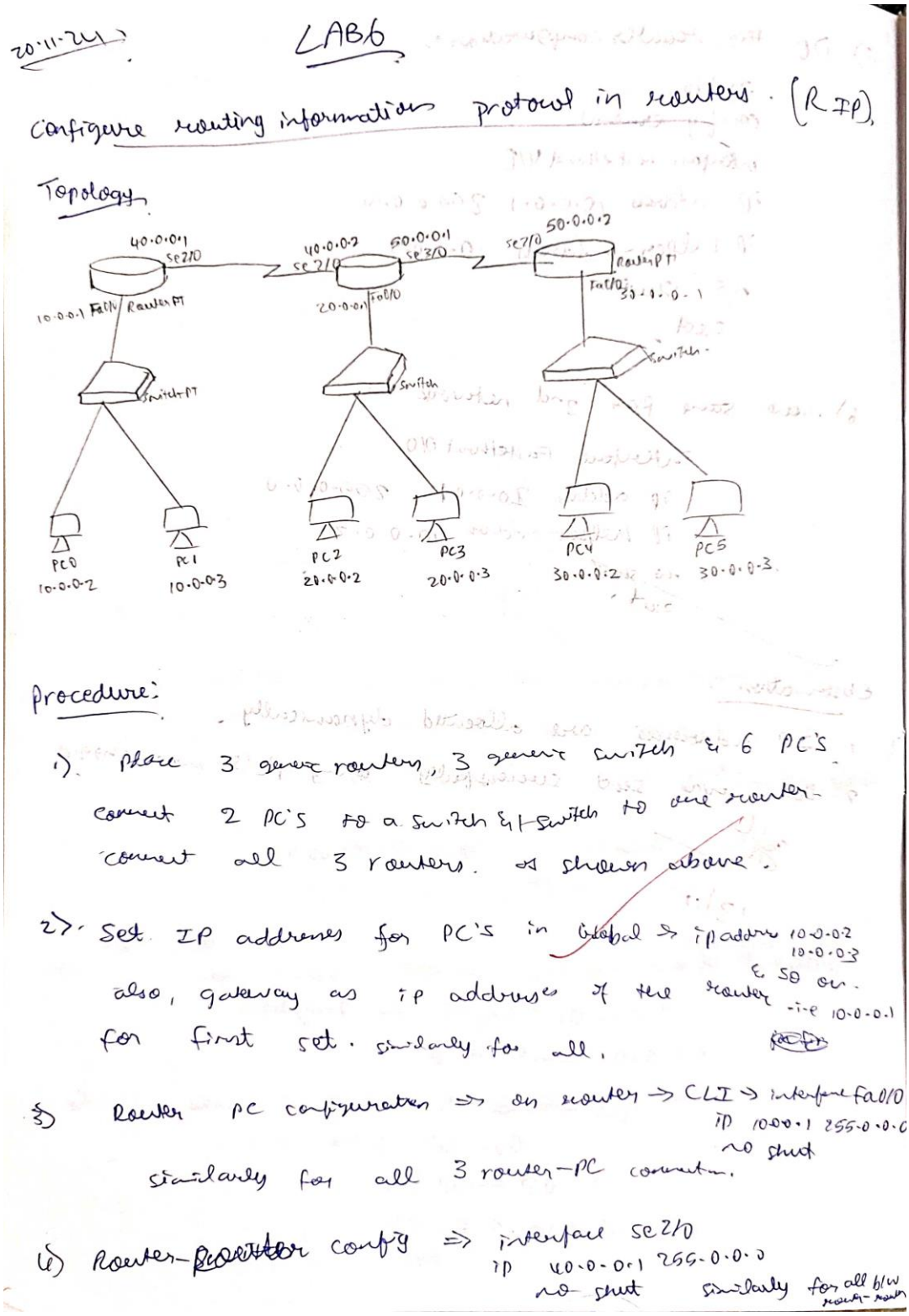



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from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

Program 6

Aim: Configure RIP routing Protocol in Routers .

Topology , Procedure and Observation:



5) Configure routing information protocol to 3 routers

in Router 0

(config)# router rip

(config-router) # network 10.0.0.0

network 40.0.0.0

Similarly for Router1 and Router 2.

Router1 → 40.0.0.0, 50.0.0.0, 20.0.0.0

Router 2 → 50.0.0.0, 30.0.0.0.

Observation:

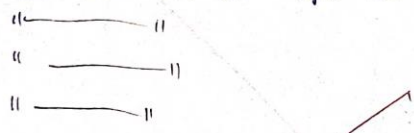
- Before rip, unable to ping → was getting Request Timed Out.
- After configuring routing information protocol in routers, able to ping successfully.

Output:

*) PC> ping 30.0.0.3

Pinging 30.0.0.3 with 32 bytes of data:

Reply from 30.0.0.3: bytes=32 time=2ms TTL=126



Ping statistics for 30.0.0.3:

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

Approximate round trip times in milli seconds:

Minimum=1ms, Maximum=3ms, Avg=2ms.

*) #show ip route → Router3.

R 10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:06, Serial2/0

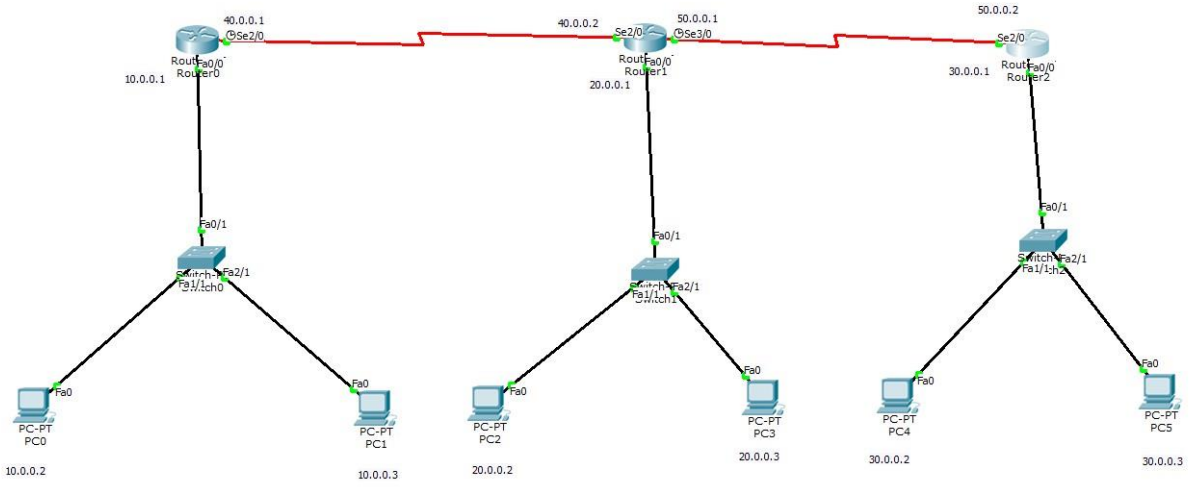
R 20.0.0.0/8 [120/1] via 50.0.0.1 00:00:06, Serial2/0.

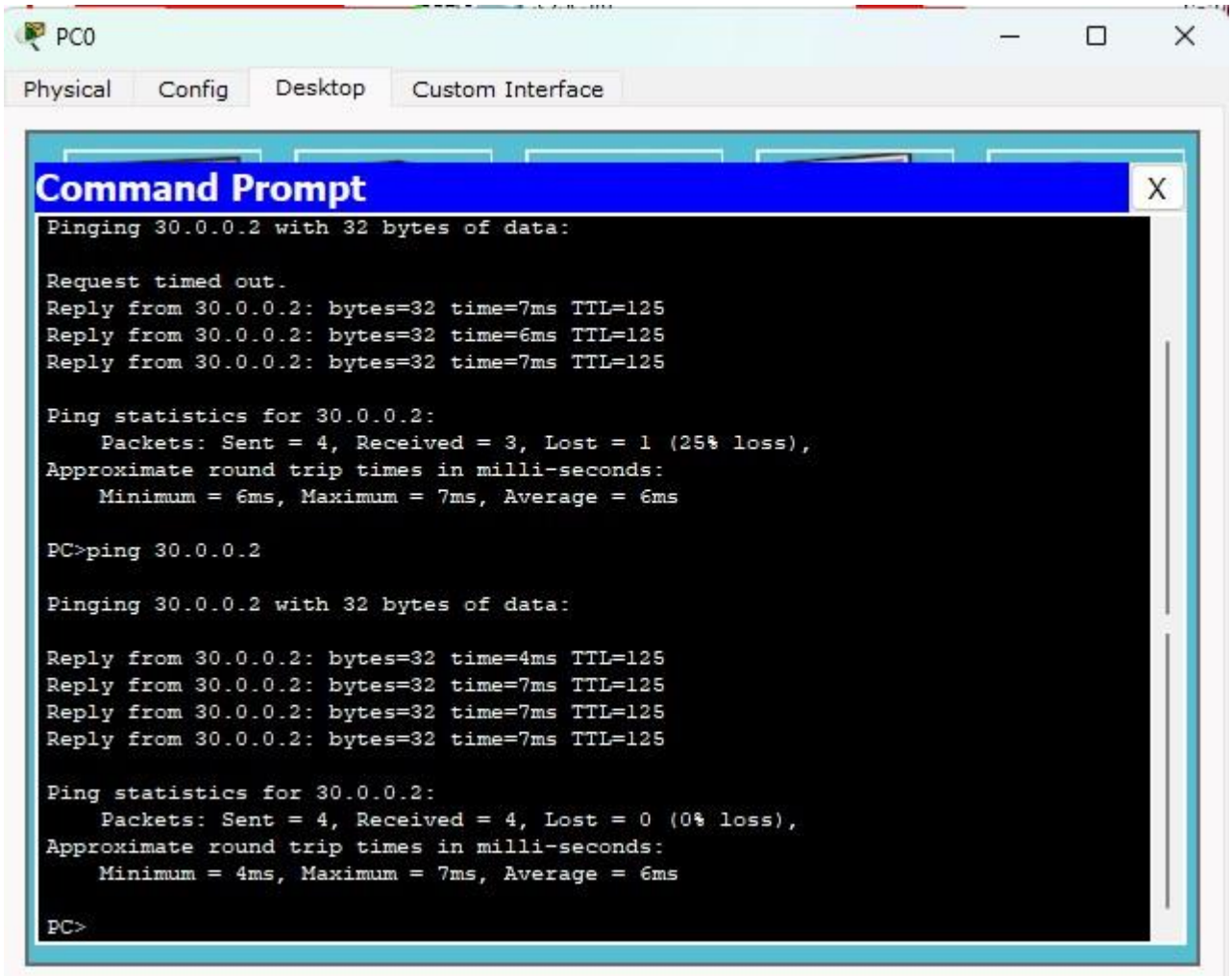
C 50.0.0.0/8 is directly connected. Fa0/0

R 40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:00, Se2/0.

