VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

(23CS5PCCON)

Submitted by

ATHMICA N (1BM22CS058)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

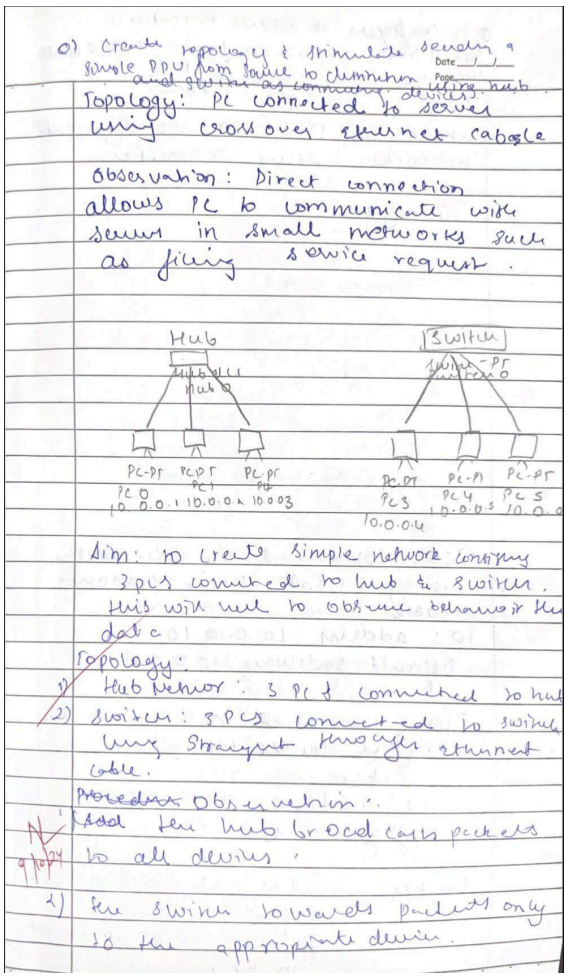
CERTIFICATE

This is to certify that the Lab work entitled "**COMPUTER NETWORKS (23CS5PCCON)" carried out by ATHMICA N (1BM22CS058),** who is a bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of a COMPUTER NETWORKS (23CS5PCCON) work prescribed for the said degree.

Dr. Nandhini Vineeth Associate Professor, Department of CSE, BMSCE	Dr. KAVITHA SOODA Professor & HOD Department of CSE, BMSCE

Index

Hub and Switch Experiment



01) Create topology & simulate sending a simple PDU from source to destination using hub and switch as connecting devices.

Topology: PC connected to server using cross over ethernet cable

Observation: Direct connection allows PC to communicate with server in small networks such as filing service request.

Hub

PC-PT PC-PT PC-PT
PC0 PC1 PC2
10.0.0.1 10.0.0.2 10.0.0.3

Switch

PC-PT PC-PT PC-PT
PC3 PC4 PC5
10.0.0.4 10.0.0.5 10.0.0.6

Aim: To create simple network consisting 3 PCs connected to hub & switch. Also to observe behaviour of the device

Topology:

1) Hub Behavior: 3 PC's connected to hub
2) Switch: 3 PCs connected to switch using Straight through ethernet cable.

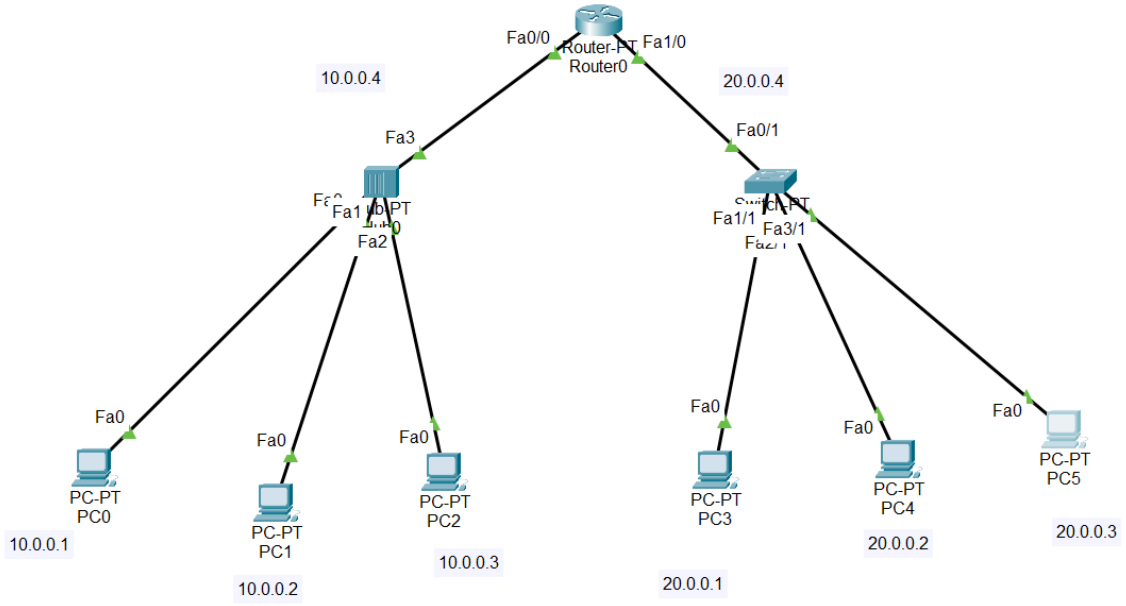
Procedure Observation:

1) Add ten hub broadcasts packets to all devices.

9/10/24

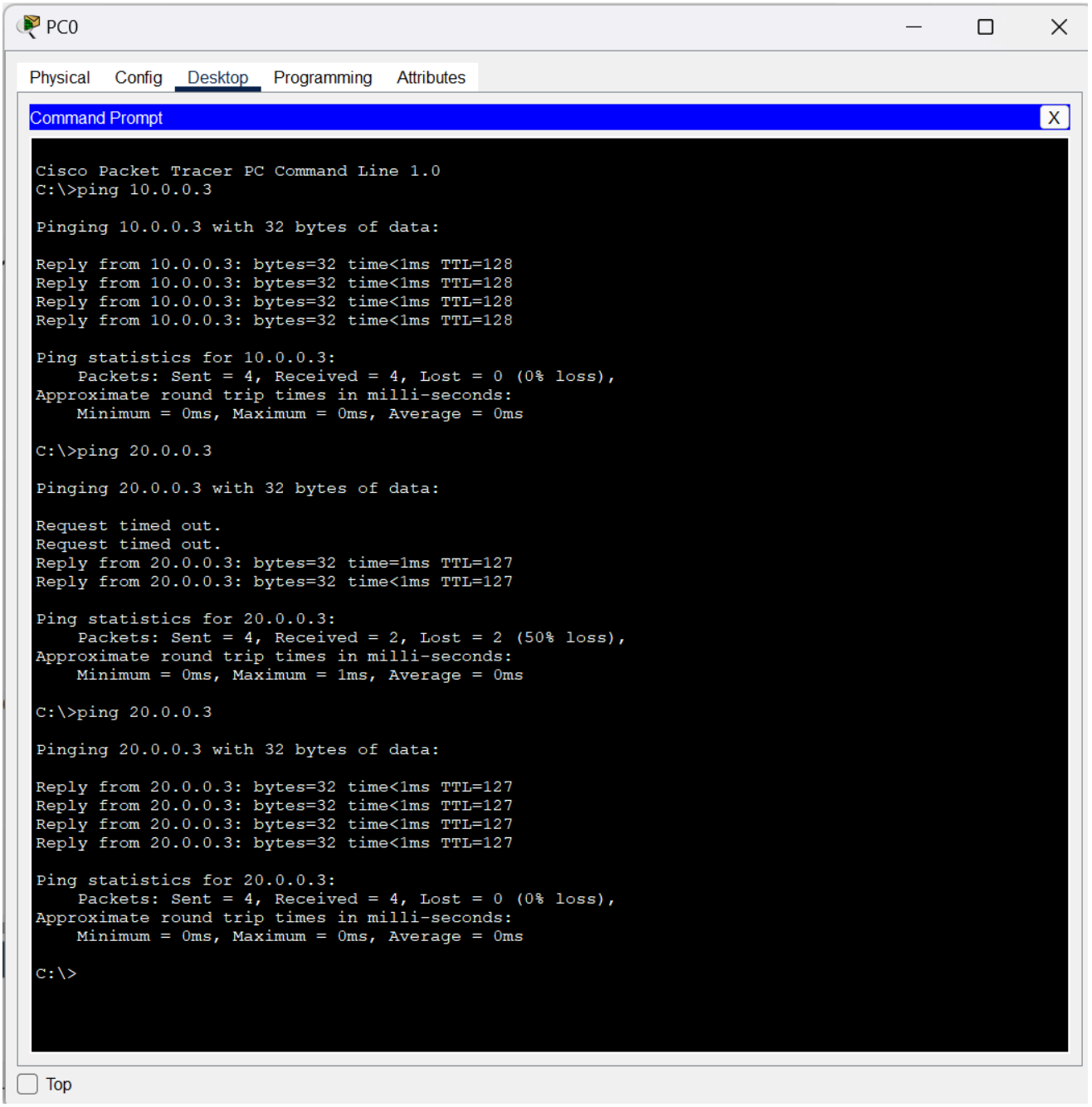
2) The switch towards packets only to the appropriate device.

Topology



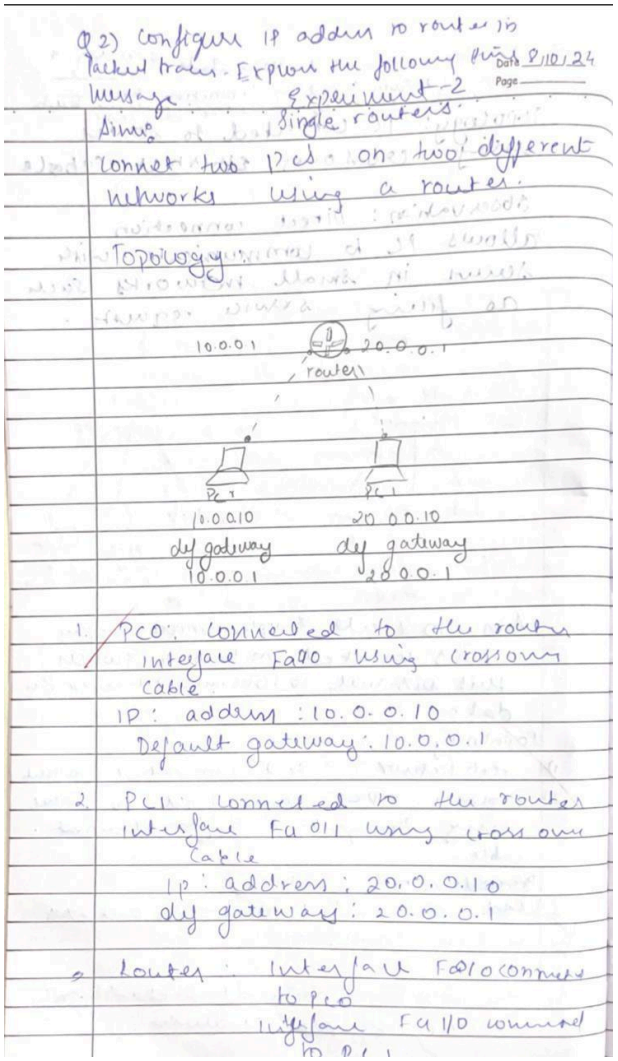
CLI Commands

```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.4 255.0.0.0
Router(config-if)#ip address 10.0.0.4 255.0.0.0
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)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 20.0.0.4 255.0.0.0
Router(config-if)#ip address 20.0.0.4 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

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

Router(config-if)#
```


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 20.0.0.3
Pinging 20.0.0.3 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 20.0.0.3: bytes=32 time=1ms TTL=127
Reply from 20.0.0.3: bytes=32 time<1ms TTL=127
Ping statistics for 20.0.0.3:
Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>ping 20.0.0.3
Pinging 20.0.0.3 with 32 bytes of data:
Reply from 20.0.0.3: bytes=32 time<1ms TTL=127
Reply from 20.0.0.3: bytes=32 time<1ms TTL=127
Reply from 20.0.0.3: bytes=32 time<1ms TTL=127
Reply from 20.0.0.3: bytes=32 time<1ms TTL=127
Ping statistics for 20.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
Top

Single Router Experiment

2a)



Q 2) Configure IP addrs to router in packet tracer. Explore the following messages Date 8/10/24

Page

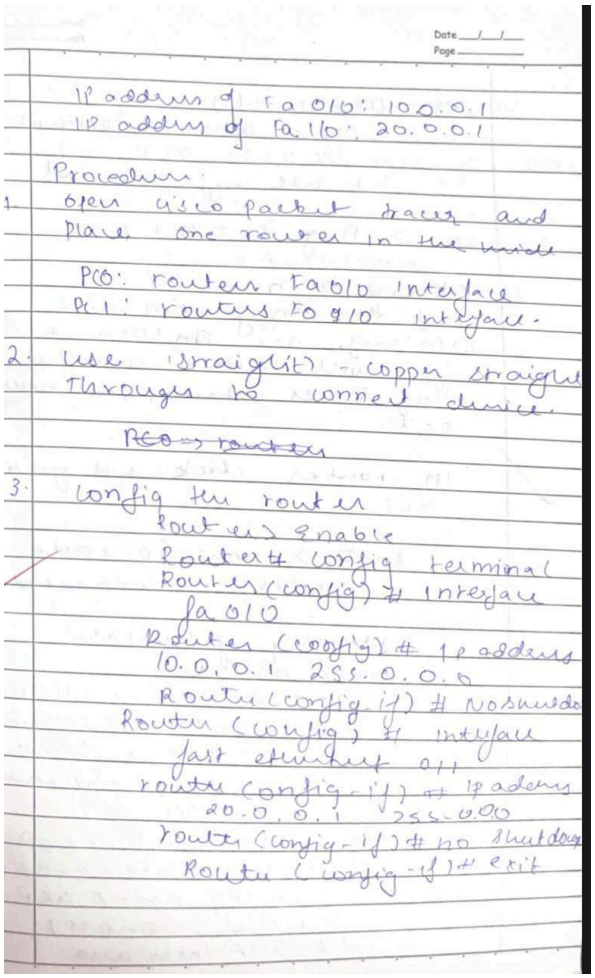
Experiment-2

Aim: Single routers.

Connect two PCs on two different networks using a router.

Topology:

1. PC0: connected to the router interface Fa0/0 using crossover cable
   IP: addrss: 10.0.0.10
   Default gateway: 10.0.0.1
2. PC1 connected to the router interface Fa 0/1 using crossover cable
   Ip: address: 20.0.0.10
   dfg gateway: 20.0.0.1
3. Router: Interface Fa0/0 connected to PC0
   Interface Fa 1/0 connected to PC1



Date

Page

IP address of Fa 0/0: 10.0.0.1
IP addrss of Fa 1/0: 20.0.0.1

Procedure:

1. Open cisco packet tracer and place one router in the middle
   PC0: router Fa0/0 interface
   PC1: routers Fa 0/0 interface.
2. Use straight - copper straight Through to connect device.
   ~~PC0 → router~~
3. Config the router
   Router> Enable
   Router# config terminal
   Router(config)# interface fa 0/0
   Router(config)# ip address 10.0.0.1 255.0.0.0
   Router(config-if)# No shutdown
   Router(config)# interface fast ethernet 0/1
   router(config-if)# ip address 20.0.0.1 255.0.0.0
   Router(config-if)# no shutdown
   Router(config-if)# exit

4. test connectivity:

Open command prompt of PC0 to PC1 & use my command

>> Ping 10.0.0.1

Observation:

1. If the configuration and cabling are correct you will recieve successful ping replies between the two PC's

In routers click and go to CLI

Router > show ip route

Codes: C - connected, ...

Gateway of the last resort is not set

C 10.0.0.0/8 is directly connected, FastEthernet0/1

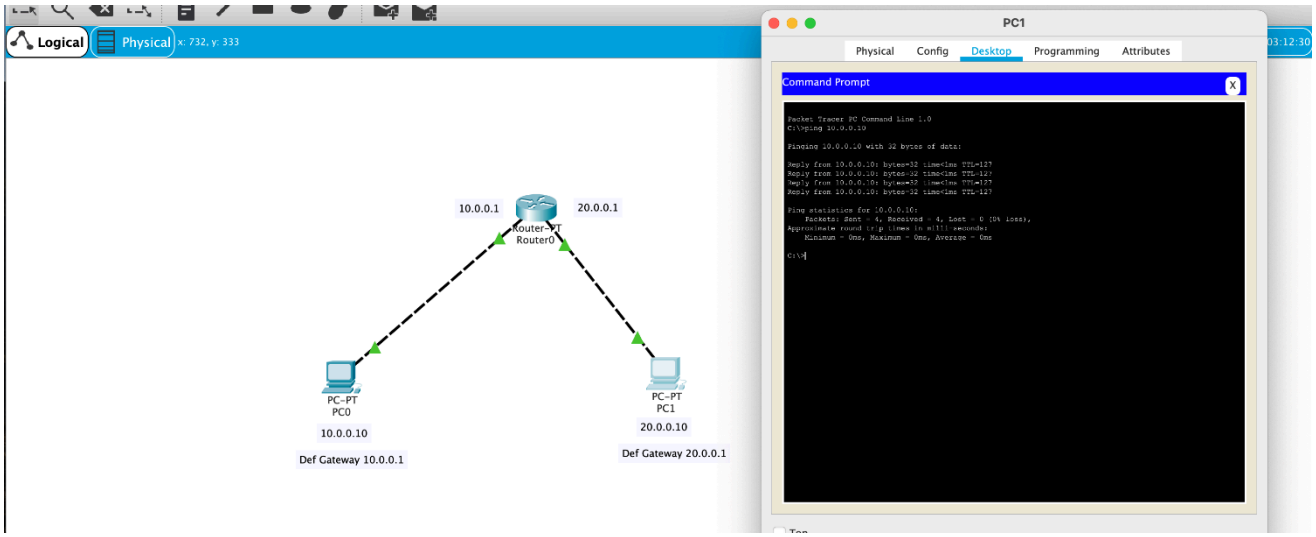
C 20.0.0.0/8 is directly connected, FastEthernet0/1

Codes: S - static, I - IGRP, R - RIP, M - mobile, B - BGP, D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

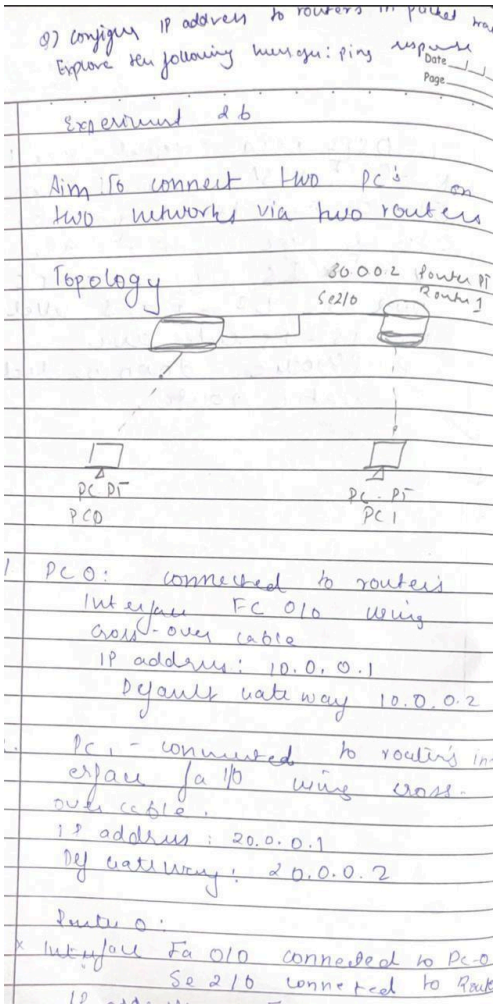
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2, E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP, i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area, P - periodic downloaded static route.

9/10/24

Topology:



2b)



Experiment 2b

Aim: To connect two PC's on two networks via two routers

Topology

30.0.0.2 Router PT Router1 Se2/0

PC PT PC0

PC PT PC1

PC 0: connected to router's Interface FC 0/0 using cross-over cable

IP address: 10.0.0.1

Default gateway 10.0.0.2

PC 1 - connected to routing interface fa 1/0 using cross-over cable.

