VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

M Rajashekar Reddy (1BM22CS138)

in partial fulfilment 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
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 **M Rajashekar Reddy (1BM22CS138),** 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.

Srushti C S Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

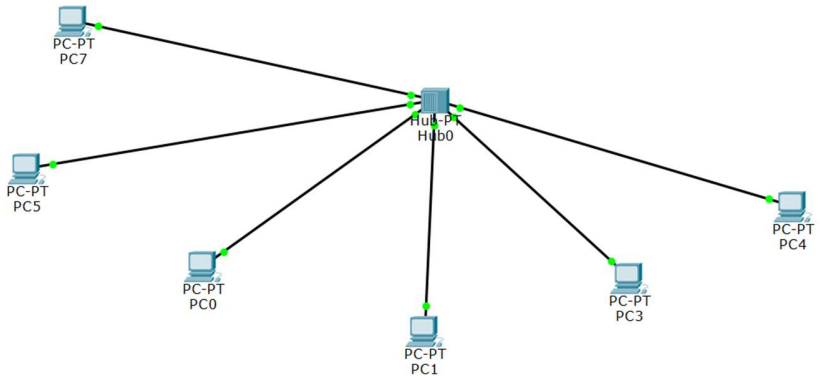
GitHub Link:
https://github.com/Raja3008/CN-1BM22CS138

CYCLE – I

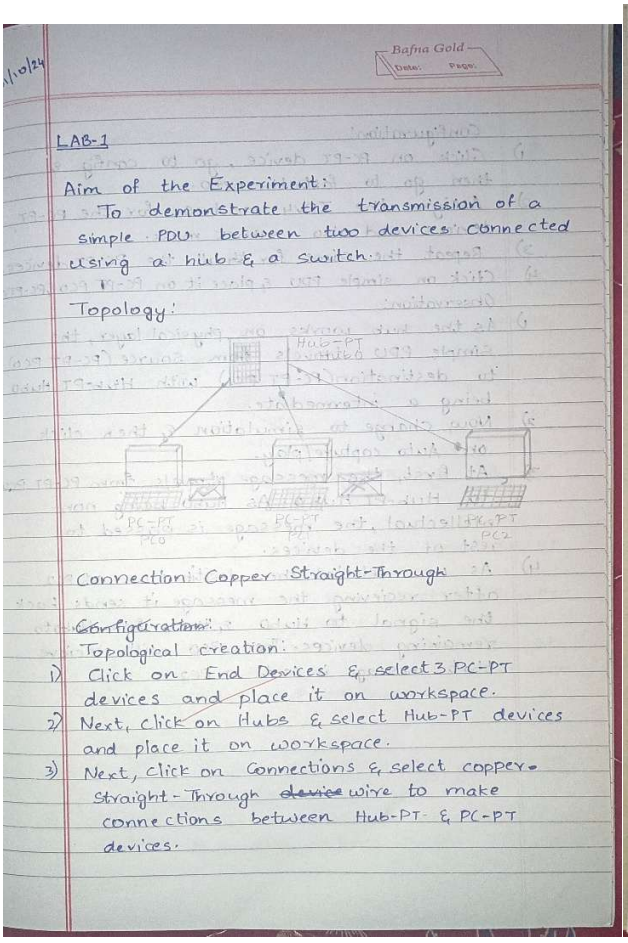
Program 1: To demonstrate the transmission of a simple PDU between 2 devices connected using a hub and a switch.

HUB:

Topology:



Observation:



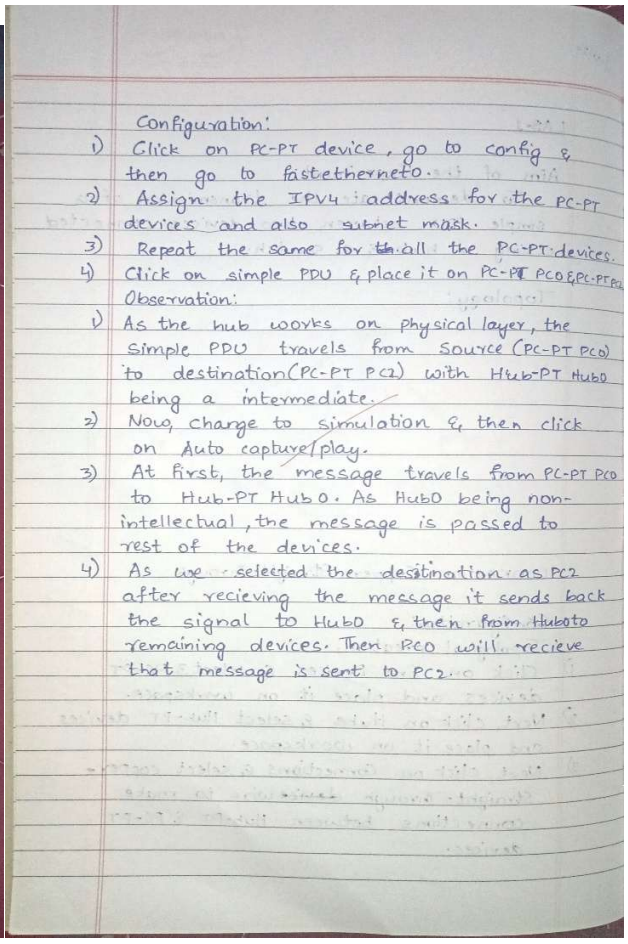
LAB-1

Aim of the Experiment:
To demonstrate the transmission of a simple PDU between two devices connected using a hub & a switch.

Topology:

Connection: Copper Straight-Through

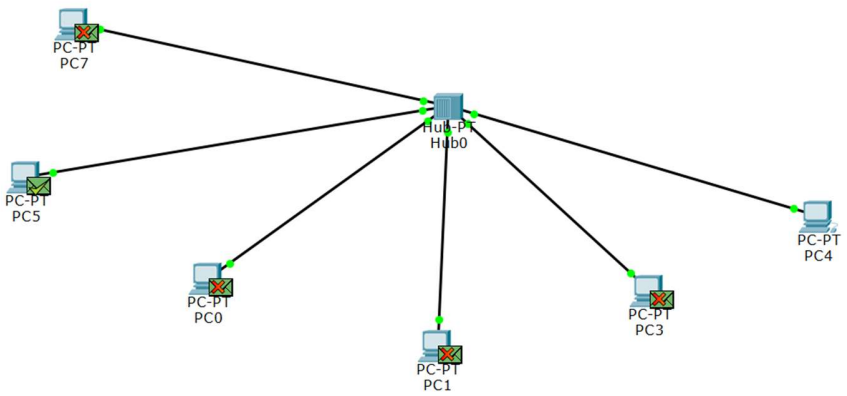
Configuration:
Topological creation:
1) Click on End Devices & select 3 PC-PT devices and place it on workspace.
2) Next, click on Hubs & select Hub-PT devices and place it on workspace.
3) Next, click on Connections & select copper-Straight-Through ~~device~~ wire to make connections between Hub-PT & PC-PT devices.



Configuration:
1) Click on PC-PT device, go to config & then go to fastethernet0.
2) Assign the IPV4 address for the PC-PT devices and also subnet mask.
3) Repeat the same for all the PC-PT devices.
4) Click on simple PDU & place it on PC-PT PC0 & PC-PT PC2

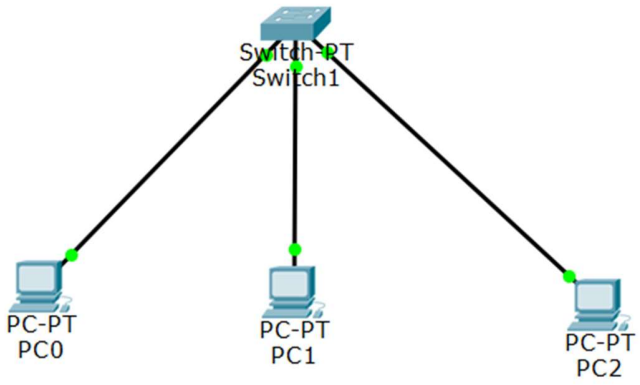
Observation:
1) As the hub works on physical layer, the simple PDU travels from source (PC-PT PC0) to destination (PC-PT PC2) with Hub-PT Hub0 being a intermediate.
2) Now, change to simulation & then click on Auto capture/play.
3) At first, the message travels from PC-PT PC0 to Hub-PT Hub0. As Hub0 being non-intellectual, the message is passed to rest of the devices.
4) As we selected the desitination as PC2 after recieving the message it sends back the signal to Hub0 & then from Hub0 to remaining devices. Then PC0 will recieve that message is sent to PC2.

Output:

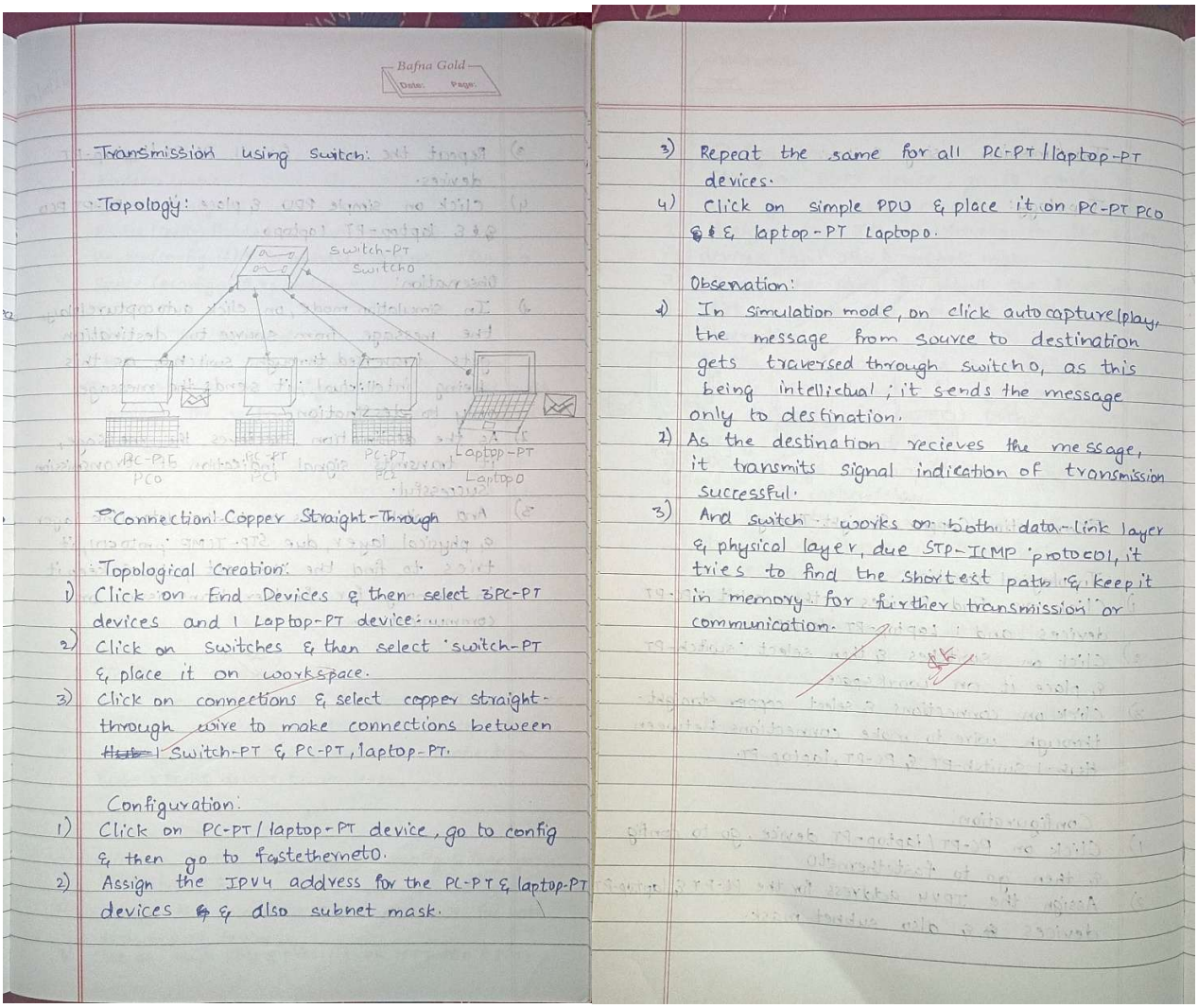


SWITCH:

Topology:



Observation:



Transmission using switch:

Topology:

Connection: Copper Straight-Through

Topological Creation:

1) Click on End Devices & then select 3PC-PT devices and 1 Laptop-PT device.
2) Click on Switches & then select 'switch-PT & place it on workspace.
3) Click on connections & select copper straight-through wire to make connections between ~~Hub~~ Switch-PT & PC-PT, laptop-PT.

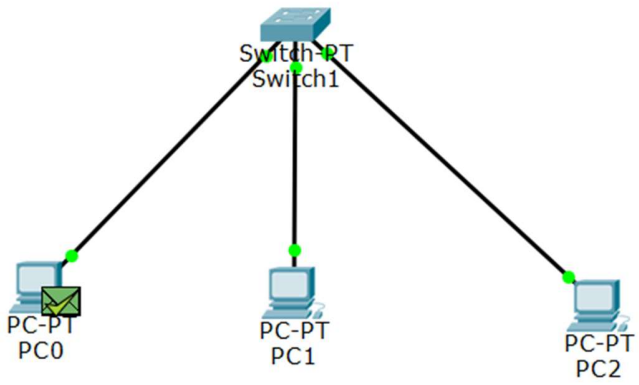
Configuration:

1) Click on PC-PT / laptop-PT device, go to config & then go to fastethernet0.
2) Assign the IPV4 address for the PC-PT & laptop-PT devices & also subnet mask.
3) Repeat the same for all PC-PT / laptop-PT devices.
4) Click on simple PDU & place it on PC-PT PC0 & & laptop-PT Laptop0.

Observation:

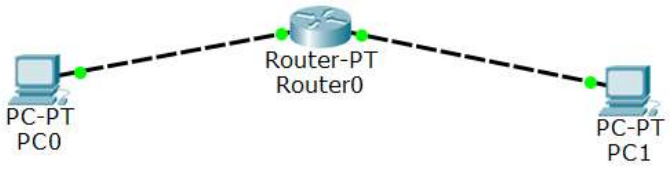
1) In simulation mode, on click auto capture/play, the message from source to destination gets traversed through switch0, as this being intellictual; it sends the message only to destination.
2) As the destination recieves the message, it transmits signal indication of transmission successful.
3) And switch works on both data-link layer & physical layer, due STP-ICMP protocol, it tries to find the shortest path & keep it in memory for further transmission or communication.

Output:



Program 2: To demonstrate configuration of IP Addresses to the routers and explore ping command.

Topology:



Observation:

08/10/24

LAB-2

Router > enable
Router # configure t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.3 255.0.0.0
Router (config-if) # no shutdown

Aim of the experiment:
~~PDU between two devices using router~~
To demonstrate the configuration of ip address with ~~Topology~~ the routers & explore ping command.
Topology:

Connection: Copper cross-over

Topological creation:
1) Click on End devices & select 2 PC-PT devices
2) Click on routers & select generic router device.
3) Click on connections & make the connection two 2 PC-PT devices to routers.

Configuration:
1) Click on PC-PT devices, go to config & then go fastethernet 0.
2) Assign the ip address & subnet mask for both PC0 & PC1 devices.
3) Click on simple PDU & place it on PC-PT PC0 & PC1.

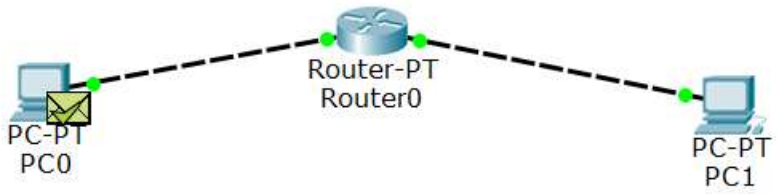
Gateway:
3) For router, given the following commands set the ip address for both Fa0/0 & Fa1/0.
Fastethernet 0/0 - 10.0.0.3 255.0.0.0
Fastethernet 1/0 - 20.0.0.3 255.0.0.0
4) Click on PC0 & PC1 devices and set the gateway
PC0 - 10.0.0.3
PC1 - 20.0.0.3
Now, in PC0, click on desktop & then command prompt.
PC> ping 20.0.0.3

Outputs possible:
1) Request timed-Out (Gateway not setup)
2) 0% loss statistics (Gateway setup successful)

Observation:
1) Pinging 20.0.0.3 with 32 bytes of data:
Reply from 20.0.0.3: bytes=32 time=0ms TTL=255
Reply from 20.0.0.3: bytes=32 time=4ms TTL=255
Reply from 20.0.0.3: bytes=32 time=0ms TTL=255
Reply from 20.0.0.3: bytes=32 time=0ms TTL=255

Ping statistics for 20.0.0.3:
Packets: Sent=4, Recieved=4, Lost=0 (0% loss)

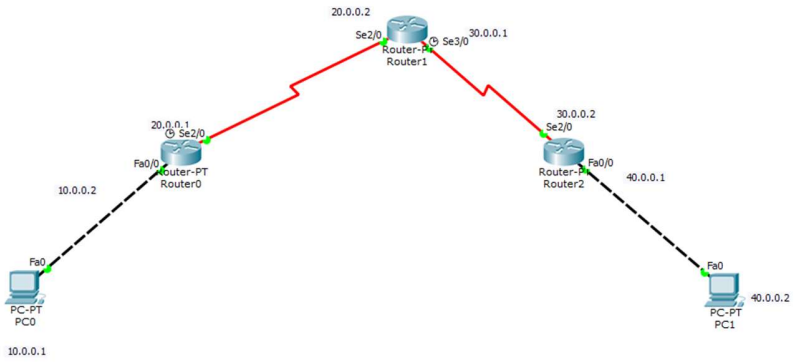
Output:


Router-PT
Router0
PC-PT
PC0
PC-PT
PC1

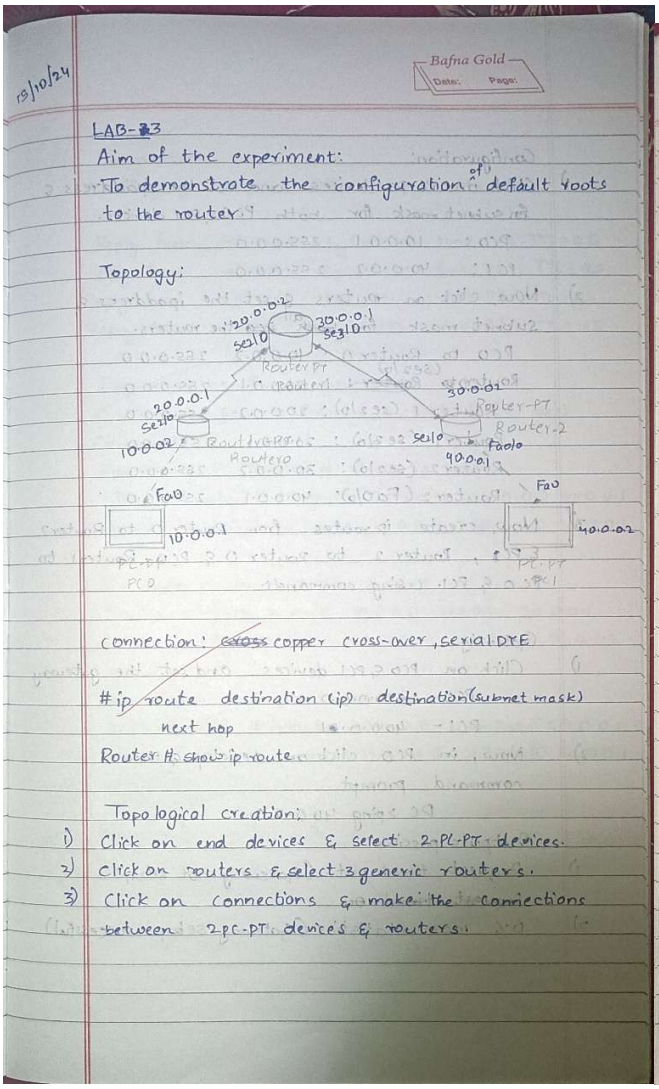

Router0
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: 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 FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
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
Copy
Paste

Program 3: To configure default and static routers to a connection of routers.

Topology:
Default:



Observation:



19/10/24

Bafna Gold
Date: Page:

LAB-3
Aim of the experiment:
To demonstrate the configuration of default routs to the router

Topology:

connection: ~~cross~~ copper cross-over, serial DTE

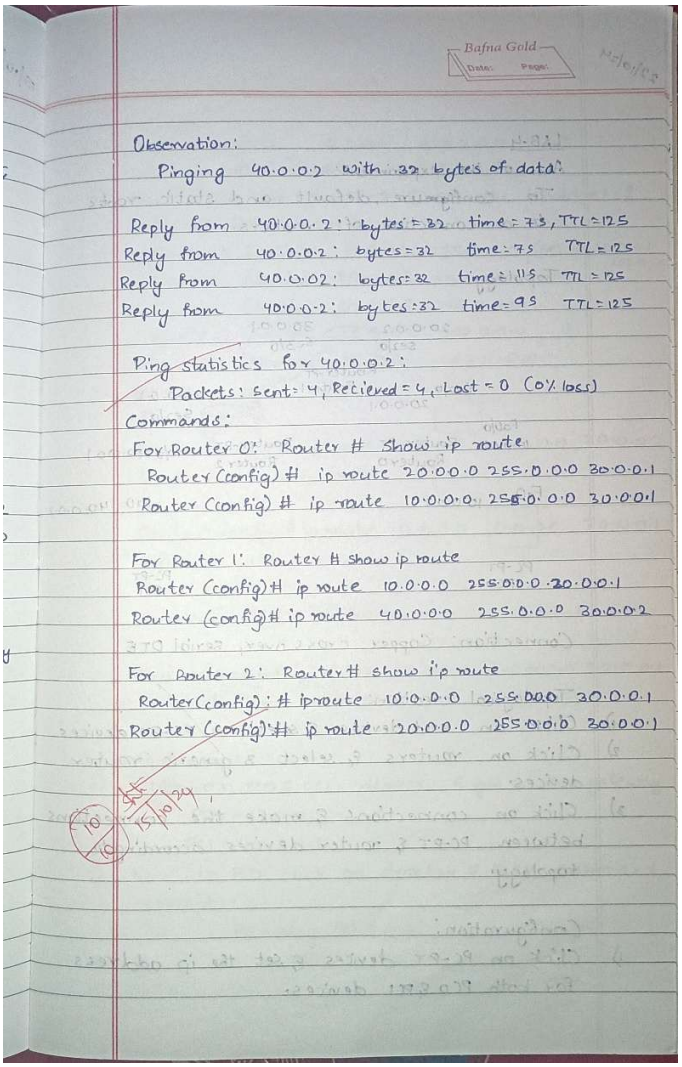
#ip route destination (ip) destination (subnet mask)
next hop
Router # show ip route

Topological creation:
1) Click on end devices & select 2 PC-PT devices.
2) Click on routers & select 3 generic routers.
3) Click on connections & make the connections between 2 pc-PT devices & routers.

