VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS
(22CS4PCCON)

Submitted by

SAATVIK S
(1BM21CS178)

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
October-2022 to Feb-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**Computer Networks**" carried out by **Saatvik S (1BM21CS178),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (22CS4PCCON)** work prescribed for the said degree.

Lohith J J
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

INDEX

PROGRAM 1

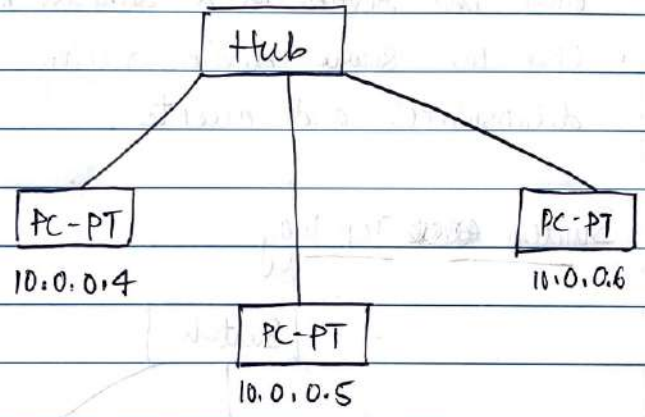
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

Date: 16 / 6 / 23

Computer Networks

Create a topology consisting of 3 or more devices connected with a hub.

Hub Topology



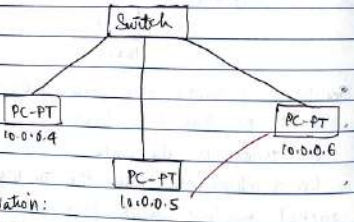
Sending a simple PDU from 10.0.0.4 to 10.0.0.6

- Source PC has 3 out layers which encapsulate and transmits the data.
- On reaching the Hub, the In layer transfers the packet to out-layer from where it is transmitted to all the other end devices connected to the hub
- On reaching the end device, its In-layers decapsulate the packet, if the IP address of the destination does not match, it is dropped.

Date: / /

- If the IP address matches, it transfers packet from in-layer to out-layer. In out-layer 2 and 2, it is encapsulated and transmitted in layer 3.
- The Hub receives and transmits again to the other end devices in a similar manner.
- Once the source device receives the PDU, it decapsulates and accepts.

Switch Topology



Observation:

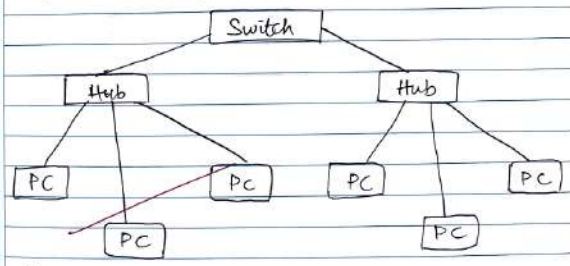
- The source device encapsulates the PDU in its three out layers.
- The switch de-encapsulates and transfers from in-layers to out-layers where it encapsulates and transmits to only the destination device.

Date: / /

unlike in hubs where the PDU is sent to all the devices.

- The destination devices demencapsulates and transfers from in-layers to out-layers where it is again encapsulated and transmitted to switch.
- When the PDU reaches the source, it is de-encapsulated and accepted.