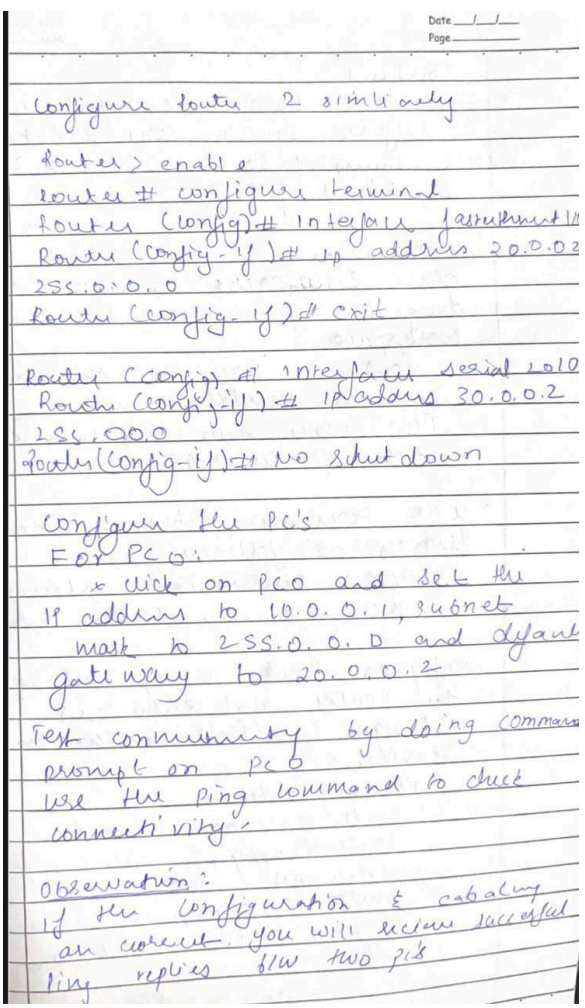
IP address: 20.0.0.1

Def gateway: 20.0.0.2

Router 0:

Interface Fa 0/0 connected to Pc-0

Se 2/0 connected to Rout



Configure router 2 similarly

Router > enable

router # configure terminal

Router (config)# interface fastethernet 1/0

Router (config-if)# ip address 20.0.0.2 255.0.0.0

Router (config-if)# exit

Router (config)# interface serial 2/0

Router (config-if)# ip address 30.0.0.2 255.00.0

Router (config-if)# no shut down

Configure the PC's

For PC 0:

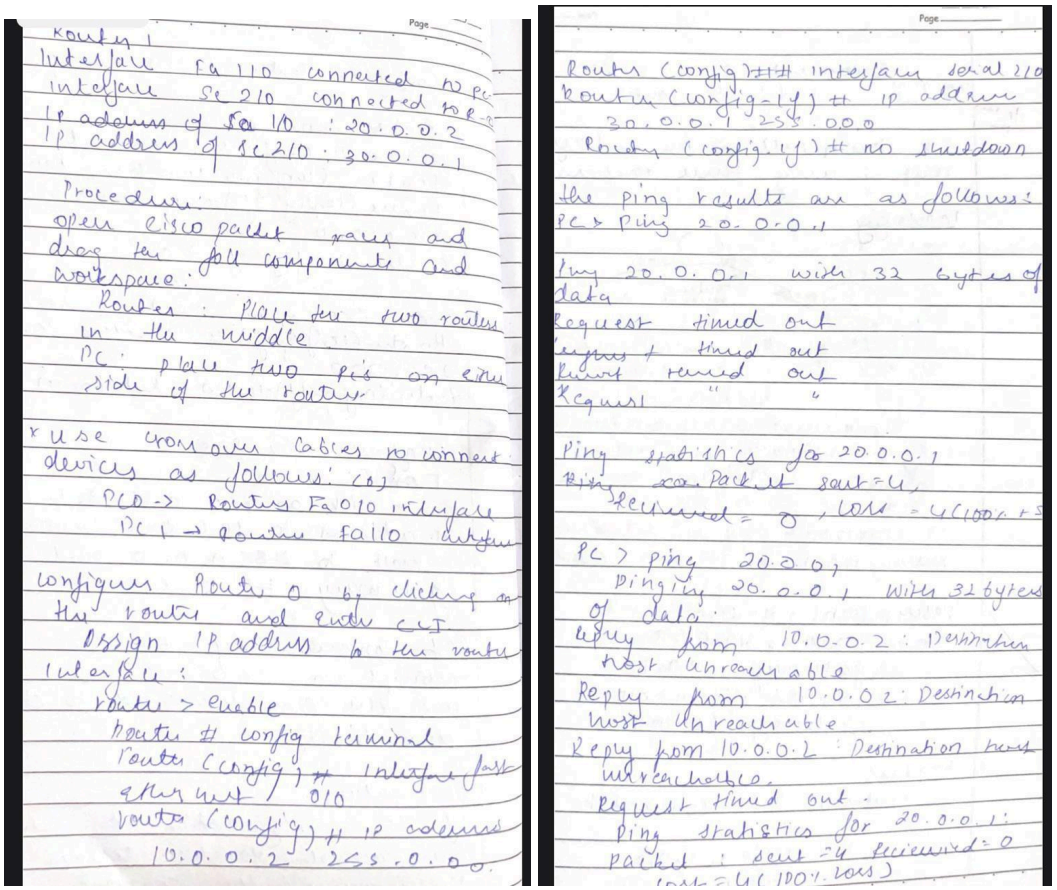
* click on PC 0 and set the IP address to 10.0.0.1, subnet mask to 255.0.0.0 and default gateway to 20.0.0.2

Test connectivity by doing command prompt on Pc 0

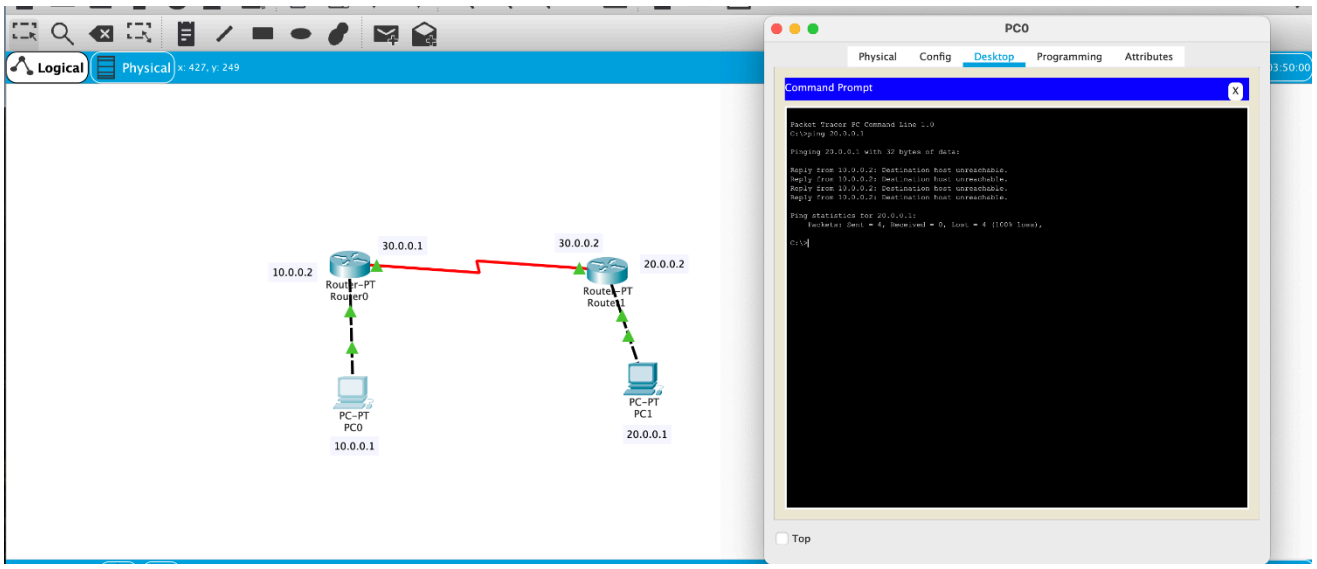
Use the Ping command to check connectivity:

Observation:

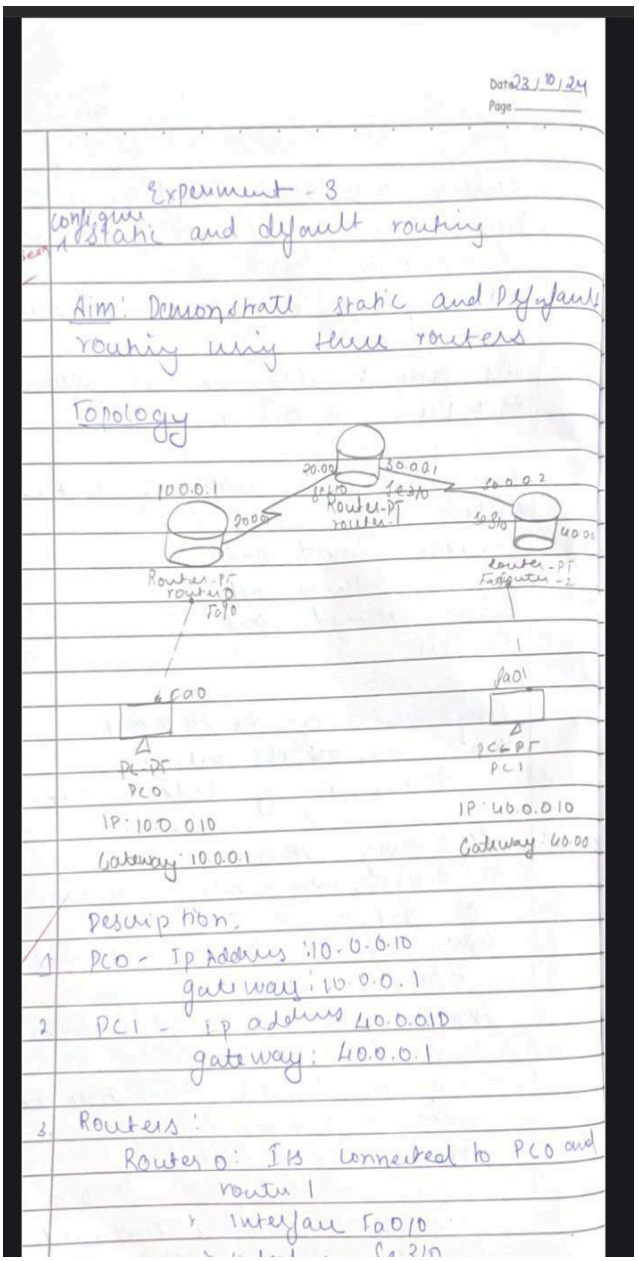
If the configuration & cabling are correct you will receive successful ping replies else two pcs



Topology


Logical
Physical
30.0.0.1
30.0.0.2
10.0.0.2
20.0.0.2
Router-PT
Router0
Router-PT
Router1
PC-PT
PC0
10.0.0.1
PC-PT
PC1
20.0.0.1
PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Top

Static and default Routing Experiment



Date 23/10/24
Page

Experiment - 3

configure static and default routing

Aim: Demonstrate static and Default routing using three routers

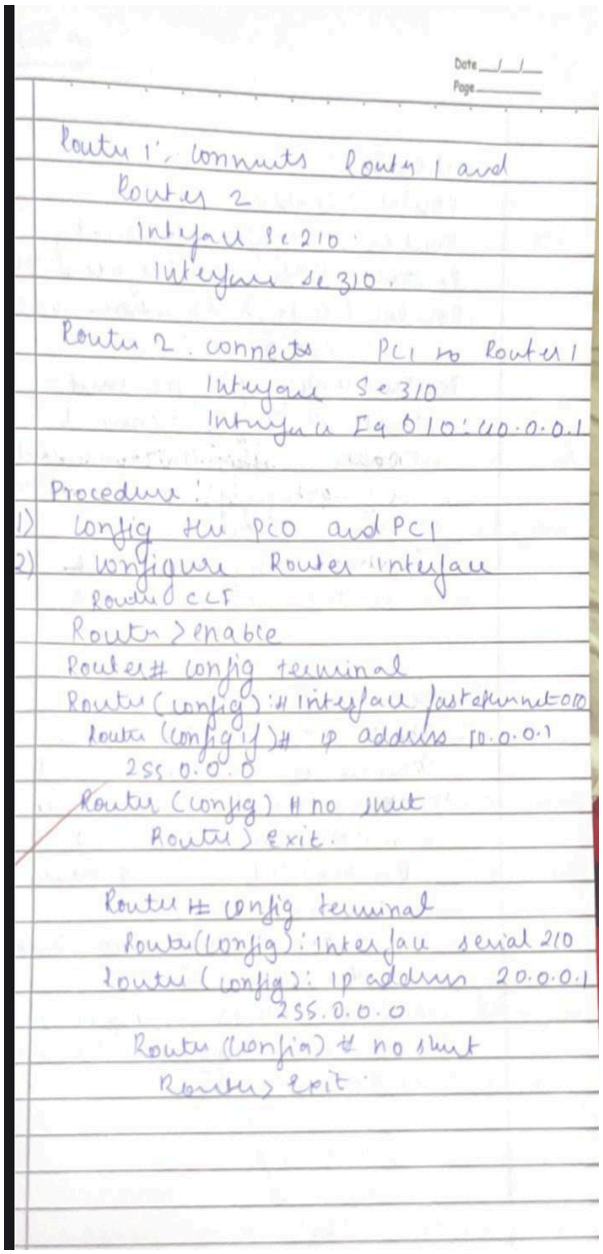
Topology

IP: 10.0.0.10
Gateway: 10.0.0.1

IP: 40.0.0.10
Gateway: 40.0.0

Description:

1. PC0 - Ip Address: 10.0.0.10
   gateway: 10.0.0.1
2. PC1 - Ip address 40.0.0.10
   gateway: 40.0.0.1
3. Routers:
   Router 0: It is connected to PC0 and router 1
   - Interface Fa0/0



Date __/__/__
Page _____

Router 1: connects Router 1 and Router 2
Interface Se 2/0
Interface Se 3/0

Router 2: connects PC1 to Router 1
Interface Se 3/0
Interface Eg 0/0: 40.0.0.1

Procedure:
1) Config the PC0 and PC1
2) Configure Router Interface
Router> CLI
Router> enable
Router# config terminal
Router(config)# interface fastethernet 0/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shut
Router> exit

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

Router 2 CLI:

Router > enable
Router # config terminal
Router (config): interface fastethernet
Router (config): ip address 40.0.0.1
255.0.0.0
Router (config-if): no shut
Router # config terminal
Router (config): interface serial 3/0
Router (config-if): ip address 30.0.0.1
255.0.0.0
Router (config) # no shut
Router: exit.

Router 1 CLI:

Router > enable
Router # config terminal
Router (config): interface serial 2/0
Router (config): IP address 20.0.0.1
255.0.0.0
router (config-if): no shut
Router # exit.
Router (config): interface serial 3/0
Router (config): no shut.
Router: exit.

Router 0

Router > show ip route
C 10.0.0.0/8 is directly connected to fast ethernet 0/0
C 20.0.0.0/8 is directly connected to serial 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.0

Router 2:

C 30.0.0.0/8 is directly connected to serial 3/0
C 40.0.0.0/8 is directly connected fastethernet 0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router 1:

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

outputs:

PC> ping 40.0.0.10
pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes 32 time = 6ms TTL = 125
Reply from 40.0.0.10: bytes 32 time = 6ms TTL = 125
Reply from 40.0.0.10: bytes 32 time = 6ms TTL = 125

Repley from 40.0.0.10 bytes=32 time 6ms TTL=125

Ping statistics for 40.0.0.10:
Packets: sent=4 Recieved=4, lost=0 (0% loss)

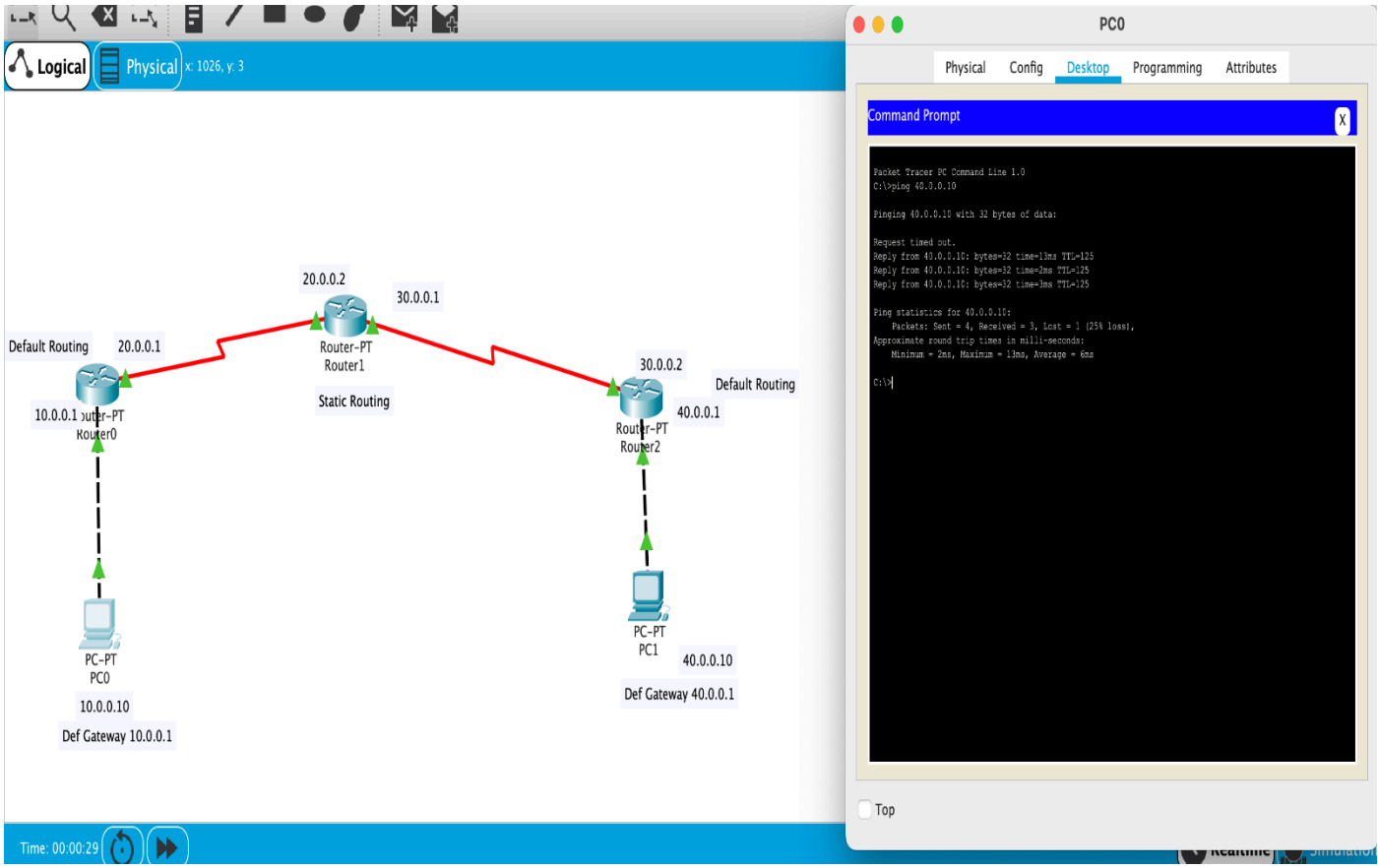
Observation:
If configuration and cabling are correct, you will recieve successful ping repies between 2 pc's

Completed 23/10/24

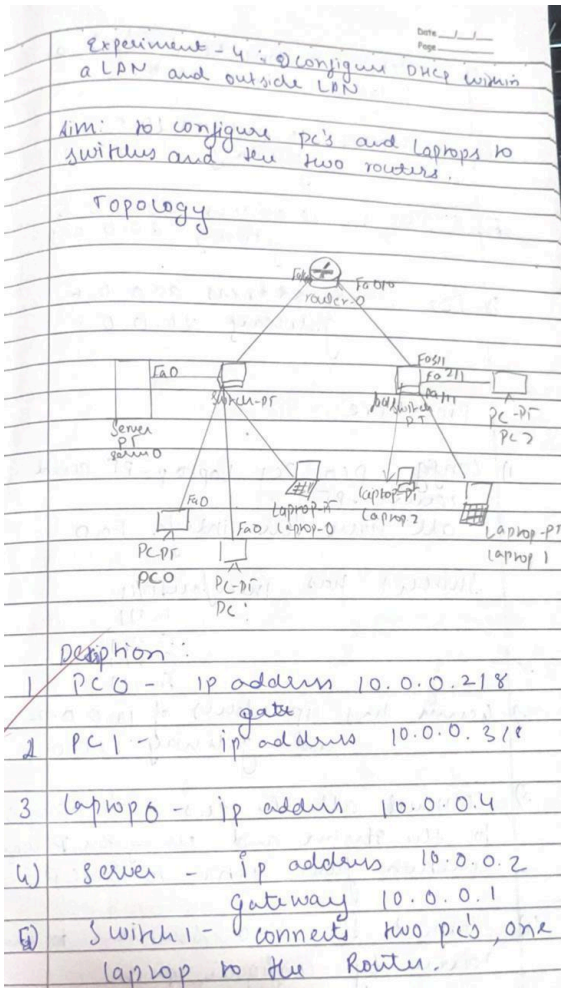
Default routing for Router 0

Enable
Config terminal
IP route 0.0.0.0 0.0.0.0 20.0.0.1
Exit
Show ip route
C 10.0.0.0/8 is directly connected Fast Ethernet 0/0
C 20.0.0.0/8 is directly connected serial 2/0
S * 0.0.0.0/0 [1/0] via 20.0.0.2

Topology


Logical
Physical
x: 1026, y: 3
20.0.0.2
30.0.0.1
Default Routing
20.0.0.1
Router-PT
Router1
Static Routing
10.0.0.1
Router0
30.0.0.2
Default Routing
40.0.0.1
Router2
PC-PT
PC0
10.0.0.10
Def Gateway 10.0.0.1
PC-PT
PC1
40.0.0.10
Def Gateway 40.0.0.1
Time: 00:00:29
PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=13ms TTL=125
Reply from 40.0.0.10: bytes=32 time=2ms TTL=125
Reply from 40.0.0.10: bytes=32 time=3ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 13ms, Average = 6ms
C:\>
Top

DHCP Inside LAN



Experiment - 4 : Configure DHCP within a LAN and outside LAN

Aim: To configure pc's and laptops to switches and the two routers.

Topology

Description:

1. PC0 - ip address 10.0.0.2 /8 gate
2. PC1 - ip address 10.0.0.3 /8
3. Laptop0 - ip addrs 10.0.0.4
4. Server - ip address 10.0.0.2 gateway 10.0.0.1
5. Switch1 - connects two pc's, one laptop to the Router.

6) Switch2: connects one PC & and 2 laptops to the router

7) Laptop2: IP address 20.0.0.4
gateway 20.0.0.1

8) Laptop1: IP address 20.0.0.5
gateway 20.0.0.1

9) PC2: IP address 20.0.0.6
gateway 20.0.0.1

Procedure

1) Config PC0 PC1 Laptop-PT and Server-PT.
all have the interface Fa0

Switch1 has interface Fa0/1
Fa 1/1
Fa 2/1
Fa 3/1
Fa 4/1

2) Server has ip address of 10.0.0.2
and gateway: 10.0.0.1

3) Connect all the PCs and laptops to the switch and change the IP config section from static to DHCP.