Configuration:
1) Click on PC-PT devices and set the ipaddress & subnet mask for both PC0 & PC1 devices.
PC0: 10.0.0.1 255.0.0.0
PC1: 40.0.0.2 255.0.0.0
2) Now, click on routers & set the ipaddress & subnet mask for all the routers.
PC0 to Router 0: 10.0.0.2 255.0.0.0
Router 0 (Se2/0): 20.0.0.1 255.0.0.0
Router 1 (Se2/0): 20.0.0.2 255.0.0.0
Router 1 (Se3/0): 30.0.0.1 255.0.0.0
Router 2 (Se2/0): 30.0.0.2 255.0.0.0
Router 2 (Fa0/0): 40.0.0.1 255.0.0.0
3) Now, create ip routes from Router 0 to Router 2 & PC2, Router 2 to Router 0 & PC0, Router 1 to PC0 & PC1 using commands.

Gateway:
1) Click on PC0 & PC1 devices and set the gateway
PC0 - 10.0.0.2
PC1 - 40.0.0.1
2) Now, in PC0 click on desktop & then command prompt.
PC > ping 40.0.0.2

Outputs possible:
1) Request timed out: (Gateway not setup)
2) 0% loss statistics (Gateway setup successful)



Observation:

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=7s, TTL=125
Reply from 40.0.0.2: bytes=32 time=7s TTL=125
Reply from 40.0.0.2: bytes=32 time=11s TTL=125
Reply from 40.0.0.2: bytes=32 time=9s TTL=125

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

Commands:

For Router 0: Router # show ip route
Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.1

For Router 1: Router # show ip route
Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

For Router 2: Router # show ip route
Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.1

Output:

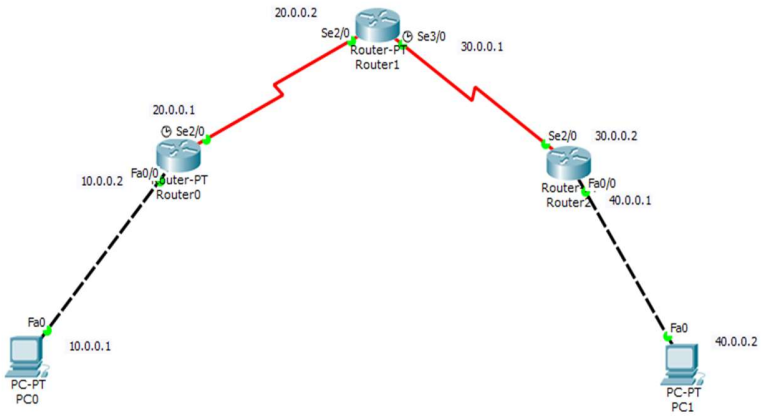
```
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, 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

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

```
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 40.0.0.0 250.0.0.0 30.0.0.2
%Inconsistent address and mask
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
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