C 50.0.0.0/8 is directly connected, Se2/0

Screen Shots:

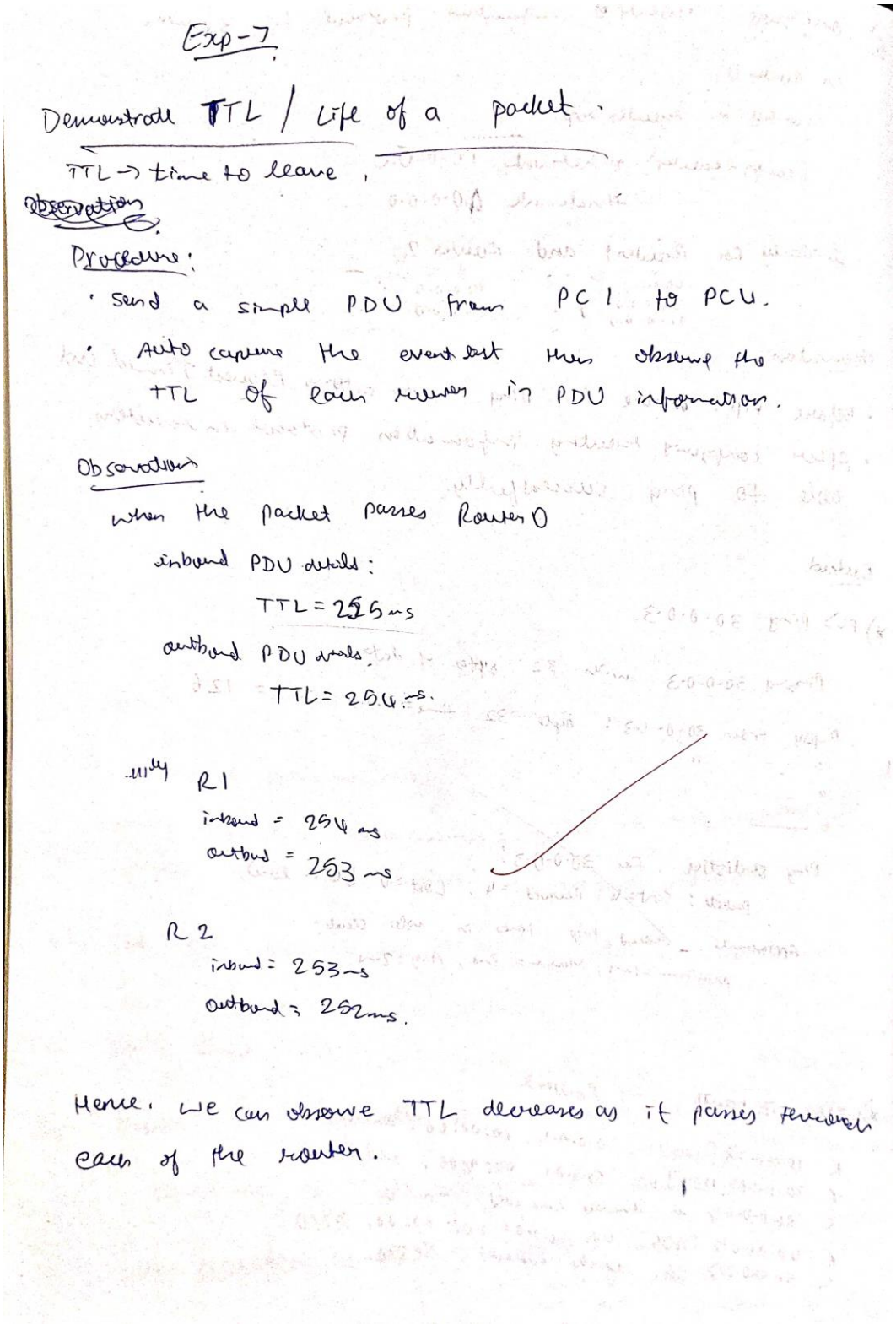




Program 7

Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:



Exp-7

Demonstrate TTL / Life of a packet.

TTL → time to leave.

Observation:

Procedure:

- Send a simple PDU from PC1 to PC4.
- Auto capture the event list then observe the TTL of each router in PDU information.

Observation

When the packet passes Router 0

inbound PDU details:

TTL = 255 ms

outbound PDU details:

TTL = 254 ms.

Then R1

inbound = 254 ms

outbound = 253 ms

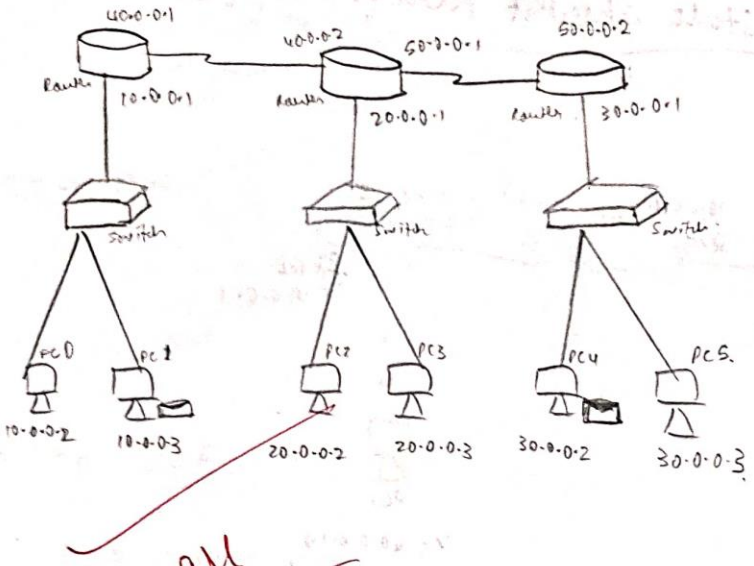
R2

inbound = 253 ms

outbound = 252 ms.

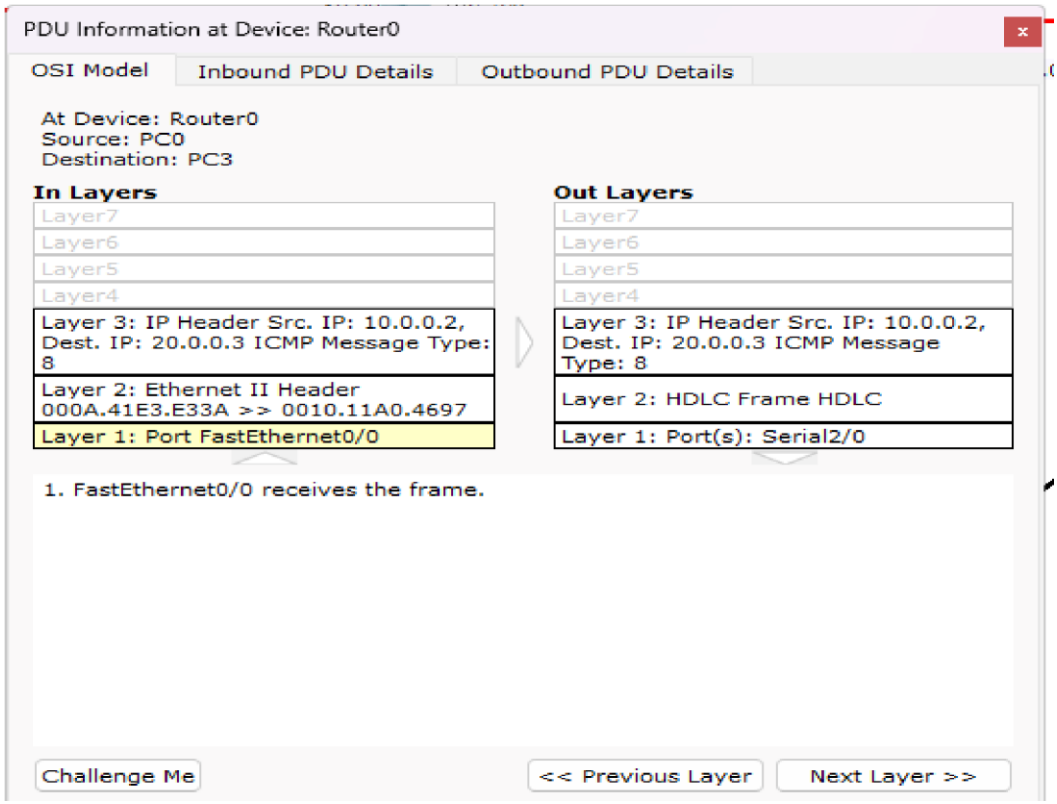
Hence, we can observe TTL decreases as it passes through each of the router.