4) Connect the two switches to the router by configuring it

with in

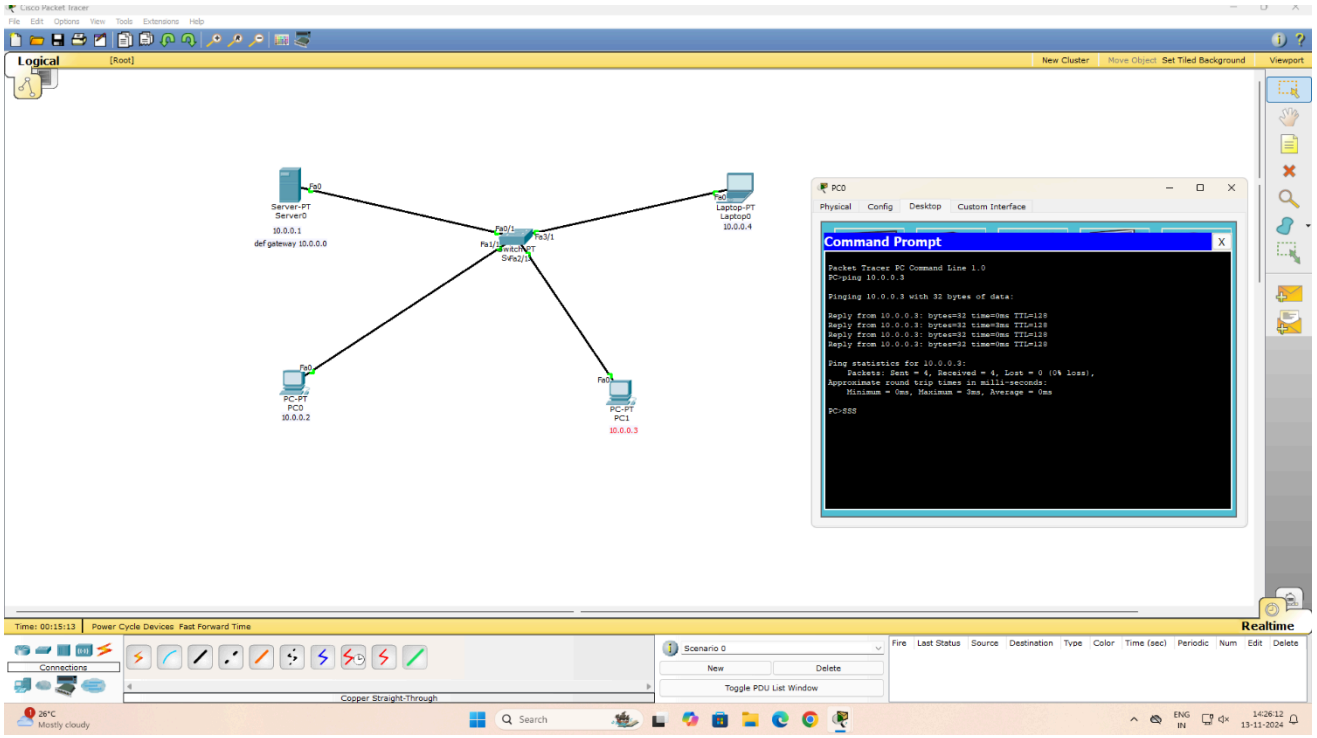
Router> CLI
router>enable
router# config terminal
router#(config)#: interface fa 4/0
router#(config-if): ip address 10.0.0.1
255.0.0.0
router(config-if): ip helper address
10.0.0.2
router: no shut
Exit

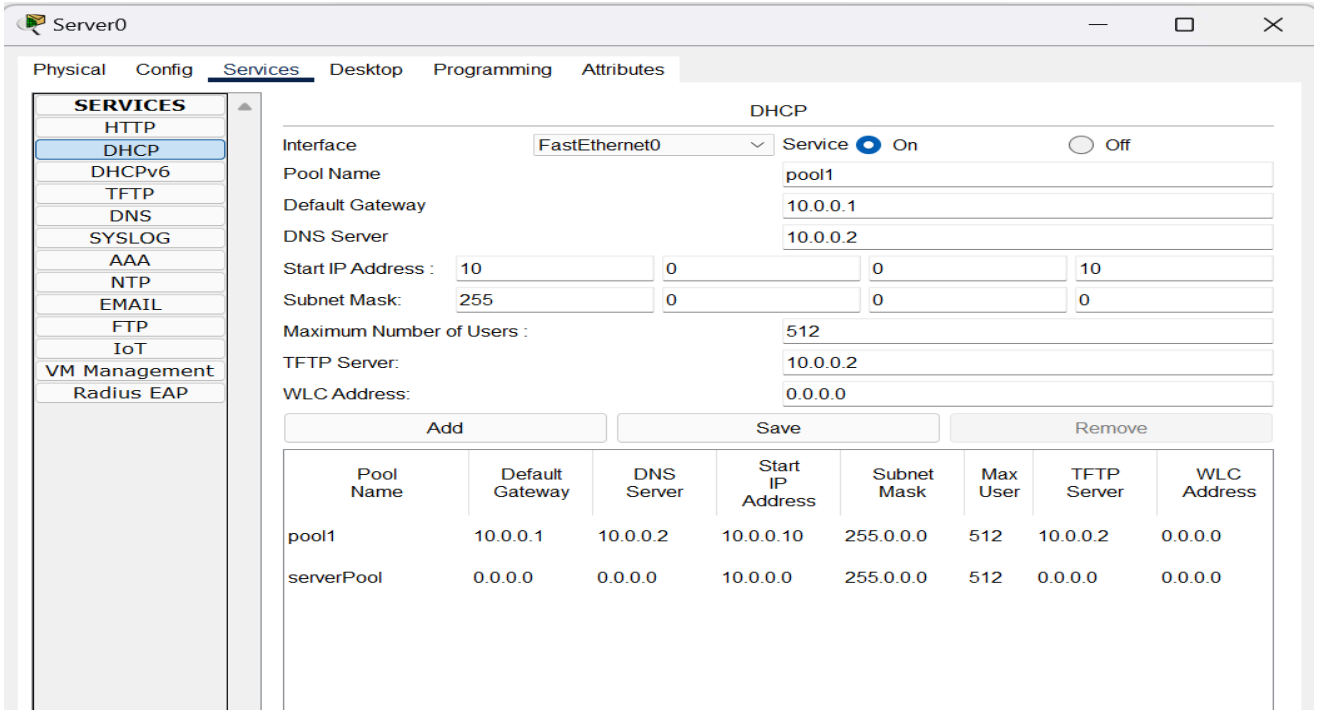
router#(config-if): interface fa0/0
router#(config-if) ip address 20.0.0.1
router#(config-if): ip helper address 10.0.0.2
no shut
exit.

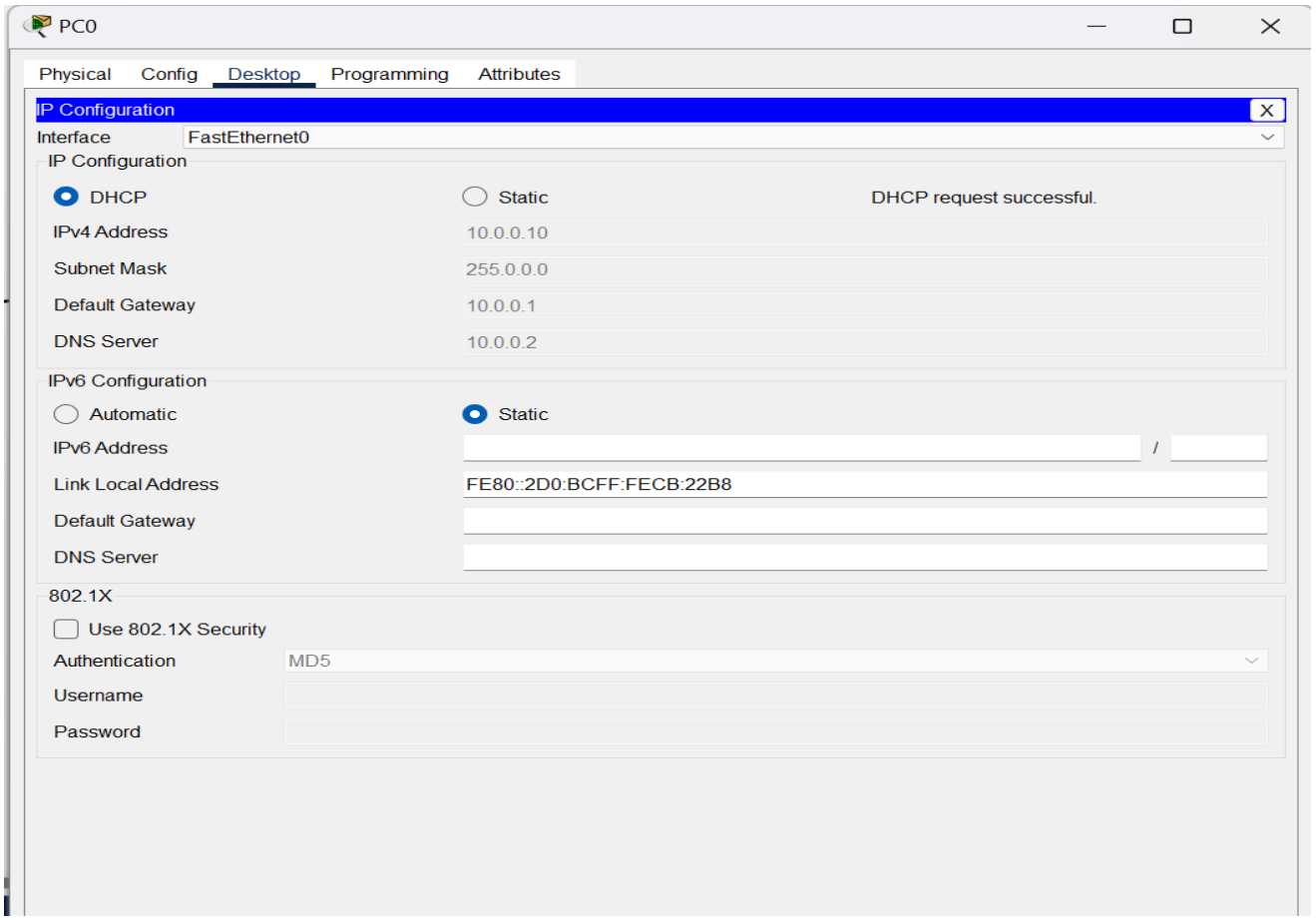
Outputs.
PC> Ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes
reply from 10.0.0.3 bytes=32 TTL=1
reply from 10.0.0.3 bytes=32 TTL=1
reply from 10.0.0.3 bytes=32 TTL=1

Ping statistics for 10.0.0.3:
packets: sent = 4, Received = 4
lost = 0 (0% loss)
Approximate round times in milliseconds:
min = 0 ms, maximum = 2 ms,
avg = 0 ms

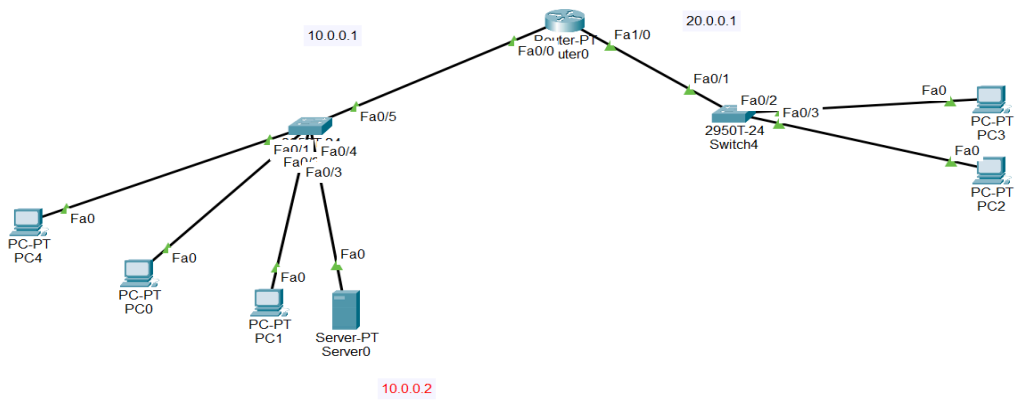
Topology

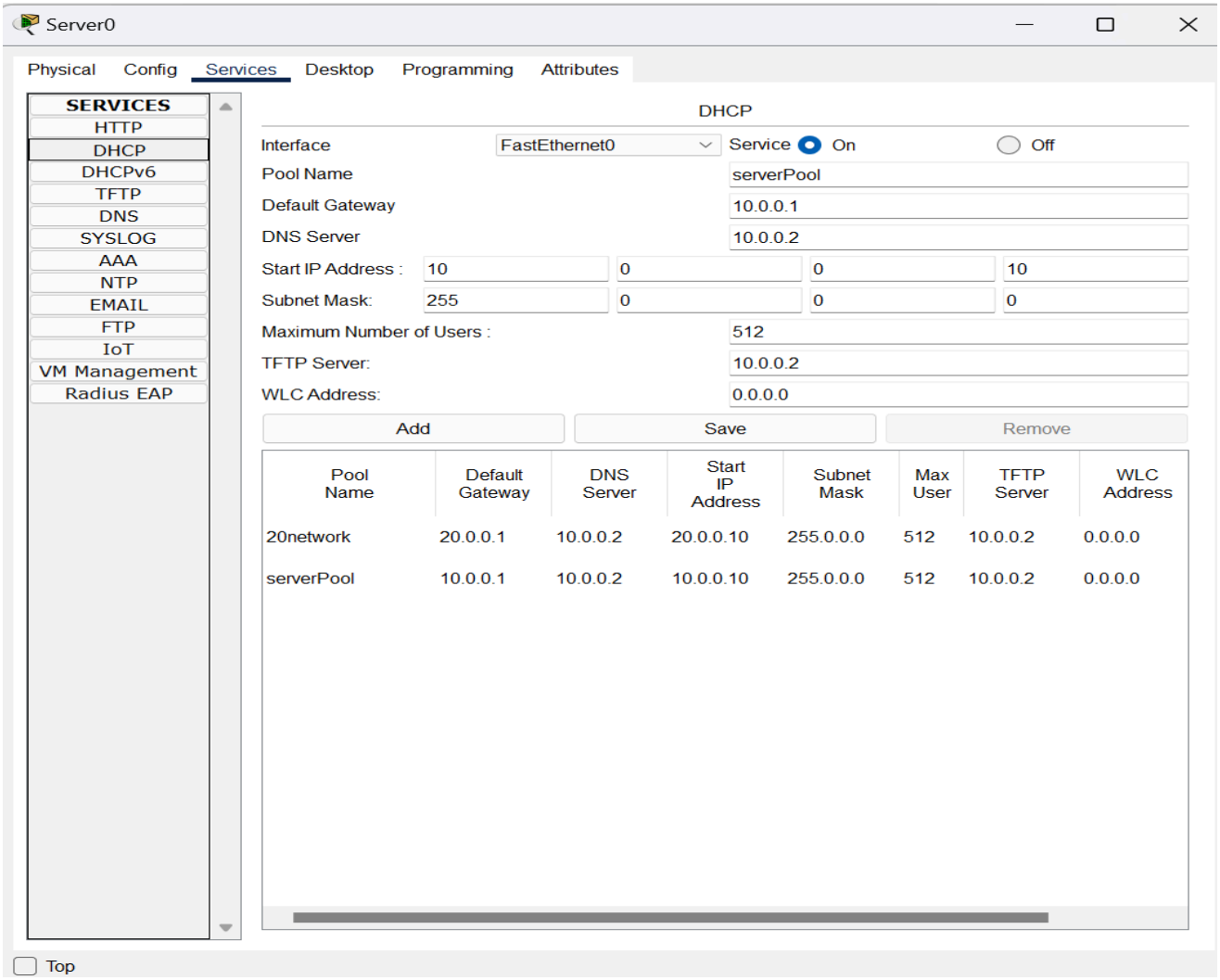

Logical
Server-PT Server0
10.0.0.1
def gateway 10.0.0.0
Laptop-PT Laptop0
10.0.0.4
PC-PT PC0
10.0.0.2
PC-PT PC1
10.0.0.3
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC-ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 0ms
PC-SSS
Realtime
Copper Straight-Through

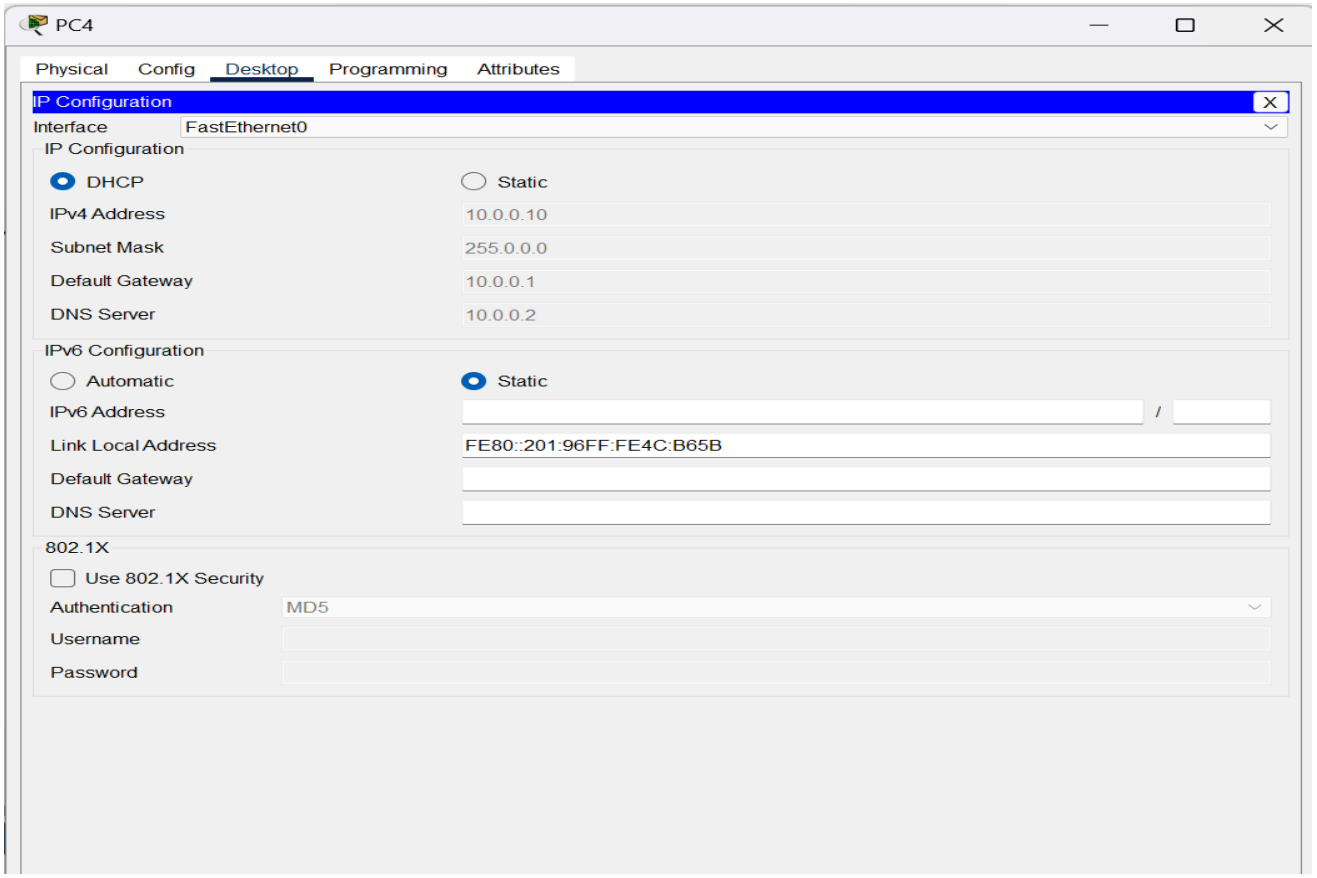

Server0
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
pool1
Default Gateway
10.0.0.1
DNS Server
10.0.0.2
Start IP Address :
10
0
0
10
Subnet Mask:
255
0
0
0
Maximum Number of Users :
512
TFTP Server:
10.0.0.2
WLC Address:
0.0.0.0
Add
Save
Remove
Pool Name
Default Gateway
DNS Server
Start IP Address
Subnet Mask
Max User
TFTP Server
WLC Address
pool1
10.0.0.1
10.0.0.2
10.0.0.10
255.0.0.0
512
10.0.0.2
0.0.0.0
serverPool
0.0.0.0
0.0.0.0
10.0.0.0
255.0.0.0
512
0.0.0.0
0.0.0.0