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(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
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    20.0.0.0/8 is directly connected, Serial2/0
S    30.0.0.0/8 [1/0] via 20.0.0.2
S    40.0.0.0/8 [1/0] via 20.0.0.2
```

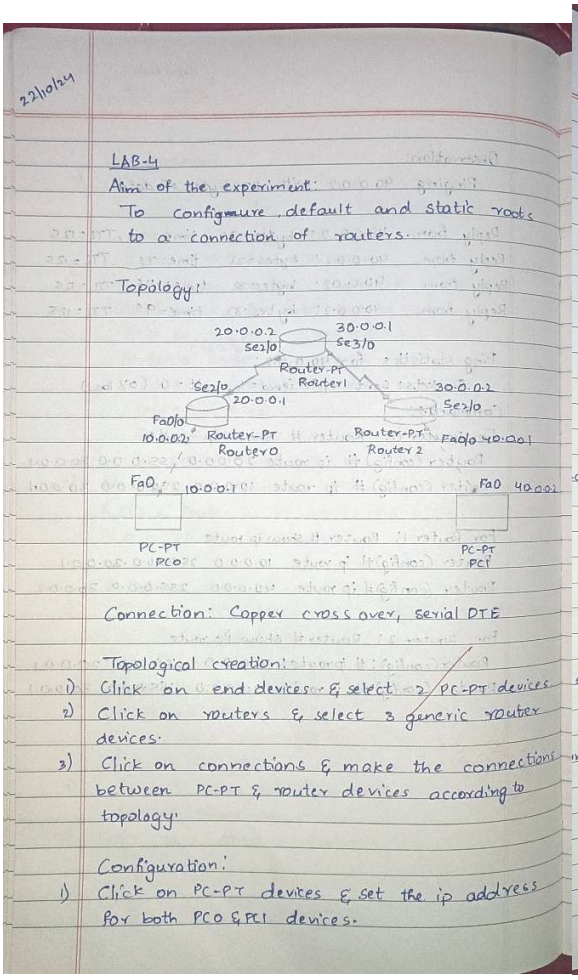
```
PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=11ms TTL=125
Reply from 40.0.0.2: bytes=32 time=9ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 11ms, Average = 8ms
```

Static:

Topology:



Observation:



22/10/24

LAB-4

Aim of the experiment:

To configure default and static route to a connection of routers.

Topology:

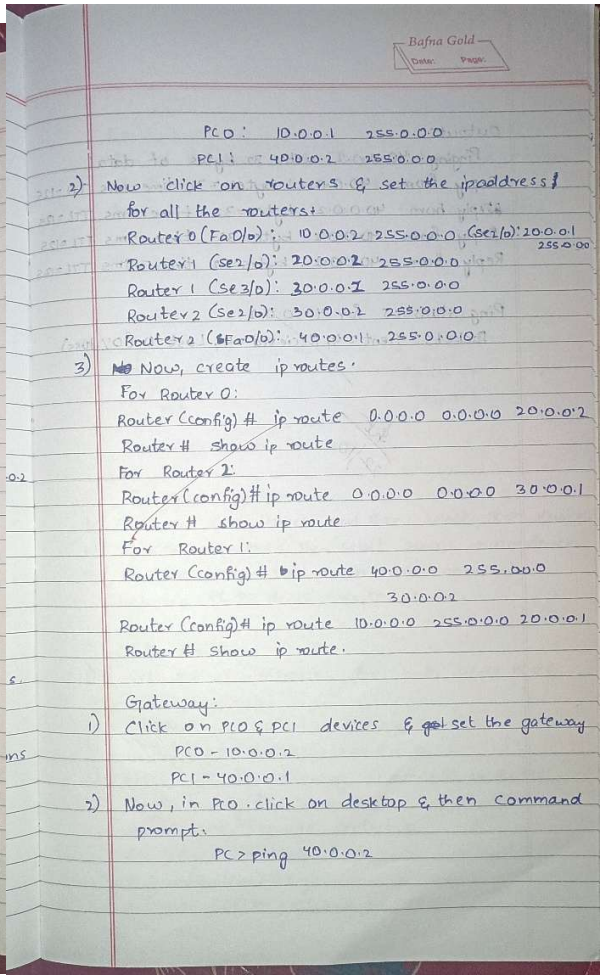
Connection: Copper cross over, serial DTE

Topological creation:

1) Click on end devices & select 2 PC-PT devices
2) Click on routers & select 3 generic router devices.
3) Click on connections & make the connections between PC-PT & router devices according to topology.

Configuration:

1) Click on PC-PT devices & set the ip address for both PC0 & PC1 devices.



Bafna Gold

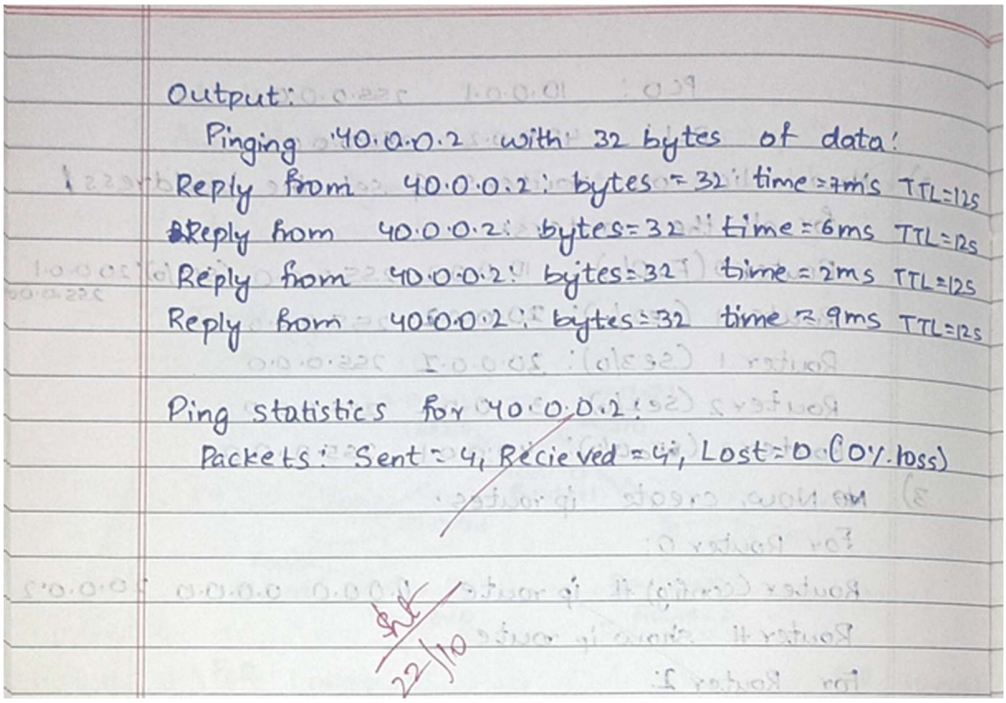
PC0: 10.0.0.1 255.0.0.0
PC1: 40.0.0.2 255.0.0.0

2) Now click on routers & set the ipaddress for all the routers:
Router 0 (Fa0/0): 10.0.0.2 255.0.0.0 (Se2/0): 20.0.0.1 255.0.0.0
Router 1 (Se2/0): 20.0.0.2 255.0.0.0
Router 1 (Se3/0): 30.0.0.1 255.0.0.0
Router 2 (Se2/0): 30.0.0.2 255.0.0.0
Router 2 (Fa0/0): 40.0.0.1 255.0.0.0

3) Now, create ip routes.
For Router 0:
Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router # show ip route
For Router 2:
Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router # show ip route
For Router 1:
Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router # show ip route.

Gateway:

1) Click on PC0 & PC1 devices & set the gateway
PC0 - 10.0.0.2
PC1 - 40.0.0.1
2) Now, in PC0 click on desktop & then command prompt.
PC > ping 40.0.0.2



Output:

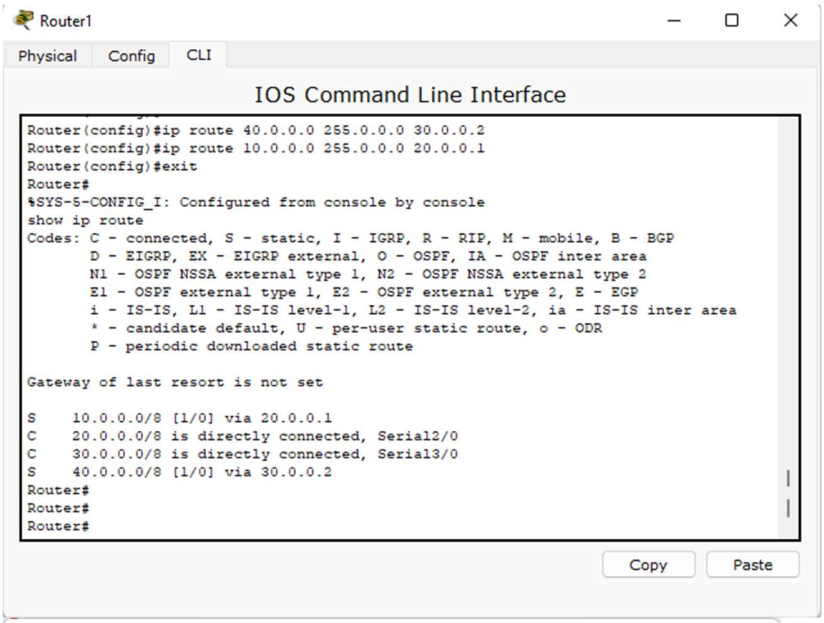
Router0

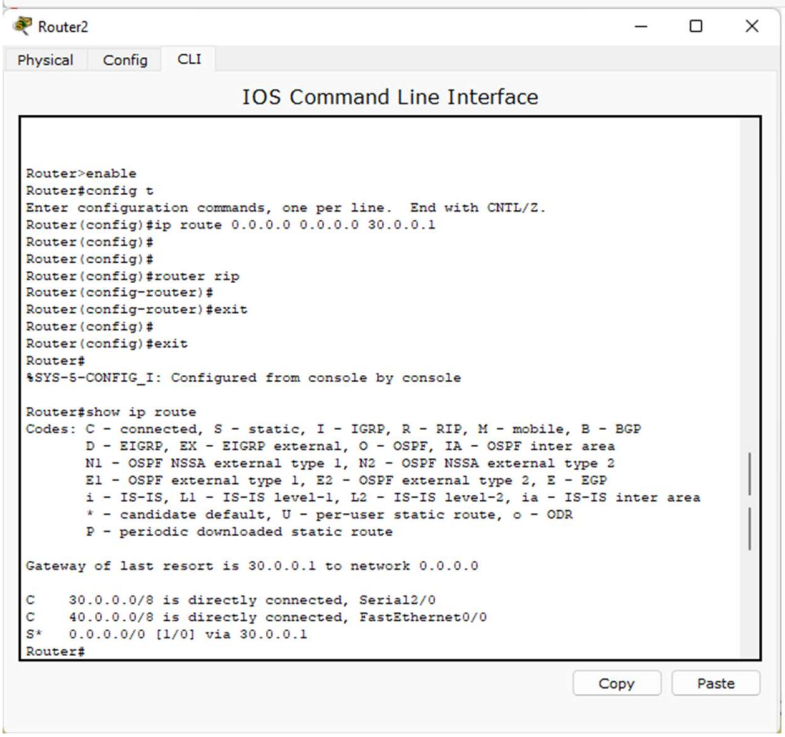
Physical | Config | CLI

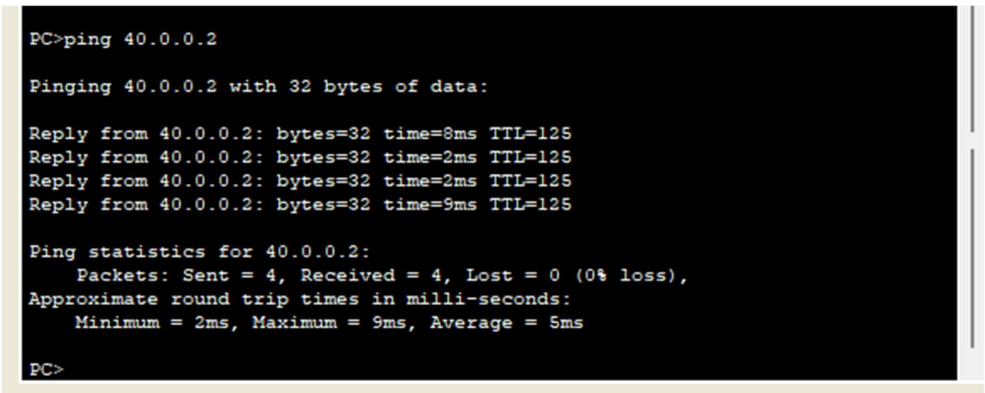
IOS Command Line Interface

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router(config)#
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

Gateway of last resort is 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#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Copy | Paste

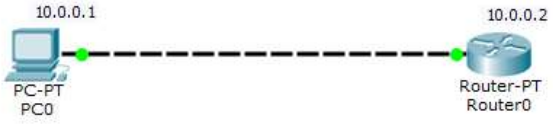

Router1
Physical
Config
CLI
IOS Command Line Interface
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, R - 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#
Router#
Router#
Copy
Paste


Router2
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router(config)#
Router(config)#
Router(config)#router rip
Router(config-router)#
Router(config-router)#exit
Router(config)#
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is 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

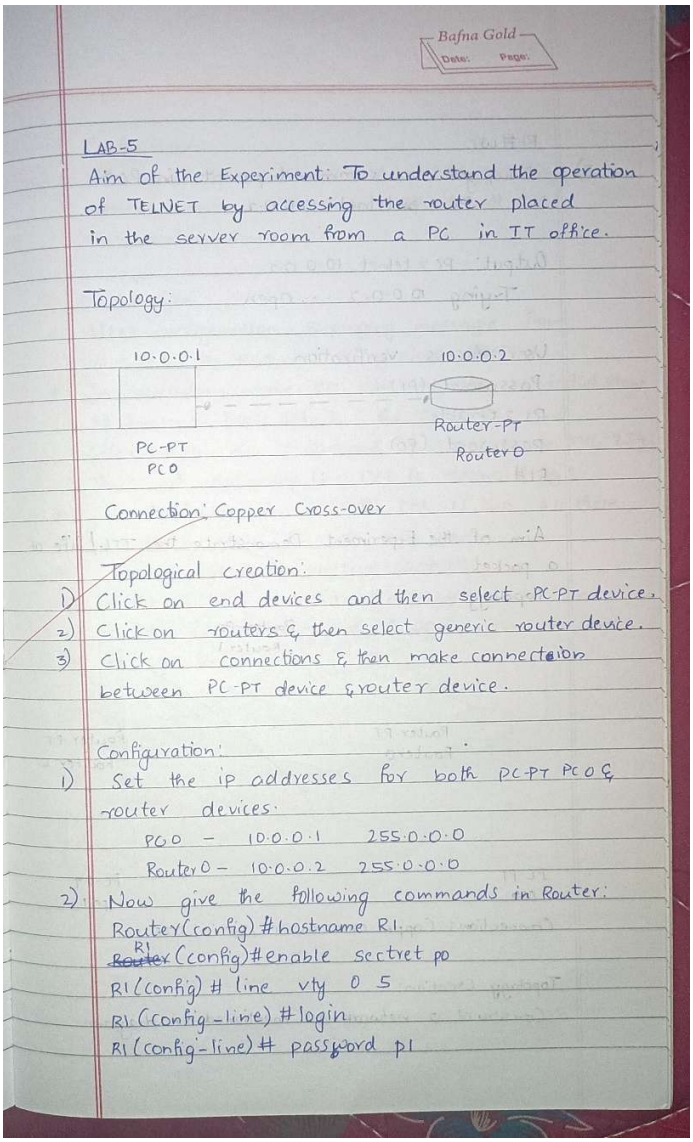

PC>ping 40.0.0.2
Pinging 40.0.0.2 with 32 bytes of data:
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=9ms TTL=125
Ping statistics for 40.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 9ms, Average = 5ms
PC>

Program 4: To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Topology:



Observation:



Bafna Gold

LAB-5

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

Topology:

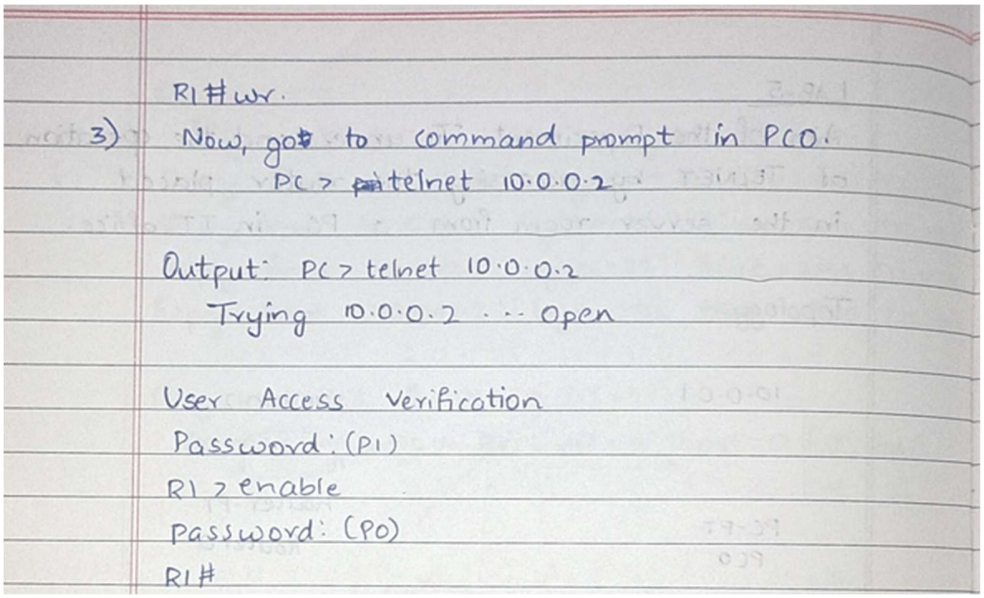
Connection: Copper Cross-over

Topological creation:

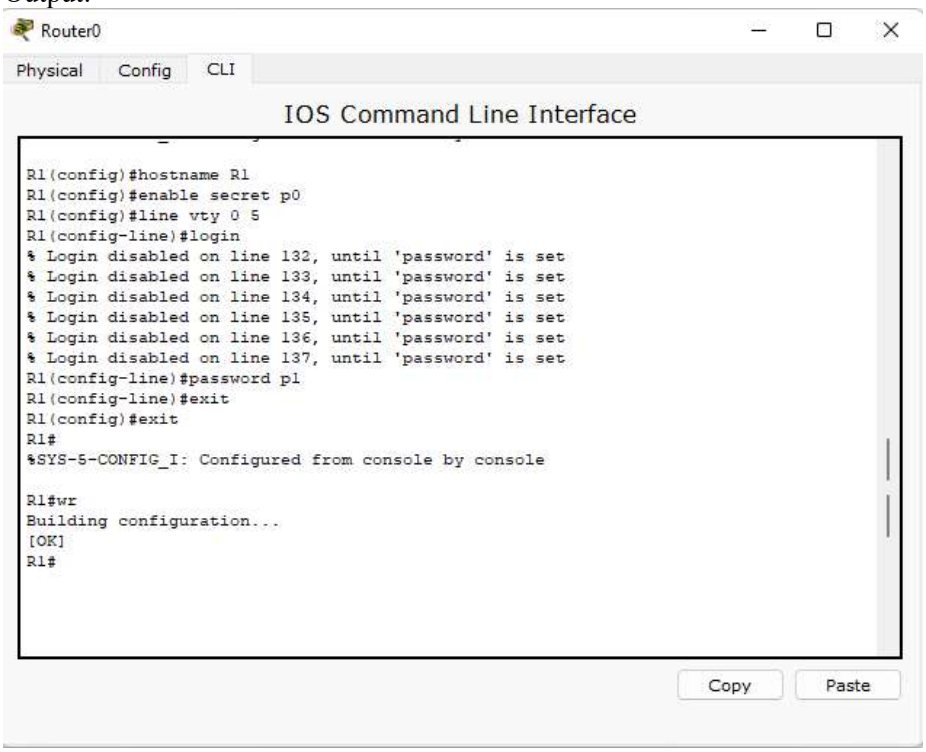
1) Click on end devices and then select PC-PT device.
2) Click on routers & then select generic router device.
3) Click on connections & then make connection between PC-PT device & router device.

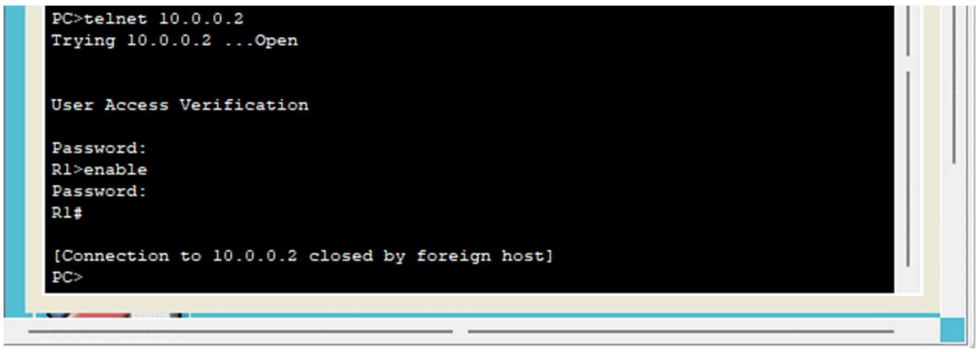
Configuration:

1) Set the ip addresses for both PC-PT PC0 & router devices.
   PC0 - 10.0.0.1 255.0.0.0
   Router0 - 10.0.0.2 255.0.0.0
2) Now give the following commands in Router:
   Router(config)#hostname R1
   ~~Router~~ R1(config)#enable secret p0
   R1(config)# line vty 0 5
   R1(config-line)#login
   R1(config-line)# password p1


R1#wr.
3) Now, go to command prompt in PC0.
PC> telnet 10.0.0.2
Output: PC> telnet 10.0.0.2
Trying 10.0.0.2 ... Open
User Access Verification
Password: (P1)
R1> enable
Password: (P0)
R1#

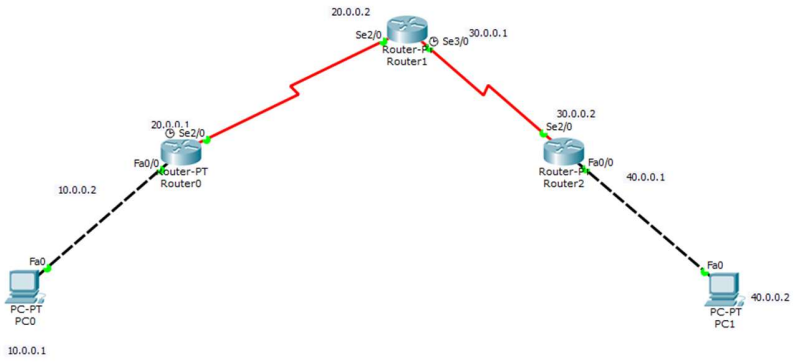
Output:


Router0
Physical
Config
CLI
IOS Command Line Interface
R1(config)#hostname R1
R1(config)#enable secret p0
R1(config)#line vty 0 5
R1(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
% Login disabled on line 136, until 'password' is set
% Login disabled on line 137, until 'password' is set
R1(config-line)#password p1
R1(config-line)#exit
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
R1#wr
Building configuration...
[OK]
R1#
Copy
Paste

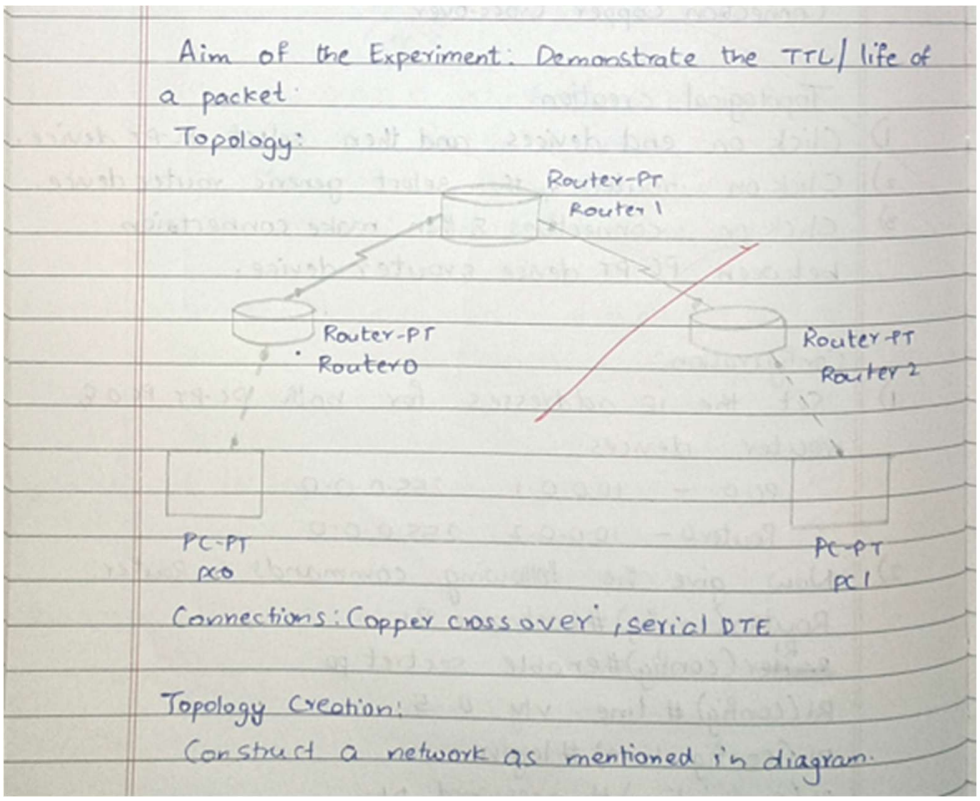

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open
User Access Verification
Password:
R1>enable
Password:
R1#
[Connection to 10.0.0.2 closed by foreign host]
PC>

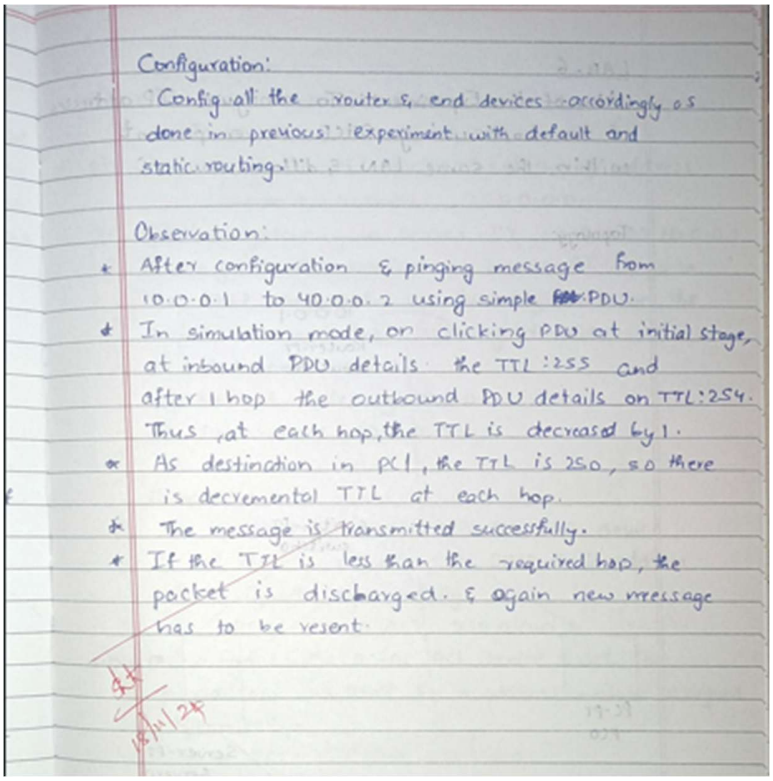
Program 5: Demonstrate the TTL/ Life of a Packet.

Topology:



Observation:





Configuration:

Config all the router & end devices accordingly as done in previous experiment with default and static routing.

Observation:

* After configuration & pinging message from 10.0.0.1 to 40.0.0.2 using simple PDU.
* In simulation mode, on clicking PDU at initial stage, at inbound PDU details the TTL:255 and after 1 hop the outbound PDU details on TTL:254. Thus, at each hop, the TTL is decreased by 1.
* As destination in PC1, the TTL is 250, so there is decremental TTL at each hop.
* The message is transmitted successfully.
* If the TTL is less than the required hop, the packet is discharged & again new message has to be resent.

Output:

PDU Information at Device: Router0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19 Byt
PREAMBLE: 101010...1011		DEST MAC: 0090.0C0A.0BA3	SRC MAC: 0003.E4D9.7C06	
TYPE: 0x800	DATA (VARIABLE LENGTH)			FCS: 0x0

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0xe			0x0	0x0	
TTL: 255	PRO: 0x1	CHKSUM			
SRC IP: 10.0.0.1					
DST IP: 40.0.0.2					
OPT: 0x0				0x0	
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x9		SEQ NUMBER: 14	

PDU Information at Device: Router0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

HDLC

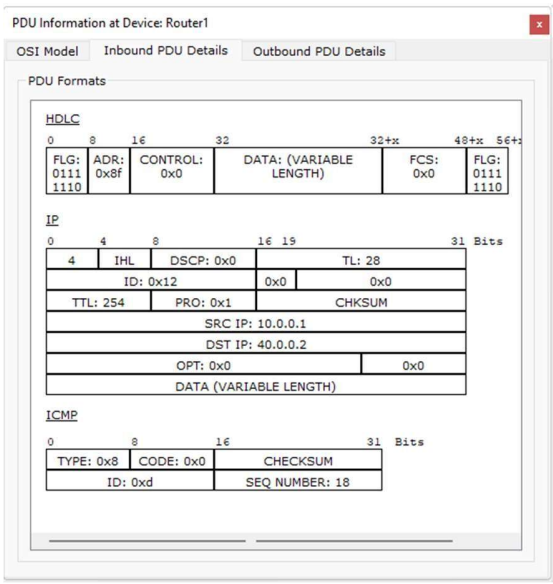
0	8	16	32	32+x	48+x	56+x
FLG: 0111 1110	ADR: 0x8f	CONTROL: 0x0	DATA: (VARIABLE LENGTH)		FCS: 0x0	FLG: 0111 1110