Hybrid - Topology



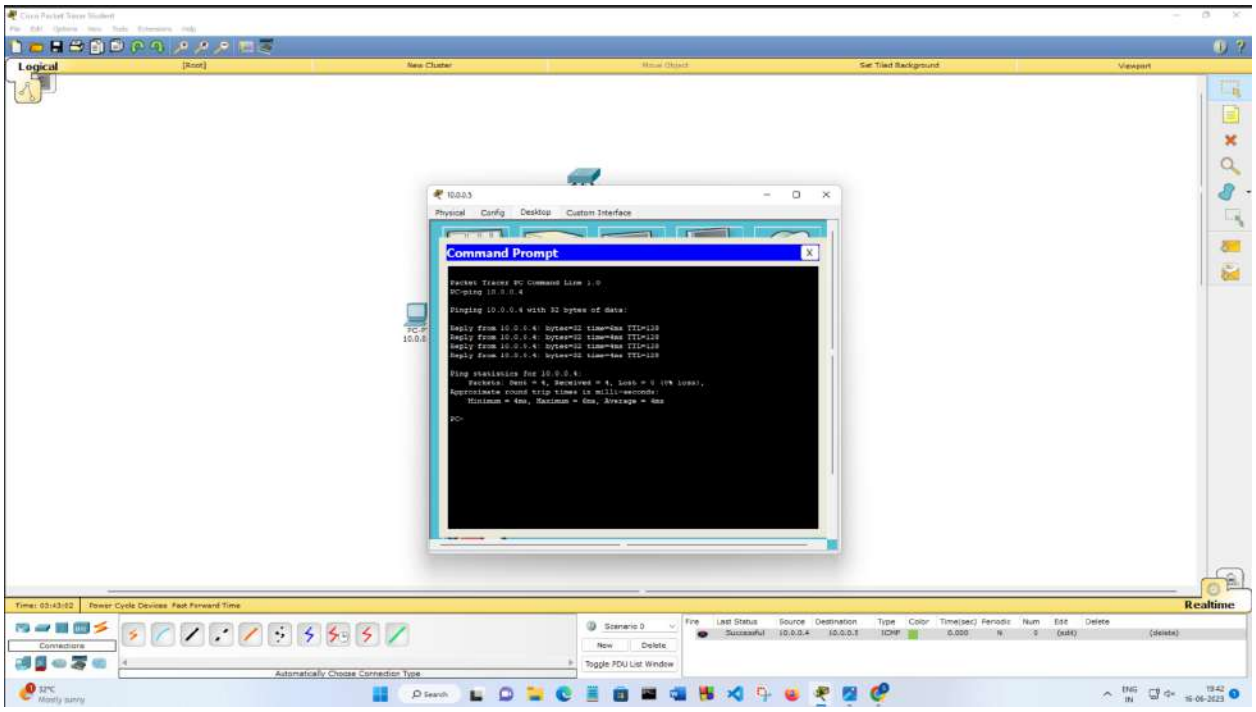
Observation:

- One of the PCs sends a PDU to the Hub which receives it and broadcasts it to the other PCs connected to the hub and the switch connected to it

Date: / /

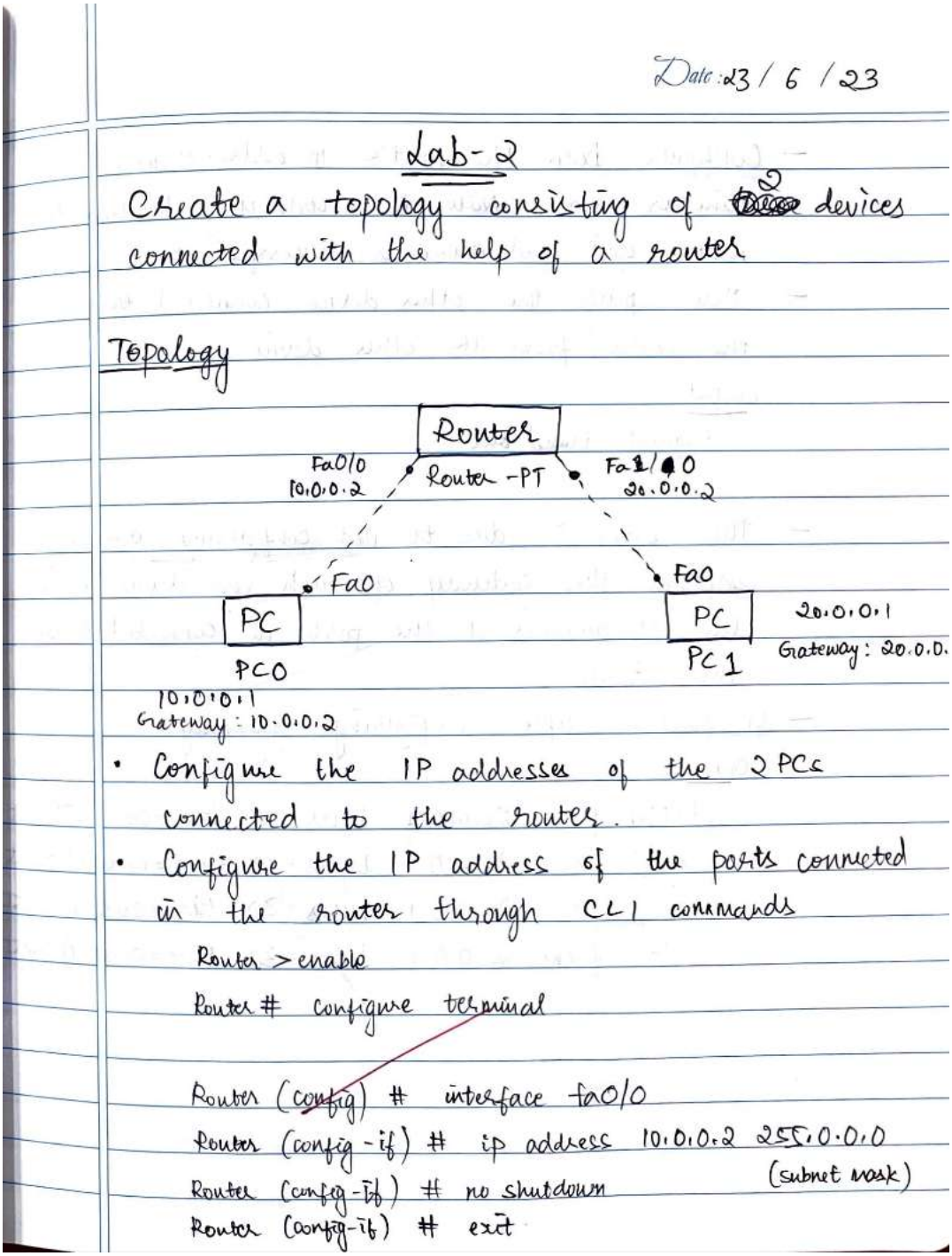
- The switch broadcasts it to the other hub connected to it.
- The hub then broadcasts it to all the PCs connected to it and the destination address which matches, accepts the message.