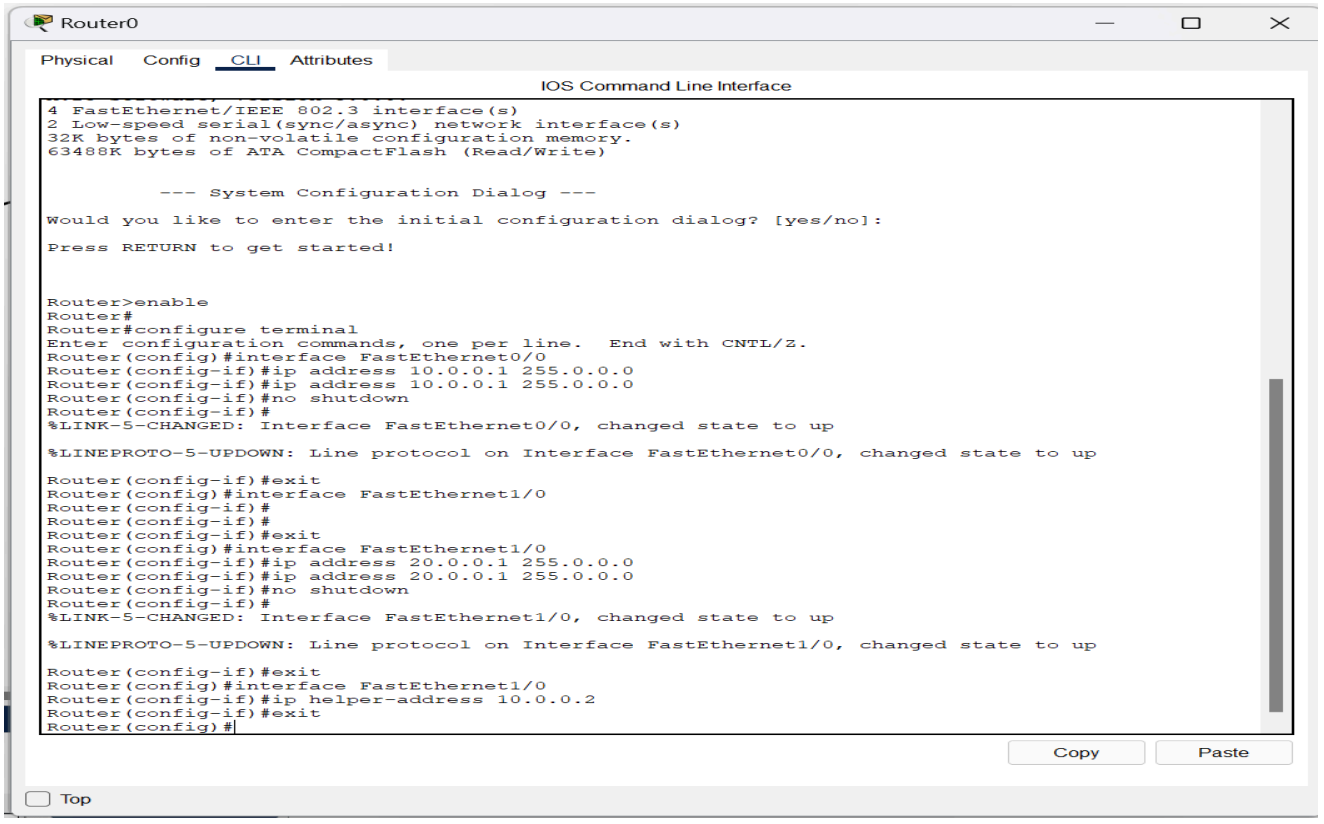

PC0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address
10.0.0.10
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
10.0.0.2
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::2D0:BCFF:FECB:22B8
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password

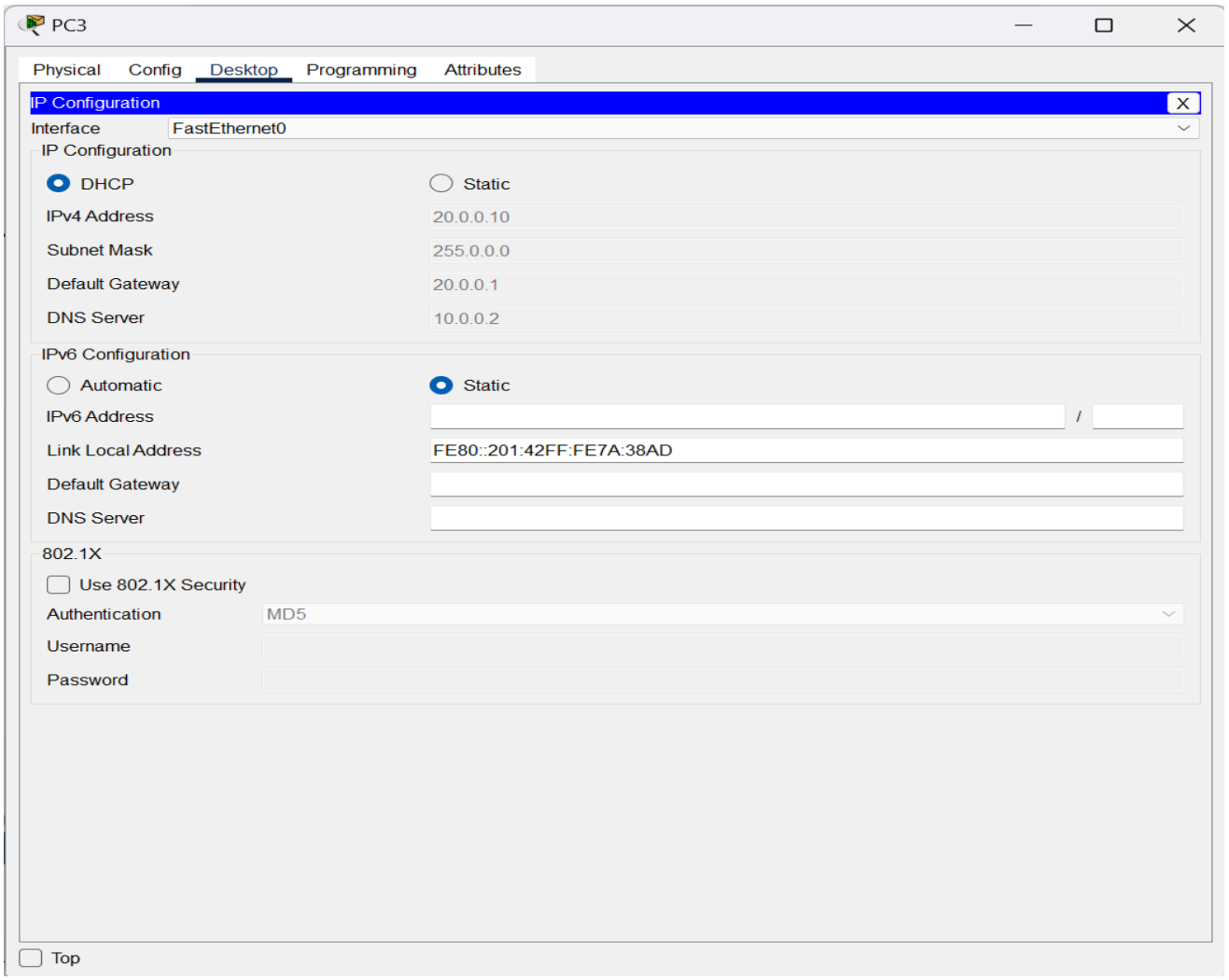
Topology


10.0.0.1
20.0.0.1
Fa1/0
Fa0/0
Fa0/1
Fa0/2
Fa0/3
Fa0/5
Fa0/4
Fa0/3
Fa0
PC-PT
PC4
PC-PT
PC0
PC-PT
PC1
Server-PT
Server0
2950T-24
Switch4
PC-PT
PC3
PC-PT
PC2
10.0.0.2


Server0
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
serverPool
Default Gateway
10.0.0.1
DNS Server
10.0.0.2
Start IP Address :
10
0
0
10
Subnet Mask:
255
0
0
0
Maximum Number of Users :
512
TFTP Server:
10.0.0.2
WLC Address:
0.0.0.0
Add
Save
Remove
Pool Name
Default Gateway
DNS Server
Start IP Address
Subnet Mask
Max User
TFTP Server
WLC Address
20network
20.0.0.1
10.0.0.2
20.0.0.10
255.0.0.0
512
10.0.0.2
0.0.0.0
serverPool
10.0.0.1
10.0.0.2
10.0.0.10
255.0.0.0
512
10.0.0.2
0.0.0.0
Top


PC4
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
10.0.0.10
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
10.0.0.2
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::201:96FF:FE4C:B65B
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password


Router0
Physical
Config
CLI
Attributes
IOS Command Line Interface
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]:
Press RETURN to get started!
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#exit
Router(config)#
Copy
Paste
Top

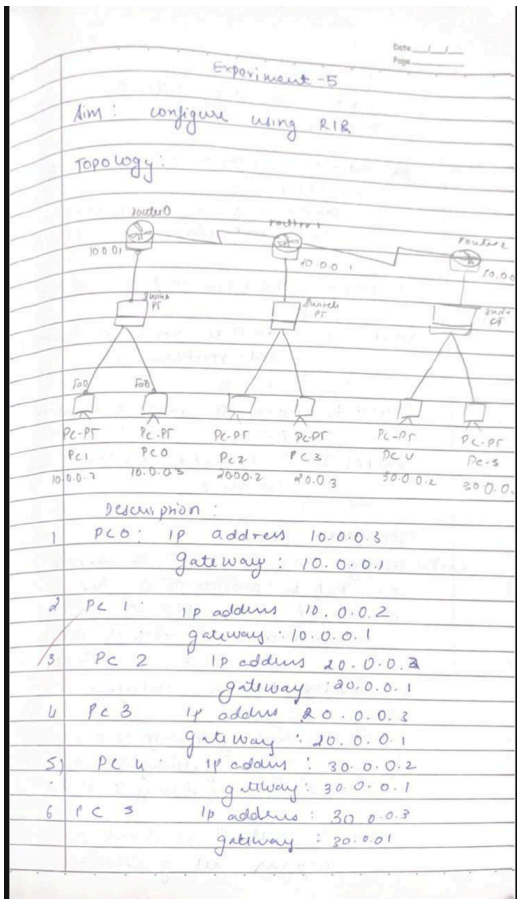

PC3
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
20.0.0.10
Subnet Mask
255.0.0.0
Default Gateway
20.0.0.1
DNS Server
10.0.0.2
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::201:42FF:FE7A:38AD
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password
Top

RIP

Experiment - 5

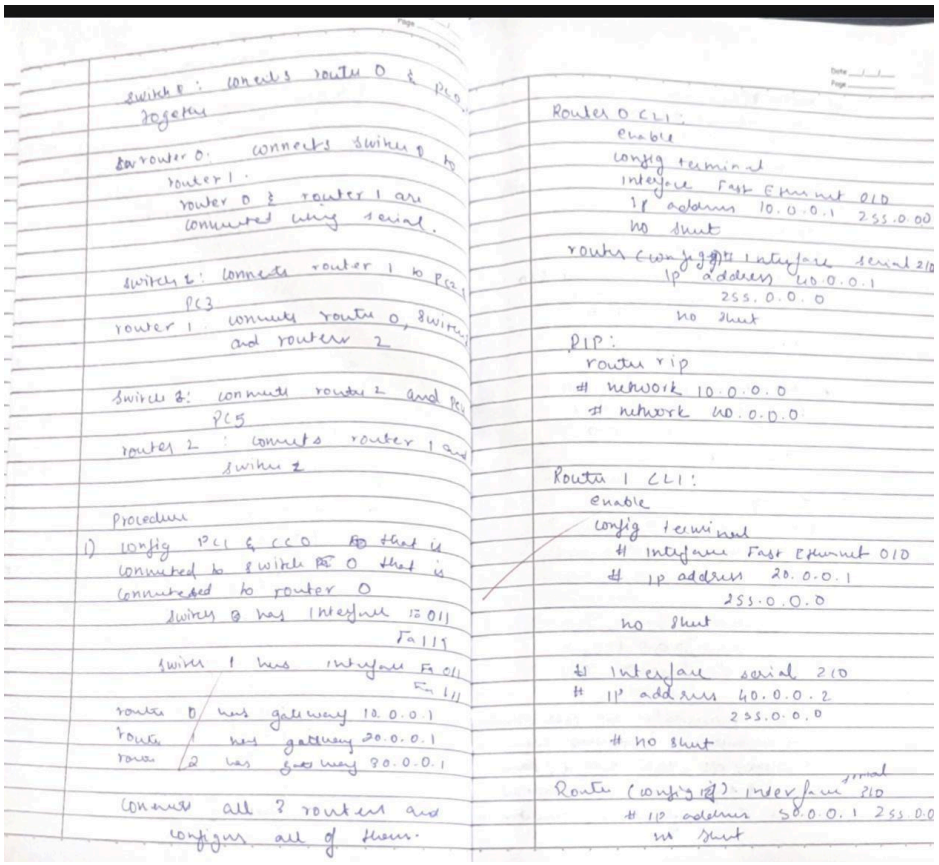
Aim: configure using RIP

Topology:



Description:

1. PC0: IP address 10.0.0.3
   Gateway: 10.0.0.1
2. PC 1: IP addrss 10.0.0.2
   gateway: 10.0.0.1
3. Pc 2 IP addrss 20.0.0.2
   gateway: 20.0.0.1
4. Pc 3 IP addrss 20.0.0.3
   gateway: 20.0.0.1
5. PC 4: IP addrss: 30.0.0.2
   gateway: 30.0.0.1
6. PC 5 IP addrss: 30.0.0.3
   gateway: 30.0.0.1



Switch 0: connects router 0 & PC0 together

router 0: connects switch 0 to router 1.
router 0 & router 1 are connected using serial.

Switch 2: connects router 1 to PC3
PC3
router 1: connects router 0, switch and router 2

Switch 1: connects router 2 and PC5
router 2: connects router 1 and switch 2

Procedure
1) config PC1 & CC0 that is connected to switch 0 that is connected to router 0
switch 0 has interface is 0/1 Fa1/1
switch 1 has interface Fa 0/1 Fa 1/1
router 0 has gateway 10.0.0.1
router 1 has gateway 20.0.0.1
router 2 has gateway 30.0.0.1

Connect all 3 routers and configure all of them.

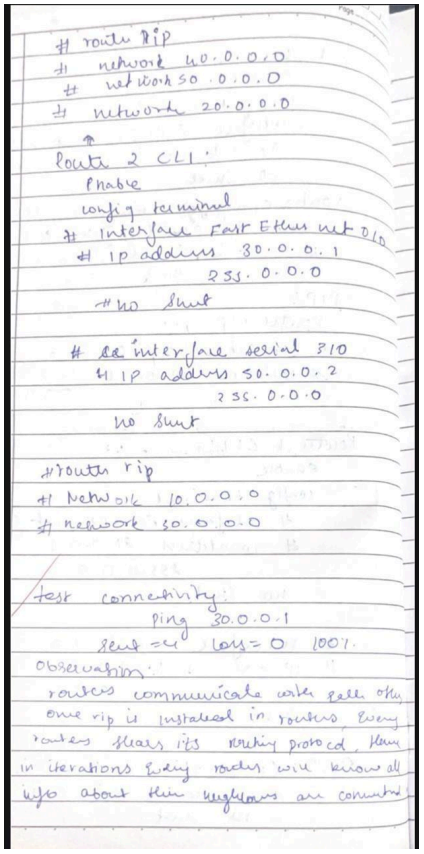
Router 0 CLI:
enable
config terminal
interface Fast Ethernet 0/0
ip address 10.0.0.1 255.0.00
no shut
router config# interface serial 2/0
ip address 40.0.0.1
255.0.0.0
no shut

RIP:
router rip
network 10.0.0.0
network 40.0.0.0

Router 1 CLI:
enable
config terminal
interface Fast Ethernet 0/0
ip address 20.0.0.1
255.0.0.0
no shut

interface serial 2/0
IP address 40.0.0.2
255.0.0.0
no shut

Router (config#) interface serial 3/0
IP address 50.0.0.1 255.0.0
no shut



router RIP
network 40.0.0.0
network 50.0.0.0
network 20.0.0.0

Router 2 CLI:
Enable
config terminal
interface Fast Ethernet 0/0
IP address 30.0.0.1
255.0.0.0
no shut

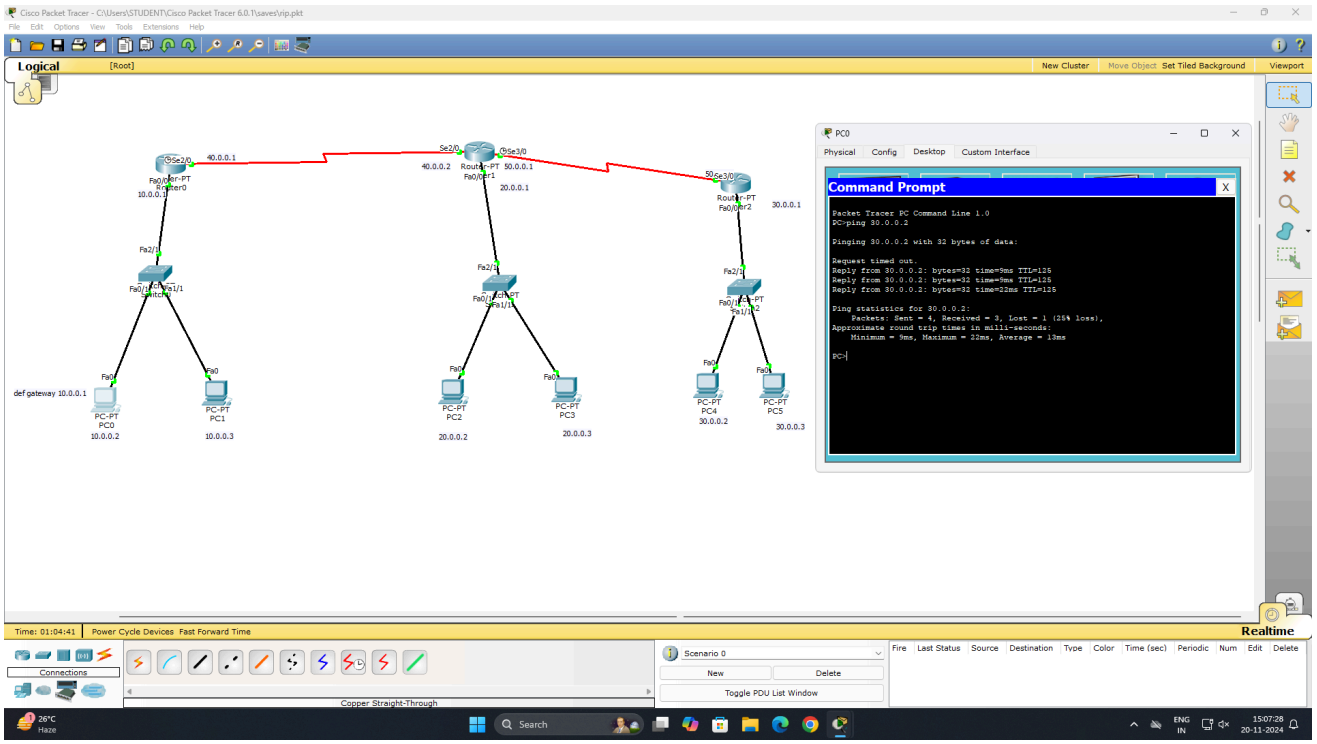
interface serial 3/0
IP address 50.0.0.2
255.0.0.0
no shut

#router rip
Network 10.0.0.0
network 30.0.0.0

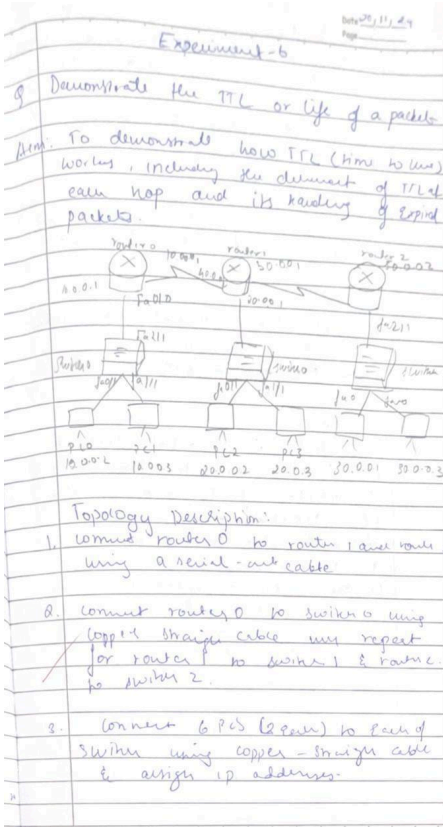
test connectivity:
ping 30.0.0.1
Sent = 4 Loss = 0 100%

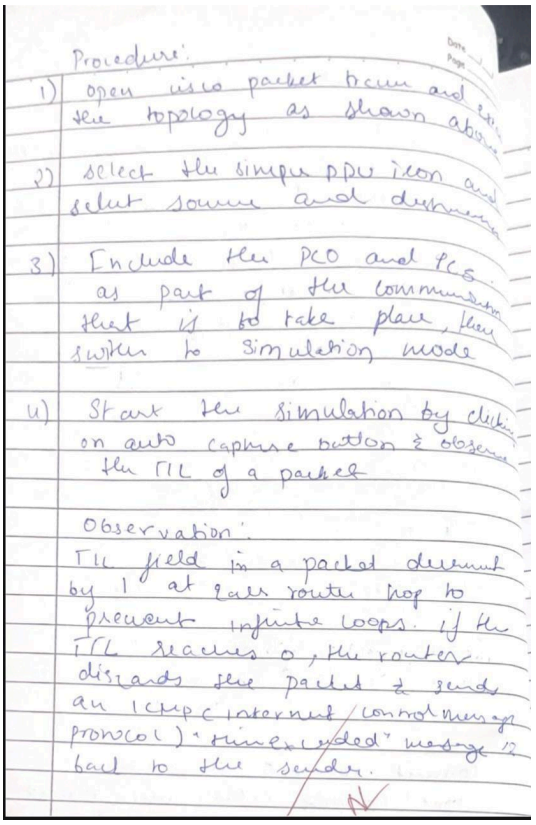
Observation:
routers communicate with each other once rip is installed in routers. Every routers shares its routing protocol. Then in iterations every router will know all info about their neighbours are connected