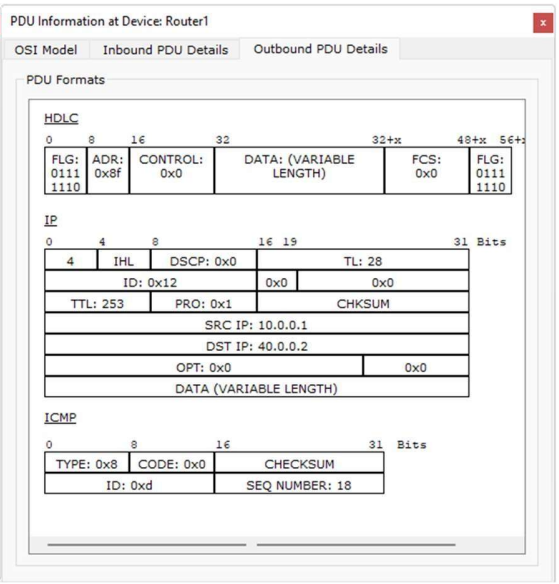
IP

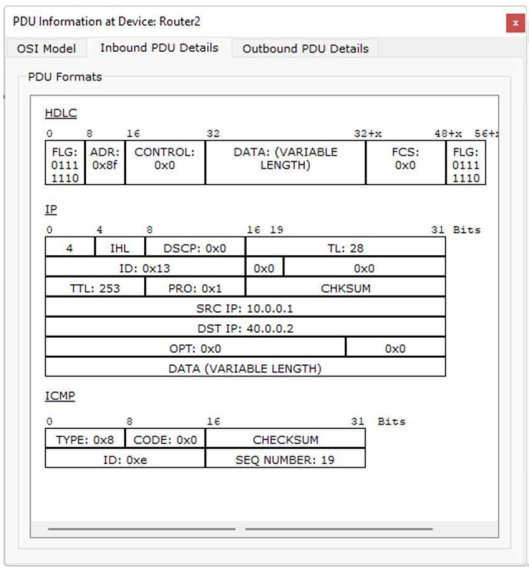
0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x10			0x0	0x0	
TTL: 254	PRO: 0x1	CHKSUM			
SRC IP: 10.0.0.1					
DST IP: 40.0.0.2					
OPT: 0x0				0x0	
DATA (VARIABLE LENGTH)					

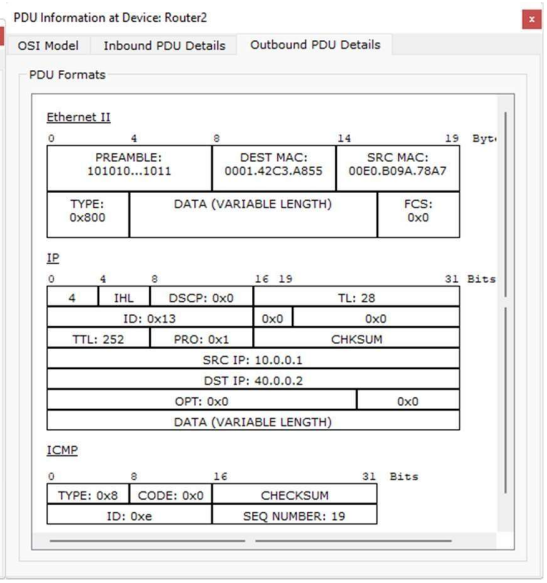
ICMP

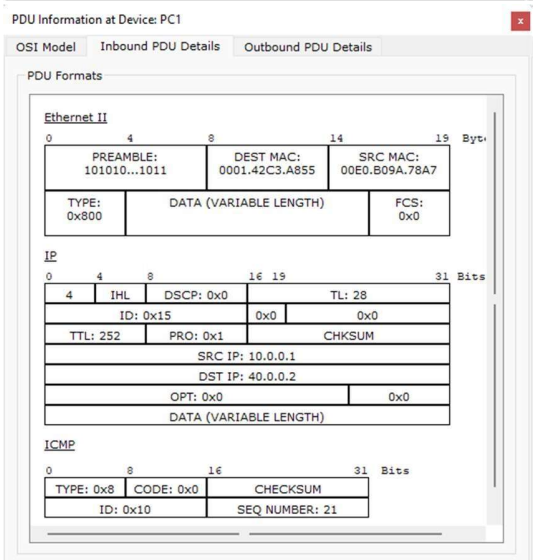
0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0xb		SEQ NUMBER: 16	

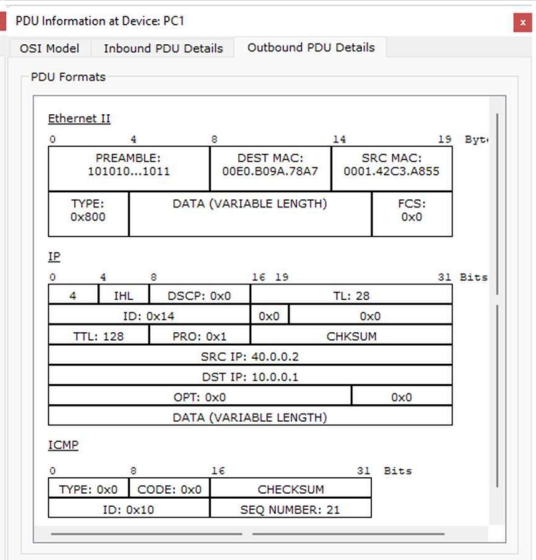

PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x12
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0xd
SEQ NUMBER: 18


PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x12
0x0
0x0
TTL: 253
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0xd
SEQ NUMBER: 18


PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x13
0x0
0x0
TTL: 253
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0xe
SEQ NUMBER: 19


PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0001.42C3.A855
SRC MAC: 00E0.B09A.78A7
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x13
0x0
0x0
TTL: 252
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0xe
SEQ NUMBER: 19

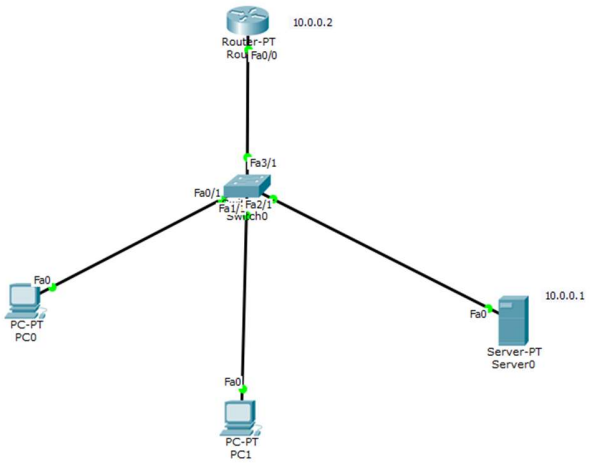

PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0001.42C3.A855
SRC MAC: 00E0.B09A.78A7
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x15
0x0
0x0
TTL: 252
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x10
SEQ NUMBER: 21


PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 00E0.B09A.78A7
SRC MAC: 0001.42C3.A855
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x14
0x0
0x0
TTL: 128
PRO: 0x1
CHKSUM
SRC IP: 40.0.0.2
DST IP: 10.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x10
SEQ NUMBER: 21

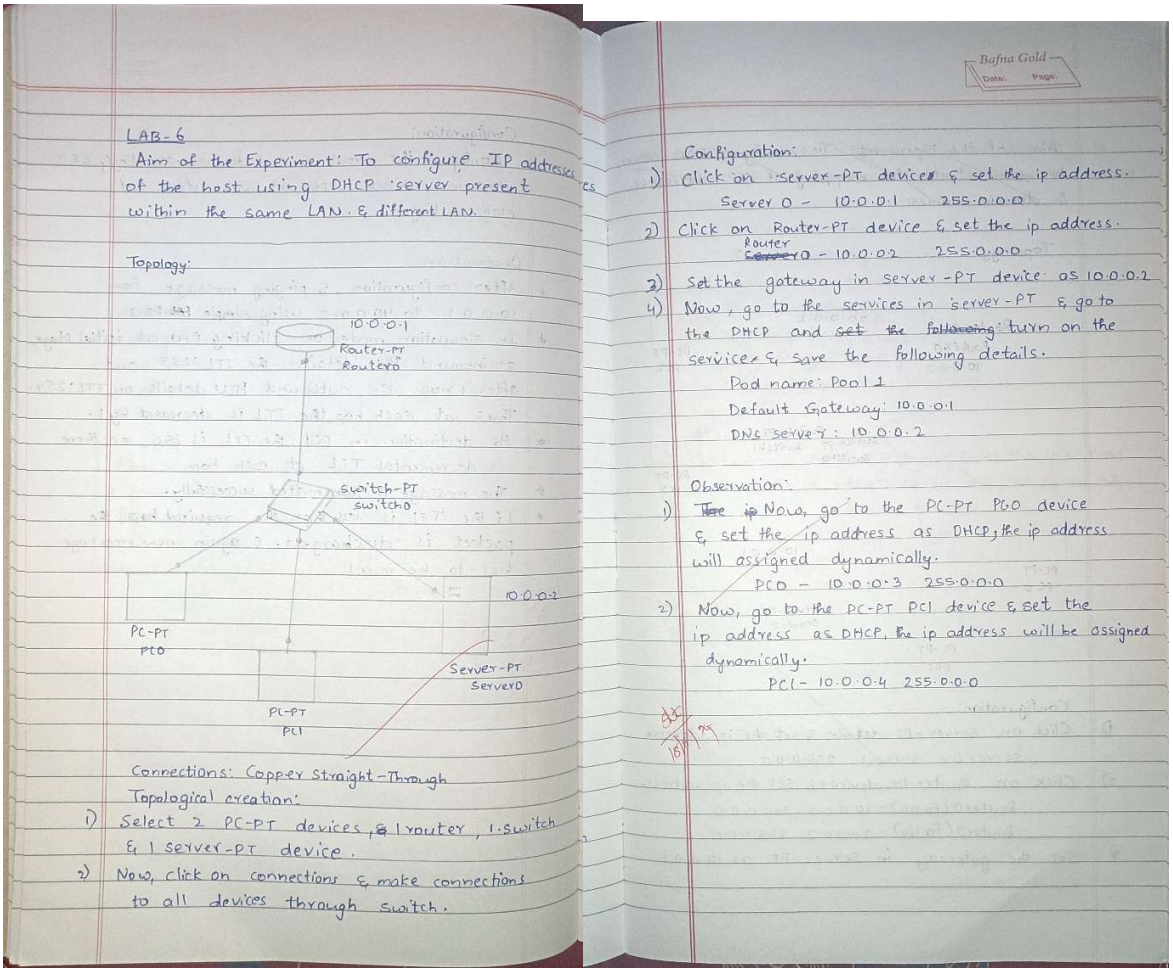
Program 6: Configure DHCP within a LAN and outside LAN.

WITHIN A LAN:

Topology:



Observation:



LAB-6

Aim of the Experiment: To configure IP addresses of the host using DHCP server present within the same LAN & different LAN.

Topology:

10.0.0.1
Router-PT
Router0
switch-PT
switch0
10.0.0.2
PC-PT
PC0
Server-PT
Server0
PC-PT
PC1

Connections: Copper Straight-Through

Topological creation:

1) Select 2 PC-PT devices, 1 router, 1 switch & 1 server-PT device.
2) Now, click on connections & make connections to all devices through switch.

Bafna Gold

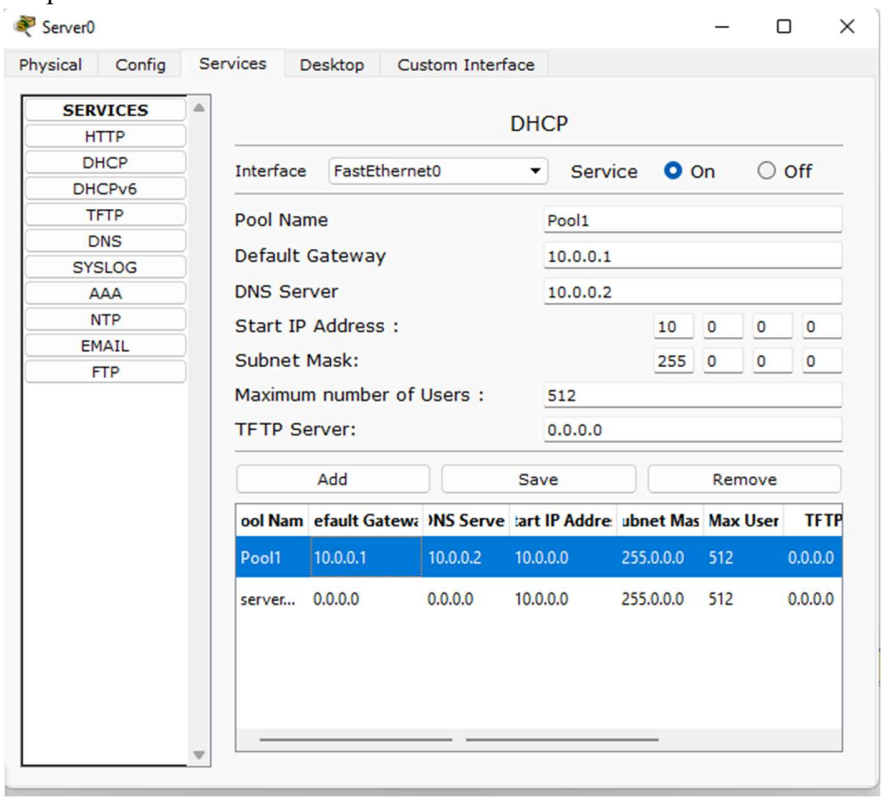
Configuration:

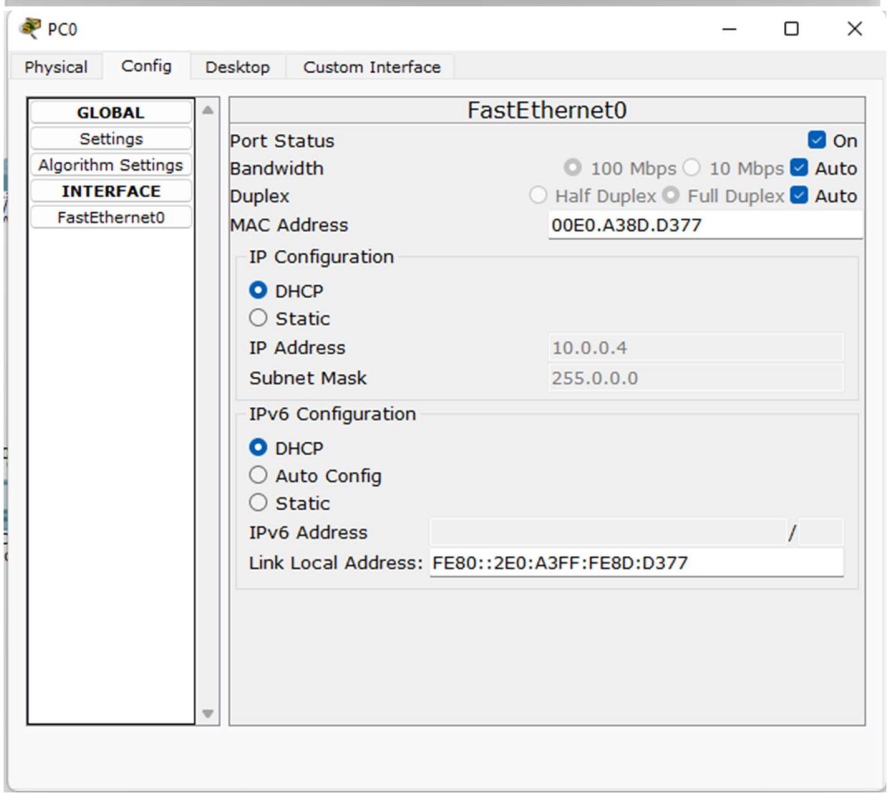
1) Click on server-PT device & set the ip address.
   Server 0 - 10.0.0.1 255.0.0.0
2) Click on Router-PT device & set the ip address.
   Router
   ~~Server~~0 - 10.0.0.2 255.0.0.0
3) Set the gateway in server-PT device as 10.0.0.2
4) Now, go to the services in server-PT & go to the DHCP and ~~set the following~~ turn on the services & save the following details.
   Pod name: Pool 1
   Default Gateway: 10.0.0.1
   DNS server: 10.0.0.2

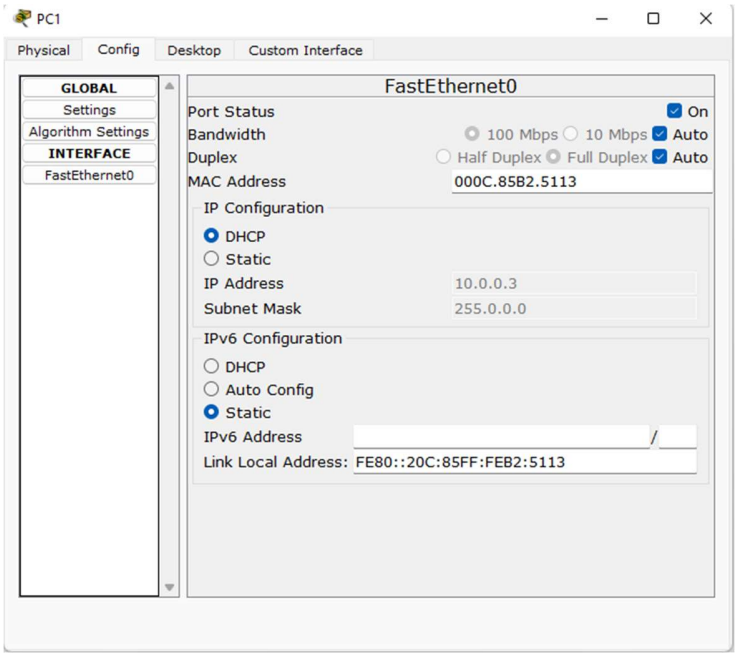
Observation:

1) ~~The~~ ip Now, go to the PC-PT PC0 device & set the ip address as DHCP, the ip address will assigned dynamically.
   PC0 - 10.0.0.3 255.0.0.0
2) Now, go to the PC-PT PC1 device & set the ip address as DHCP, the ip address will be assigned dynamically.
   PC1 - 10.0.0.4 255.0.0.0

Output:

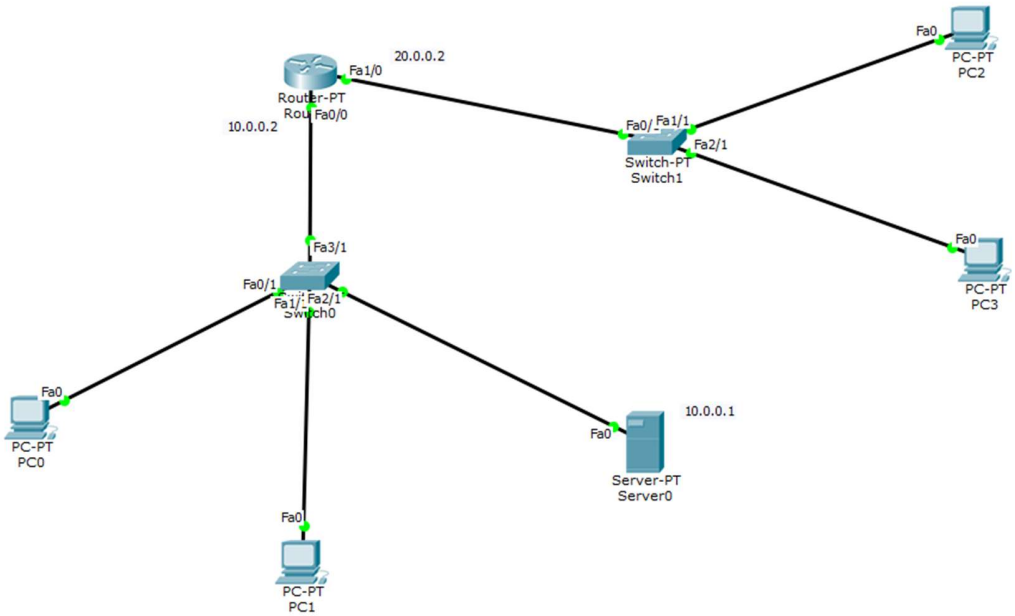

Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
Pool1
Default Gateway
10.0.0.1
DNS Server
10.0.0.2
Start IP Address :
10 0 0 0
Subnet Mask:
255 0 0 0
Maximum number of Users :
512
TFTP Server:
0.0.0.0
Add
Save
Remove
ool Nam
efault Gatew
NS Serve
art IP Addre
ubnet Mas
Max User
TFTP
Pool1
10.0.0.1
10.0.0.2
10.0.0.0
255.0.0.0
512
0.0.0.0
server...
0.0.0.0
0.0.0.0
10.0.0.0
255.0.0.0
512
0.0.0.0


PC0
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00E0.A38D.D377
IP Configuration
DHCP
Static
IP Address
10.0.0.4
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
/
Link Local Address:
FE80::2E0:A3FF:FE8D:D377

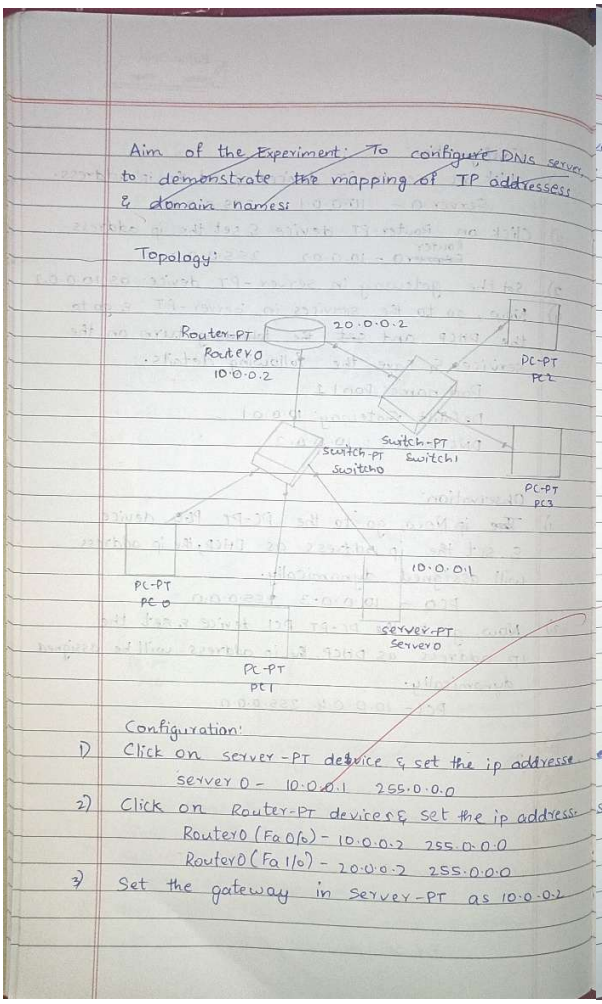


OUTSIDE LAN:

Topology:



Observation:

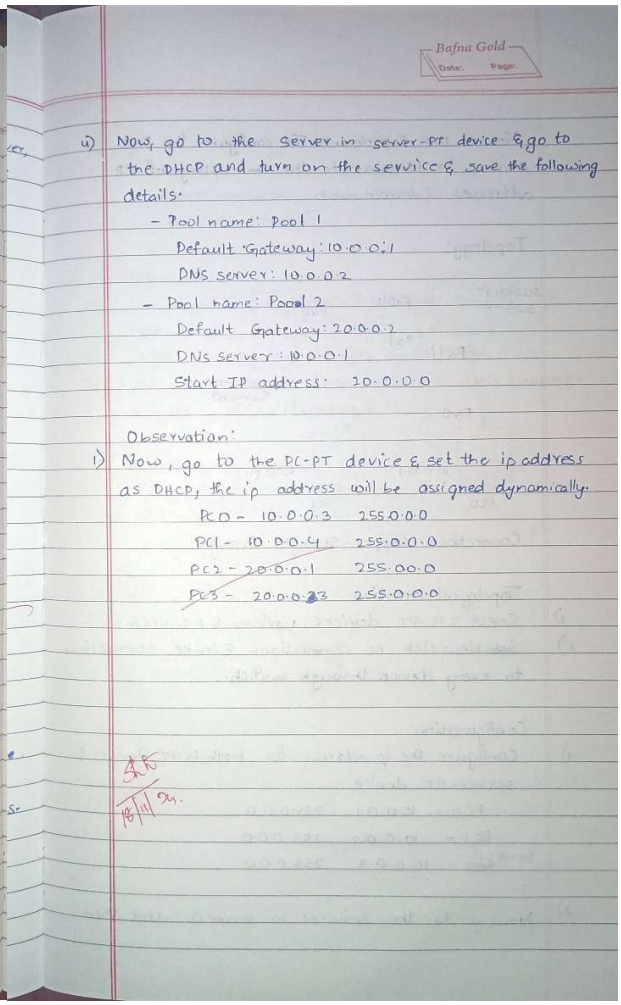


Aim of the Experiment: To configure DNS server to demonstrate the mapping of IP addresses & domain names.

Topology:

Configuration:

1) Click on server-PT device & set the ip address
   server 0 - 10.0.0.1 255.0.0.0
2) Click on Router-PT device & set the ip address.
   Router0 (Fa0/0) - 10.0.0.2 255.0.0.0
   Router0 (Fa1/0) - 20.0.0.2 255.0.0.0
3) Set the gateway in server-PT as 10.0.0.2



Bafna Gold

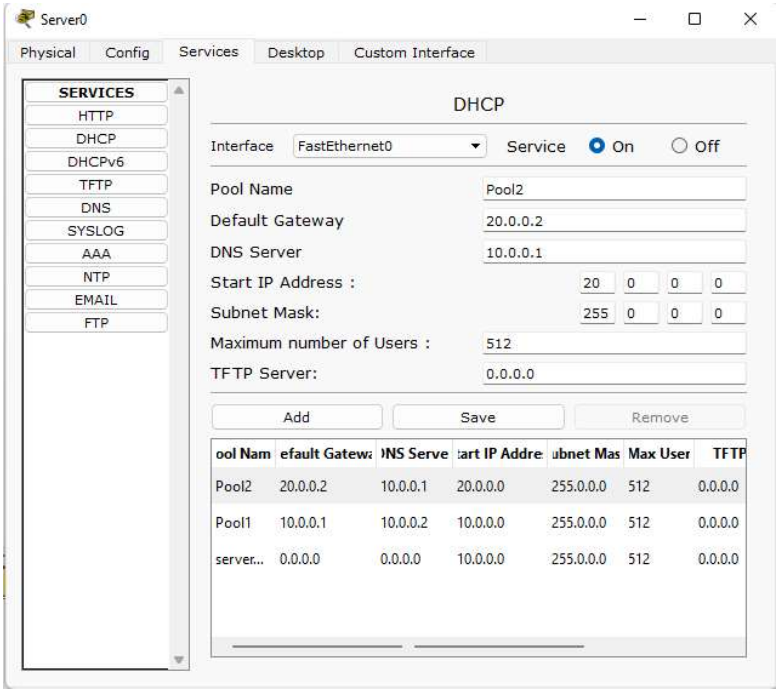
4) Now, go to the server in server-PT device & go to the DHCP and turn on the service & save the following details:
   - Pool name: Pool 1
     Default Gateway: 10.0.0.1
     DNS server: 10.0.0.2
   - Pool name: Pool 2
     Default Gateway: 20.0.0.2
     DNS server: 10.0.0.1
     Start IP address: 20.0.0.0

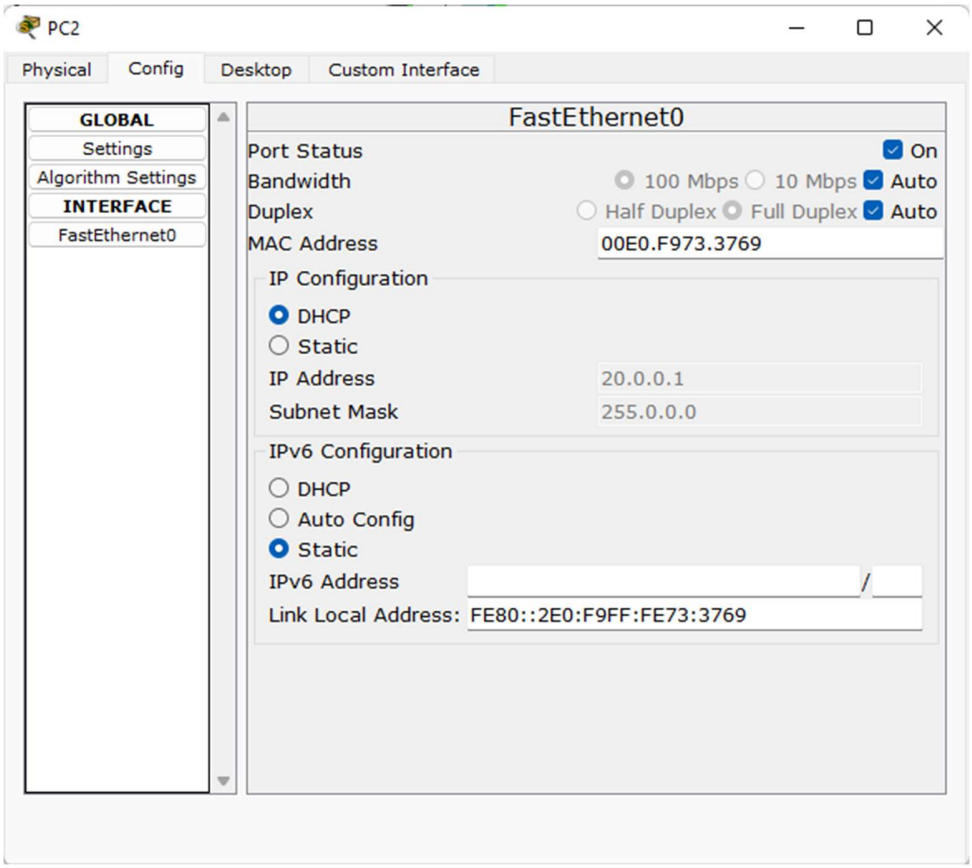
Observation:

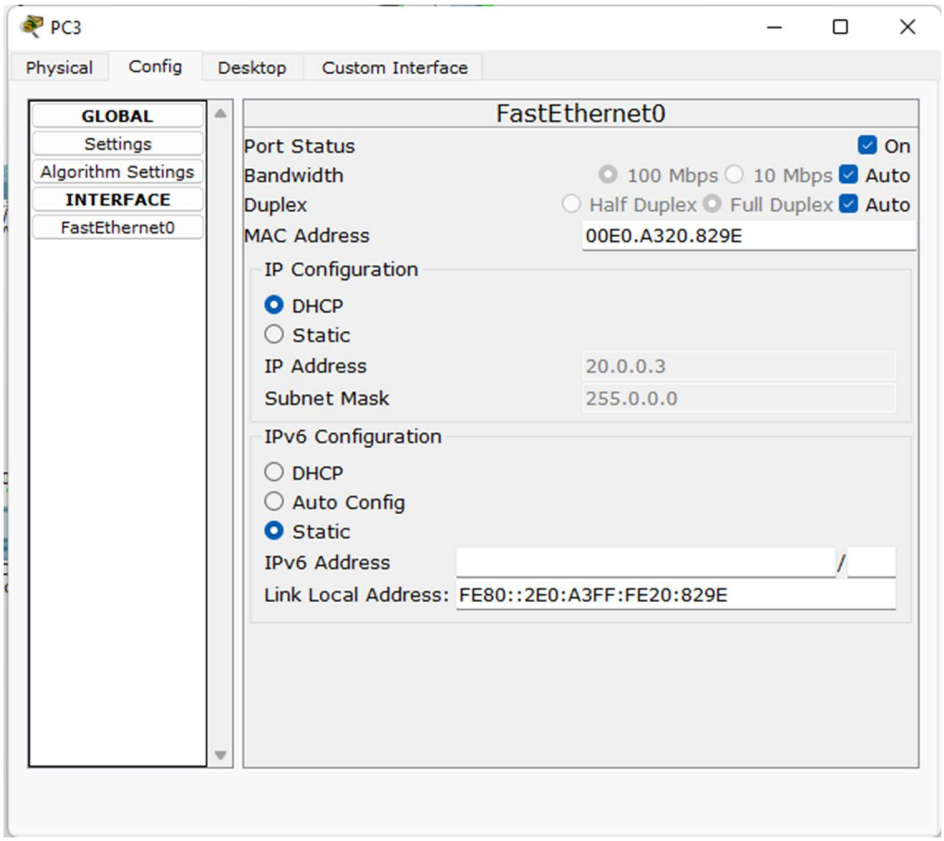
1) Now, go to the PC-PT device & set the ip address as DHCP, the ip address will be assigned dynamically.
   PC0 - 10.0.0.3 255.0.0.0
   PC1 - 10.0.0.4 255.0.0.0
   PC2 - 20.0.0.1 255.0.0.0
   PC3 - 20.0.0.3 255.0.0.0

18/11/24

Output:

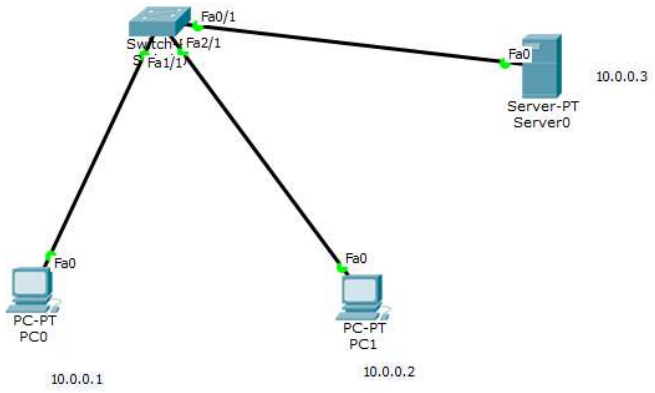



PC2
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00E0.F973.3769
IP Configuration
DHCP
Static
IP Address
20.0.0.1
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address:
FE80::2E0:F9FF:FE73:3769

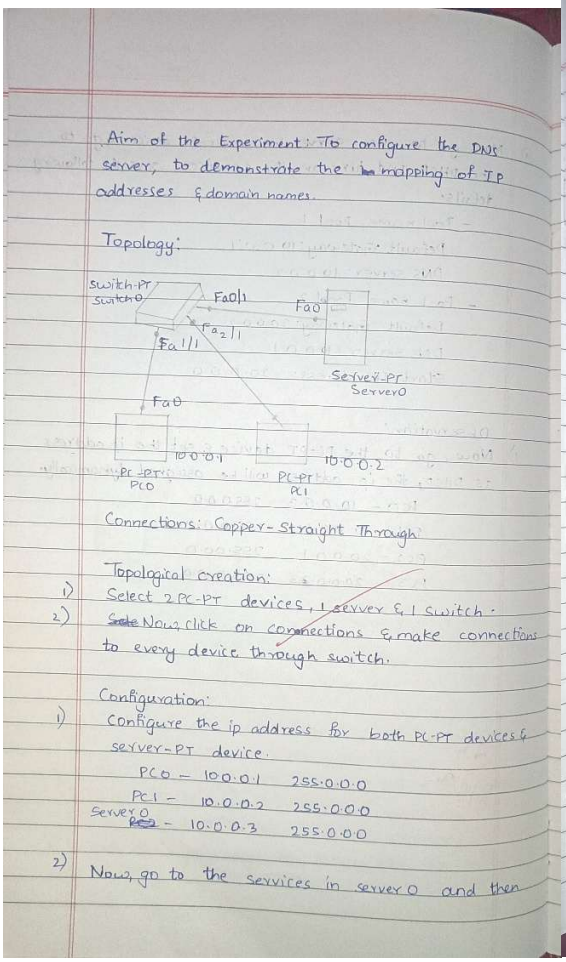

PC3
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00E0.A320.829E
IP Configuration
DHCP
Static
IP Address
20.0.0.3
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address:
FE80::2E0:A3FF:FE20:829E

Program 7: To configure DNS server to demonstrate the mapping of IP addresses and domain names.

Topology:



Observation:



Aim of the Experiment: To configure the DNS server, to demonstrate the mapping of IP addresses & domain names.

Topology:

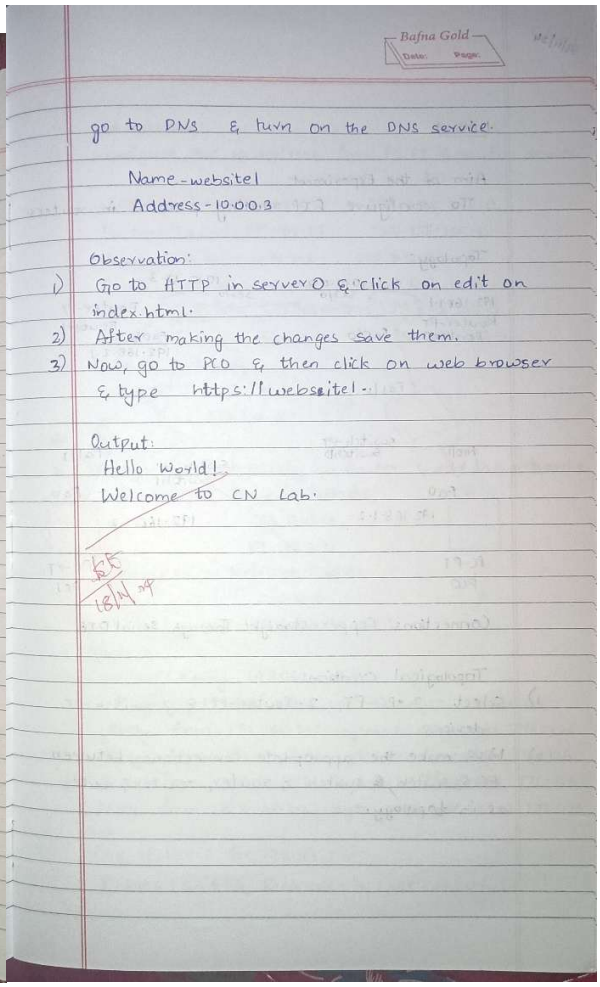
Connections: Copper-Straight Through

Topological creation:

1) Select 2 PC-PT devices, 1 server & 1 switch.
2) Now, click on connections & make connections to every device through switch.

Configuration:

1) Configure the ip address for both PC-PT devices & server-PT device.
   PC0 - 10.0.0.1 255.0.0.0
   PC1 - 10.0.0.2 255.0.0.0
   Server0 - 10.0.0.3 255.0.0.0
2) Now, go to the services in server 0 and then



Bafna Gold

go to DNS & turn on the DNS service.

Name - website
Address - 10.0.0.3

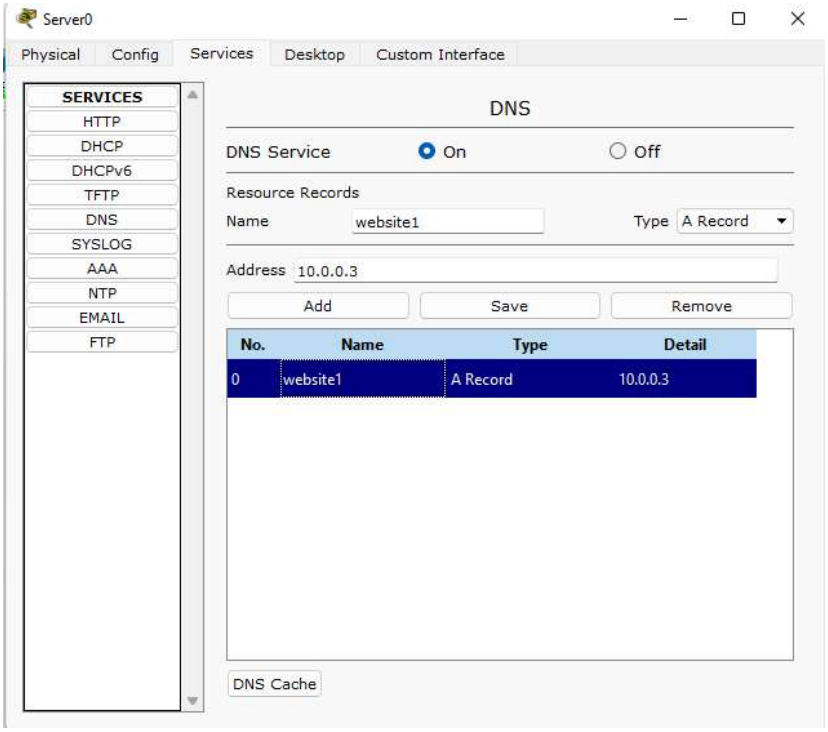