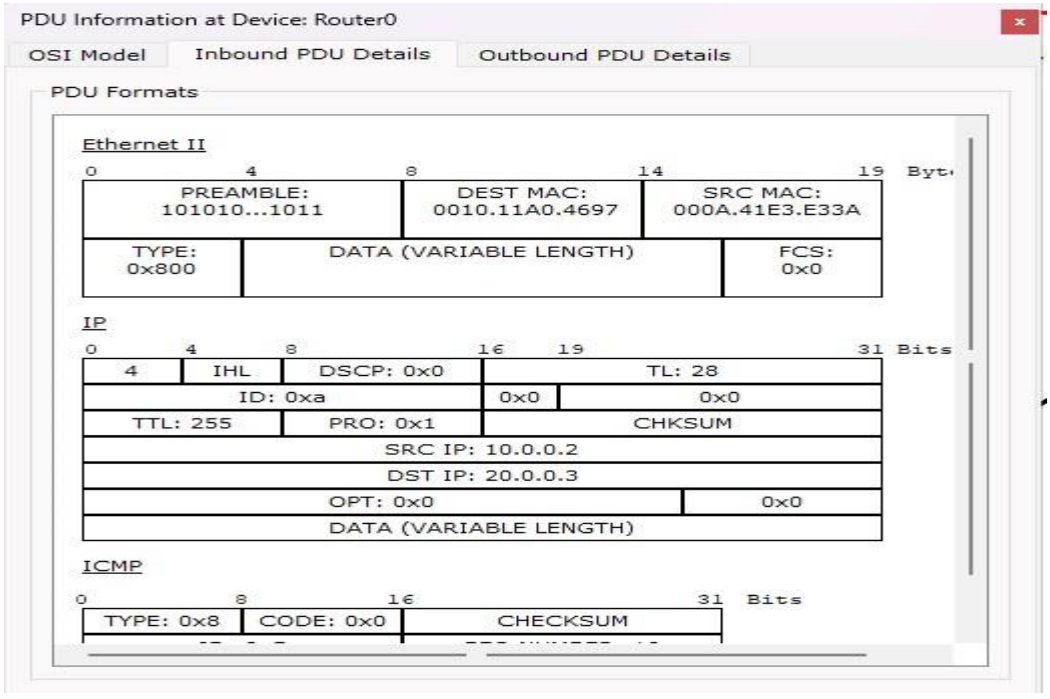
Topology

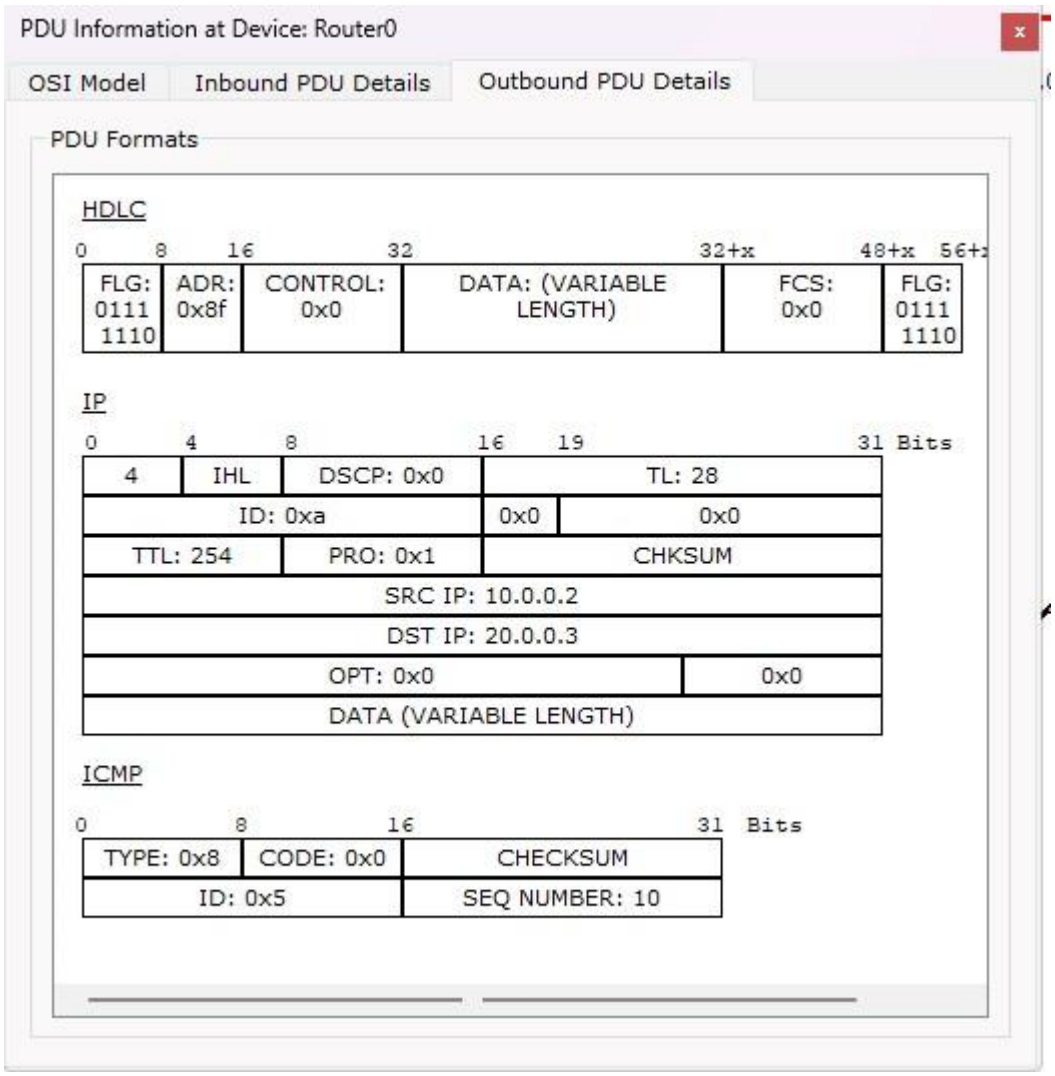


Star
20m

Screen Shots:






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

Program 8

Aim: Configure OSPF routing protocol .

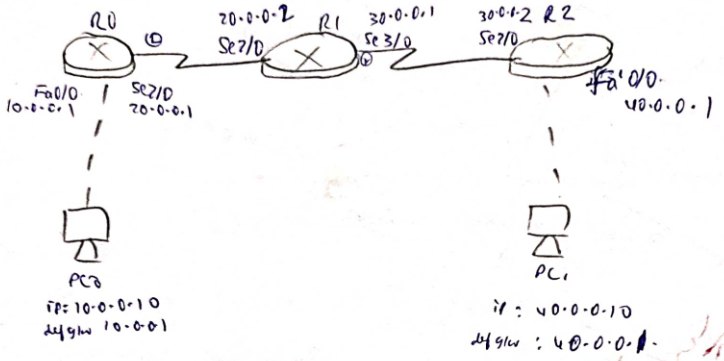
Topology , Procedure and Observation:

27-11-24

Exp 8

Open state Shortest path first. (OSPF)

Topology



Procedure

1) 3 generic router & 2PC's & configure them as shown in above topology

2). Configure PC & router using command prompt.

interface Fa0/0
ip address 10.0.0.1 255.0.0.0 } for R0.
no shut

similarly R2.

3) For Router - Router configuration

R0 → interface Serial 2/0
ip address 20.0.0.1 255.0.0.0
encapsulation PPP
clock rate 64000 [clock on which side for that only]
no shut
exit

R1 → interface Serial 2/0
ip address 20.0.0.2 255.0.0.0
encapsulation PPP
no shut
exit

R1 ⇒ serial 3/0
ip address 30.0.0.1 255.0.0.0
encapsulation PPP
clock rate 64000
no shut
exit.

R2 ⇒ serial 2/0
ip addr 30.0.0.2 255.0.0.0
encapsulation PPP
no shut
exit.

4) DO OSPF.

R0 ⇒ router OSPF 1
router-id 1.1.1.1
network 10.0.0.0 0.255.255.255 area 3
network 20.0.0.0 0.255.255.255 area 1
exit.

R1 ⇒ router ospf 1
router-id 2.2.2.2
network 20.0.0.0 0.255.255.255 area 1
network 30.0.0.0 0.255.255.255 area 0
exit.

R2 ⇒ router OSPF 2
router-id 3.3.3.3
network 30.0.0.0 0.255.255.255 area 0
network 40.0.0.0 0.255.255.255 area 2
exit.

5) Loopback

R0 ⇒ interface loopback 0
ip add 172.16.1.252 255.255.0.0
no shut.

R1 ⇒ interface loopback 0
ip add 172.16.1.253 255.255.0.0

R2 > interface loopback0

ip add 172.16.1.254 255.255.0.0

no shut.

6). Virtual link between R1 & R2.

R0> router ospf 1

area 1 virtual-link 2.2.2.2

R1> router ospf 1

area 1 virtual-link 1.1.1.1

exit.

Observation:

When pinged, message was successfully sent from PC 0 to PC 1.

Output:

ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes = 32 time = 0ms TTL = 126

" ——— "

" ——— "

" ——— "

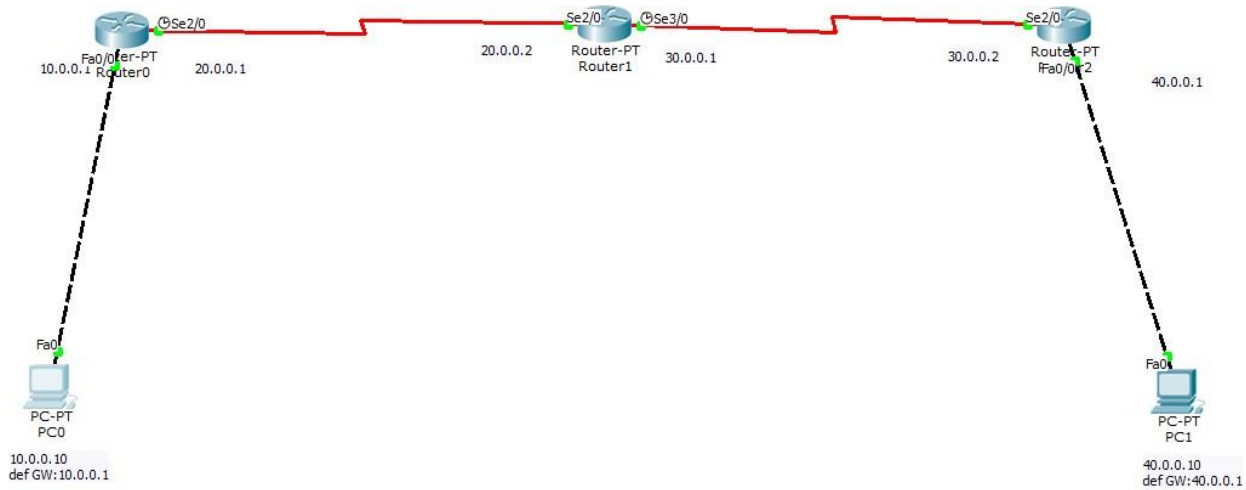
Ping statistics for 40.0.0.10:

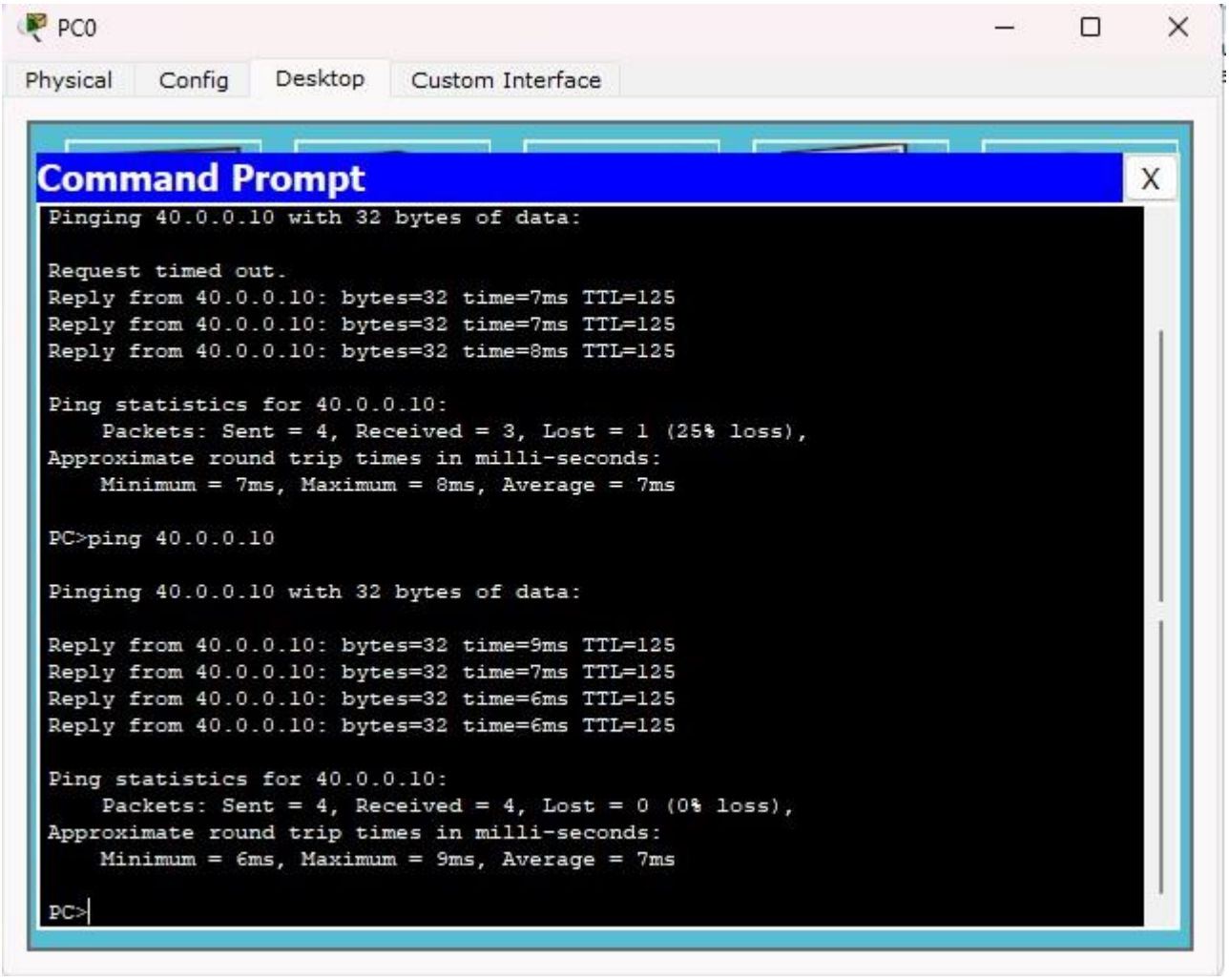
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

Show ip route : [for R2]

O IA 10.0.0.0/8 [110/174] via 30.0.0.1, 00:28:26 serial0/0
O IA 20.0.0.0/8 [110/178] via 30.0.0.1, 00:02:22, serial0/0.
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks.
C 30.0.0.0/8 is directly connected, serial0/0
C 30.0.0.1/32 is directly connected, serial0/0
C 40.0.0.0/8 is directly connected, Fa 0/0
C 172.16.0.0/16 is directly connected, Loopback 0

Screen Shots:





Program 9

Aim: Configure Web Server, DNS within a LAN.