OUTPUT


Logical
Physical
Config
Desktop
Custom Interface
Command Prompt
Realtime

PROGRAM 2

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



Date: 23 / 6 / 23

Lab-2

Create a topology consisting of 2 devices connected with the help of a router

Topology

- Configure the IP addresses of the 2 PCs connected to the routers.
- Configure the IP address of the ports connected in the router through CLI commands

```
Router > enable
Router # configure terminal
Router (config) # interface fa0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
                                              (subnet mask)
Router (config-if) # no shutdown
Router (config-if) # exit
```

Date: / /

- Configure both the part's ip addresses in a similar way. Now the connection between the router and end device is successful.
- Now ping the other device connected through the router from the other device

Output:

Request timed out

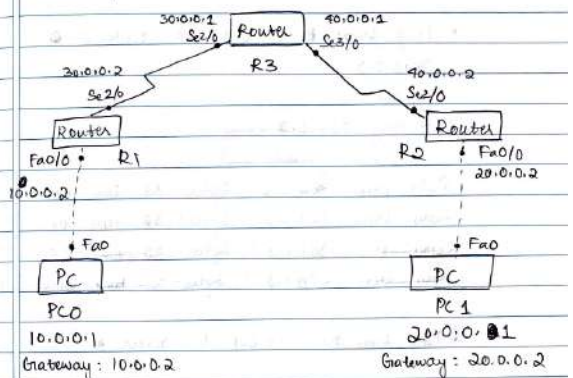
- This error is due to not configuring gateway configure the gateway of each end device with the ip address of the port its connected to in the router.
- On pinging after configuring gateway

Output:

Reply from 20.0.0.1 bytes=32 time=0ms TTL=2
Reply from 20.0.0.1 bytes=32 time=0ms TTL=2
Reply from 20.0.0.1 bytes=32 time=0ms TTL=
Reply from 20.0.0.1 bytes=32 time=0ms TTL=

Date: 23/ 6 /23

Topology



Observation:

- Connect the routers with serial cable
- Configure the ip addresses of the se2 and se3 connections of the appropriate routers with appropriate addresses.
- The routers connected with a single connection must have appropriate address. Example: The se2/0 of R1 and se2/0 of R2 have addresses 30.0.0.2 and 30.0.0.1 respectively.