Observation:

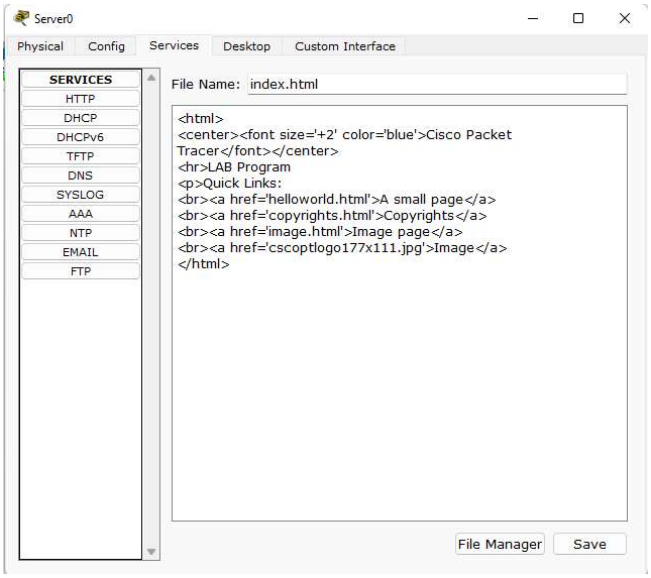
1) Go to HTTP in server 0 & click on edit on index.html.
2) After making the changes save them.
3) Now, go to PC0 & then click on web browser & type https://website.

Output:

Hello World!
Welcome to CN Lab.

Output:


Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DNS
DNS Service
On
Off
Resource Records
Name
website1
Type
A Record
Address
10.0.0.3
Add
Save
Remove
No.
Name
Type
Detail
0
website1
A Record
10.0.0.3
DNS Cache

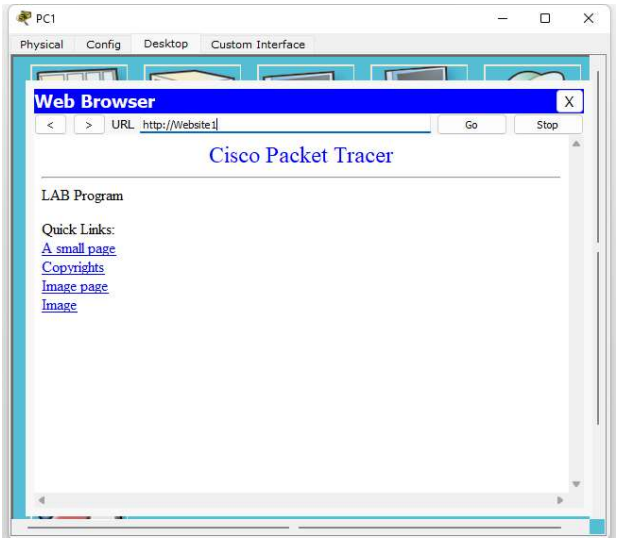

Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
File Name:
index.html
<html>
Cisco Packet Tracer
<hr>LAB Program
<p>Quick Links:

A small page

Copyrights

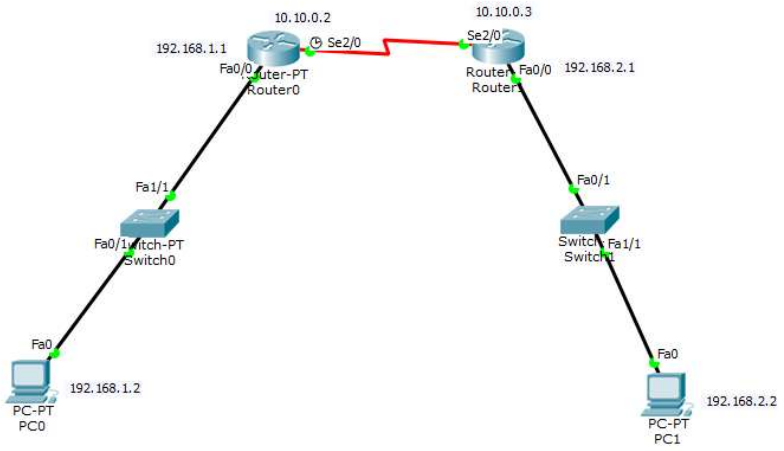
Image page

Image
</html>
File Manager
Save

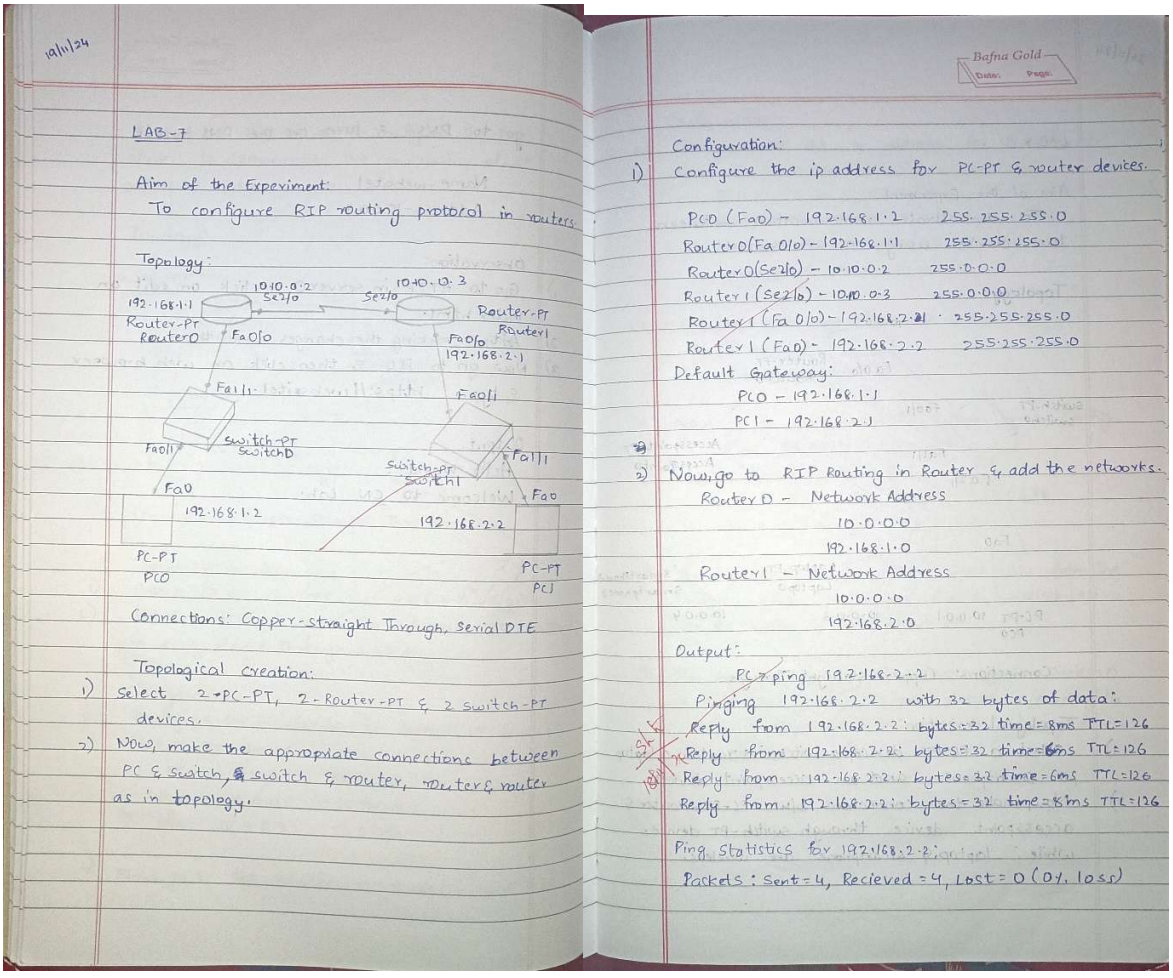

PC1
Physical
Config
Desktop
Custom Interface
Web Browser
URL
http://Website1
Go
Stop
Cisco Packet Tracer
LAB Program
Quick Links:
A small page
Copyrights
Image page
Image

Program 8: Configure RIP routing Protocol in Routers.

Topology:



Observation:



19/11/24

LAB-7

Aim of the Experiment:

To configure RIP routing protocol in routers

Topology:

Connections: Copper-straight Through, Serial DTE

Topological creation:

1) Select 2-PC-PT, 2-Router-PT & 2 switch-PT devices.
2) Now, make the appropriate connections between PC & switch, switch & router, router & router as in topology.

Bafna Gold

Configuration:

1) Configure the ip address for PC-PT & router devices.

PC0 (Fa0) - 192.168.1.2 255.255.255.0
Router0(Fa0/0) - 192.168.1.1 255.255.255.0
Router0(Se2/0) - 10.10.0.2 255.0.0.0
Router1 (Se2/0) - 10.10.0.3 255.0.0.0
Router1 (Fa0/0) - 192.168.2.1 255.255.255.0
Router1 (Fa0) - 192.168.2.2 255.255.255.0

Default Gateway:

PC0 - 192.168.1.1
PC1 - 192.168.2.1

2) Now, go to RIP Routing in Router & add the networks.

Router0 - Network Address
10.0.0.0
192.168.1.0

Router1 - Network Address
10.0.0.0
192.168.2.0

Output:

PC> ping 192.168.2.2

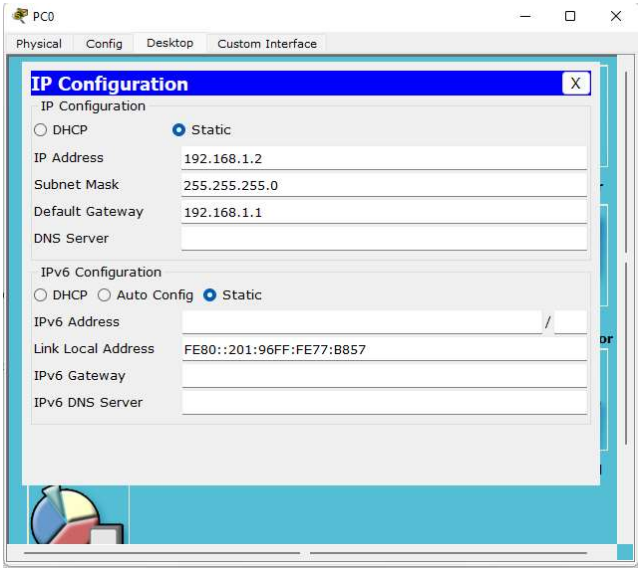
Pinging 192.168.2.2 with 32 bytes of data:

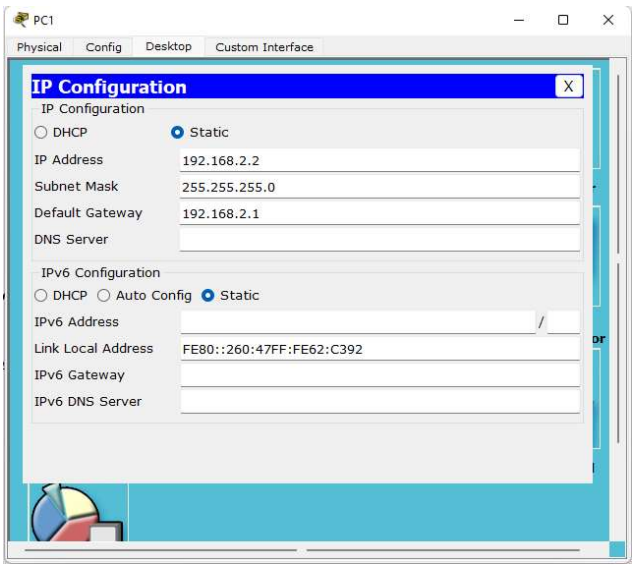
Reply from 192.168.2.2: bytes=32 time=8ms TTL=126
Reply from 192.168.2.2: bytes=32 time=6ms TTL=126
Reply from 192.168.2.2: bytes=32 time=6ms TTL=126
Reply from 192.168.2.2: bytes=32 time=8ms TTL=126

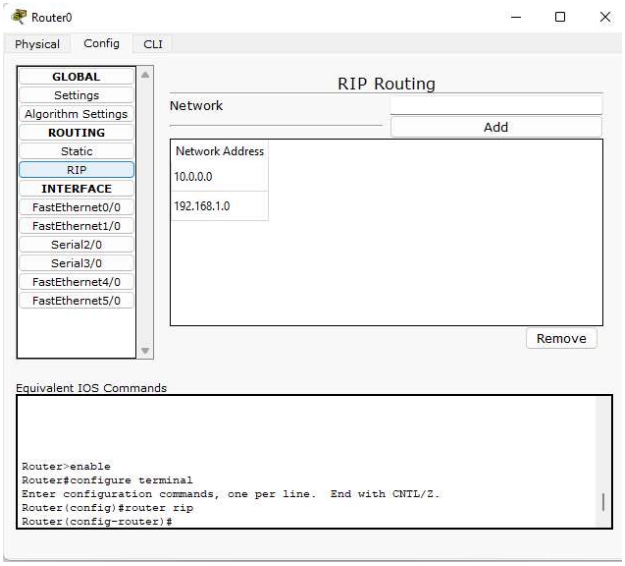
Ping Statistics for 192.168.2.2:

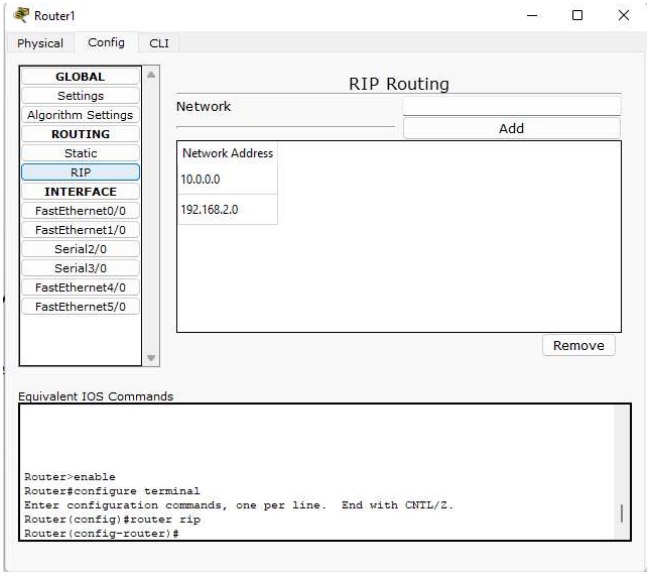
Packets: Sent=4, Recieved=4, Lost=0 (0% loss)

Output:


PC0
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
IP Address
192.168.1.2
Subnet Mask
255.255.255.0
Default Gateway
192.168.1.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::201:96FF:FE77:B857
IPv6 Gateway
IPv6 DNS Server

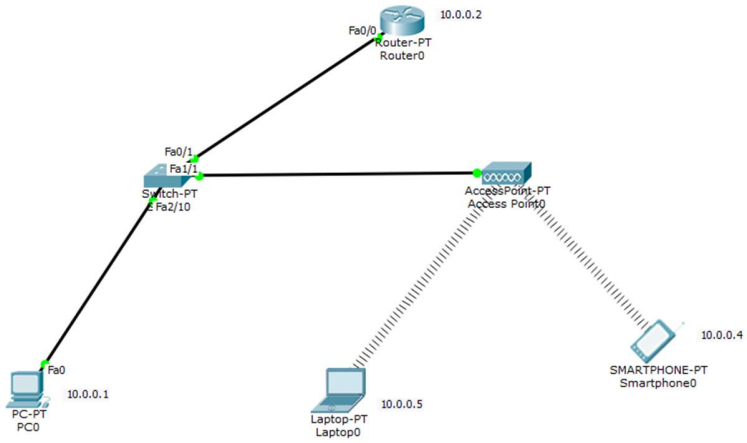

PC1
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
IP Address
192.168.2.2
Subnet Mask
255.255.255.0
Default Gateway
192.168.2.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::260:47FF:FE62:C392
IPv6 Gateway
IPv6 DNS Server


Router0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
RIP Routing
Network
Add
Network Address
10.0.0.0
192.168.1.0
Remove
Equivalent IOS Commands
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#

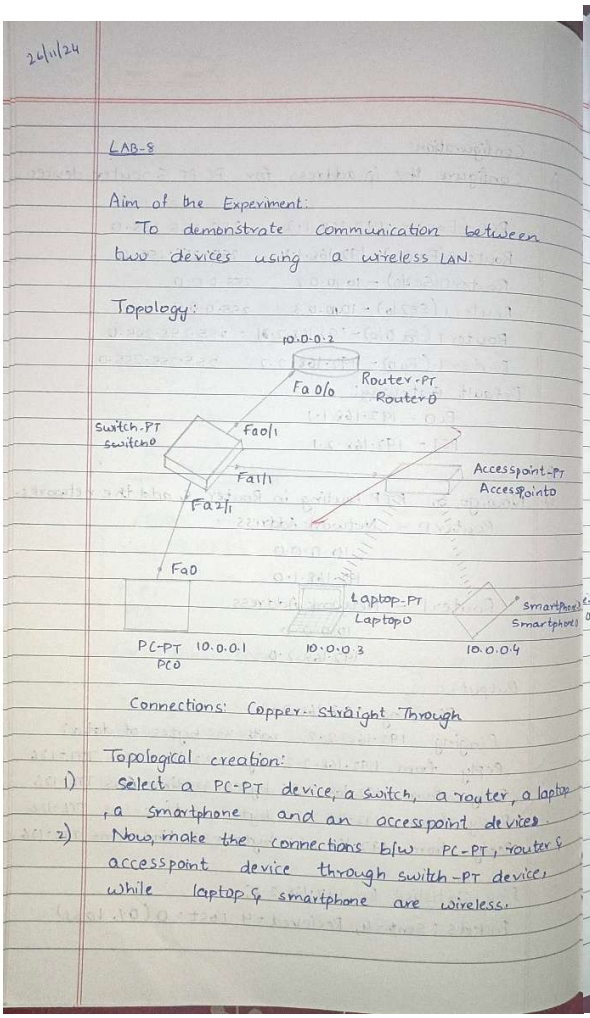

Router1
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
RIP Routing
Network
Add
Network Address
10.0.0.0
192.168.2.0
Remove
Equivalent IOS Commands
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#

Program 9: To demonstrate communication between 2 devices using a wireless LAN.

Topology:



Observation:


24/11/24

LAB-8

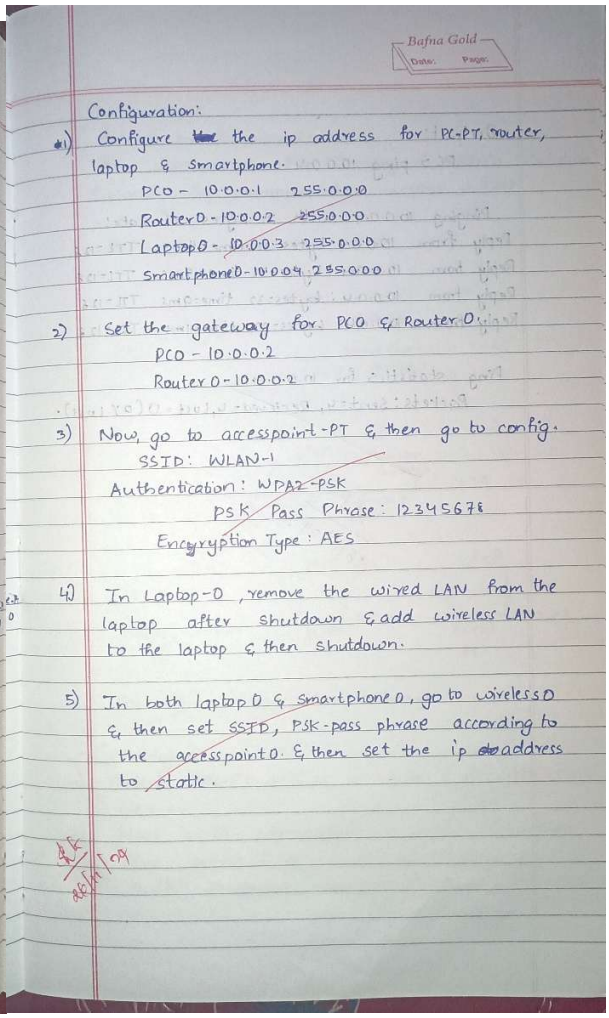
Aim of the Experiment:
To demonstrate communication between two devices using a wireless LAN.

Topology:

Connections: Copper Straight Through

Topological creation:

1) Select a PC-PT device, a switch, a router, a laptop, a smartphone and an accesspoint device.
2) Now, make the connections b/w PC-PT, router & accesspoint device through switch-PT device, while laptop & smartphone are wireless.


Bafna Gold
Date: Page:

Configuration:

1) Configure the ip address for PC-PT, router, laptop & smartphone.
PC0 - 10.0.0.1 255.0.0.0
Router0 - 10.0.0.2 255.0.0.0
Laptop0 - 10.0.0.3 255.0.0.0
Smartphone0 - 10.0.0.4 255.0.0.0

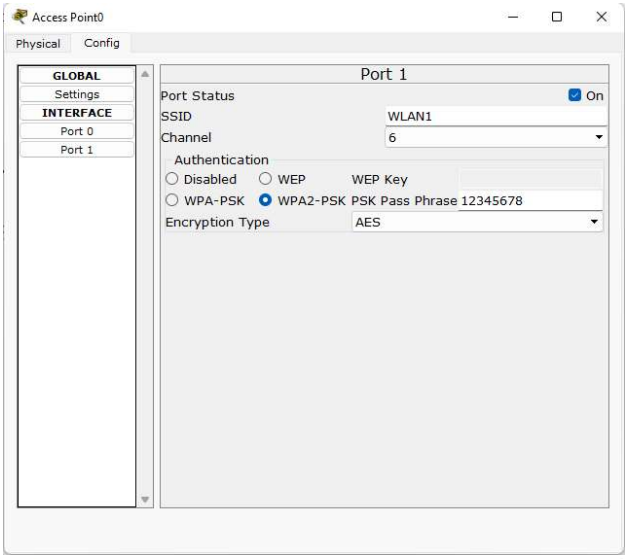
2) Set the gateway for PC0 & Router0.
PC0 - 10.0.0.2
Router0 - 10.0.0.2

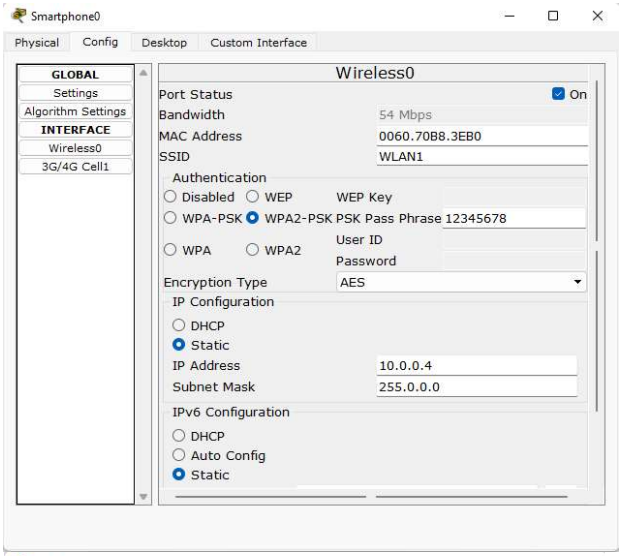
3) Now, go to accesspoint-PT & then go to config.
SSID: WLAN-1
Authentication: WPA2-PSK
PSK Pass Phrase: 12345678
Encryption Type: AES

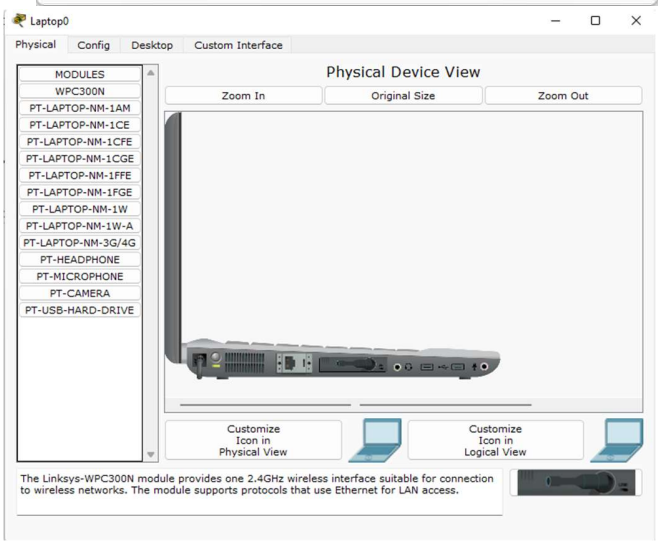
4) In Laptop-0, remove the wired LAN from the laptop after shutdown & add wireless LAN to the laptop & then shutdown.

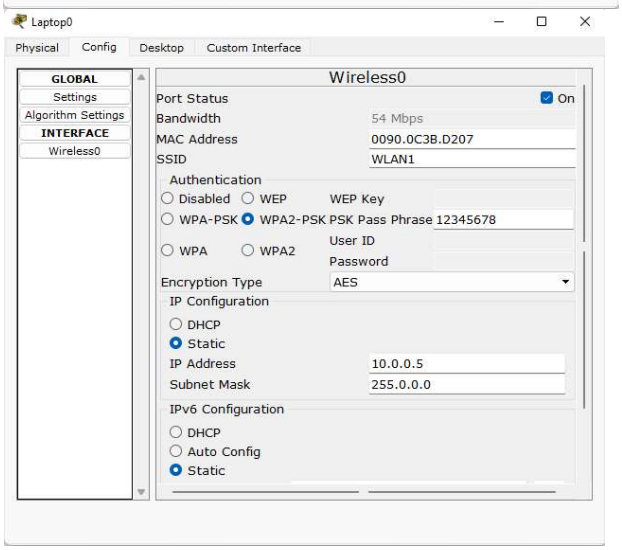
5) In both laptop 0 & smartphone 0, go to wireless 0 & then set SSID, PSK-pass phrase according to the accesspoint 0. & then set the ip address to static.

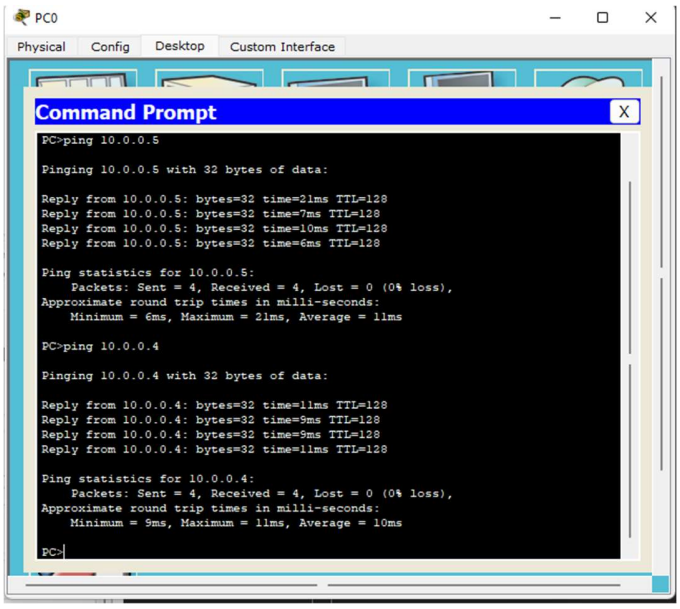
Output:


Access Point0
Physical
Config
GLOBAL
Settings
INTERFACE
Port 0
Port 1
Port 1
Port Status
On
SSID
WLAN1
Channel
6
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
12345678
Encryption Type
AES


Smartphone0
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
3G/4G Cell1
Wireless0
Port Status
On
Bandwidth
54 Mbps
MAC Address
0060.70B8.3EB0
SSID
WLAN1
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
12345678
WPA
WPA2
User ID
Password
Encryption Type
AES
IP Configuration
DHCP
Static
IP Address
10.0.0.4
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static

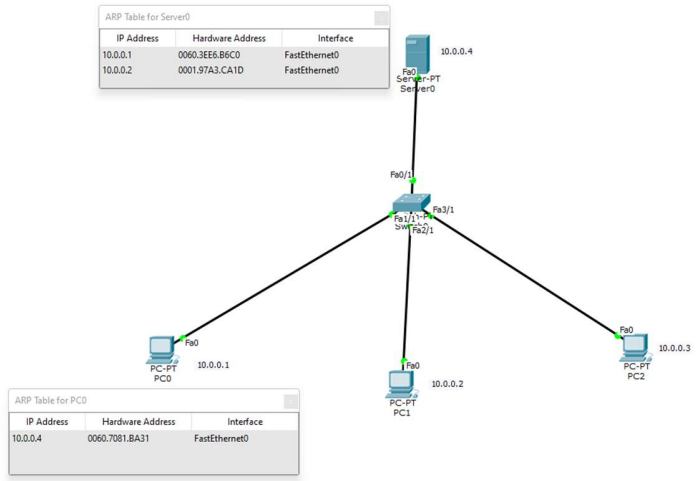

Laptop0
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize
Icon in
Physical View
Customize
Icon in
Logical View
The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.


Laptop0
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
Wireless0
Port Status
On
Bandwidth
54 Mbps
MAC Address
0090.0C3B.D207
SSID
WLAN1
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
12345678
WPA
WPA2
User ID
Password
Encryption Type
AES
IP Configuration
DHCP
Static
IP Address
10.0.0.5
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static

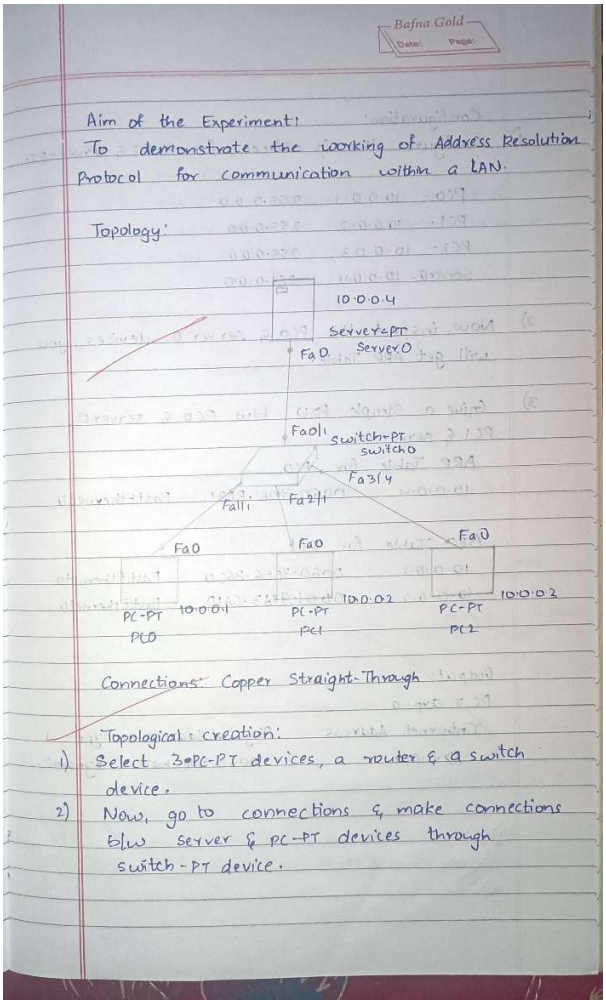

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
PC>ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time=21ms TTL=128
Reply from 10.0.0.5: bytes=32 time=7ms TTL=128
Reply from 10.0.0.5: bytes=32 time=10ms TTL=128
Reply from 10.0.0.5: bytes=32 time=6ms TTL=128