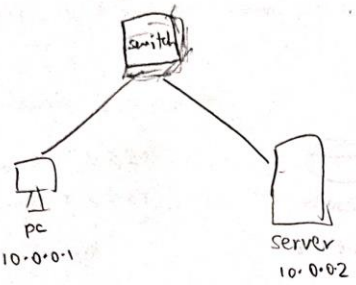
Topology , Procedure and Observation:

18/12/24

Lab No 11.

DNS

Topology



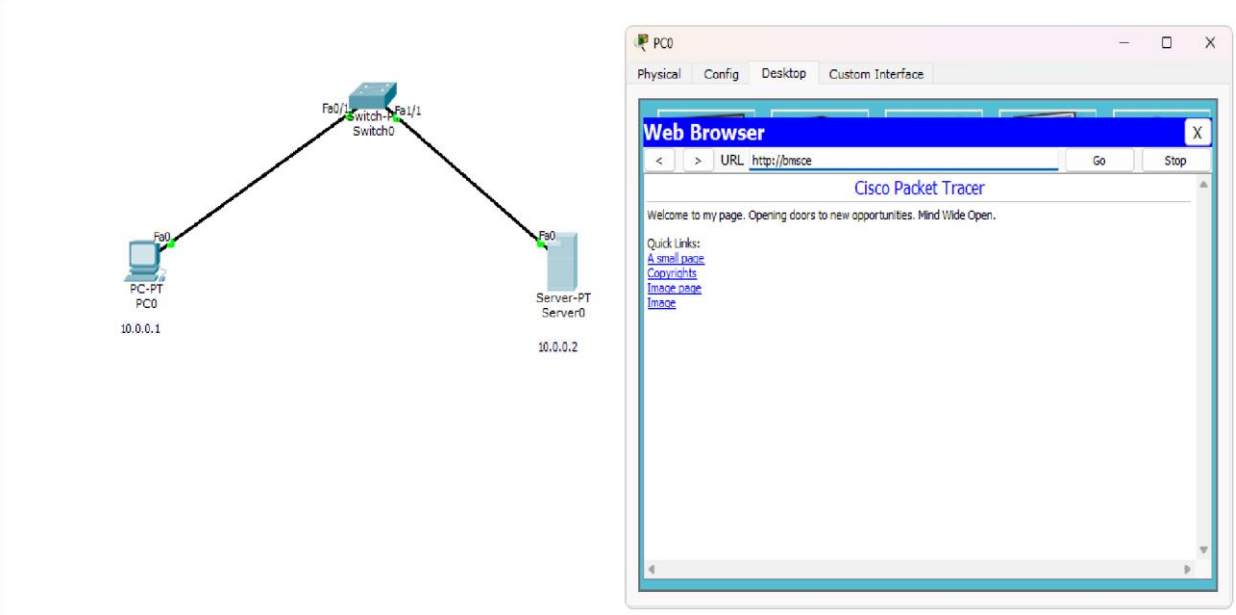
Procedure

1. Set up LAN as per topology as above & configure device.
2. Go to Server → Service → DNS:
   Name: bmsce
   Address: 10.0.0.2
   Add the mapping of domain name to address.
3. Go to PC → Config → ~~global~~ → settings → DNS server: 10.0.0.2
4. Go to PC → Desktop → web browser:
   type the url: http://bmsce.

Observations:

1. The webpages hosted by the server were visible on the house.
2. The DNS was successful in mapping the domain name to ip addr.
3. DNS server is a server that contains a domain name: ip address mapping to which the end device send request to map the name to IP address.

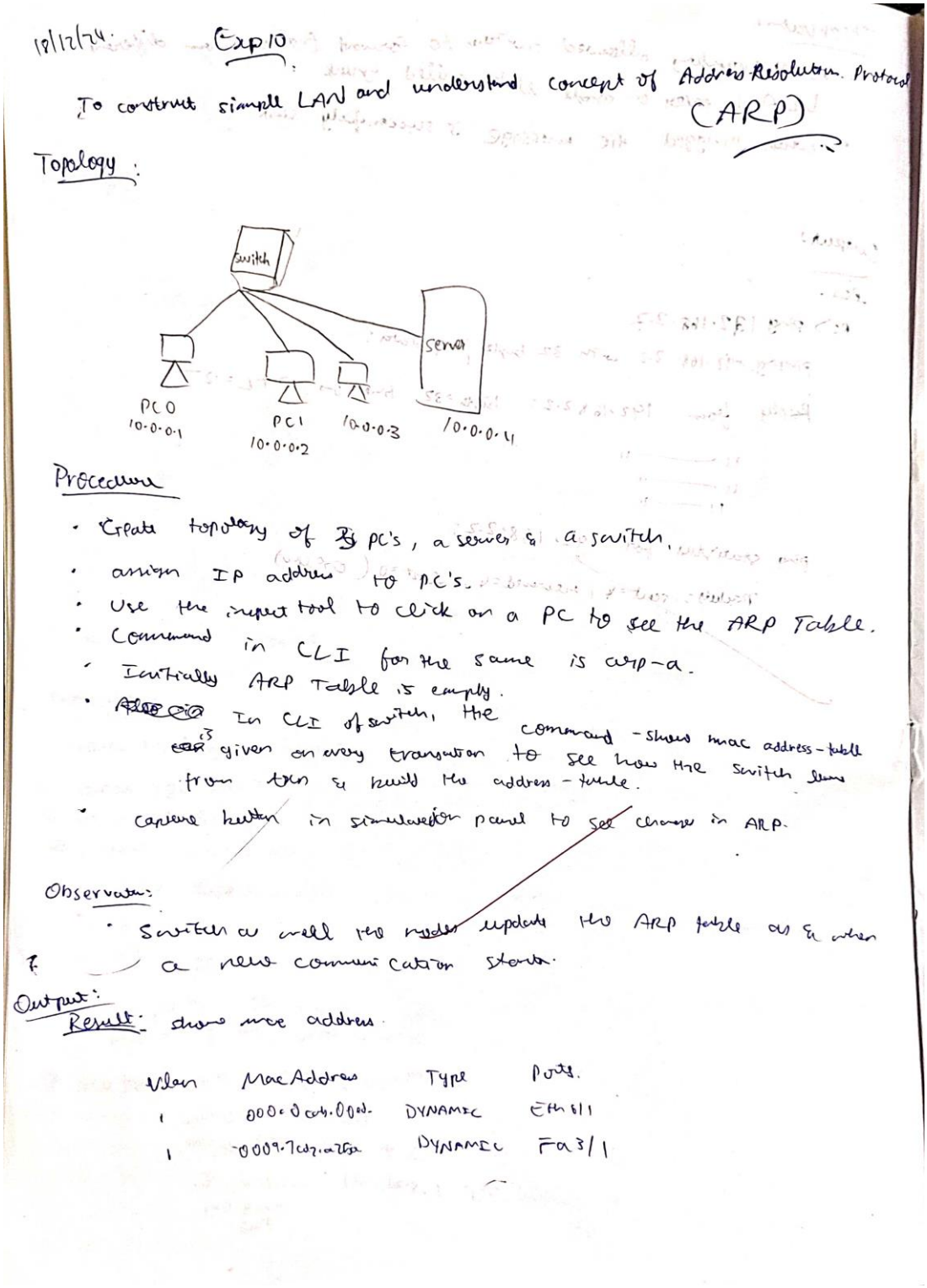
Screen Shots:



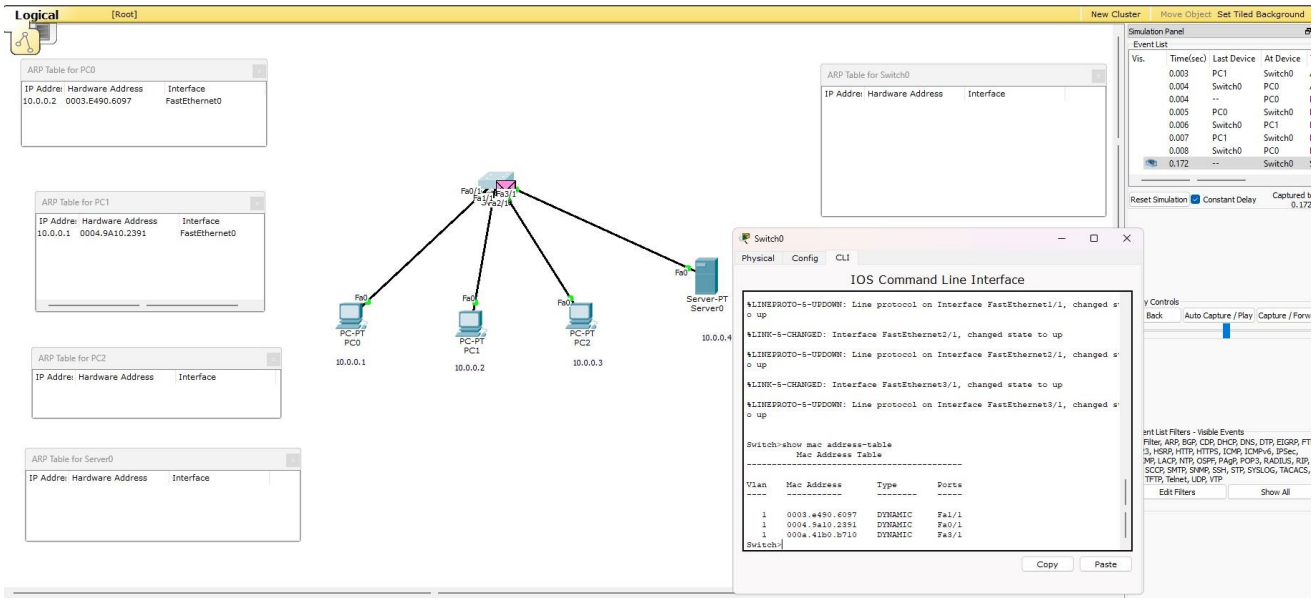
Program 10

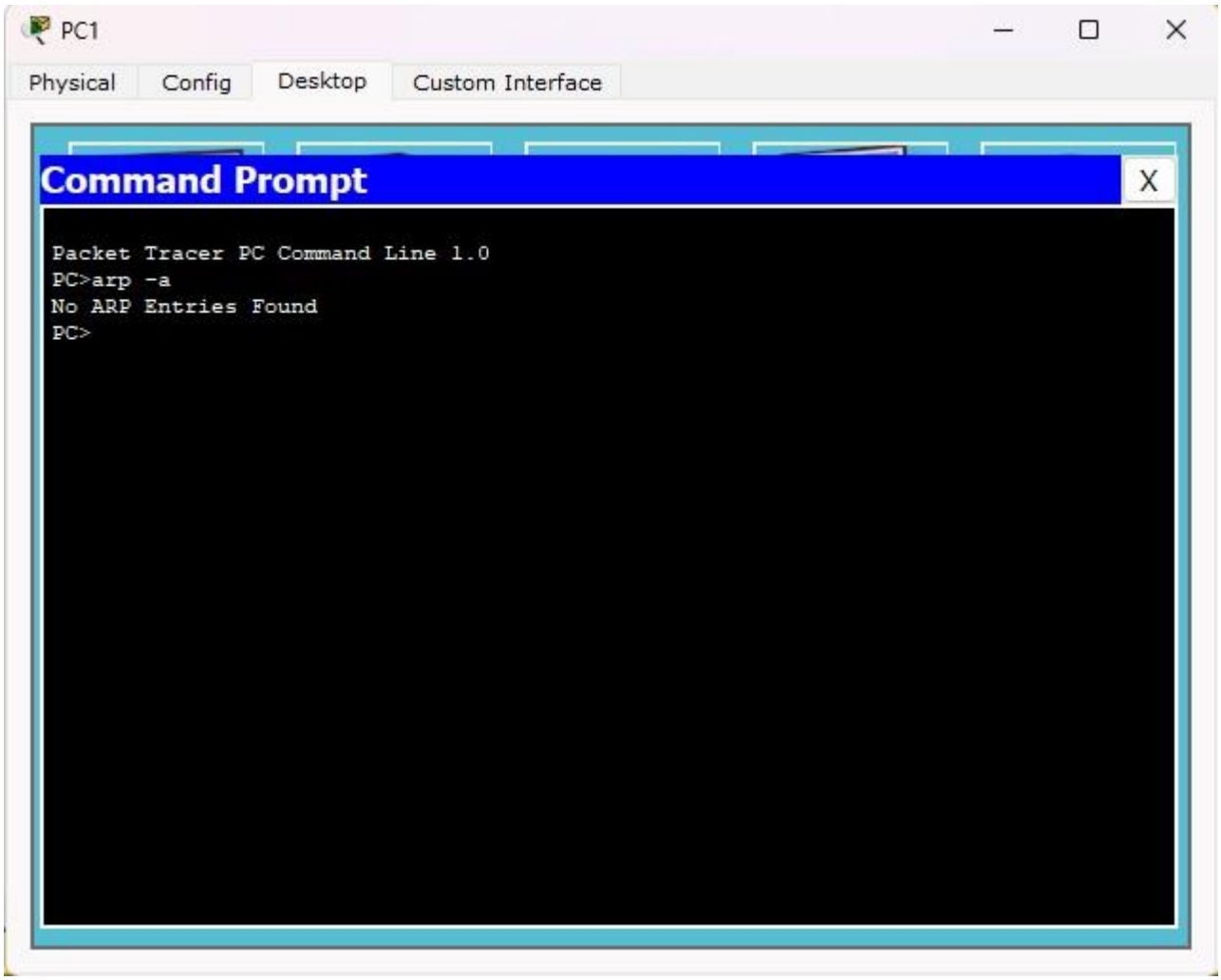
Aim: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Topology , Procedure and Observation:



Screen Shots:




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

Program 11

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

Topology , Procedure and Observation:

Lab 12

TELNET

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 above & configure the devices.
2. Commands in Router:

```
Router > enable.
Router # config terminal.
Router (config) # hostname R1
R1 (config) #      enable secret 1234
   "  —  "        interface fastethernet 0/0
R1 (config-if) # ip address 10.0.0.2 255.0.0.0
                 no shut
R1 (config-if) # line vty 0 3
               # login
```

% login disabled on line 194, until 'password' is set.

```
R1 (config-line) # password 4321
R1 (config-line) # exit.
R1 (config) # exit.
R1 # wr
Building configuration...
[OK]
```

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 : U321

password : U321

R1 > enable

password : 1234

R1 # show ip route.

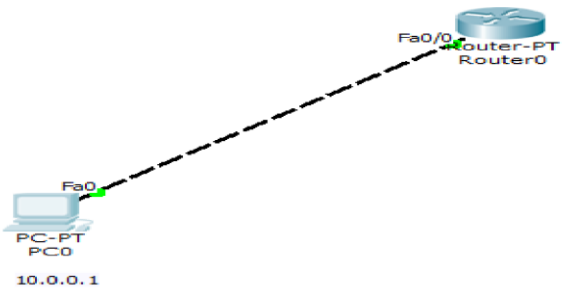
C 10.0.0.0/8 is directly connected, Fast Ethernet 0/0.

R1 #.

Observations:

1. The admin in PC is able to run commands as seen in router CLI and see the results from PC.
2. Telnet allows users to establish a remote sessions with another device like router, over a TCP/IP network.
3. Using Telnet, we can access and control the remote devices CLI as if you were physically connected to it.

Screen Shots:



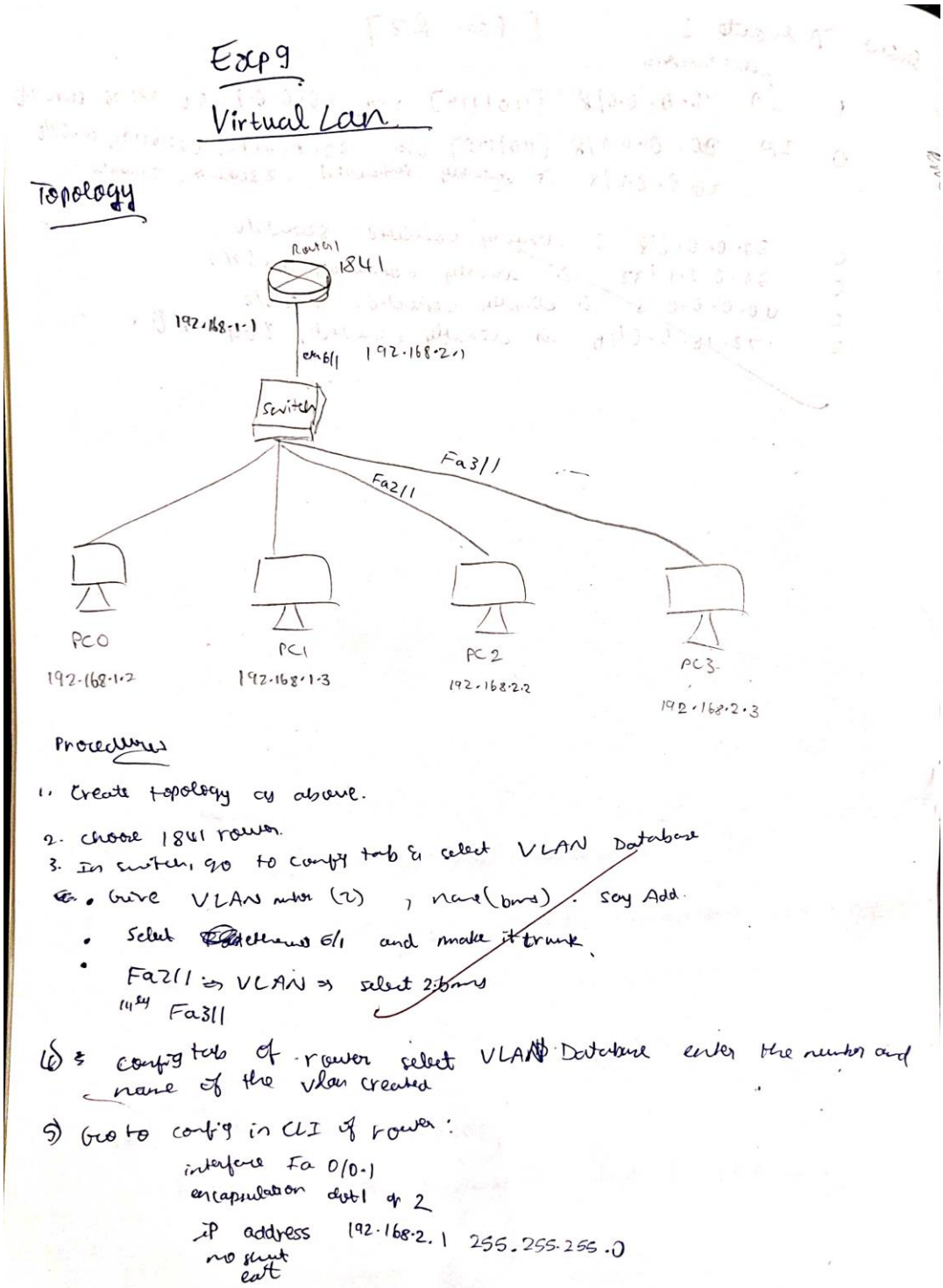
Command Prompt

```
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#
```

Program 12

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

Topology , Procedure and Observation:



Observation:

- VLAN trunking allowed switches to forward frames from different VLANs over a single link called trunk.
- when pinged the message is successfully sent

Output:

PC1:

PC> ping 192.168.2.2

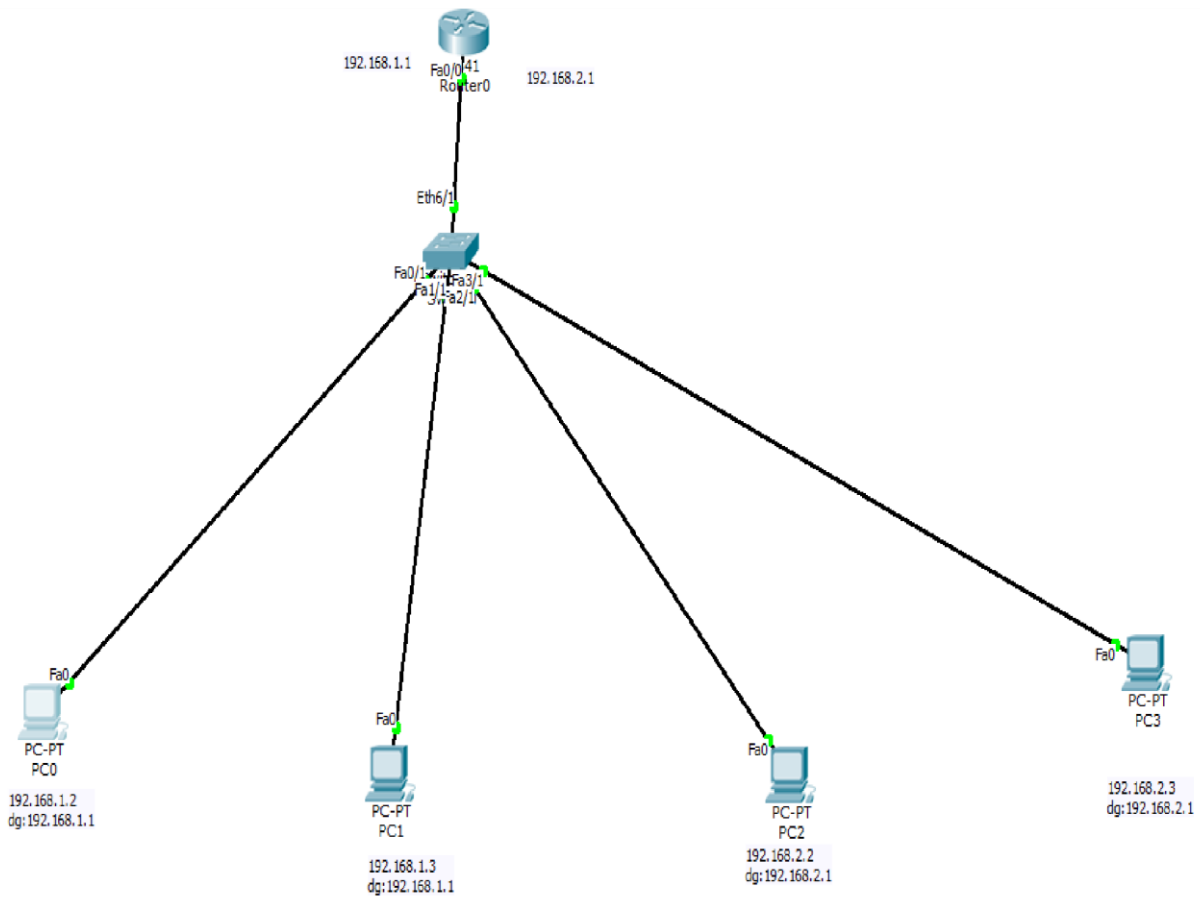
pinging 192.168.2.2 with 32 bytes of data:

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

ping statistics for 192.168.2.2:

Packets: sent=4, received=4, lost=0 (0% loss)

Screen Shots:



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>
```

Program 13

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

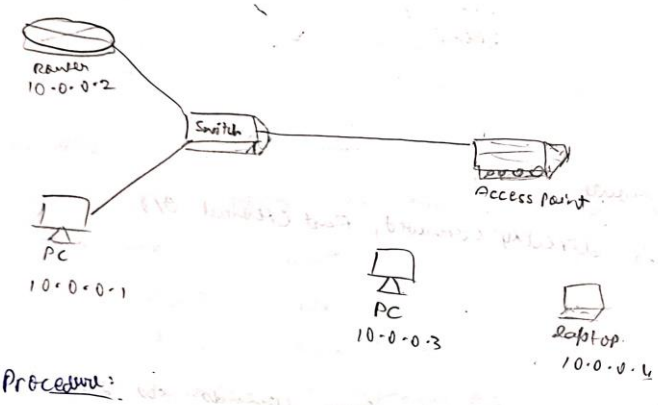
Topology , Procedure and Observation:

18.12.24 Lab 13

WLAN

Aim: To construct a wireless LAN and make the nodes communicate wirelessly.

Topology:

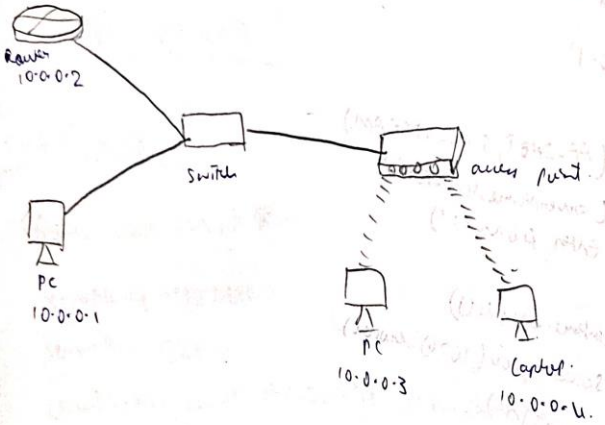


Procedure:

1. Create topology as given above & configure the devices.
2. Configure access point:
   Click access point → config → port1:
   SSID: bmsce
   Select ⊙ WEP
   Set key : 12 34 56 78 90.
3. Configure PC & laptop with wireless standards:
   - switch off device.
   - Drag the existing PT-HOST-NM-1 AM to the Components listed in the LHS of physical.
   - Drag WMP300N wireless interface to the empty port.
   - switch on the device.
4. In config tab, a new wireless interface was added.

5. Configure the device by entering SSID, WEP, WEP key, IP address & gateway.

Topology after wireless configuration:



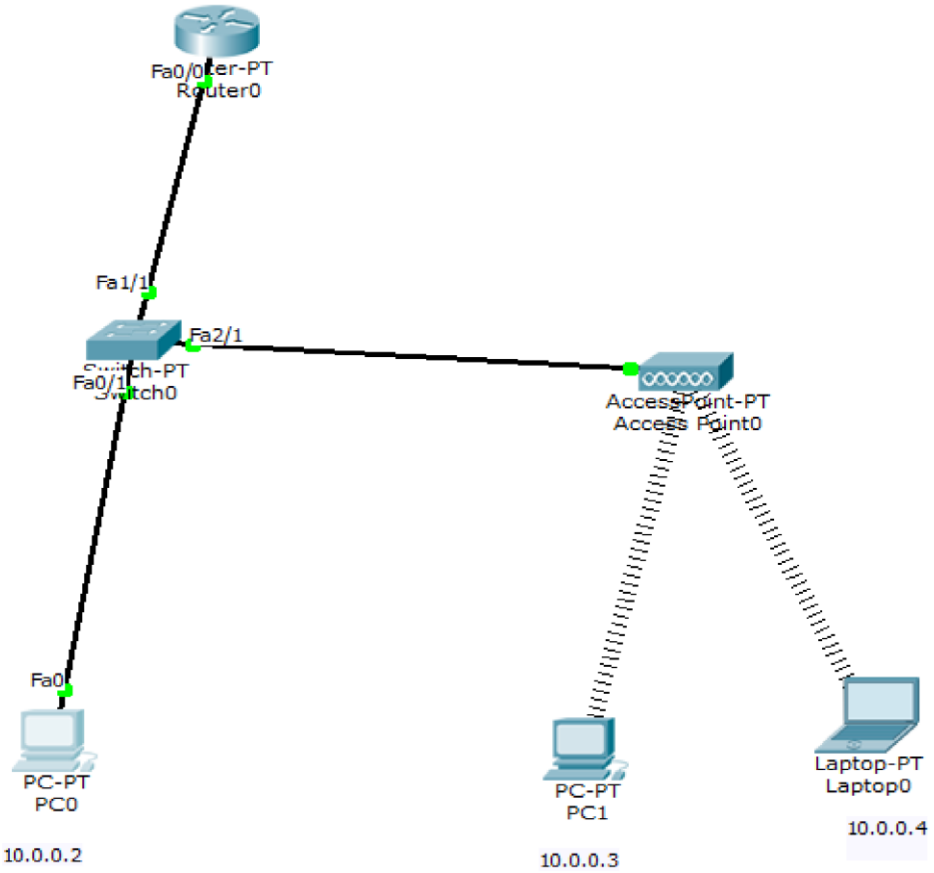
6. Ping from every device to every other device to check for connection.

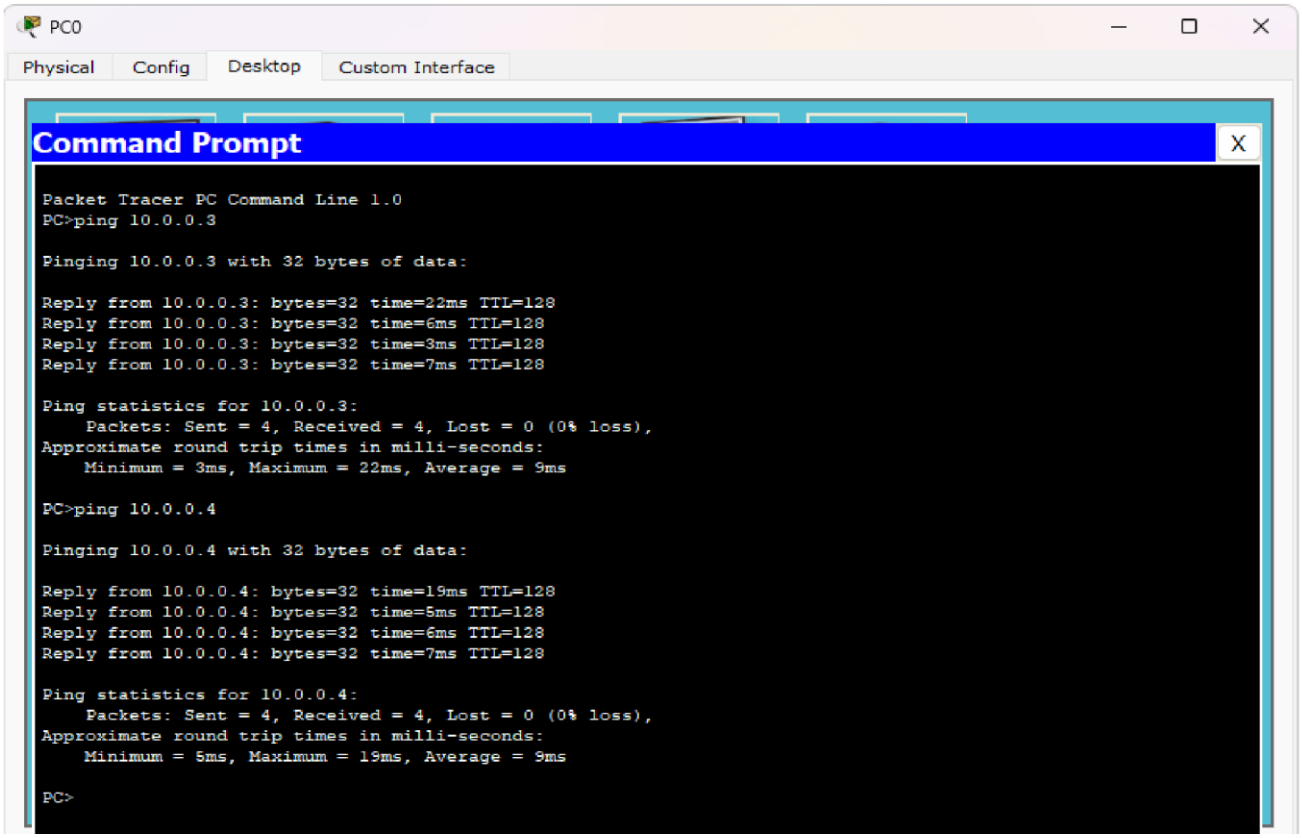
Observation:

1. We were able to ping from every device to every other device.
2. Every device is now connected to every other device in the WLAN.

26/12

Screen Shots:



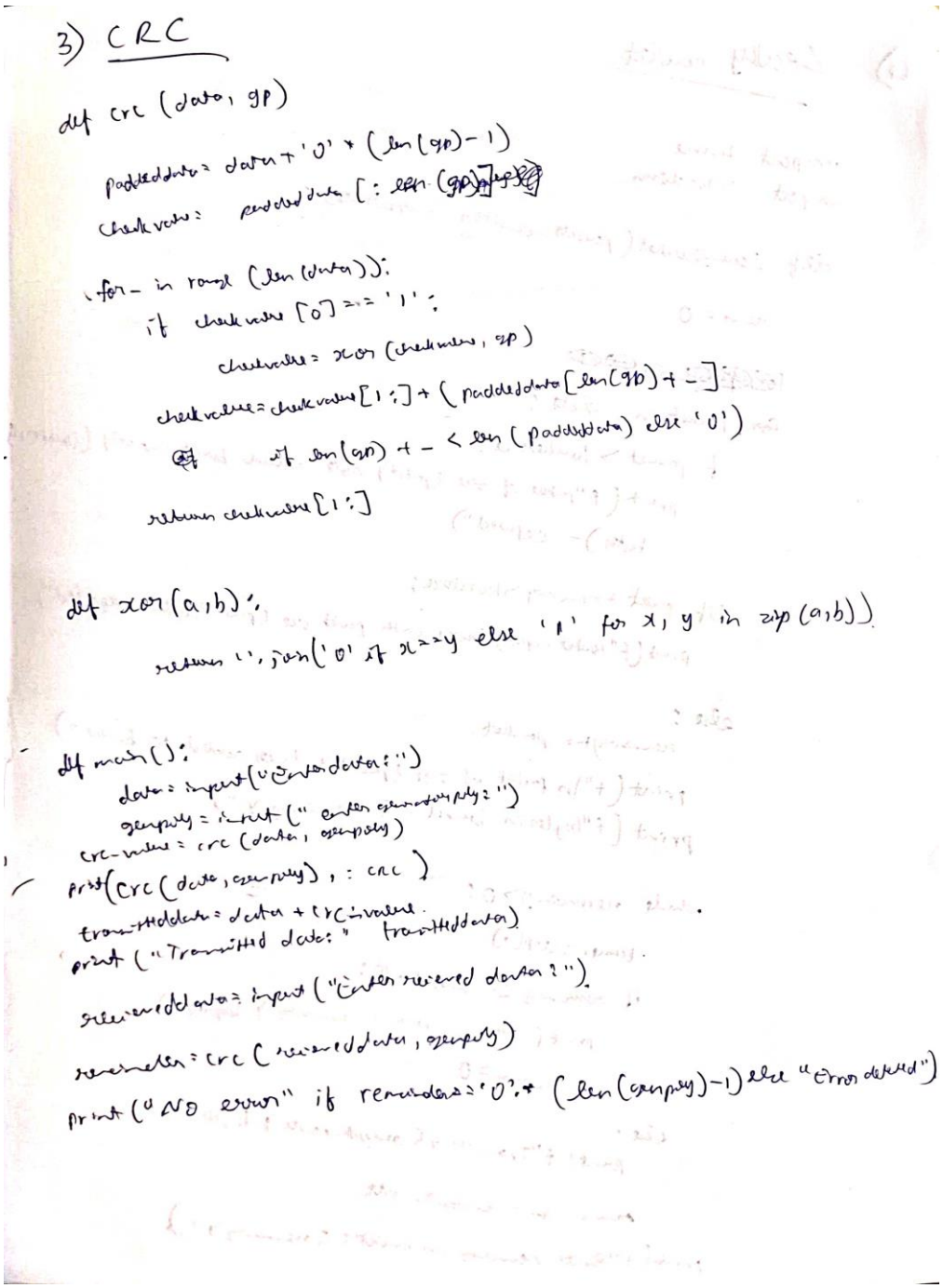


PART-B

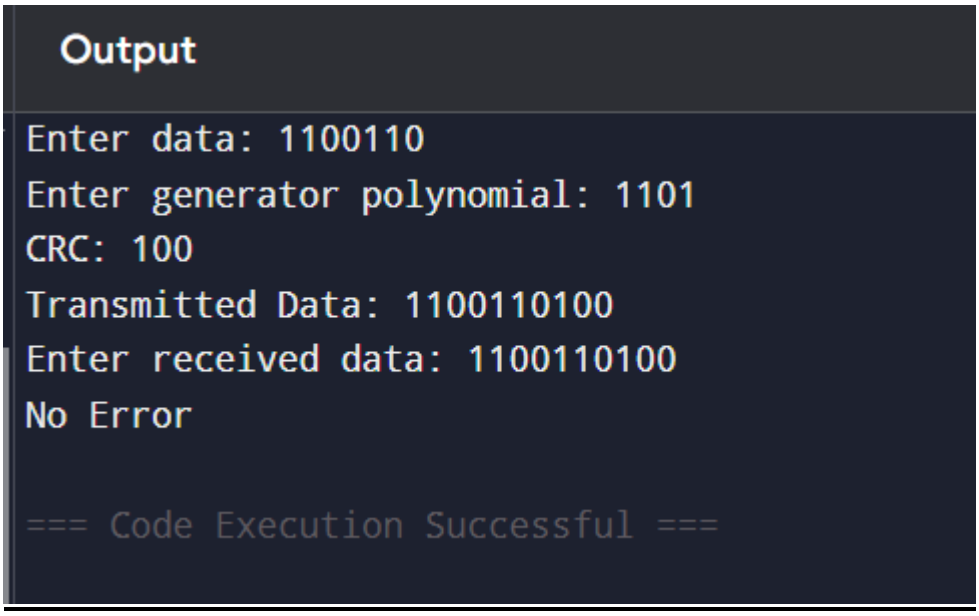
Program 14

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

Code :



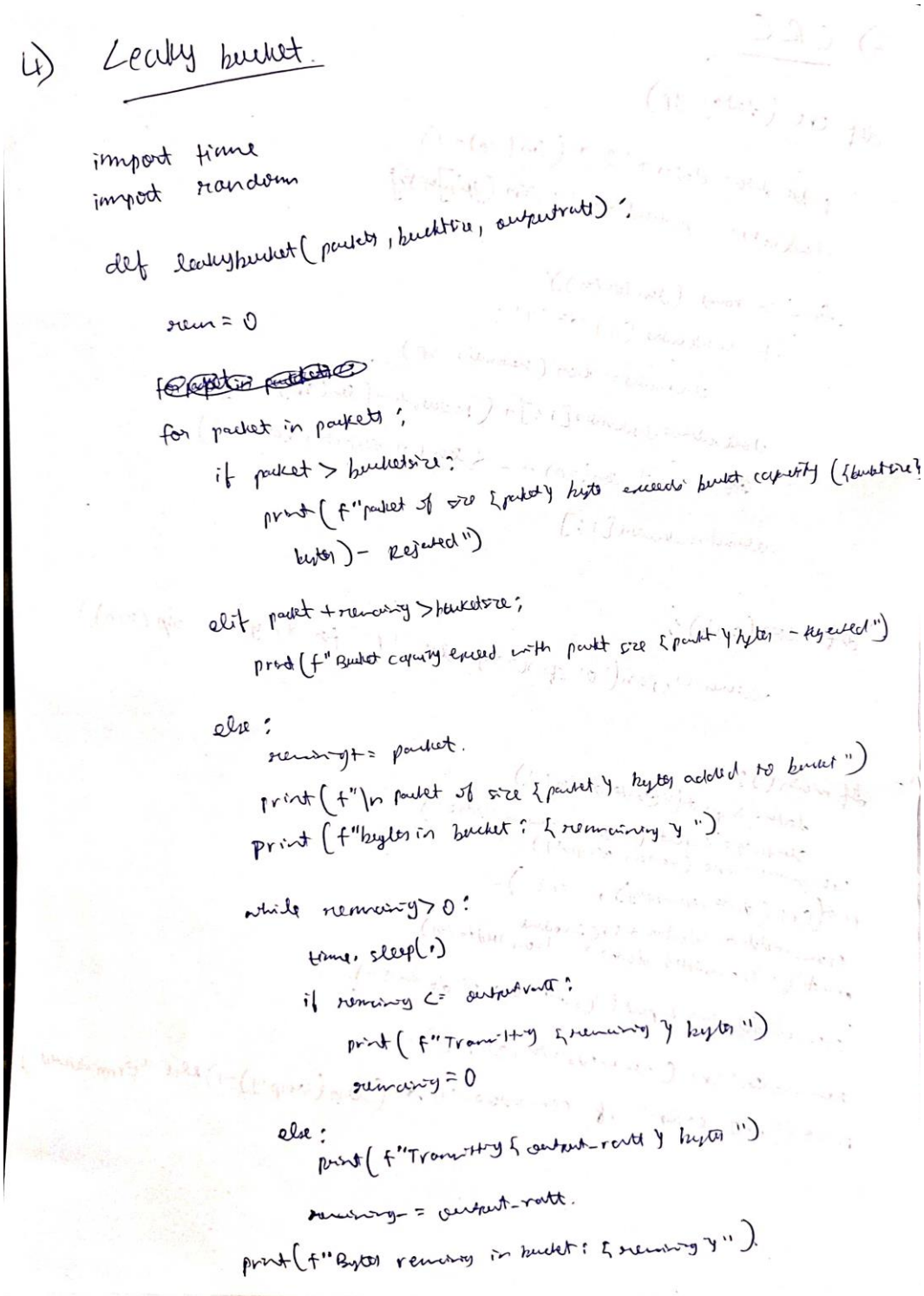
```
3) CRC

def crc (data, gp)
    paddeddata = data + '0' * (len(gp)-1)
    checkvalue = paddeddata [: len(gp)]

    for _ in range (len(data)):
        if checkvalue[0] == '1':
            checkvalue = xor(checkvalue, gp)
        checkvalue = checkvalue[1:] + (paddeddata[len(gp)+_]
            if len(gp)+_ < len(paddeddata) else '0')
    return checkvalue[1:]

def xor(a,b):
    return ''.join('0' if x==y else '1' for x, y in zip(a,b))

def main():
    data = input("Enter data: ")
    gpoly = input("enter generator poly: ")
    crc_value = crc(data, gpoly)
    print(crc(data, gpoly), : crc)
    transmitteddata = data + crc_value
    print("Transmitted data: " transmitteddata)
    receiveddata = input("Enter received data? ")
    remainder = crc(receiveddata, gpoly)
    print("NO error" if remainder == '0'* (len(gpoly)-1) else "Error detected")
```


Output
Enter data: 1100110
Enter generator polynomial: 1101
CRC: 100
Transmitted Data: 1100110100
Enter received data: 1100110100
No Error
=== Code Execution Successful ===

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code :



```
4) Leaky bucket

import time
import random

def leakybucket(packets, bucketsize, outputrate):
    rem = 0
    for packet in packets:
        if packet > bucketsize:
            print(f"packet of size {packet} bytes exceeds bucket capacity ({bucketsize}
              bytes) - Rejected")
        elif packet + remaining > bucketsize:
            print(f"Bucket capacity exceed with packt size {packet} bytes - Rejected")
        else:
            remaining += packet.
            print(f"\n packet of size {packet} bytes added to bucket")
            print(f"bytes in bucket : {remaining} ")
            while remaining > 0:
                time. sleep(1)
                if remaining <= outputrate:
                    print(f"Transmitting {remaining} bytes")
                    remaining = 0
                else:
                    print(f"Transmitting {outputrate} bytes")
                    remaining -= outputrate.
            print(f"Bytes remaining in bucket: {remaining}").
```