Topology

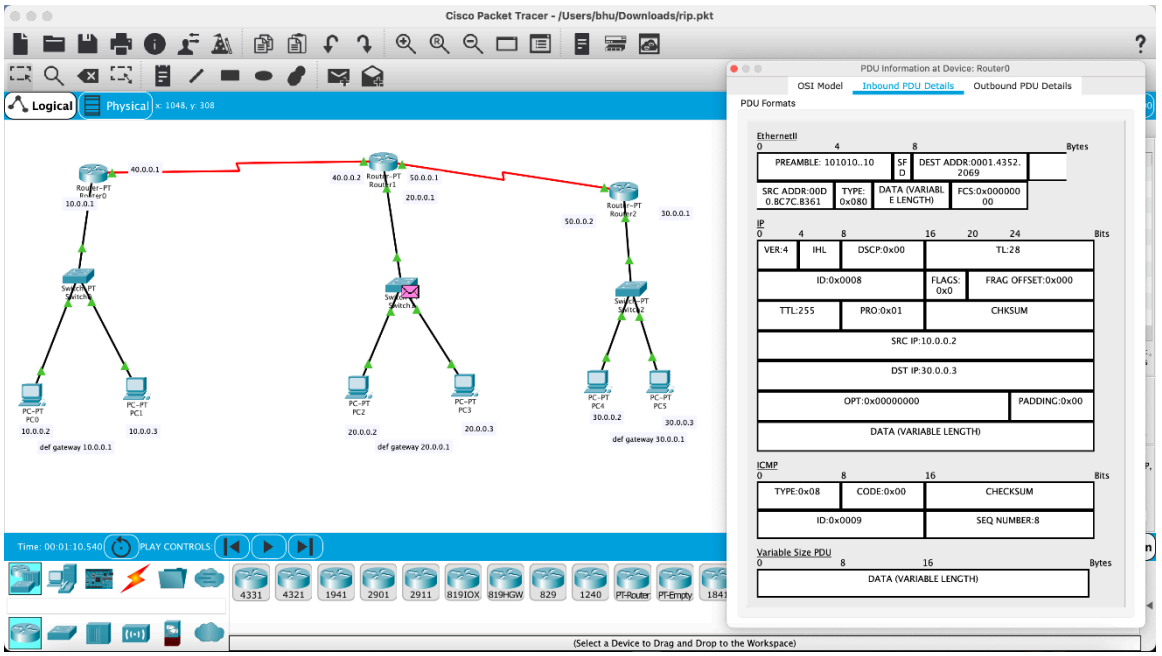

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 30.0.0.2: bytes=32 time=9ms TTL=125
Reply from 30.0.0.2: bytes=32 time=9ms TTL=125
Reply from 30.0.0.2: bytes=32 time=22ms TTL=125
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 9ms, Maximum = 22ms, Average = 13ms
PC>
40.0.0.1
40.0.0.2
50.0.0.1
20.0.0.1
30.0.0.1
10.0.0.1
def gateway 10.0.0.1
10.0.0.2
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3
Time: 01:04:41
Power Cycle Devices
Fast Forward Time
Realtime
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time (sec)
Periodic
Num
Edit
Delete
26°C
Haze
Search
ENG
IN
15:07:28
20-11-2024

TTL

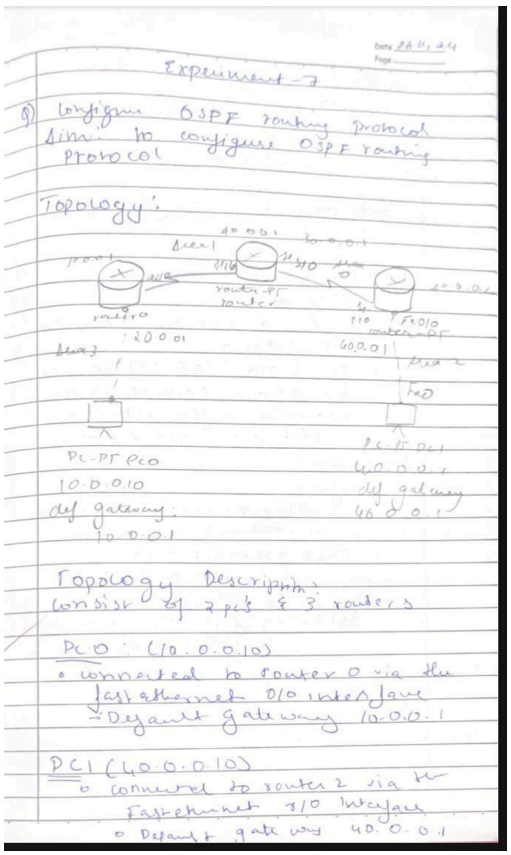

Experiment-6
Topology Description:


Procedure:
Observation:

Topolgy


Cisco Packet Tracer - /Users/bhu/Downloads/rip.pkt
Logical
Physical
PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
(Select a Device to Drag and Drop to the Workspace)

OSPF Routing Protocol



Experiment - 7

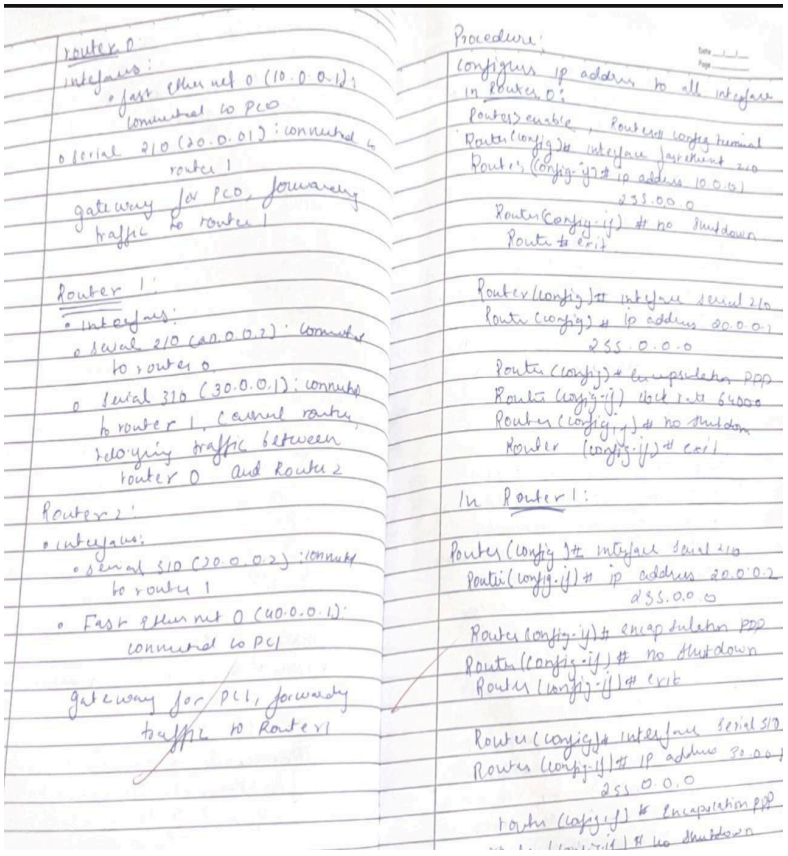
Q) Configure OSPF routing protocol
Aim: To configure OSPF routing protocol

Topology:

Topology Description:
Consist of 2 pc's & 3 routers

PC0 : (10.0.0.10)
- connected to router 0 via the fastethernet 0/0 interface
- Default gateway 10.0.0.1

PC1 (40.0.0.10)
- connected to router 2 via the Fastethernet 3/0 interface
- Default gateway 40.0.0.1



Router 0:
Interfaces:
- fast ethernet 0 (10.0.0.1): connected to PC0
- Serial 2/0 (20.0.0.1): connected to router 1

gateway for PC0, forwarding traffic to router 1

Router 1:
- Interfaces:
  - Serial 2/0 (20.0.0.2): connected to router 0
  - Serial 3/0 (30.0.0.1): connected to router 1 (central router relaying traffic between router 0 and Router 2

Router 2:
- Interfaces:
  - Serial 3/0 (30.0.0.2): connected to router 1
  - Fast ethernet 0 (40.0.0.1): connected to PC1

gateway for PC1, forwarding traffic to Router 1

Procedure:
Configure IP address to all interface in Router 0:
Router> enable, Router# config terminal
Router(config)# interface fastethernet 2/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shutdown
Router to exit

Router(config)# interface serial 2/0
Router(config)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# encapsulation PPP
Router(config-if)# clock rate 64000
Router(config-if)# no shutdown
Router(config-if)# exit

In Router 1:

Router(config)# interface serial 2/0
Router(config-if)# ip address 20.0.0.2 255.0.0.0
Router(config-if)# encapsulation PPP
Router(config-if)# no shutdown
Router(config-if)# exit

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

Step 3: Now, Enable IP routing by configuring ospf routing protocol in all routers

Router 0:

Router(config)# router ospf 1
Router(config-router)# router-id 1.1.1.1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# network 20.0.0.0 0.255.255.255 area 0
Router(config-router)# exit

Router 1:

Router(config)# router ospf 1
Router(config-router)# router-id 2.2.2.2
Router(config-router)# network 20.0.0.0 0.255.255.255 area 0
Router(config-router)# network 30.0.0.0 0.255.255.255 area 0
Router(config-router)# exit

Router 2:

Router(config)# router ospf 1
Router(config-router)# router-id 3.3.3.3
Router(config-router)# network 30.0.0.0 0.255.255.255 area 2

Step 4: Now check routing table of R1

Router# show ip route
C 20.0.0.0/8 is directly connected, FastEthernet0/0
C 30.0.0.0/8 is directly connected, Serial2/0

O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:04:33, Serial2/0

O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:07:29, Serial2/0

Configure Loopback:

Router 0:

Router(config)# interface loopback 0
Router(config-if)# ip address 172.16.2.2 255.255.255.0

Router 1:

Router(config)# interface loopback 0
Router(config-if)# ip address 172.16.1.253 255.255.255.0
Router(config-if)# no shutdown

Router# 2:

Router(config)# interface loopback 0
Router(config-if)# ip address 172.16.1.254 255.255.255.0
Router(config-if)# no shutdown

Step 5: Now check Routing table of R3
→ show ip route

C 40.0.0.0/8 is directly connected, FastEthernet0/0
C 30.0.0.0/8 is directly connected, Serial3/0

Step 6: Create virtual link between R1, R2 by this we create a virtual link to connect area 3 to area 0

Router 0:

Router(config)# router ospf 1
Router(config-router)# area 1 virtual-link 2.2.2.2

Router(config-router)# *Feb 10 ... from LOADING to FULL, Loading Done

Router 1:

Router(config)# router ospf 1
Router(config-router)# area 1 virtual-link 1.1.1.1
Router(config-router)# exit

Step 7: R1 & R3 get updates about area 3:

Router 2: show ip routes
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:00:01:36, Serial2/0
C 40.0.0.0/8 is directly connected FastEthernet2/0
O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:01:56, Serial1/0
C 30.0.0.0/8 is directly connected, Serial3/0

Step 8: Now check connectivity between Host 40.0.0.10 to 40.0.0.10

PC0 command prompt

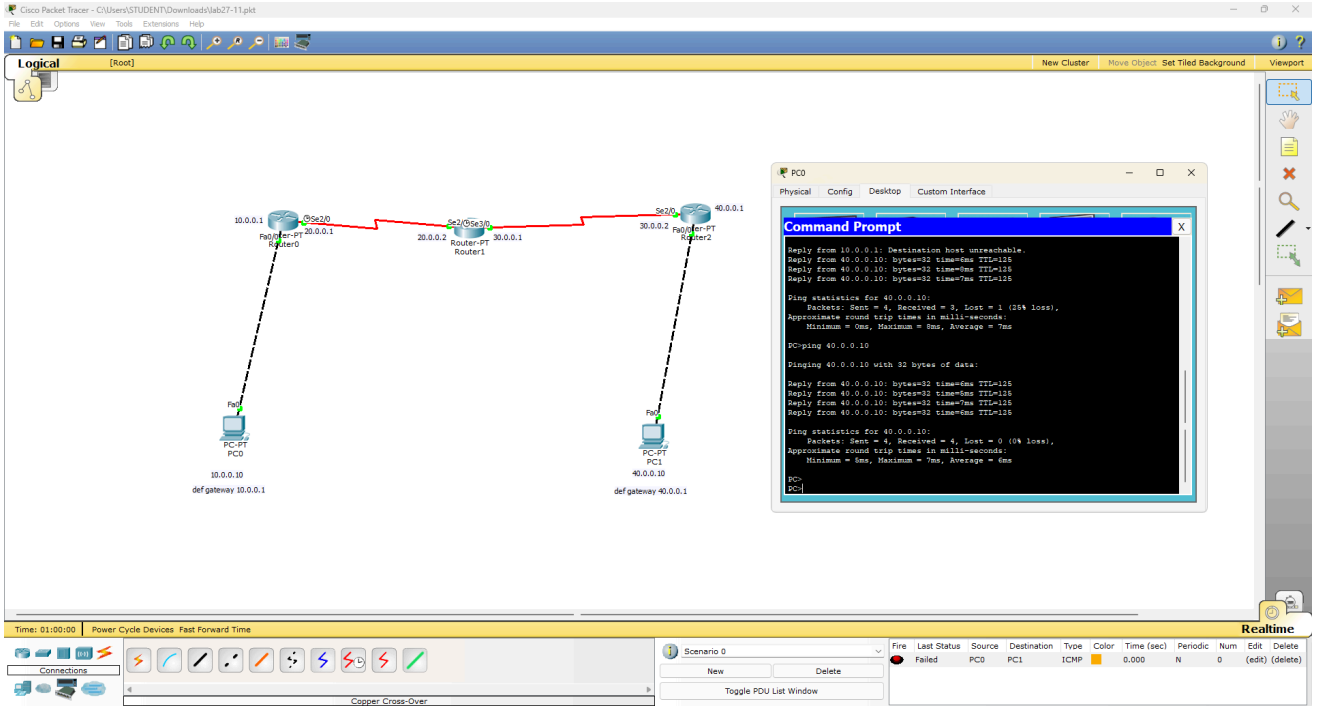
PC> ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes=32 time=9ms TTL=125
Reply from 40.0.0.10 bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time<1ms TTL=125
Reply from 40.0.0.10 bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.10:
Packets: Sent=4 Received=4 Lost=0 (0% loss)

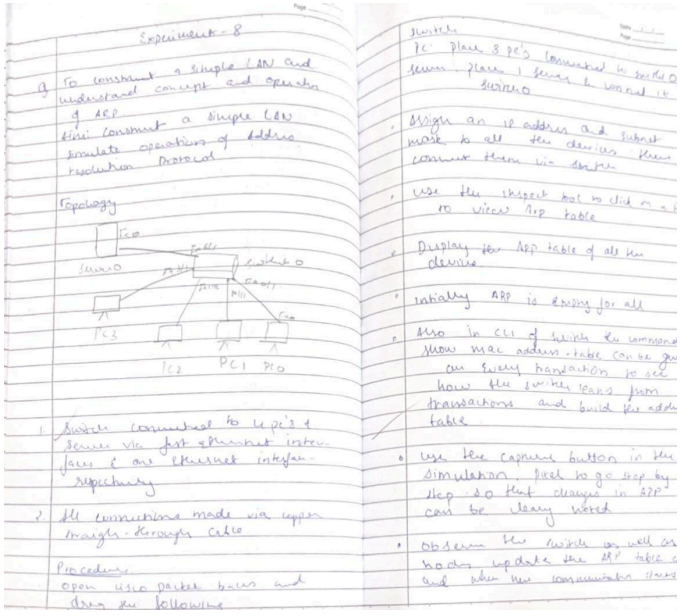
Observation:

The experiment demonstrates how OSPF dynamically learns and advertise router routers, enabling efficient and scalable routing across multiple area
→ router table on all routes must display network from all areas with OIA indicating

Topology


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
10.0.0.1
Fa0/0
Router-PT
Router0
Se2/0
20.0.0.1
Se2/0
Se3/0
20.0.0.2
Router-PT
Router1
30.0.0.1
Se2/0
30.0.0.2
40.0.0.1
Fa0/0
Router-PT
Router2
Fa0
PC-PT
PC0
10.0.0.10
def gateway 10.0.0.1
Fa0
PC-PT
PC1
40.0.0.10
def gateway 40.0.0.1
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Reply from 10.0.0.1: Destination host unreachable.
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 8ms, Average = 7ms
PC>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 7ms, Average = 6ms
PC>
PC>
Time: 01:00:00
Power Cycle Devices
Fast Forward Time
Realtime
Connections
Copper Cross-Over
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time (sec)
Periodic
Num
Edit
Delete
Failed
PC0
PC1
ICMP
0.000
N
0
(edit)
(delete)

Simple LAN and ARP Protocol



Experiment - 8

Aim: To construct a simple LAN and understand concept and operation of ARP

To construct a simple LAN and simulate operations of Address resolution Protocol

Topology

1. Switch connected to 4 pc's & Server via fast ethernet interfaces & one ethernet interface respectively
2. All connections made via copper straight-through cable

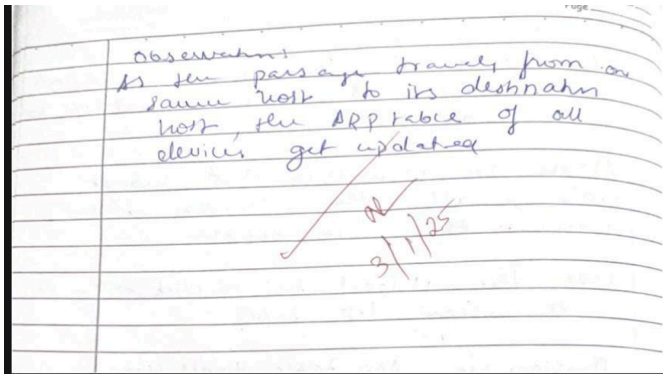
Procedure:

open cisco packet tracer and draw the following

Switch

To Place 5 pc's connected to switch0 server. Place 1 server to connect to switch0

- Assign an IP address and subnet mask to all the devices then communicate them via switch
- Use the inspect tool to click on a to view its table
- Display the ARP table of all the devices
- Initially ARP is empty for all
- Also in CLI of switch the command show mac address-table can be given on every handshake to see how the switch learns from transactions and build the address table
- Use the capture button in the simulation. First to go step by step so that changes in ARP can be clearly noted
- Observe the switch as well as nodes update the ARP table and when the communication starts

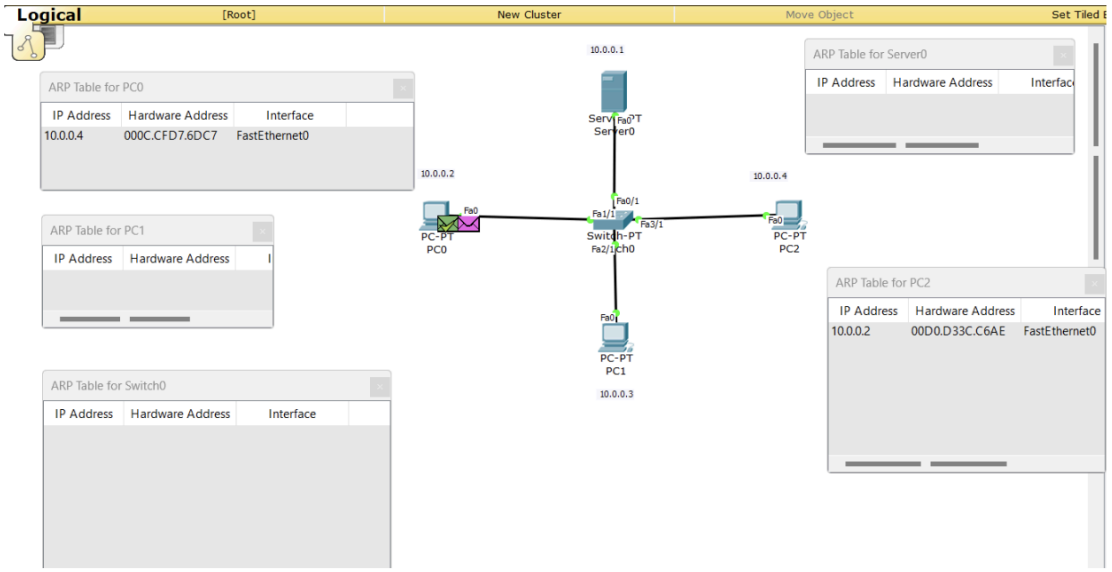


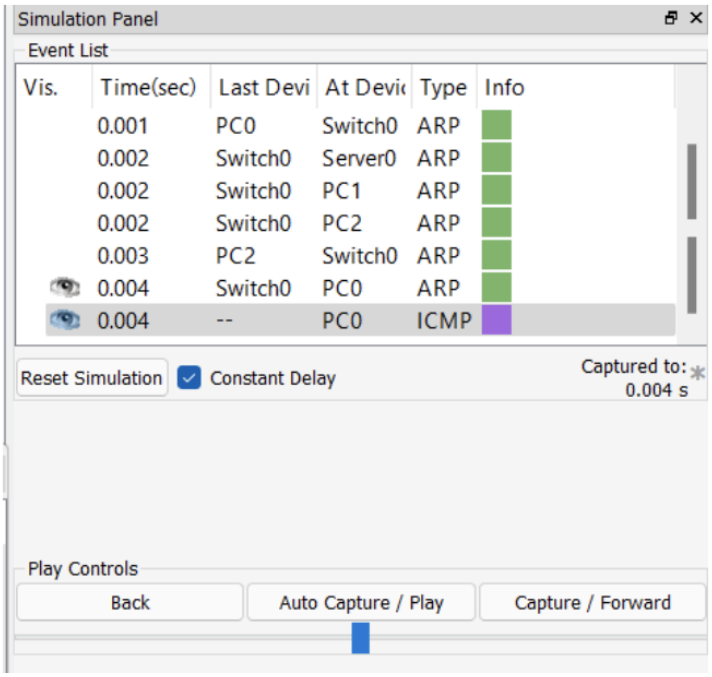
Observation:

As the passage travels from one source host to its destination host, the ARP tables of all devices get updated