Ping statistics for 10.0.0.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 21ms, Average = 11ms
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=11ms TTL=128
Reply from 10.0.0.4: bytes=32 time=9ms TTL=128
Reply from 10.0.0.4: bytes=32 time=9ms TTL=128
Reply from 10.0.0.4: bytes=32 time=11ms 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 = 9ms, Maximum = 11ms, Average = 10ms
PC>

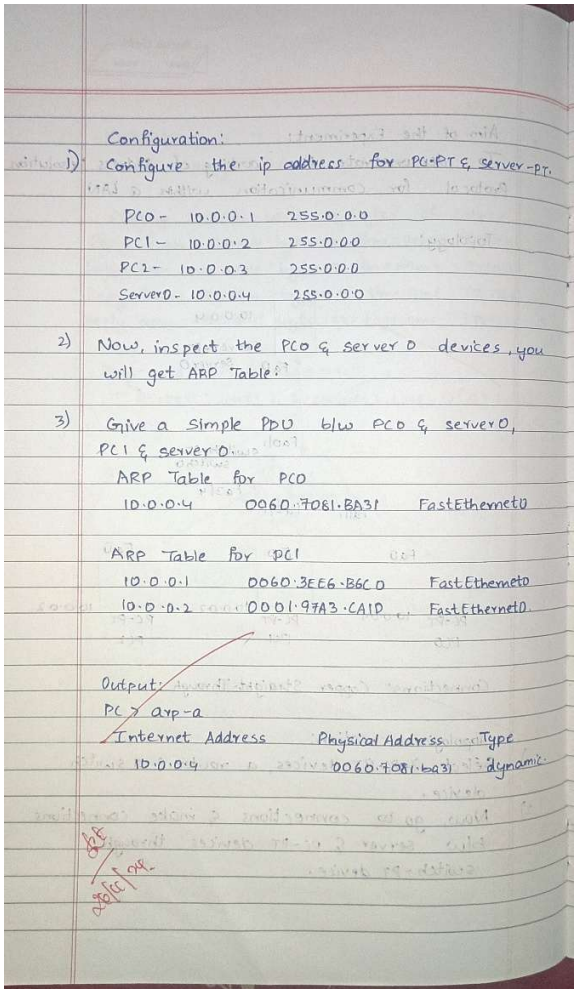
Program 10: To demonstrate the working of Address Resolution Protocol for communication within a LAN.

Topology:

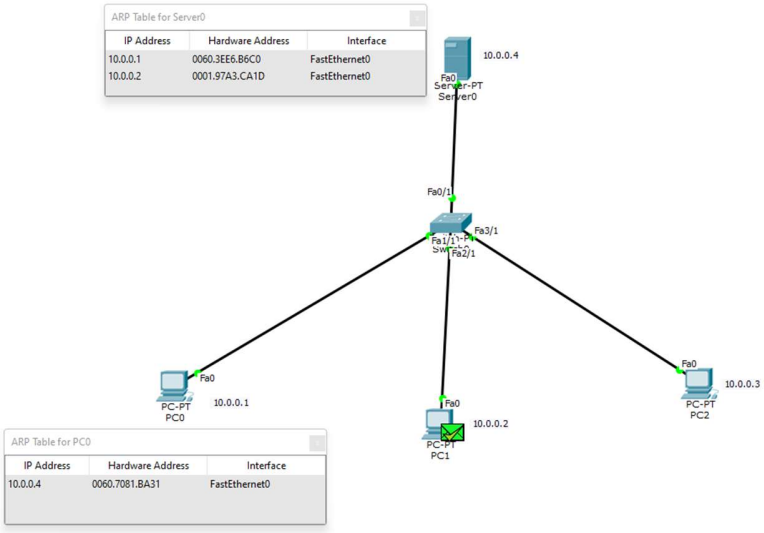


Observation:





Output:



ARP Table for PC0

IP Address	Hardware Address	Interface
10.0.0.4	0060.7081.BA31	FastEthernet0

ARP Table for Server0

IP Address	Hardware Address	Interface
10.0.0.1	0060.3EE6.B6C0	FastEthernet0
10.0.0.2	0001.97A3.CA1D	FastEthernet0

PDU Information at Device: PC1

OSI Model | Inbound PDU Details

PDU Formats

Ethernet II

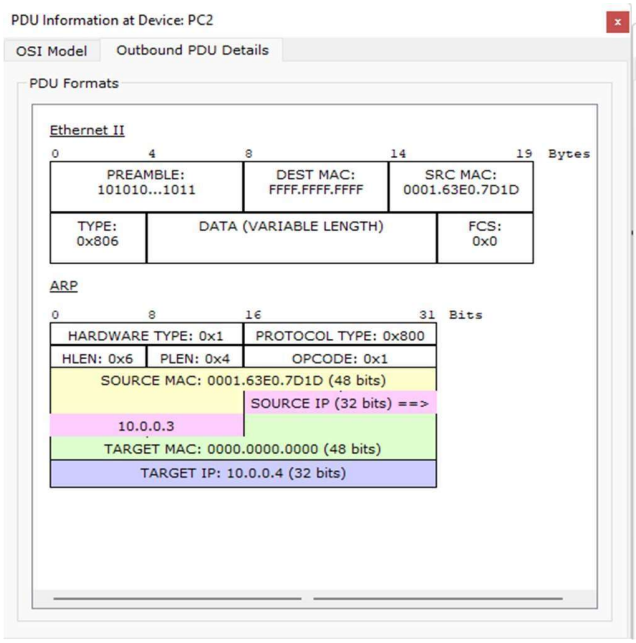
0	4	8	14	19 Byt
PREAMBLE: 101010...1011		DEST MAC: 0001.97A3.CA1D	SRC MAC: 0060.7081.BA31	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

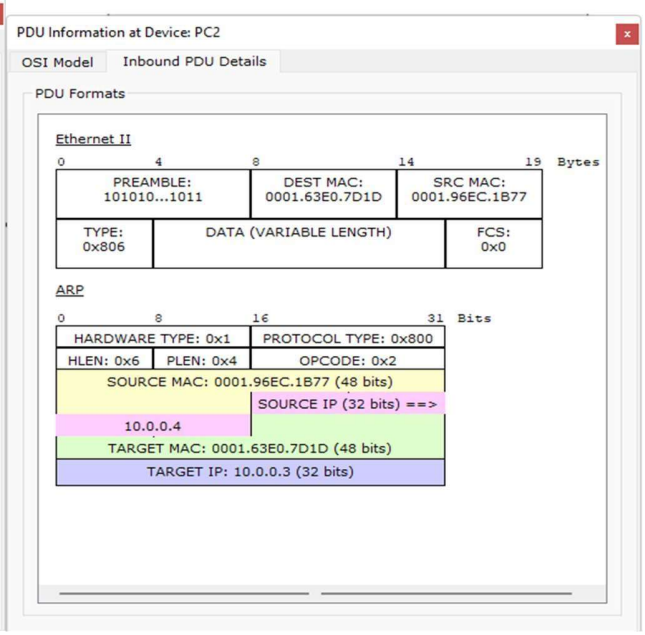
IP

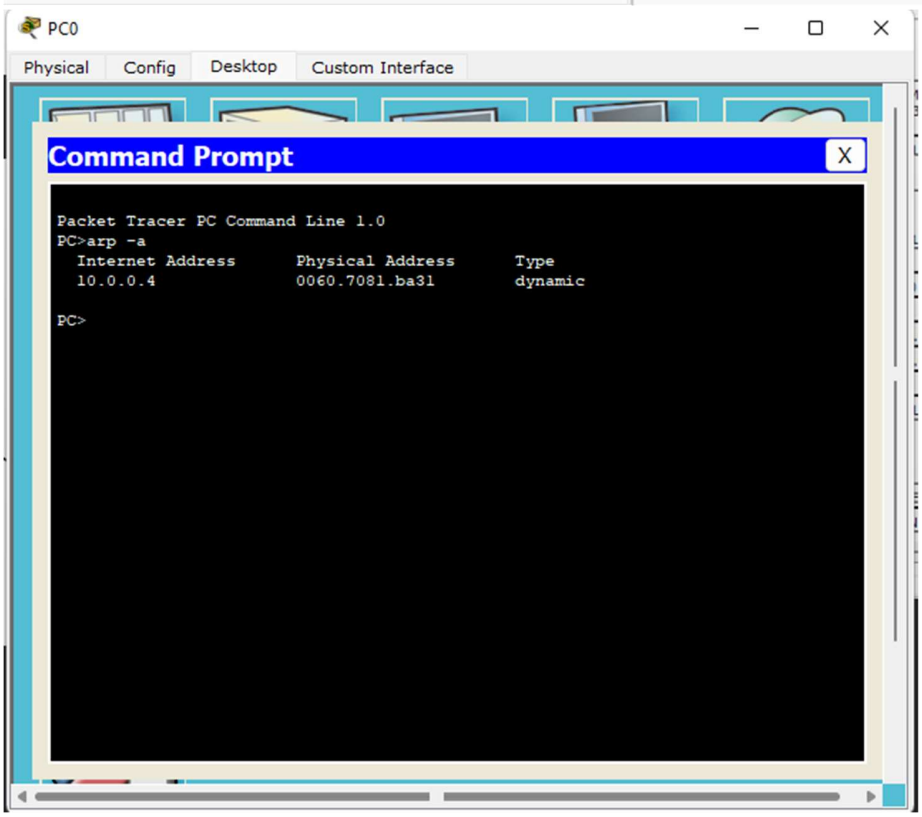
0	4	8	16 19	31 Bits
4	IHL	DSCP: 0x0	TL: 28	
ID: 0xd		0x0	0x0	
TTL: 128		PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.4				
DST IP: 10.0.0.2				
OPT: 0x0			0x0	
DATA (VARIABLE LENGTH)				

ICMP

0	8	16	31 Bits
TYPE: 0x0	CODE: 0x0	CHECKSUM	
ID: 0x7		SEQ NUMBER: 6	

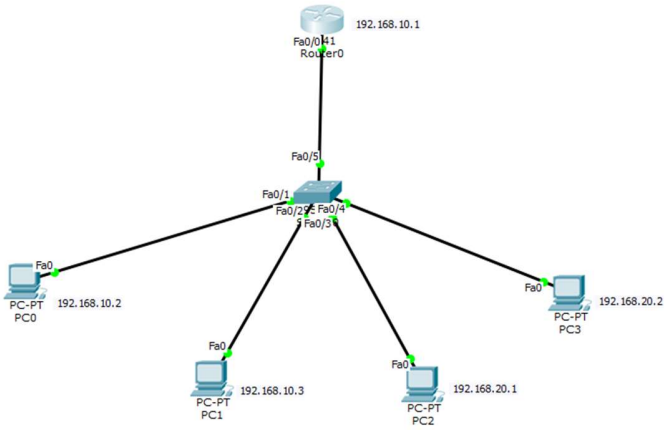

PDU Information at Device: PC2
OSI Model
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Bytes
PREAMBLE: 101010...1011
DEST MAC: FFFF.FFFF.FFFF
SRC MAC: 0001.63E0.7D1D
TYPE: 0x806
DATA (VARIABLE LENGTH)
FCS: 0x0
ARP
0 8 16 31 Bits
HARDWARE TYPE: 0x1
PROTOCOL TYPE: 0x800
HLEN: 0x6
PLEN: 0x4
OPCODE: 0x1
SOURCE MAC: 0001.63E0.7D1D (48 bits)
SOURCE IP (32 bits) ==>
10.0.0.3
TARGET MAC: 0000.0000.0000 (48 bits)
TARGET IP: 10.0.0.4 (32 bits)


PDU Information at Device: PC2
OSI Model
Inbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Bytes
PREAMBLE: 101010...1011
DEST MAC: 0001.63E0.7D1D
SRC MAC: 0001.96EC.1B77
TYPE: 0x806
DATA (VARIABLE LENGTH)
FCS: 0x0
ARP
0 8 16 31 Bits
HARDWARE TYPE: 0x1
PROTOCOL TYPE: 0x800
HLEN: 0x6
PLEN: 0x4
OPCODE: 0x2
SOURCE MAC: 0001.96EC.1B77 (48 bits)
SOURCE IP (32 bits) ==>
10.0.0.4
TARGET MAC: 0001.63E0.7D1D (48 bits)
TARGET IP: 10.0.0.3 (32 bits)

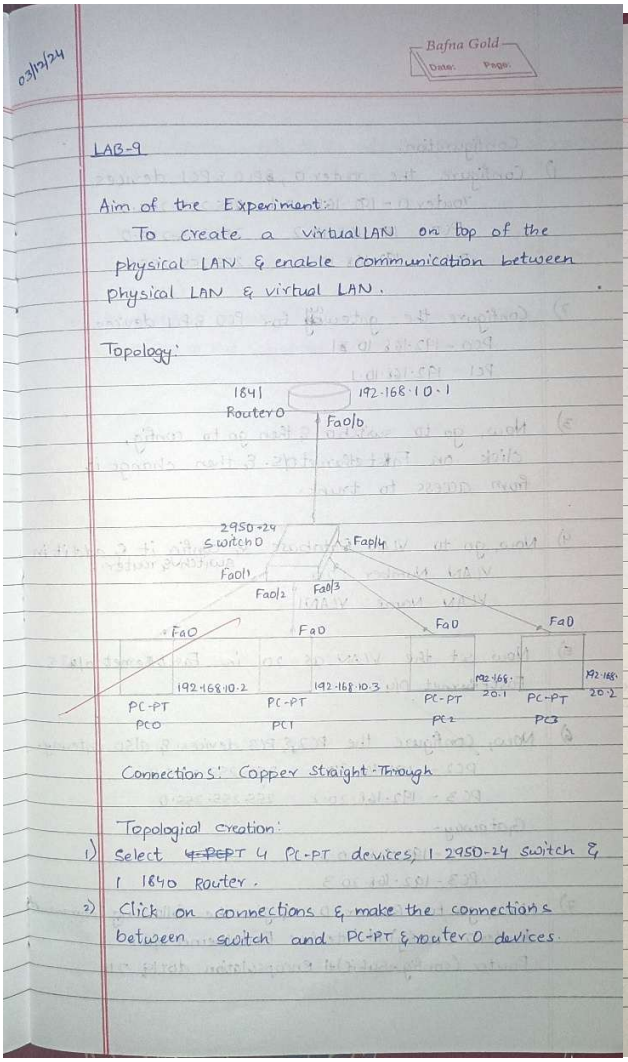

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>arp -a
Internet Address Physical Address Type
10.0.0.4 0060.7081.ba31 dynamic
PC>

Program 11: To create a Virtual LAN on top of physical LAN and enable communication between physical LAN and VLAN.

Topology:



Observation:



03/12/24

Bafna Gold

LAB-9

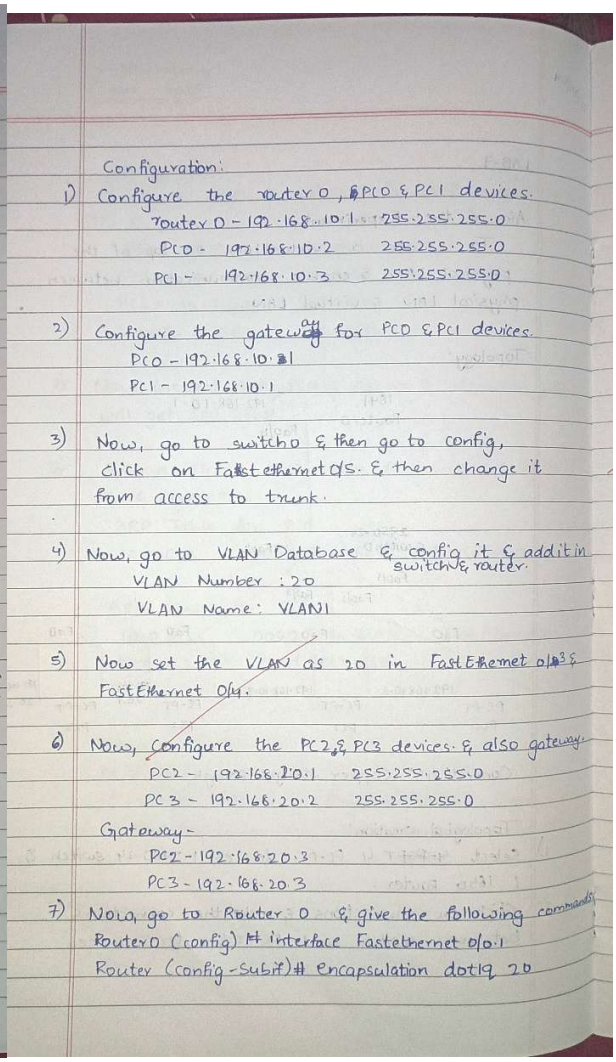
Aim of the Experiment:
To create a virtual LAN on top of the physical LAN & enable communication between physical LAN & virtual LAN.

Topology:

1841 Router0 — 192.168.10.1 — Fa0/0
2950-24 Switch0 — Fa0/4, Fa0/1, Fa0/2, Fa0/3
Fa0 — PC-PT PC0 — 192.168.10.2
Fa0 — PC-PT PC1 — 192.168.10.3
Fa0 — PC-PT PC2 — 192.168.20.1
Fa0 — PC-PT PC3 — 192.168.20.2

Connections: Copper Straight-Through

Topological creation:
1) Select 4 PC-PT & PC-PT devices, 1 2950-24 switch & 1 1840 Router.
2) Click on connections & make the connections between switch and PC-PT & router 0 devices.



Configuration:
1) Configure the router 0, PC0 & PC1 devices.
Router 0 - 192.168.10.1 255.255.255.0
PC0 - 192.168.10.2 255.255.255.0
PC1 - 192.168.10.3 255.255.255.0

2) Configure the gateway for PC0 & PC1 devices.
PC0 - 192.168.10.1
PC1 - 192.168.10.1

3) Now, go to switch0 & then go to config, click on Fastethernet 0/5. & then change it from access to trunk.

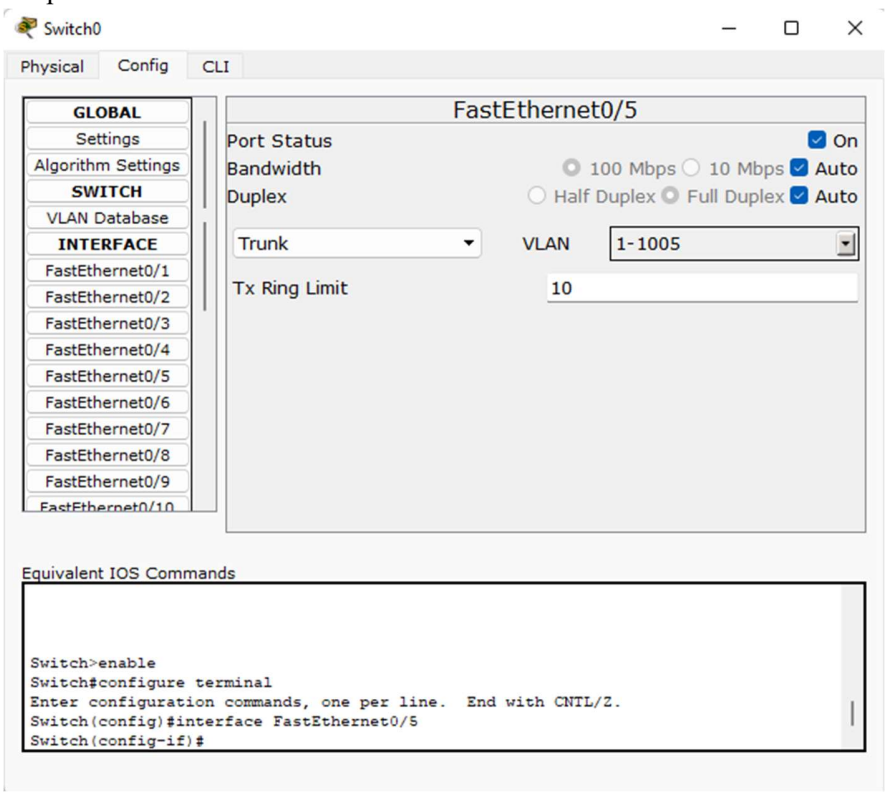
4) Now, go to VLAN Database & config it & add it in switch & router.
VLAN Number : 20
VLAN Name: VLAN1

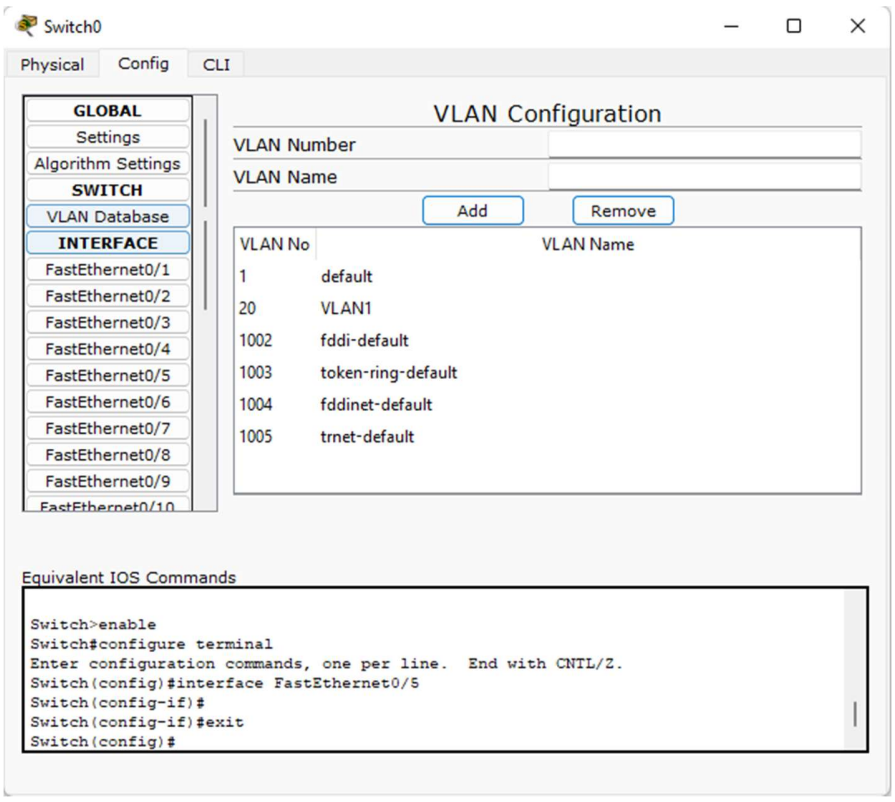
5) Now set the VLAN as 20 in FastEthernet 0/3 & FastEthernet 0/4.

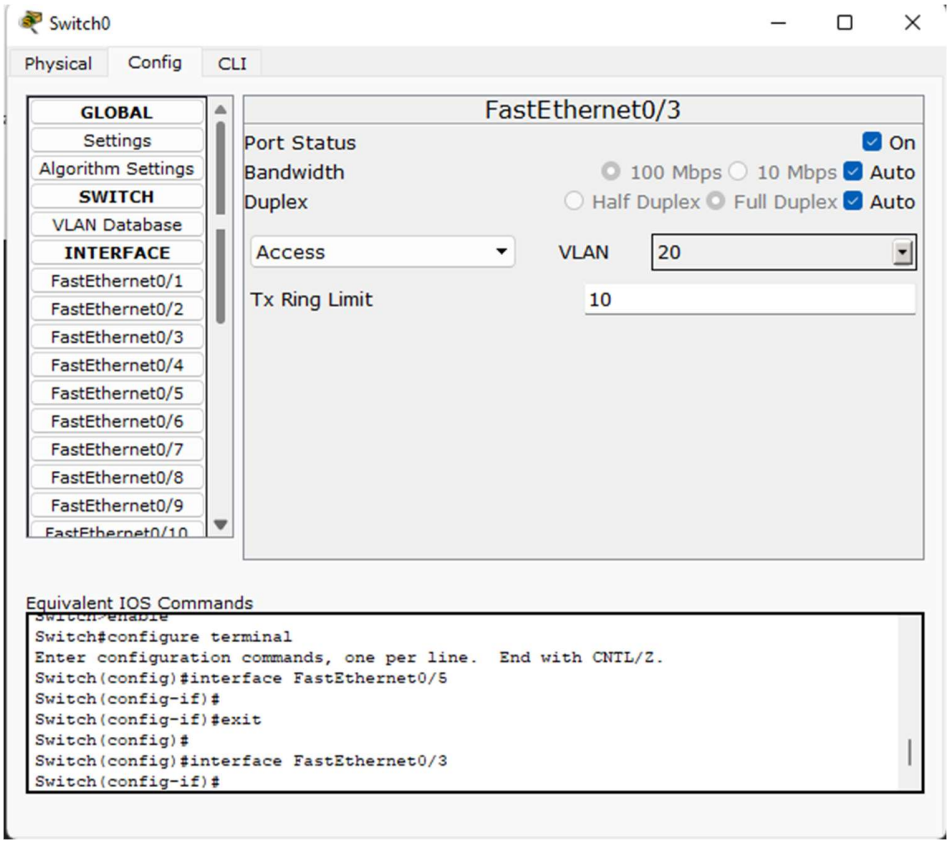
6) Now, configure the PC2 & PC3 devices. & also gateway.
PC2 - 192.168.20.1 255.255.255.0
PC3 - 192.168.20.2 255.255.255.0
Gateway -
PC2 - 192.168.20.3
PC3 - 192.168.20.3

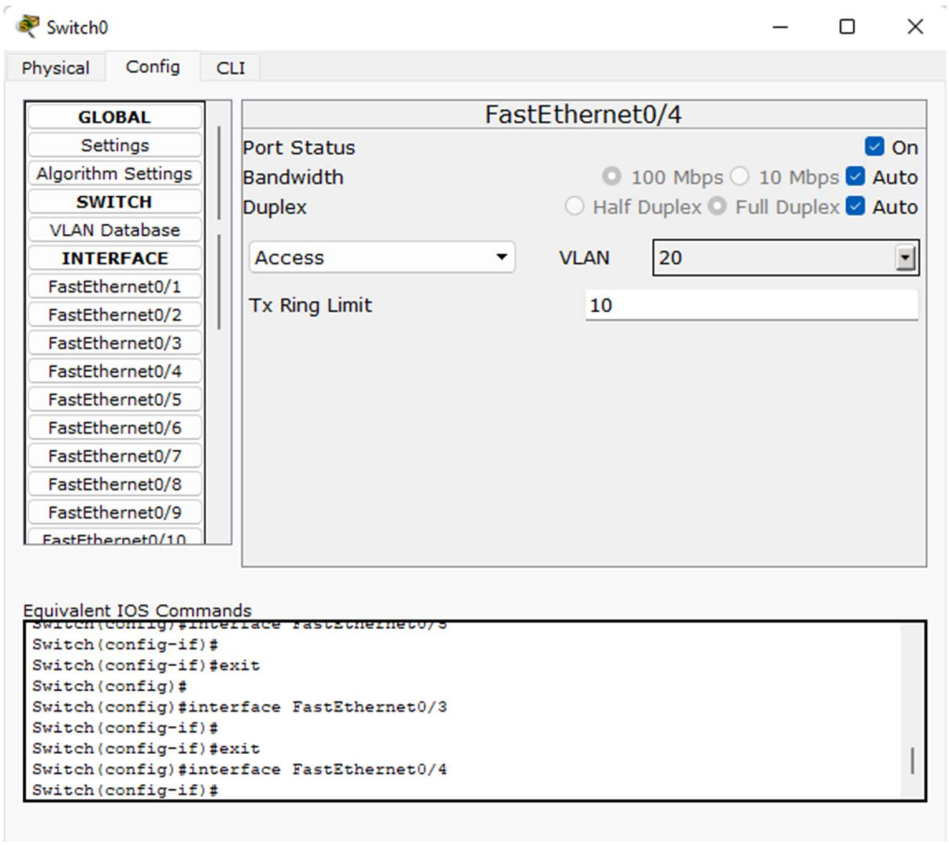
7) Now, go to Router 0 & give the following commands
Router0 (config) # interface Fastethernet 0/0.1
Router (config-subif)# encapsulation dot1q 20

Output:


Switch0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet0/2
FastEthernet0/3
FastEthernet0/4
FastEthernet0/5
FastEthernet0/6
FastEthernet0/7
FastEthernet0/8
FastEthernet0/9
FastEthernet0/10
FastEthernet0/5
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
Trunk
VLAN
1-1005
Tx Ring Limit
10
Equivalent IOS Commands
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface FastEthernet0/5
Switch(config-if)#


Switch0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet0/2
FastEthernet0/3
FastEthernet0/4
FastEthernet0/5
FastEthernet0/6
FastEthernet0/7
FastEthernet0/8
FastEthernet0/9
FastEthernet0/10
VLAN Configuration
VLAN Number
VLAN Name
Add
Remove
VLAN No
VLAN Name
1 default
20 VLAN1
1002 fddi-default
1003 token-ring-default
1004 fddinet-default
1005 trnet-default
Equivalent IOS Commands
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface FastEthernet0/5
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#


Switch0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet0/2
FastEthernet0/3
FastEthernet0/4
FastEthernet0/5
FastEthernet0/6
FastEthernet0/7
FastEthernet0/8
FastEthernet0/9
FastEthernet0/10
FastEthernet0/3
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
Access
VLAN
20
Tx Ring Limit
10
Equivalent IOS Commands
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface FastEthernet0/5
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#
Switch(config)#interface FastEthernet0/3
Switch(config-if)#


Switch0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet0/2
FastEthernet0/3
FastEthernet0/4
FastEthernet0/5
FastEthernet0/6
FastEthernet0/7
FastEthernet0/8
FastEthernet0/9
FastEthernet0/10
FastEthernet0/4
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
Access
VLAN
20
Tx Ring Limit
10
Equivalent IOS Commands
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#
Switch(config)#interface FastEthernet0/3
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/4
Switch(config-if)#

Router0

Physical | Config | CLI

IOS Command Line Interface