```
def main():
    packets = [random.randint(1,100) for _ in range(5)]
    print(f"Generated packets: {packets}")

    bucket_size = int(input("Enter bucket size: "))
    output_rate = int(input("Enter output rate: "))

    leaky_bucket(packets, bucket_size, output_rate)
```

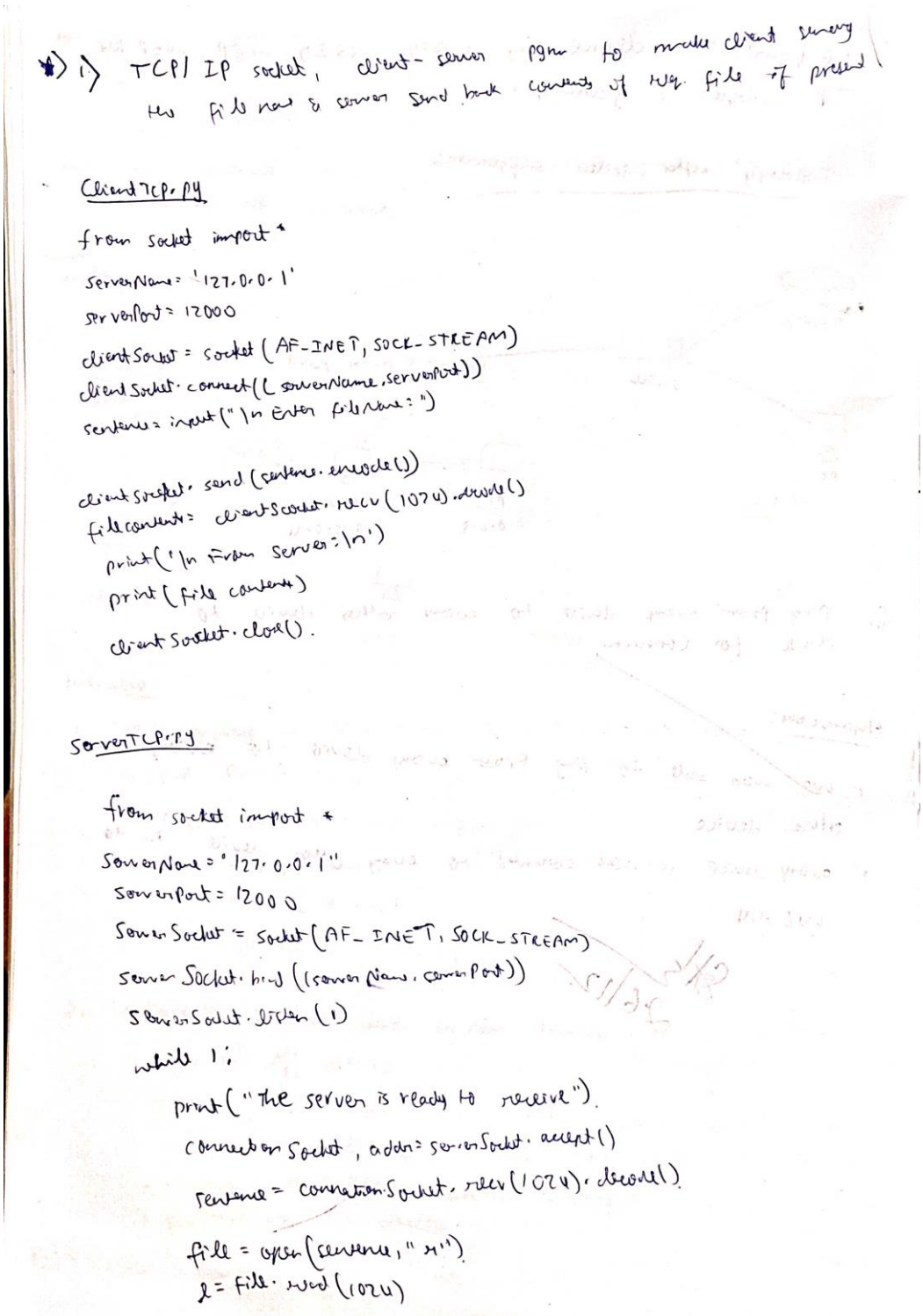
Output Clear

```
Generated packets: [80, 63, 57, 12, 69]
Enter bucket size: 60
Enter output rate: 30
Packet of size 80 bytes exceeds bucket capacity (60 bytes) - REJECTED
Packet of size 63 bytes exceeds bucket capacity (60 bytes) - REJECTED

Packet of size 57 bytes added to bucket
Bytes in bucket: 57
Transmitting 30 bytes
Bytes remaining in bucket: 27
Transmitting 27 bytes
Bytes remaining in bucket: 0

Packet of size 12 bytes added to bucket
Bytes in bucket: 12
Transmitting 12 bytes
Bytes remaining in bucket: 0
Packet of size 69 bytes exceeds bucket capacity (60 bytes) - REJECTED

=== Code Execution Successful ===
```

Program 16

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

Code and Output:



1) TCP/IP socket, client-server pgm to make client send file name & server send back contents of req. file if present

ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\n Enter filename: ")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('\n From Server:\n')
print(filecontents)
clientSocket.close()
```

ServerTCP.py

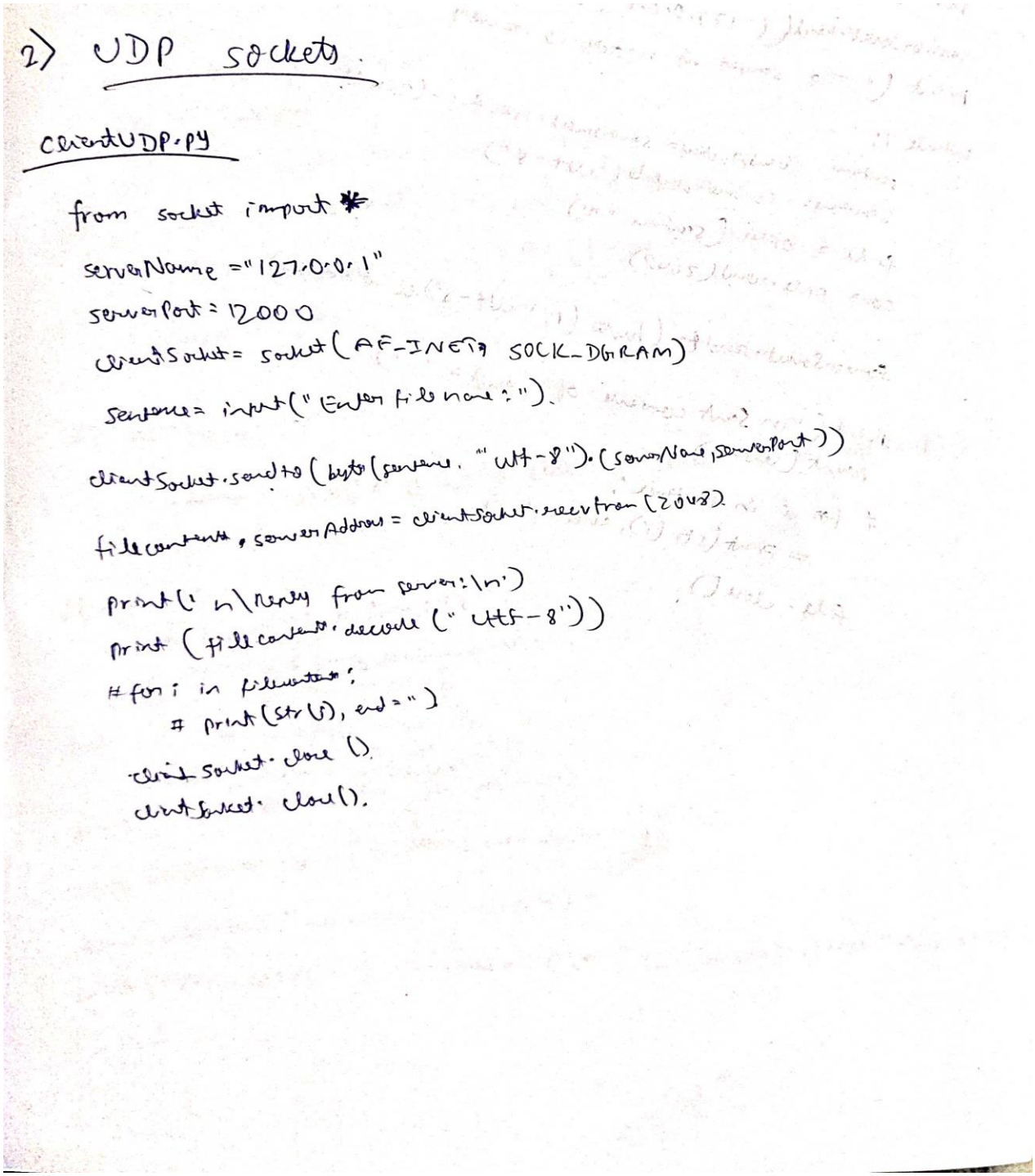
```
from socket import *
ServerName = '127.0.0.1'
ServerPort = 12000
ServerSocket = socket(AF_INET, SOCK_STREAM)
ServerSocket.bind((serverName, serverPort))
ServerSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
```

```
connectionSocket.send(l.encode())
print ("\n sent contents of ' + sentence)

file.close()
connectionSocket.close()
```

Program 17

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

Code and Output:



2) UDP sockets.

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(bytes(sentence, "utf-8"), (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom(2048)
print('\n\nReply from server:\n')
print(fileContents.decode("utf-8"))
# for i in fileContents:
    # print(str(i), end="")
clientSocket.close()
clientSocket.close().
```

ServerUDP.py.

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
    serverSocket.sendto(bytes(1, "utf-8"),clientAddress)
    print('\n Sent contents of', end = ' ')
    print(sentence)
    # for i in sentence:
        # print(str(i), end = ' ')
    file.close()
```

WIRESHARK

Wireshark

It is a powerful used network protocol analyzer. It allows you to capture and inspect data packets traveling over a network in real time, making it a crucial tool for studying computer networks, troubleshooting network issues and understanding protocols.

Features:

1. Packet capture: captures live network traffic from various interfaces (eg. Wifi)
2. Protocol analysis: supports 100's of protocols like TCP, UDP.
3. Filtering: isolate specific packets.
6. Visualization: display packet details with hierarchical layers

Use cases:

1. Network Troubleshooting:
   - Diagnosing slow network speeds.
   - Identifying bottlenecks or misconfigurations.
2. Security analysis:
   - detecting malicious traffic or intrusions.
3. Protocol Study:
   - understanding packet structures & communication flows.

common filters:

- http: shows only HTTP traffic
- tcp.port == 80 : shows traffic on TCP port 80
- ip.addr == 192.168.1.1 : shows pkt to or from a single specific IP address.
- UDP : shows only UDP traffic.