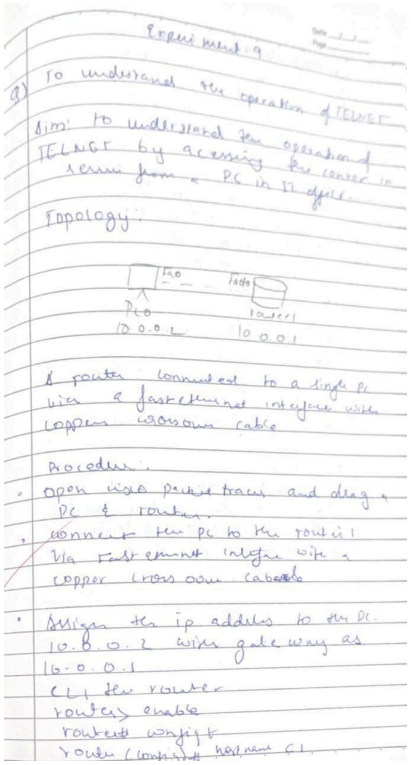
3/11/25

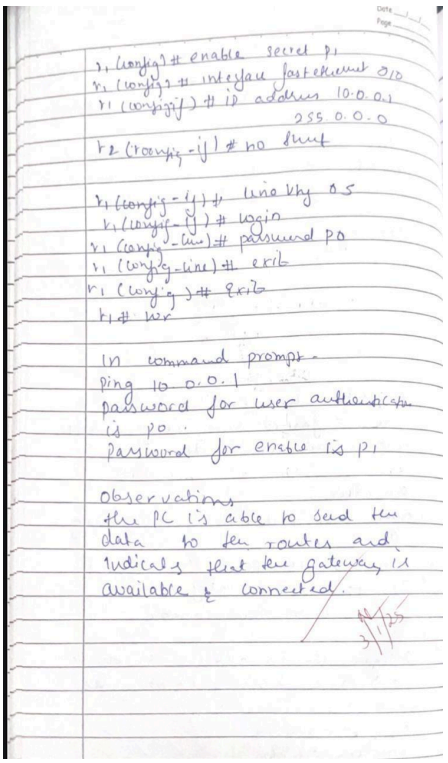
Topology


Logical
[Root]
New Cluster
Move Object
ARP Table for PC0
IP Address
Hardware Address
Interface
10.0.0.4
000C.CFD7.6DC7
FastEthernet0
ARP Table for PC1
ARP Table for Switch0
ARP Table for Server0
ARP Table for PC2
10.0.0.2
00D0.D33C.C6AE
10.0.0.1
10.0.0.3
PC-PT PC0
PC-PT PC1
PC-PT PC2
Switch-PT
Server0

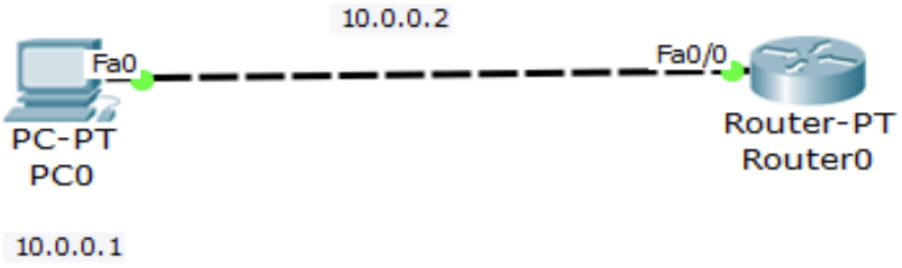

Simulation Panel
Event List
Vis.
Time(sec)
Last Devi
At Devi
Type
Info
0.001 PC0 Switch0 ARP
0.002 Switch0 Server0 ARP
0.002 Switch0 PC1 ARP
0.002 Switch0 PC2 ARP
0.003 PC2 Switch0 ARP
0.004 Switch0 PC0 ARP
0.004 -- PC0 ICMP
Reset Simulation
Constant Delay
Captured to: 0.004 s
Play Controls
Back
Auto Capture / Play
Capture / Forward

Simple TELNET Protocol





Topology



VLAN Protocol

Experiment 10

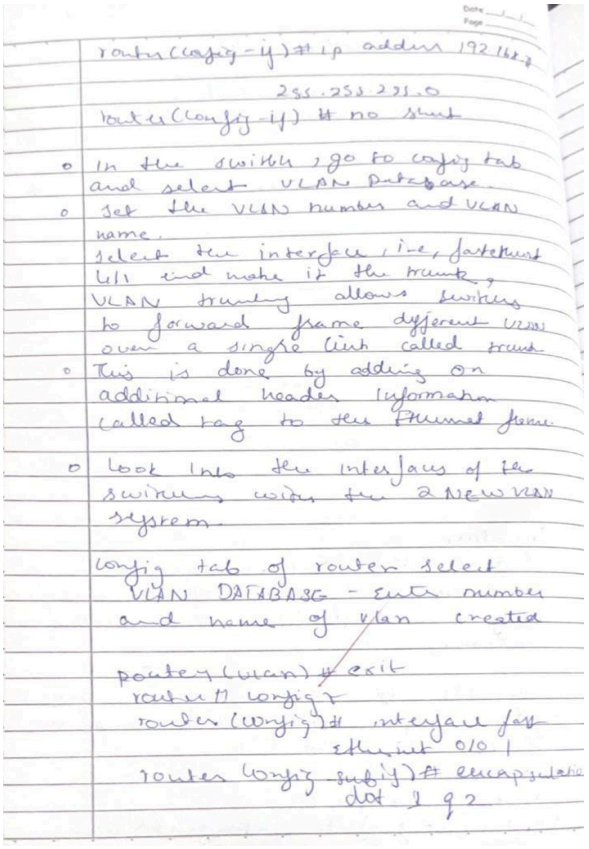
Aim: To construct VLAN and the PC's communicate among a VLAN

Aim: construct a VLAN and enable communication between PC's among a VLAN

Topology

Procedure

- Choose the 1841 router and connect a switch and 4 pcs via ethernet interface and fastethernet interface respectively
- Set the ip address of the PC's and configure the router 1 with ip address 192.168.1.1
  router> enable
  router> config t
  router(config)# interface FA 0/0

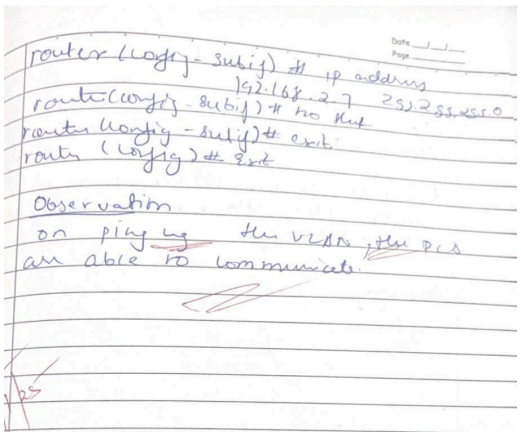


router(config-if)# ip addrss 192.168.1.1 255.255.255.0
router(config-if)# no shut

- In the switch, go to config tab and select VLAN Database
- Set the VLAN number and VLAN name.
  Select the interface i.e., fastethernet 1/1 and make it the trunk.
  VLAN trunking allows switches to forward frame different VLANs over a single link called trunk
- This is done by adding on additional header information called tag to the Ethernet frame
- Look into the interfaces of the switches with the 2 new VLAN system.

Config tab of router select VLAN DATABASE - Enter number and name of vlan created

router(vlan)# exit
router# config t
router(config)# interface fast ethernet 0/0.1
router(config-subif)# encapsulation dot1q 2

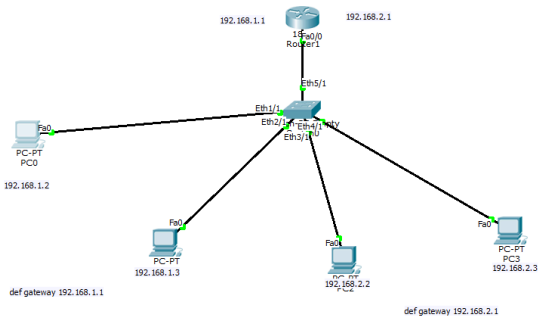


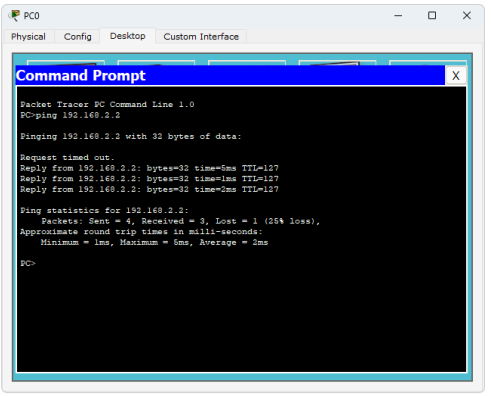
router(config-subif)# ip address 192.168.2.1 255.255.255.0
router(config-subif)# no shut
router(config-subif)# exit
router(config)# exit

Observation

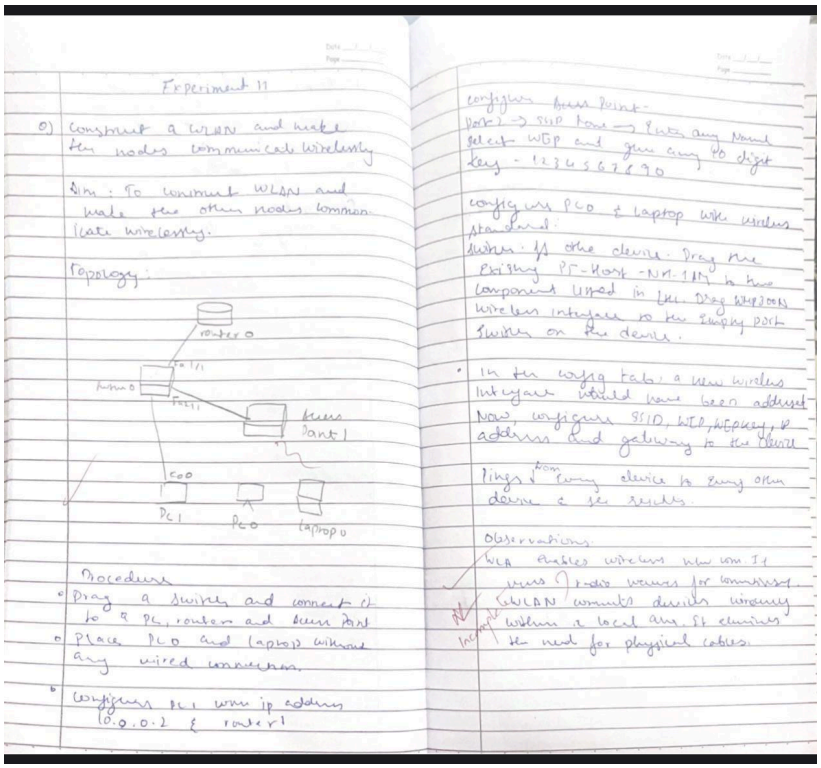
On pinging the VLAN, the PCs are able to communicate.

Topology

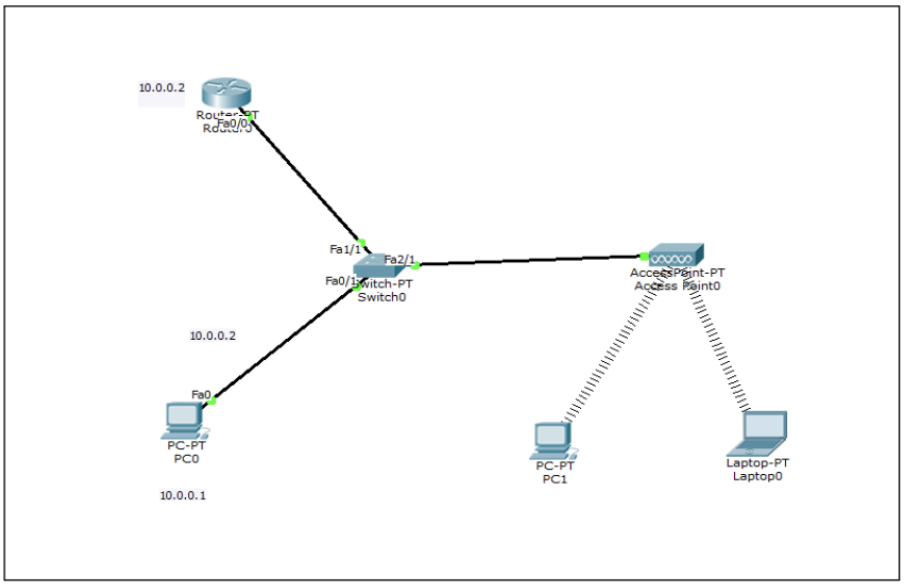

192.168.1.1
192.168.2.1
Fa0/0
Router1
Eth5/1
Eth1/1
Eth2/1
Eth3/1
Eth4/1
Fa0
PC-PT
PC0
192.168.1.2
Fa0
PC-PT
192.168.1.3
Fa0
192.168.2.2
Fa0
PC-PT
PC3
192.168.2.3
def gateway 192.168.1.1
def gateway 192.168.2.1

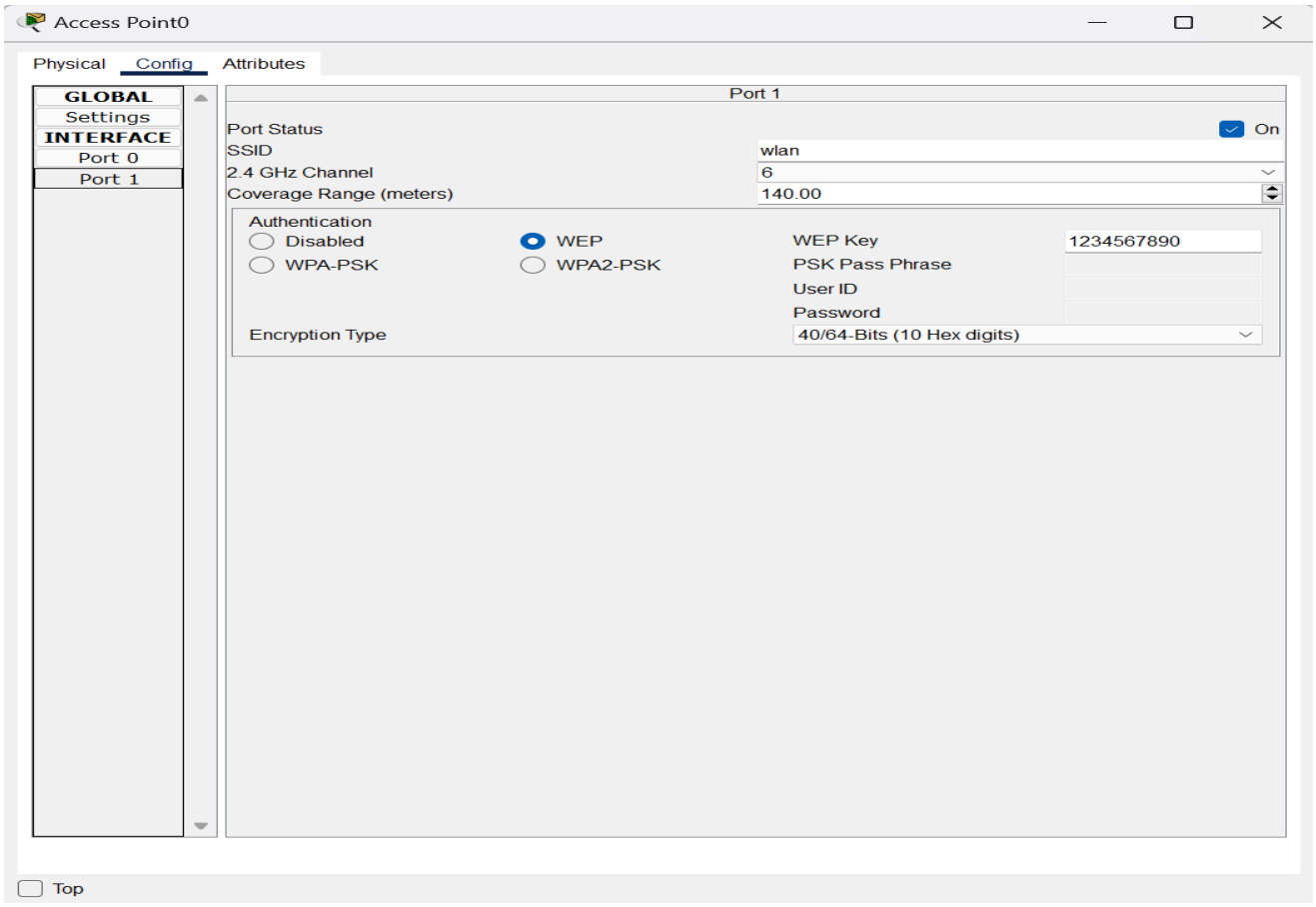

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.2.2: bytes=32 time=5ms TTL=127
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms 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 = 1ms, Maximum = 5ms, Average = 2ms
PC>

WLAN Protocol

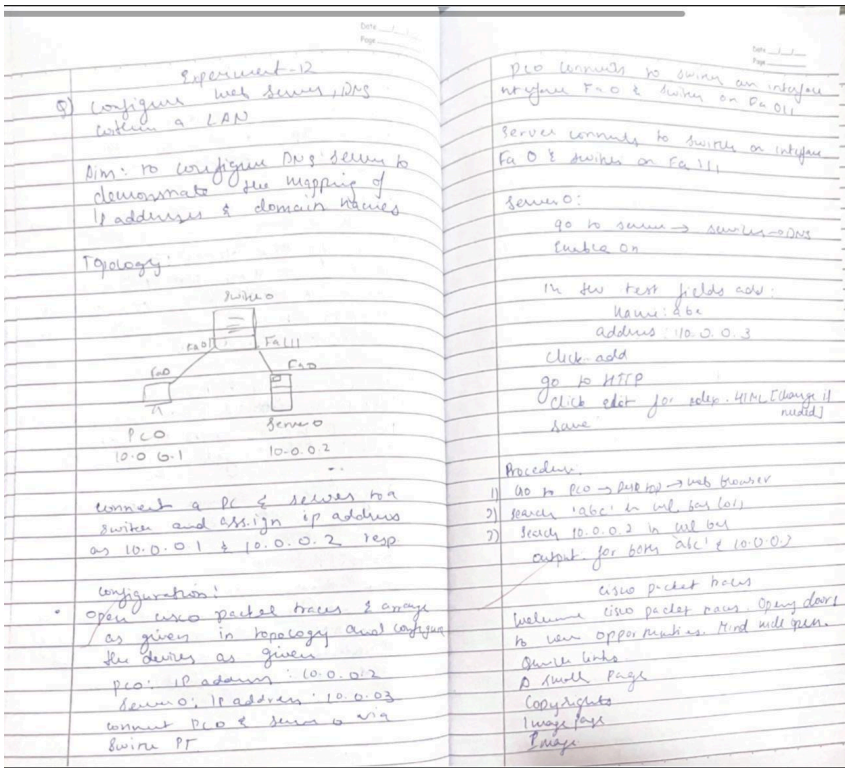


Topology




Access Point0
Physical
Config
Attributes
GLOBAL
Settings
INTERFACE
Port 0
Port 1
Port 1
Port Status
On
SSID
wlan
2.4 GHz Channel
6
Coverage Range (meters)
140.00
Authentication
Disabled
WEP
WPA-PSK
WPA2-PSK
WEP Key
1234567890
PSK Pass Phrase
User ID
Password
Encryption Type
40/64-Bits (10 Hex digits)
Top

DNS within a LAN



Experiment-12

Q) Configure web server, DNS within a LAN

Aim: To configure DNS server to demonstrate the mapping of IP address & domain names

Topology:

connect a PC & server to a switch and assign ip address as 10.0.0.1 & 10.0.0.2 resp.

Configuration:

- Open cisco packet tracer & create as given in topology and configure the devices as given
  PC0: IP address : 10.0.0.2
  Server 0: IP address : 10.0.0.3
  connect PC0 & Server 0 via Switch PT

PC0 connects to switch on interface Fa0 & switch on Fa0/1

Server connects to switch on interface Fa0 & switch on Fa0/11

Server 0:

go to server → services → DNS

Enable on

In the text fields add:

Name: abc

address: 10.0.0.3

Click add

go to HTTP

Click edit for index.html [change if needed]

Save

Procedure:

1) Go to PC0 → Desktop → web browser
2) Search 'abc' in url box (or)
3) Search 10.0.0.2 in url box

output for both abc & 10.0.0.2

cisco packet tracer

Welcome cisco packet tracer. Opens doors to new opportunities. Mind will open.