```
Router0(config)#interface FastEthernet0/0.1
Router0(config-subif)#ip address 192.168.20.3 255.255.255.0

% Configuring IP routing on a LAN subinterface is only allowed if that
subinterface is already configured as part of an IEEE 802.10, IEEE 802.1Q,
or ISL vLAN.

Router0(config-subif)#
Router0(config-subif)#exit
```

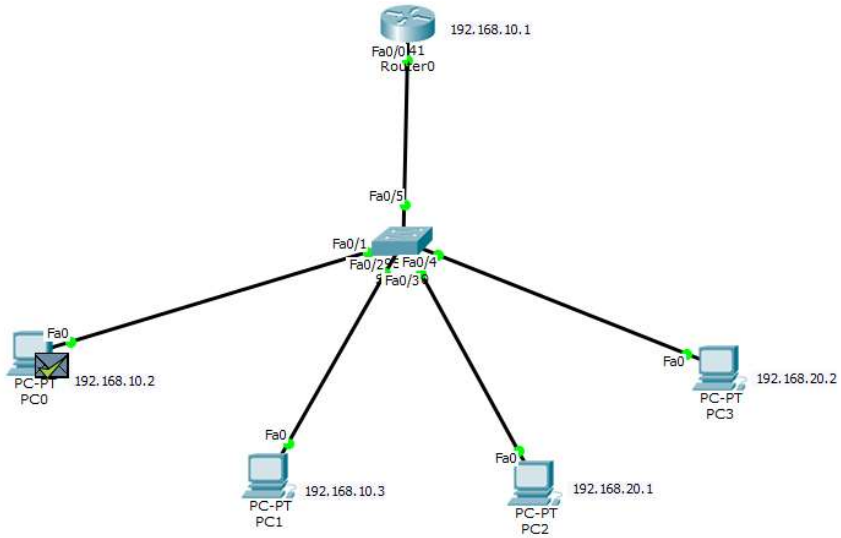
Router0

Physical | Config | CLI

IOS Command Line Interface