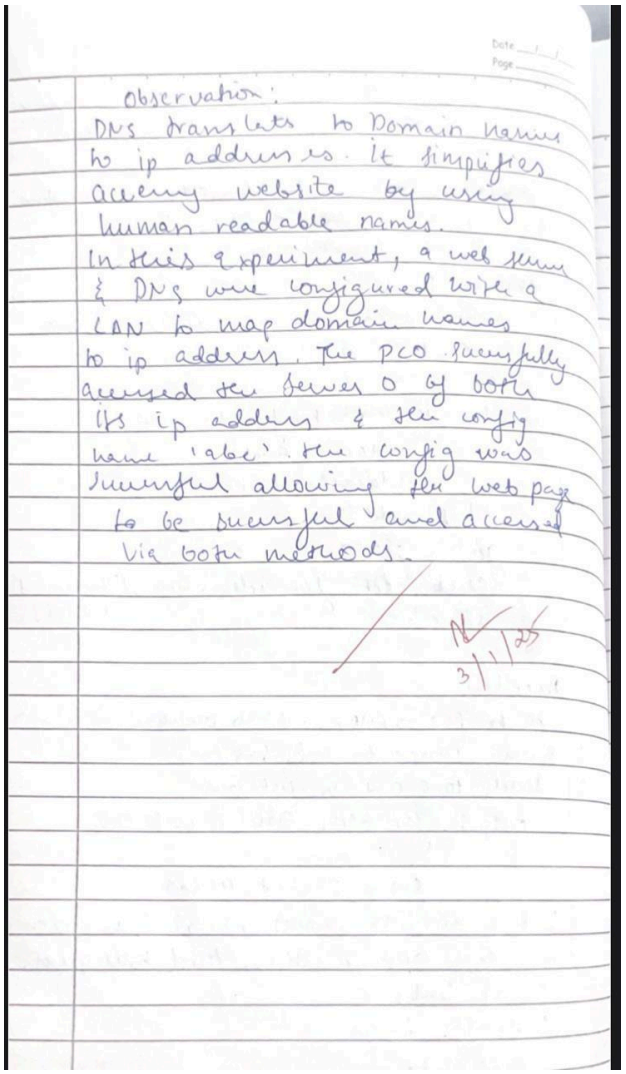
Quick links

A small Page

Copyrights

Image Page

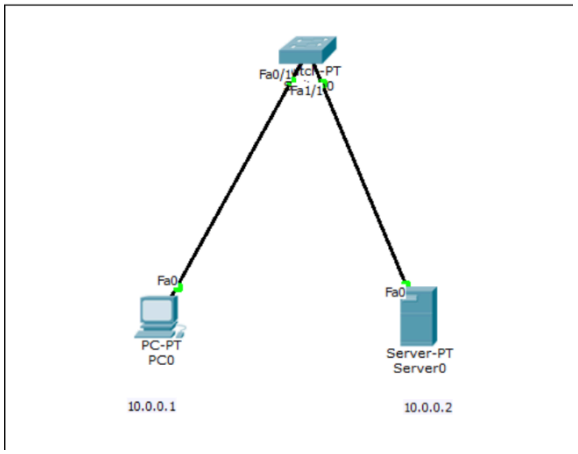
Image

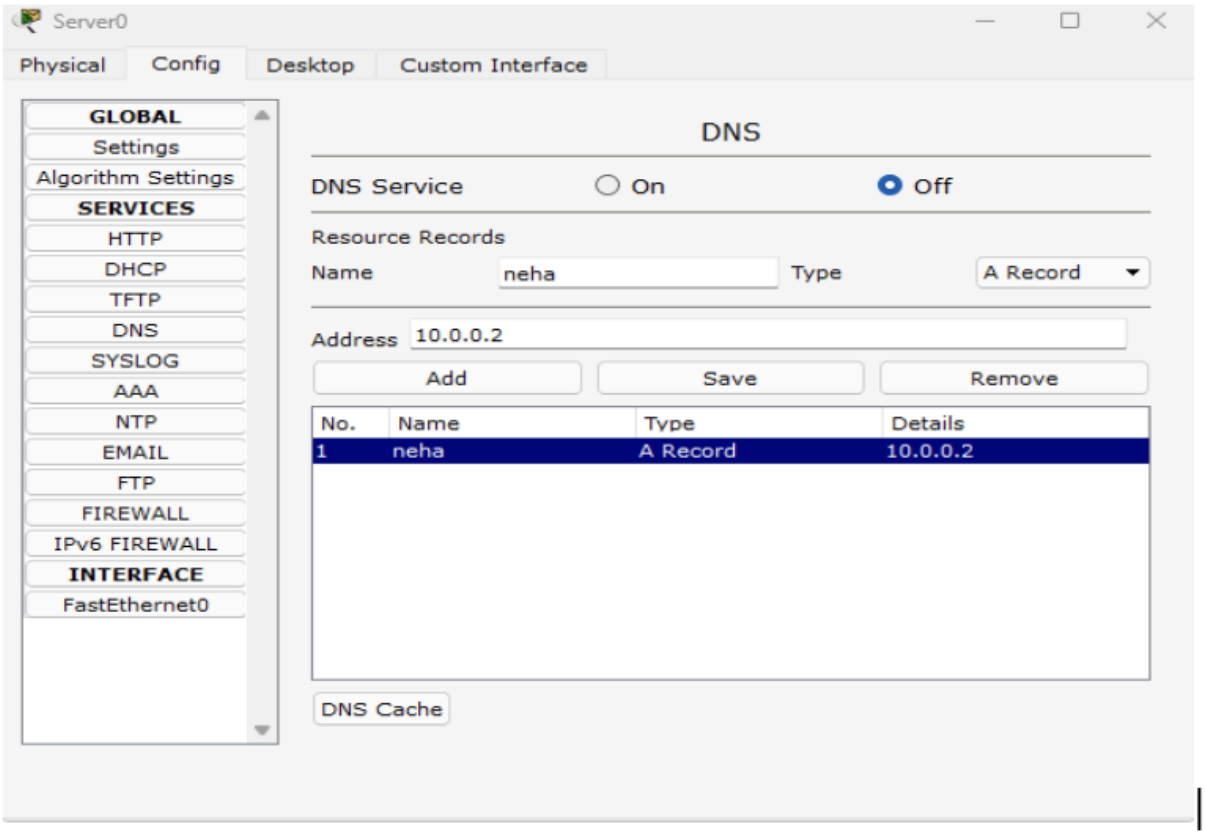


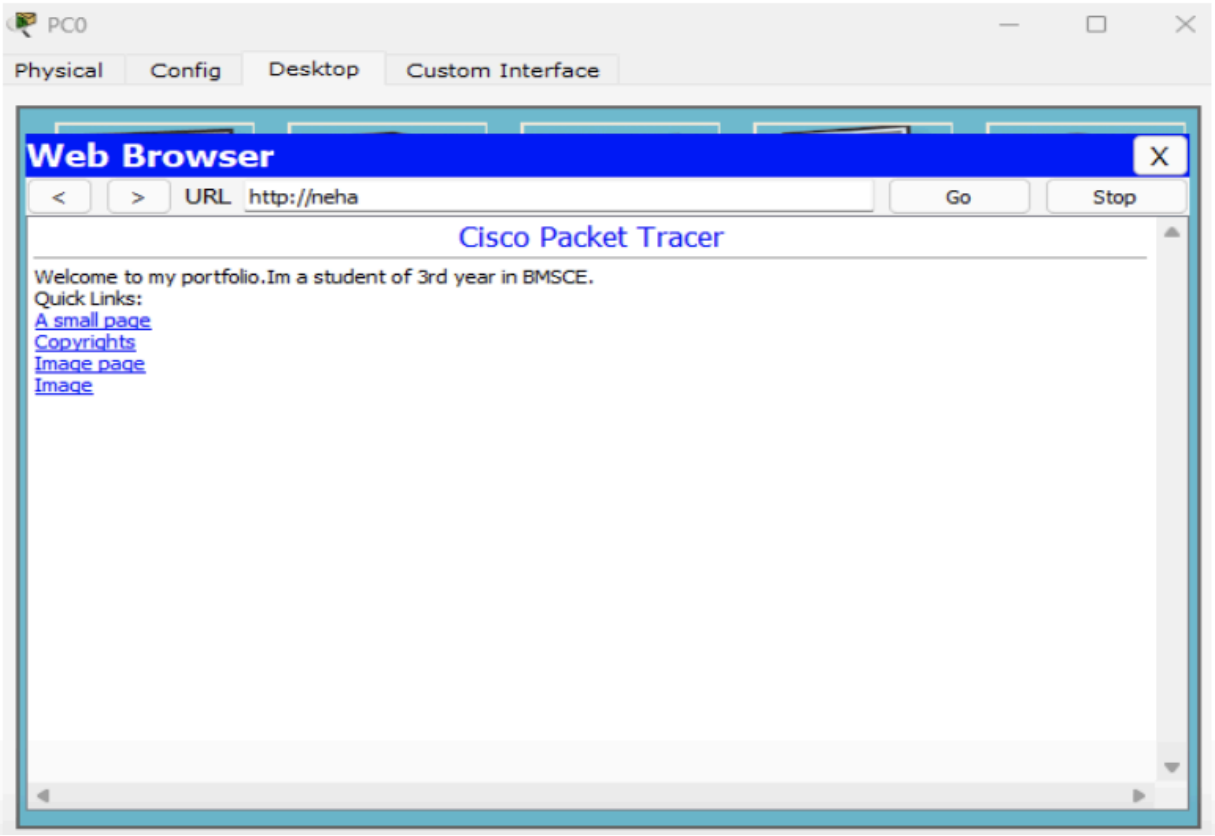
Observation:
DNS translates to Domain names to ip addresses. It simplifies accessing website by using human readable names.
In this experiment, a web server & DNS were configured with a LAN to map domain names to ip address. The PC0 successfully accessed the server 0 by both its ip address & the config name 'abc'. The config was successful allowing the web page to be successful and accessed via both methods.

3/1/25

Topology

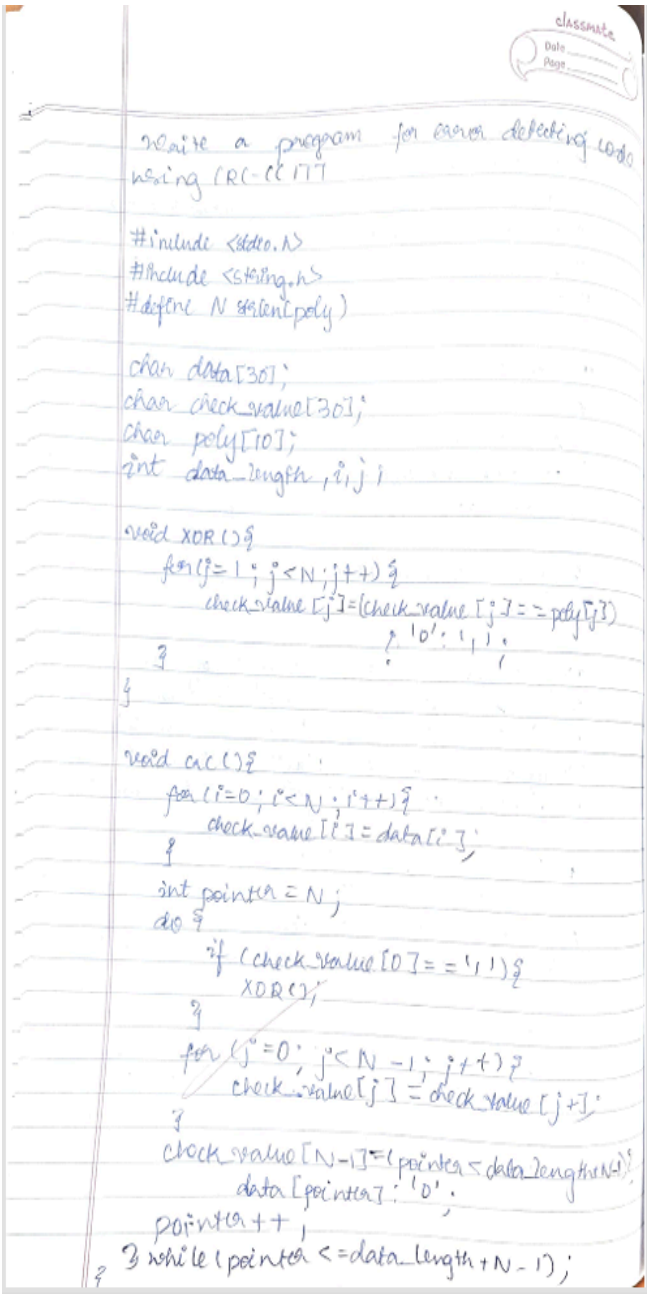



Server0
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
SERVICES
HTTP
DHCP
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
FIREWALL
IPv6 FIREWALL
INTERFACE
FastEthernet0
DNS
DNS Service
On
Off
Resource Records
Name
neha
Type
A Record
Address
10.0.0.2
Add
Save
Remove
No.
Name
Type
Details
1
neha
A Record
10.0.0.2
DNS Cache

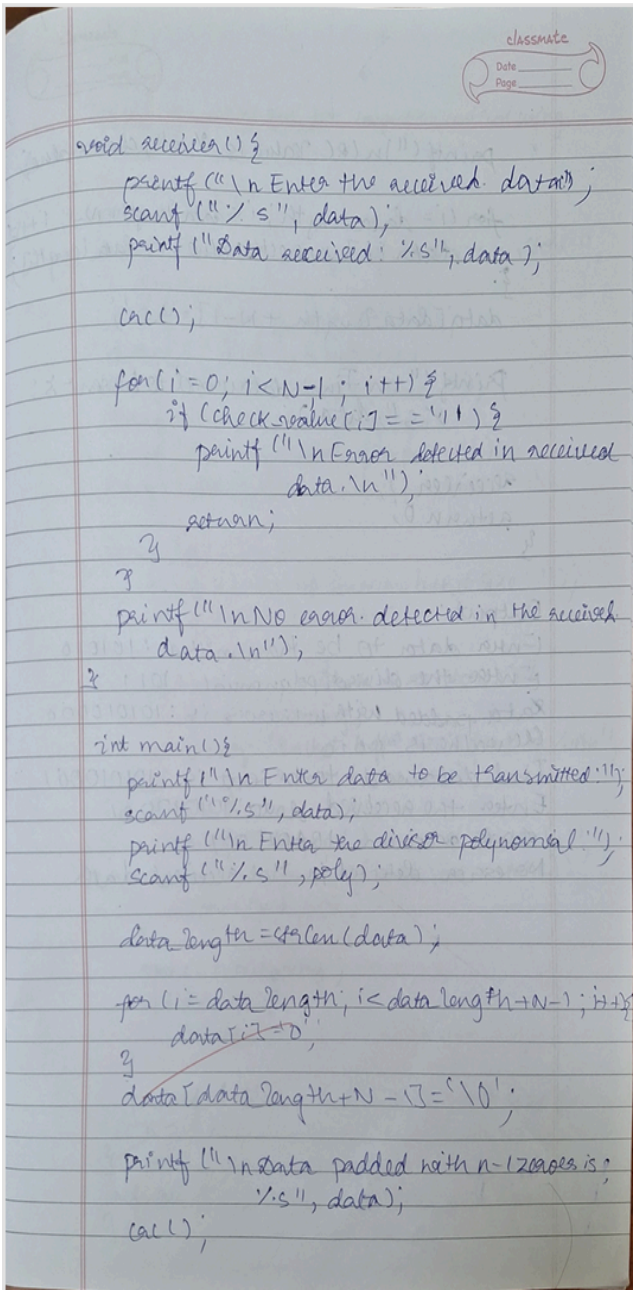

PC0
Physical
Config
Desktop
Custom Interface
Web Browser
URL
http://neha
Go
Stop
Cisco Packet Tracer
Welcome to my portfolio.Im a student of 3rd year in BMSCE.
Quick Links:
A small page
Copyrights
Image page
Image

Cycle-2

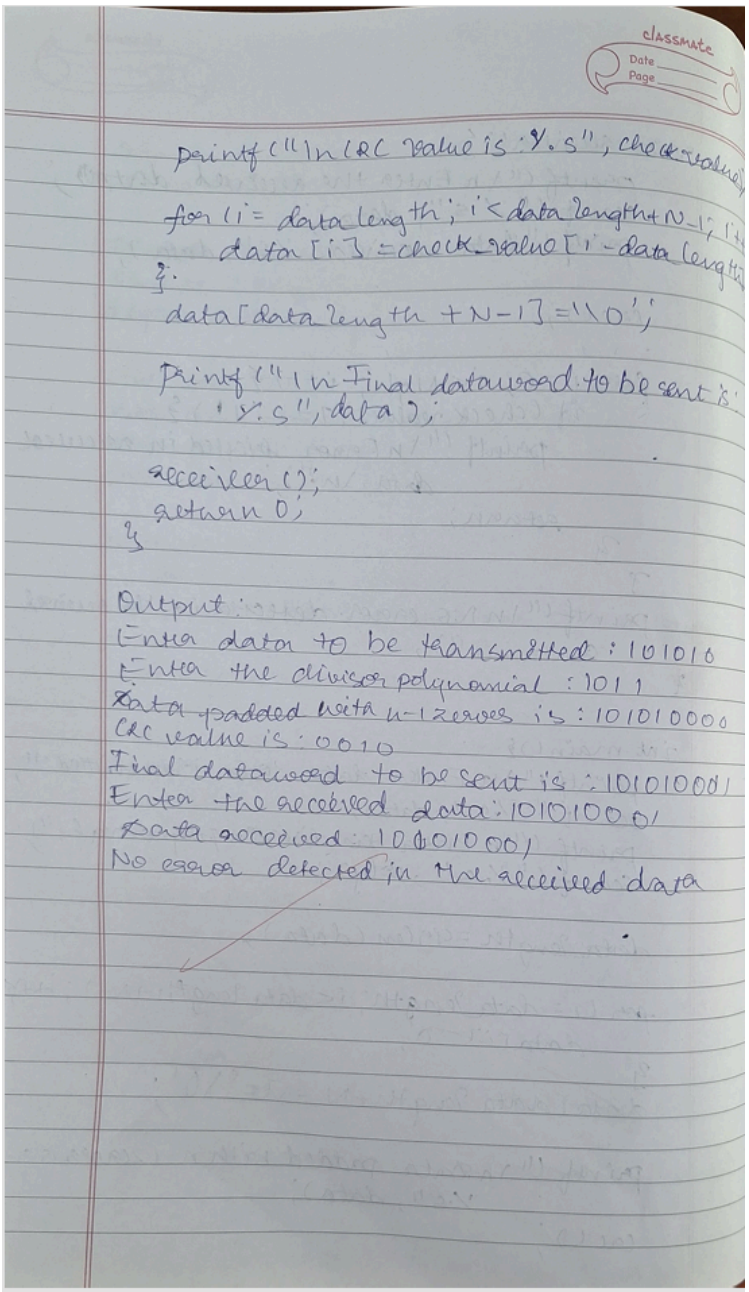
CRC implementation



```
write a program for error detecting code
using CRC-CCITT

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

char data[30];
char check_value[30];
char poly[10];
int data_length,i,j;

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

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



```
void receiver(){
    printf("\n Enter the received data: ");
    scanf("%s", data);
    printf("Data received: %s", data);

    crc();

    for(i=0; i<N-1; i++){
        if(check_value[i]=='1'){
            printf("\n Error detected in received
                    data.\n");
            return;
        }
    }
    printf("\n No error detected in the received
        data.\n");
}

int main(){
    printf("\n Enter data to be transmitted: ");
    scanf("%s", data);
    printf("\n Enter the divisor polynomial: ");
    scanf("%s", poly);

    data_length = strlen(data);

    for(i=data_length; i<data_length+N-1; i++){
        data[i]='0';
    }
    data[data_length+N-1]='\0';

    printf("\n Data padded with n-1 zeroes is:
            %s", data);
    crc();
```



printf("\n CRC value is : %s", check_value);

for (i = data_length; i < data_length + N - 1; i++)
    data[i] = check_value[i - data_length];

data[data_length + N - 1] = '\0';

printf("\n Final dataword to be sent is : %s", data);

receiver();
return 0;
}

Output:
Enter data to be transmitted : 101010
Enter the divisor polynomial : 1011
Data padded with n-1 zeroes is : 101010000
CRC value is : 0010
Final dataword to be sent is : 101010001
Enter the received data : 101010001
Data received : 101010001
No error detected in the received data

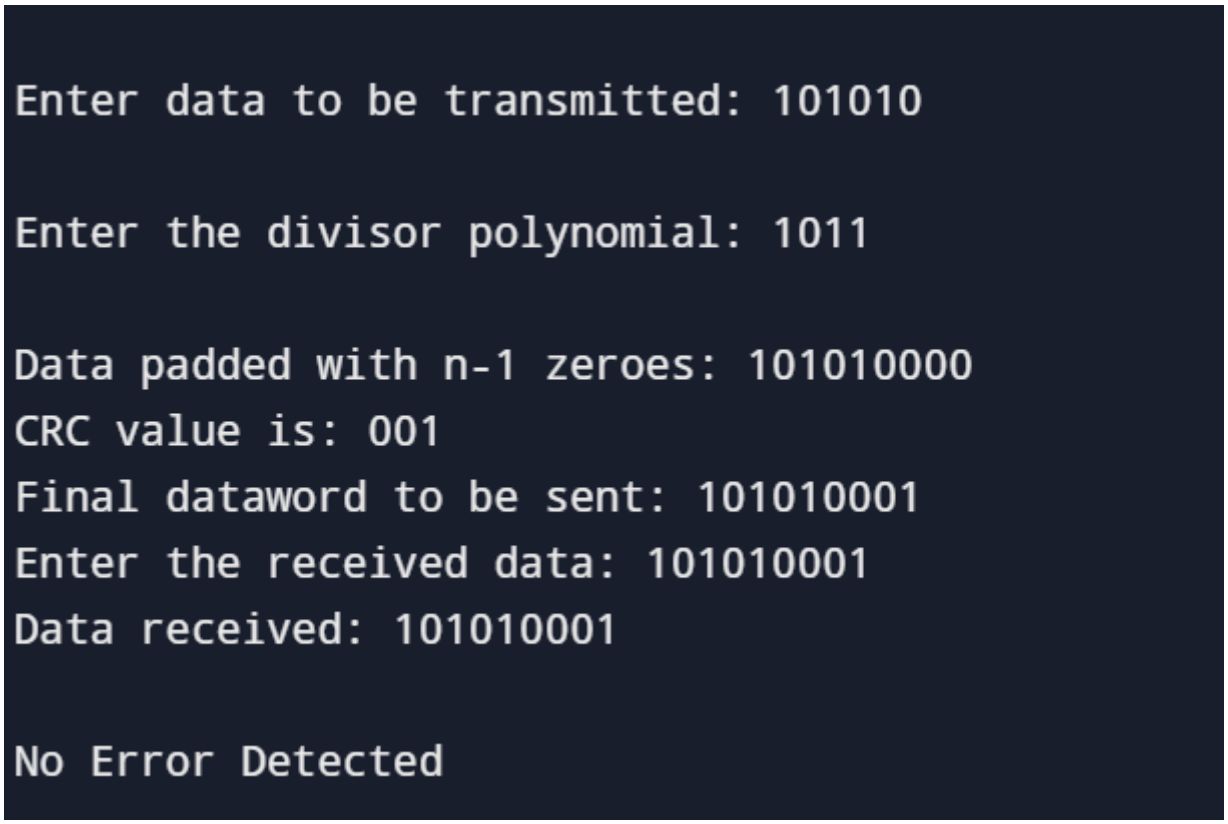
Code:

```
#include <stdio.h>
#include <string.h>

#define N strlen(poly)

char d[30];
char cv[30];
char poly[10];
int dl, i, j;

// Function prototypes
void crc();
void XOR();
```

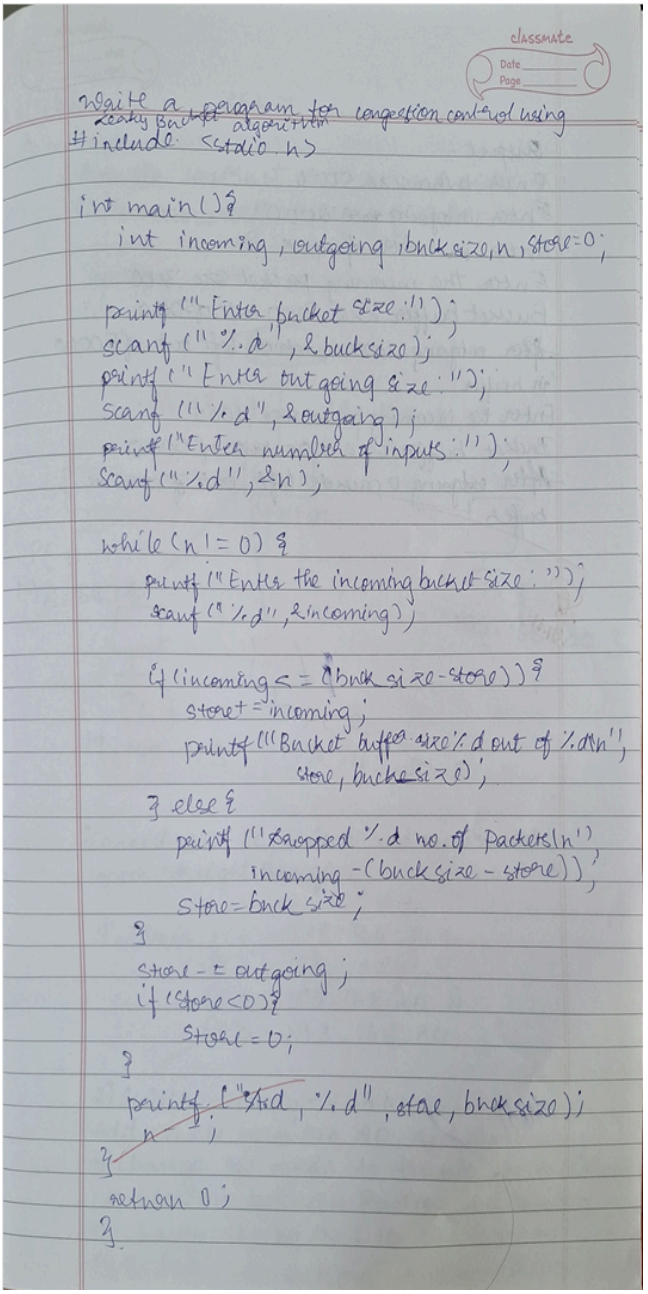
```
void r();

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

void r() {
   printf("\nEnter the received data: ");
   scanf("%s", d);
   printf("Data received: %s\n", d);
   crc();
   for (i = 0; (i < N - 1) && (cv[i] != '1'); i++);
   if (i < N - 1)
      printf("\nError Detected\n");
   else
      printf("\nNo Error Detected\n");
}

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

int main() {
   printf("\nEnter data to be transmitted: ");
   scanf("%s", d);
   printf("\nEnter the divisor polynomial: ");
   scanf("%s", poly);

   dl = strlen(d);
   for (i = dl; i < dl + N - 1; i++) {
      d[i] = '0';
   }
   d[i] = '\0';
   printf("\nData padded with n-1 zeroes: %s", d);
   crc();
```

```
    printf("\nCRC value is: %s", cv);

    for (i = dl; i < dl + N - 1; i++) {
        d[i] = cv[i - dl];
    }
    d[i] = '\0';
    printf("\nFinal dataword to be sent: %s", d);
    r();
    return 0;
}
```

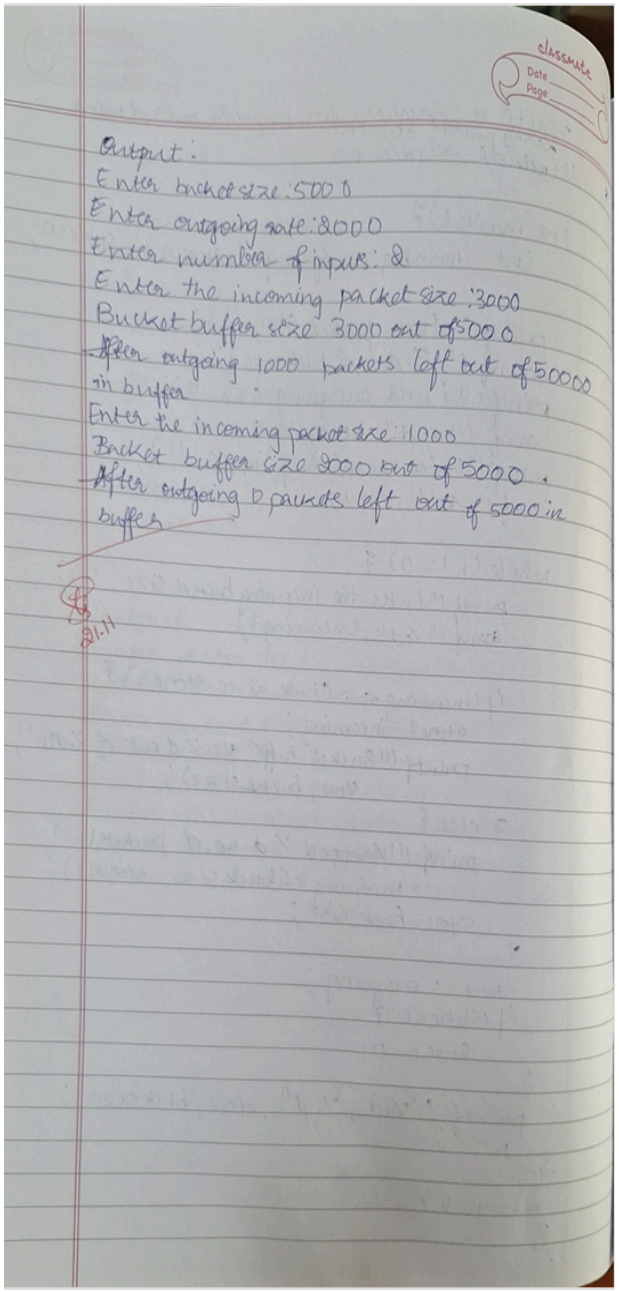
Output:



Leaky Bucket implementation



```
Write a program for congestion control using Leaky Bucket algorithm
#include <stdio.h>

int main(){
    int incoming, outgoing, bucksize, n, store=0;

    printf("Enter bucket size:");
    scanf("%d", &bucksize);
    printf("Enter outgoing size:");
    scanf("%d", &outgoing);
    printf("Enter number of inputs:");
    scanf("%d", &n);

    while(n!=0){
        printf("Enter the incoming bucket size:");
        scanf("%d", &incoming);

        if(incoming<=(bucksize-store)){
            store+=incoming;
            printf("Bucket buffer size %d out of %d\n",
                    store, bucksize);
        }else{
            printf("Dropped %d no. of Packets\n",
                    incoming-(bucksize-store));
            store=bucksize;
        }
        store-=outgoing;
        if(store<0){
            store=0;
        }
        printf("%d, %d", store, bucksize);
        n--;
    }
    return 0;
}
```



Code:

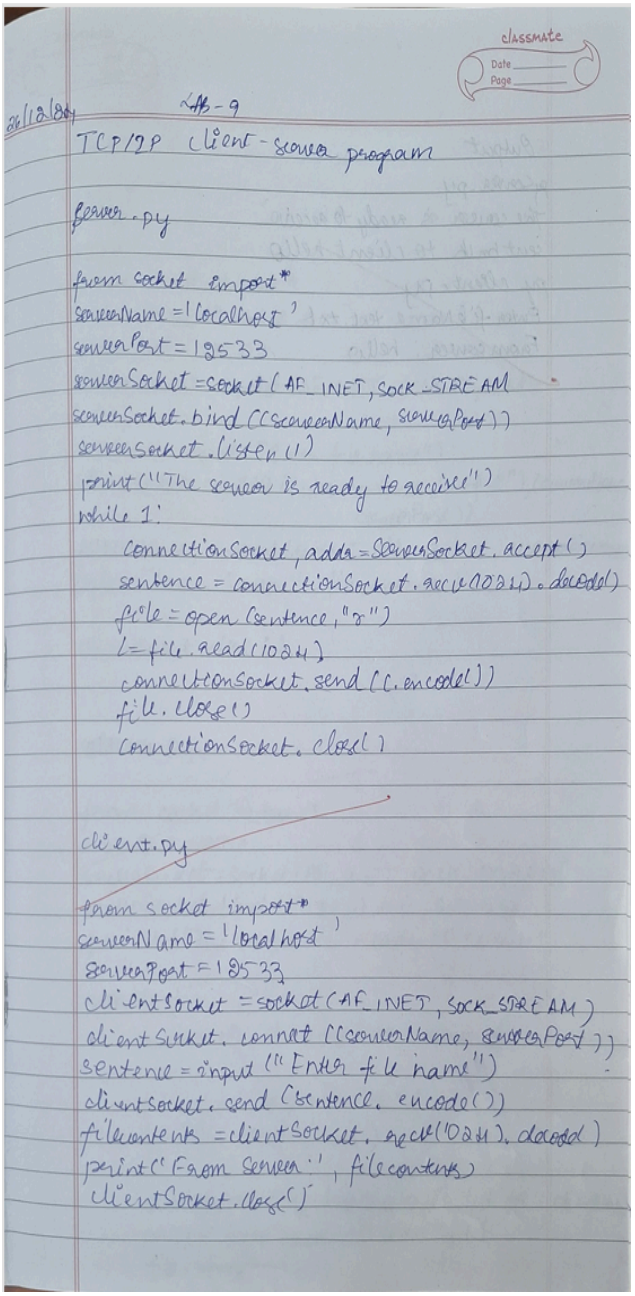
```
#include <stdio.h>

int main() {
    int i,o,bs,n,s=0;
    printf("Enter bucket size");
    scanf("%d",&bs);
    printf("Enter outgoing size");
    scanf("%d",&o);
    printf("Enter the number of inputs:");
    scanf("%d",&n);
    while(n!=0)
    {
```

```
        printf("Enter the incoming bucket size");
        scanf("%d",&i);
        if(i<=(bs-s))
        {
           s+=i;
           printf("Bucket buffer size %d out of %d\n",s,bs);
        }
        else
        {
           printf("Dropped %d no. of packets\n",i-(bs-s));
           printf("Bucket buffer size %d out of %d\n",s,bs);
           s=bs;
        }
        s=s-o;
        printf("After outgoing %d packets left out of %d in buffer\n",s,bs);
        n--;
    }
}
```

Output:

```
Enter bucket size5000
Enter outgoing size2000
Enter the number of inputs:2
Enter the incoming bucket size3000
Bucket buffer size 3000 out of 5000
After outgoing 1000 packets left out of 5000 in buffer
Enter the incoming bucket size1000
Bucket buffer size 2000 out of 5000
After outgoing 0 packets left out of 5000 in buffer
```

TCP Packet Transfer



26/12/24 Lab-9

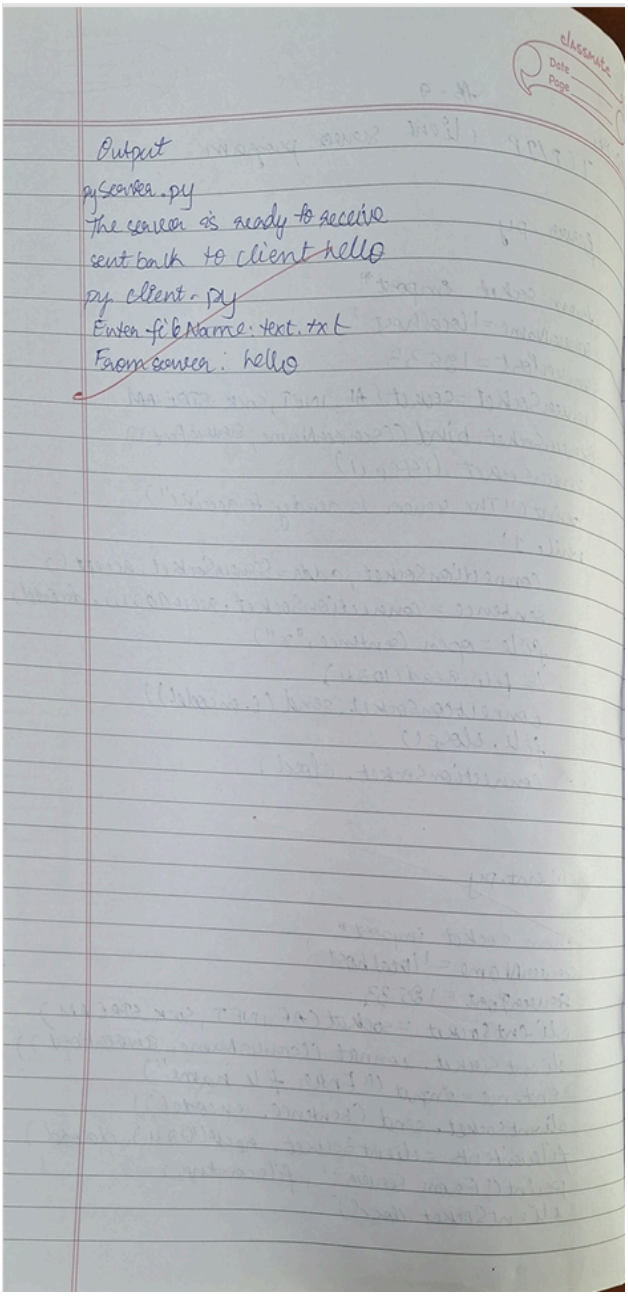
TCP/IP client-server program

server.py

```
from socket import*
serverName = 'localhost'
serverPort = 12533
serverSocket = socket(AF_INET, SOCK_STREAM
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
print("The server is ready to receive")
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()
```

client.py

```
from socket import*
serverName = 'localhost'
serverPort = 12533
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('From Server:', filecontents)
clientSocket.close()
```



server.py

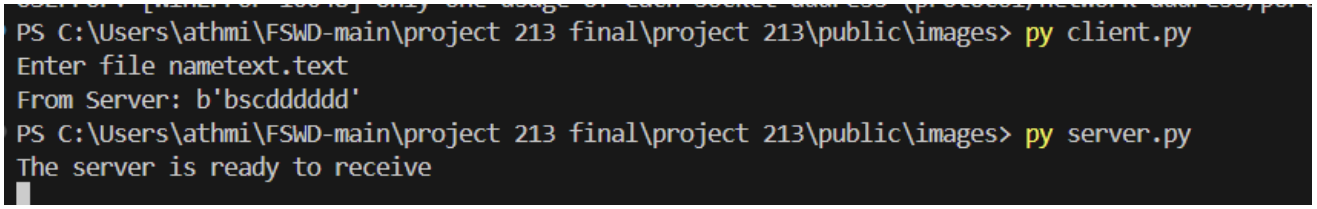
```
from socket import *
serverName='localhost'
serverPort = 12534
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
print ("The server is ready to receive")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
```

```
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    file.close()
    connectionSocket.close()
```

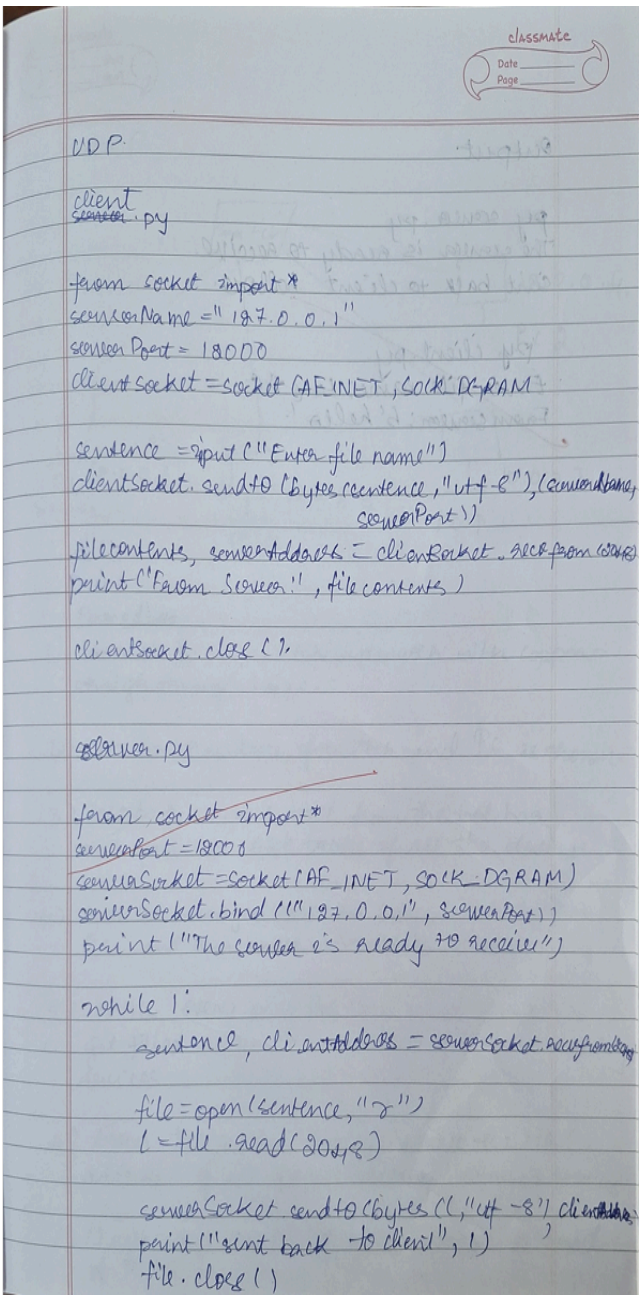
client.py

```
from socket import *
serverName = 'localhost'
serverPort = 12534
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("Enter file name")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('From Server:', filecontents)
clientSocket.close()
```

Output



UDP Packet Transfer



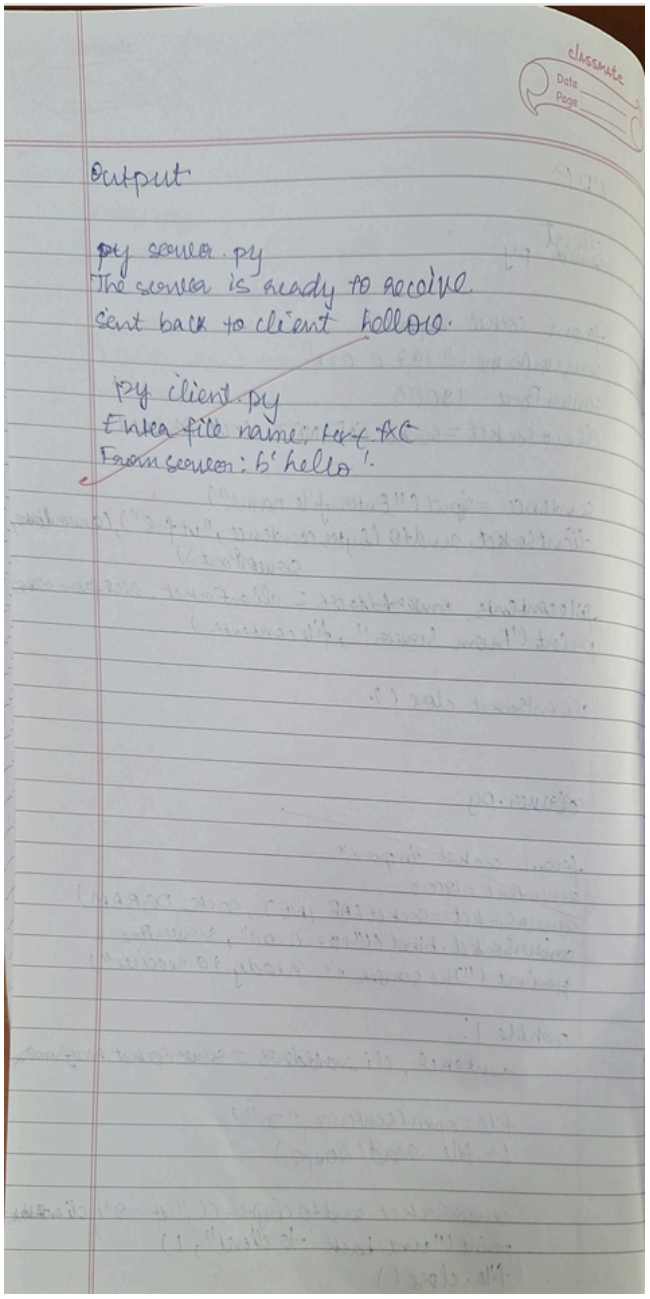
UDP

client.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 18000
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:", fileContents)

clientSocket.close()
```

server.py

```
from socket import *
serverPort = 18001
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)

    file = open(sentence, "r")
    l = file.read(2048)

    serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
    print("sent back to client", l)
    file.close()
```



server.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)

    file=open(sentence,"r")
    l=file.read(2048)
```

```
    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print("sent back to client",l)
    file.close()
```

client.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:', filecontents)

clientSocket.close()
```

Output