```
Router0#vlan database
% Warning: It is recommended to configure VLAN from config mode,
  as VLAN database mode is being deprecated. Please consult user
  documentation for configuring VTP/VLAN in config mode.

Router0(vlan)#
%SYS-5-CONFIG_I: Configured from console by console
vlan 20 name VLAN1
VLAN 20 modified:
    Name: VLAN1
Router0(vlan)#exit
APPLY completed.
Exiting....
Router0#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router0(config)#interface fastethernet 0/0.1
Router0(config-subif)#encapsulation dot1q 20
Router0(config-subif)#ip address 192.168.20.3 255.255.255.0
Router0(config-subif)#no shutdown
Router0(config-subif)#


Router0 con0 is now available
```

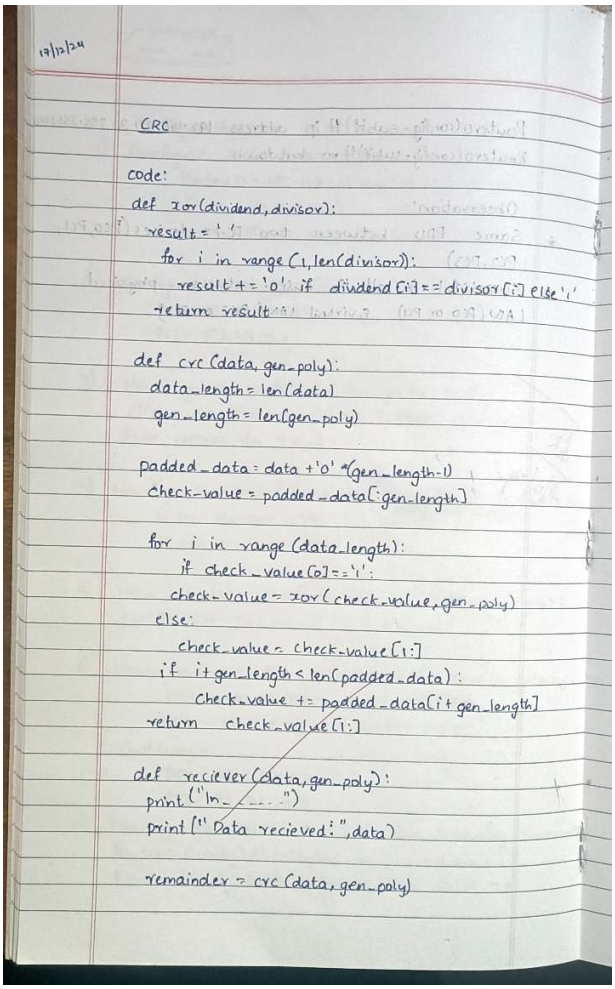
Copy | Paste



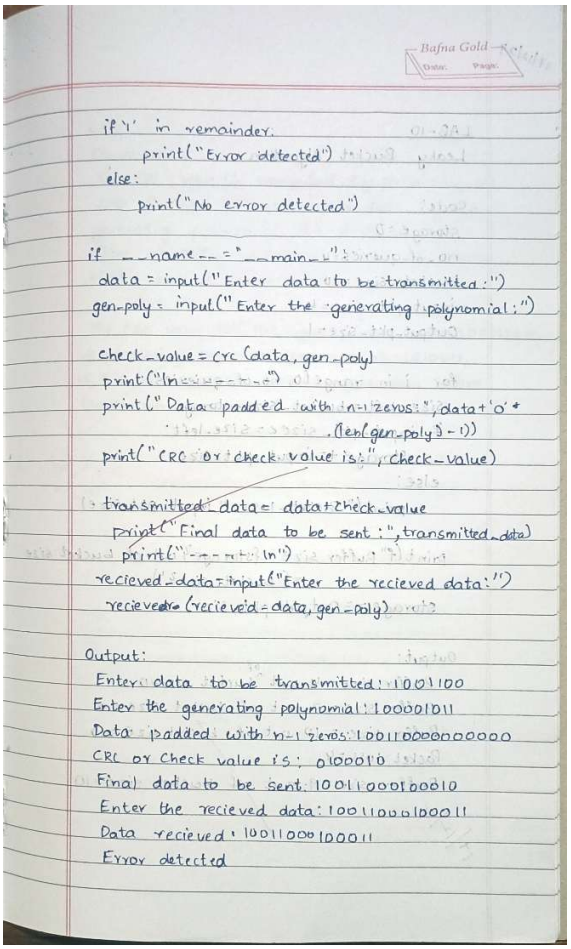
CYCLE - II

Program 1: Write a program for error detecting code using CRC (8-bits).

Observation:



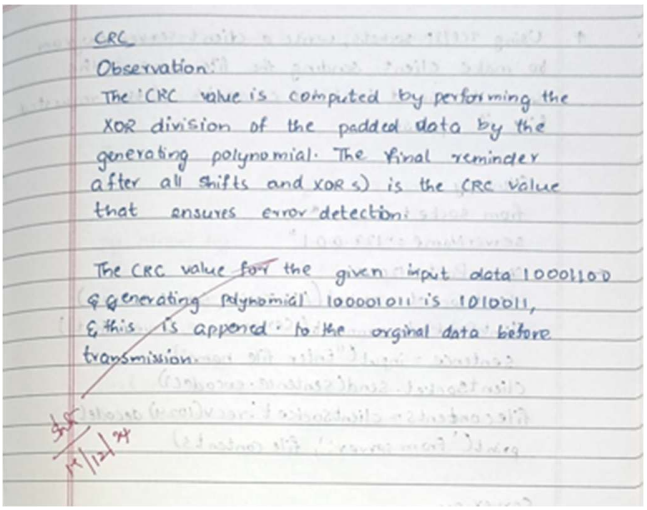
```
CRC

code:
def xor(dividend, 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):
    data_length = len(data)
    gen_length = len(gen_poly)

    padded_data = data + '0' * (gen_length-1)
    check_value = padded_data[:gen_length]

    for i in range(data_length):
        if check_value[0] == '1':
            check_value = xor(check_value, gen_poly)
        else:
            check_value = check_value[1:]
        if i+gen_length < len(padded_data):
            check_value += padded_data[i+gen_length]
    return check_value[1:]

def reciever(data, gen_poly):
    print("\n-------")
    print("Data recieved:", data)

    remainder = crc(data, gen_poly)
```



```
    if '1' in remainder:
        print("Error detected")
    else:
        print("No error detected")

if __name__ == "__main__":
    data = input("Enter data to be transmitted:")
    gen_poly = input("Enter the generating polynomial:")

    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)

    transmitted_data = data + check_value
    print("Final data to be sent:", transmitted_data)
    print("-------\n")
    recieved_data = input("Enter the recieved data:")
    reciever(recieved_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 1001100 0100010
Enter the recieved data: 1001100 0100011
Data recieved: 1001100 0100011
Error detected
```



CRC

Observation:

The CRC value is computed by performing the XOR division of the padded data by the generating polynomial. The final reminder after all shifts and XORs) is the CRC value that ensures error detection.

The CRC value for the given input data 1000110 & generating polynomial 100001011 is 1010011, & this is appened to the orginal data before transmission.

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
```

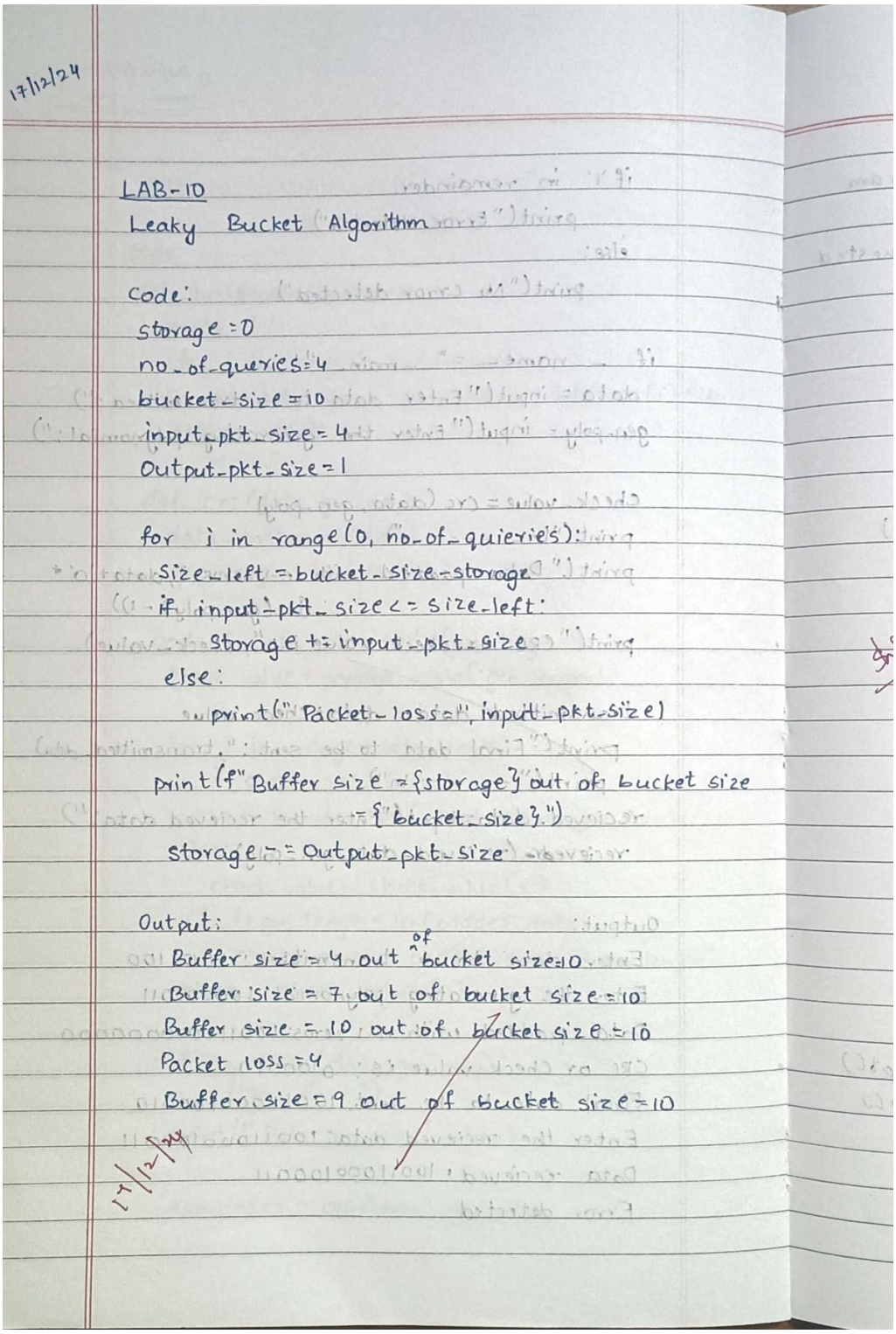
```
data = input("Enter data to be transmitted: ")
gen_poly = input("Enter the Generating polynomial: ")

# 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: 10001100
Enter the Generating polynomial: 100001011

----------------------------------------
Data padded with n-1 zeros: 1000110000000000
CRC or Check value is: 1010011
Final data to be sent: 100011001010011
----------------------------------------

Enter the received data: 10011000100011

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

Program 2: Write a program for congestion control using Leaky bucket algorithm.

Observation:



17/12/24

LAB-10

Leaky Bucket Algorithm

Code:

```
storage = 0
no_of_queries = 4
bucket_size = 10
input_pkt_size = 4
Output_pkt_size = 1

for i in range(0, no_of_quieries):
    size_left = bucket_size - storage
    if input_pkt_size <= size_left:
        storage += input_pkt_size
    else:
        print("Packet_loss=", input_pkt_size)
    print(f"Buffer size = {storage} out of bucket size = {bucket_size}.")
    storage -= Output_pkt_size
```

Output:

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

17/12/24

Code:

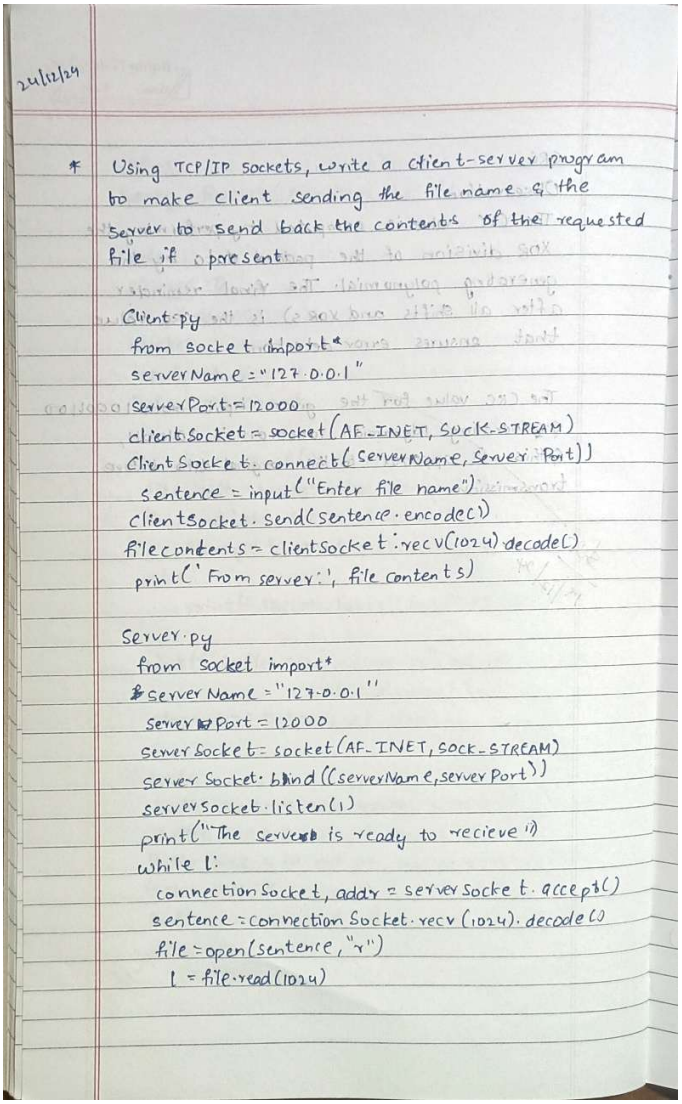
```
# initial packets in the bucket
storage = 0

# total no. of times bucket content is checked
no_of_queries = 4

# total no. of packets that can
# be accommodated in the bucket
bucket_size = 10

# no. of packets that enters the bucket at a time
input_pkt_size = 4

# no. of packets that exits the bucket at a time
output_pkt_size = 1
for i in range(0, 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:

```
Buffer size= 4 out of bucket size = 10
Buffer size= 7 out of bucket size = 10
Buffer size= 10 out of bucket size = 10
Packet loss =  4
Buffer size= 9 out of bucket size = 10
```

Program 3: 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.

Observation:



24/12/24

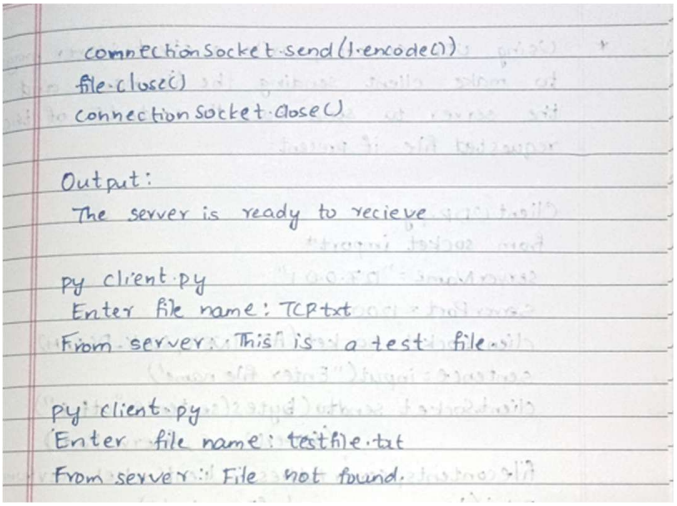
* Using TCP/IP sockets, write a client-server program to make client sending the file name & the server to send back the contents of the requested file if present

Client.py

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

Server.py

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



```
    connectionSocket.send(l.encode())
    file.close()
    connectionSocket.close()
```

Output:

The server is ready to recieve

py client.py
Enter file name: TCP.txt
From server: This is a test file.

py client.py
Enter file name: testfile.txt
From server: File not found.

Code:
Client.py

```
from socket import *
serverName = “127.0.0.1”
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input(&amp;quot;Enter file name&amp;quot;)
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print (&amp;#39;From Server:&amp;#39;, filecontents)
clientSocket.close()
```

Server.py

```
from socket import *
serverName=”127.0.0.1”
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
print (&amp;quot;The server is ready to receive&amp;quot;)
while 1:
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,&amp;quot;r&amp;quot;)
l=file.read(1024)
connectionSocket.send(l.encode())
file.close()
connectionSocket.close()
```

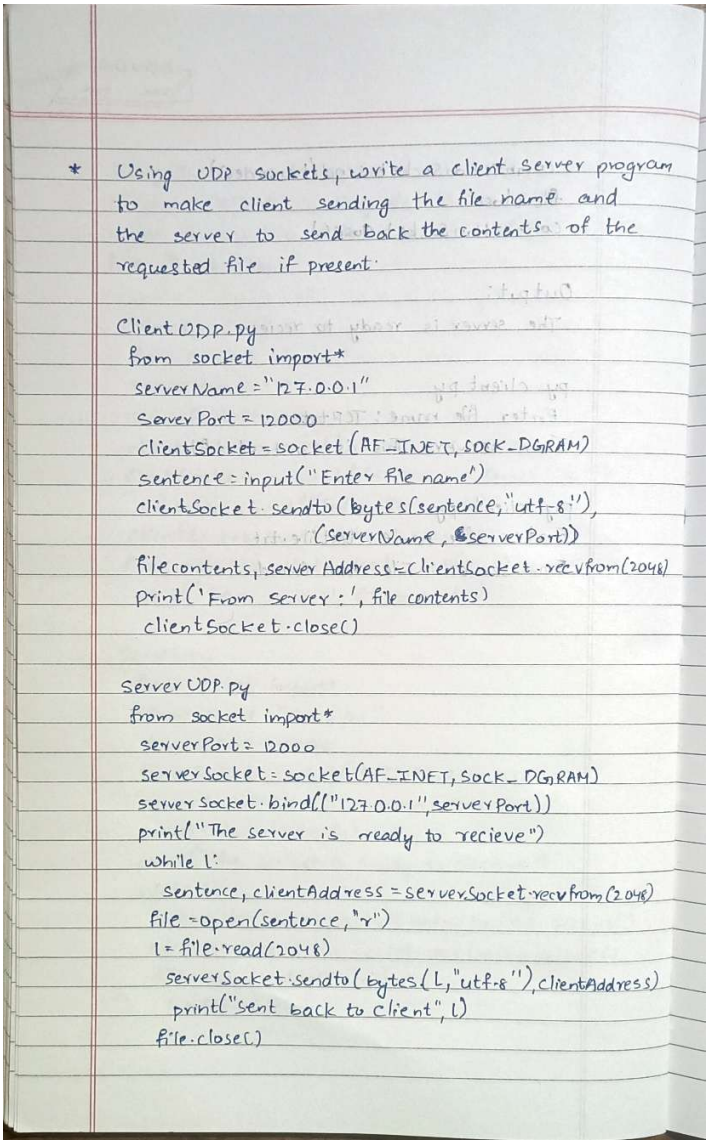
Output:

```
C:\Users\rajas\OneDrive\Documents\Server and Client>python Server.py
The server is ready to receive
```

```
PS C:\Users\rajas\OneDrive\Documents\Server and Client> python Client.py
Enter file name: TCP.txt
From Server: This is a test file.

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.

PS C:\Users\rajas\OneDrive\Documents\Server and Client>
```

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.

Observation:



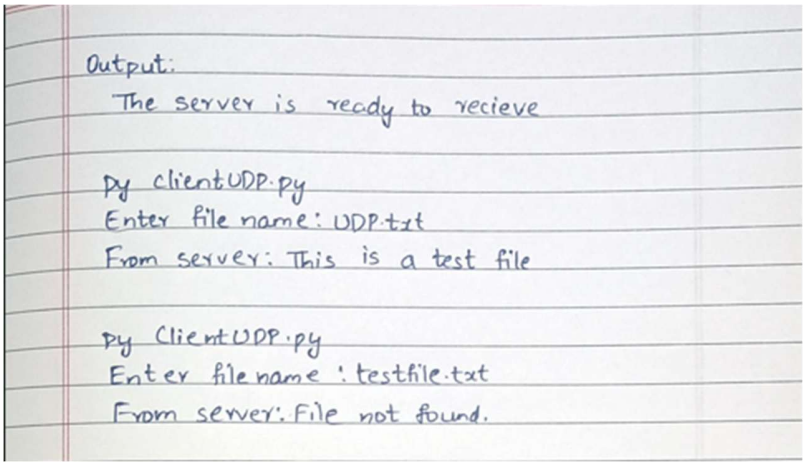
* Using UDP sockets, write a client server program to make client sending the file name and the server to send back the contents of the requested file if present.

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('From server:', file contents)
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 recieve")
while 1:
  sentence, clientAddress = serverSocket.recvfrom(2048)
  file = open(sentence,"r")
  l = file.read(2048)
  serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
  print("Sent back to client", l)
  file.close()
```



Output:

```
The server is ready to recieve

py clientUDP.py
Enter file name: UDP.txt
From server: This is a test file

py ClientUDP.py
Enter file name : testfile.txt
From server: File not found.
```

Code:
ClientUDP.py

```
from socket import *
serverName = &amp;quot;127.0.0.1&amp;quot;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input(&amp;quot;Enter file name&amp;quot;)
clientSocket.sendto(bytes(sentence,&amp;quot;utf-8&amp;quot;),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print (&amp;#39;From Server:&amp;#39;, filecontents)
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((&amp;quot;127.0.0.1&amp;quot;, serverPort))
print (&amp;quot;The server is ready to receive&amp;quot;)
while 1:
sentence,clientAddress = serverSocket.recvfrom(2048)
file=open(sentence,&amp;quot;r&amp;quot;)
l=file.read(2048)
serverSocket.sendto(bytes(l,&amp;quot;utf-8&amp;quot;),clientAddress)
print(&amp;quot;sent back to client&amp;quot;,l)
file.close()
```

Output:

```
C:\Users\rajas\OneDrive\Documents\Server and Client>python ServerUDP.py
The server is ready to receive
```

```
PS C:\Users\rajas\OneDrive\Documents\Server and Client> python ClientUDP.py
Enter file name: UDP.txt
From Server: This is a test file.

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.
PS C:\Users\rajas\OneDrive\Documents\Server and Client>
```