Date: / /

- Configure both the part's ip addresses in a similar way. Now the connection between the router and end device is successful.
- Now ping the other device connected through the router from the other device

Output:

Request timed out

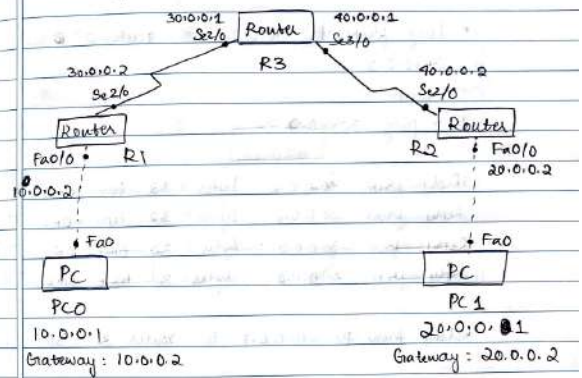
- This error is due to not configuring gateway configure the gateway of each end device with the ip address of the port its connected to in the router.
- On pinging after configuring gateway

Output:

Reply from 20.0.0.1 bytes=32 time=0ms TTL=2
Reply from 20.0.0.1 bytes=32 time=0ms TTL=2
Reply from 20.0.0.1 bytes=32 time=0ms TTL=
Reply from 20.0.0.1 bytes=32 time=0ms TTL=

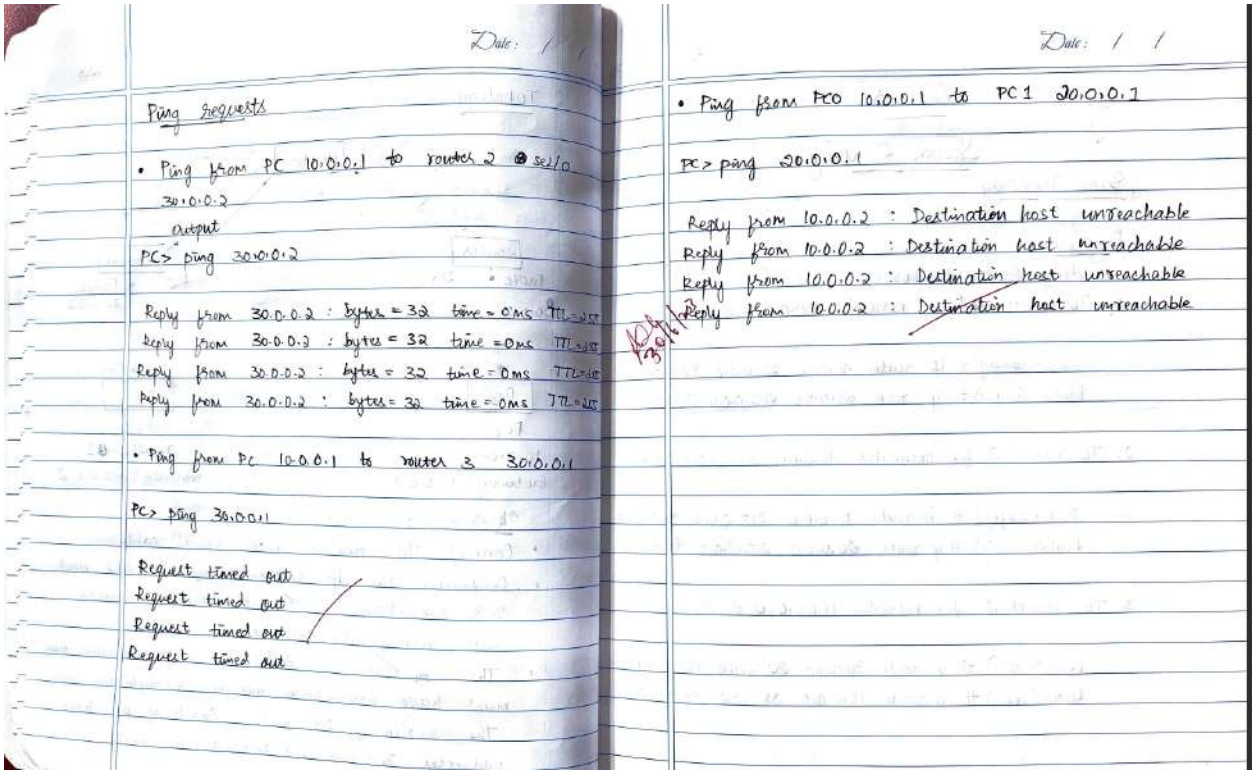
Date: 23/ 6 /23

Topology



Observation:

- Connect the routers with serial cable
- Configure the ip addresses of the se2 and se3 connections of the appropriate routers with appropriate addresses.
- The routers connected with a single connection must have appropriate address. Example: The se2/0 of R1 and se2/0 of R2 have addresses 30.0.0.2 and 30.0.0.1 respectively.



Ping requests

- Ping from PC 10.0.0.1 to router 2 se2/0 30.0.0.2

output

PC> ping 30.0.0.2

Reply from 30.0.0.2 : bytes = 32 time = 0ms TTL=255
Reply from 30.0.0.2 : bytes = 32 time = 0ms TTL=255
Reply from 30.0.0.2 : bytes = 32 time = 0ms TTL=255
Reply from 30.0.0.2 : bytes = 32 time = 0ms TTL=255

- Ping from PC 10.0.0.1 to router 3 30.0.0.1

PC> ping 30.0.0.1

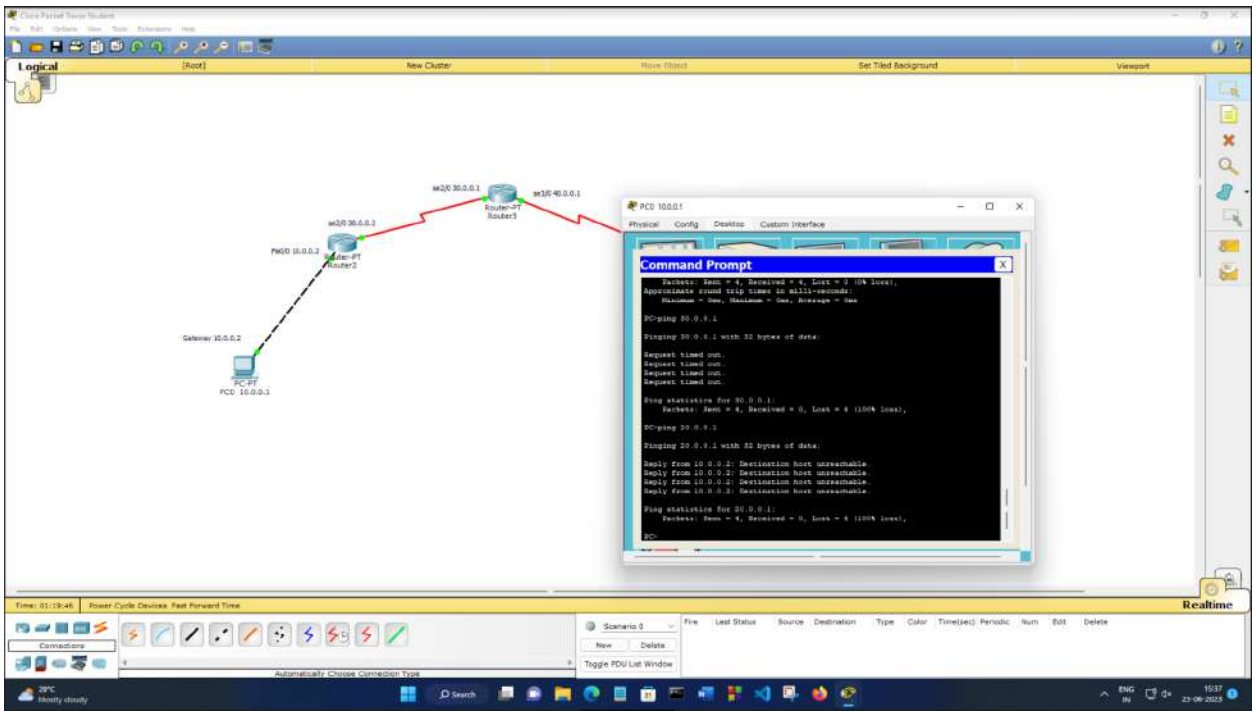
Request timed out
Request timed out
Request timed out
Request timed out

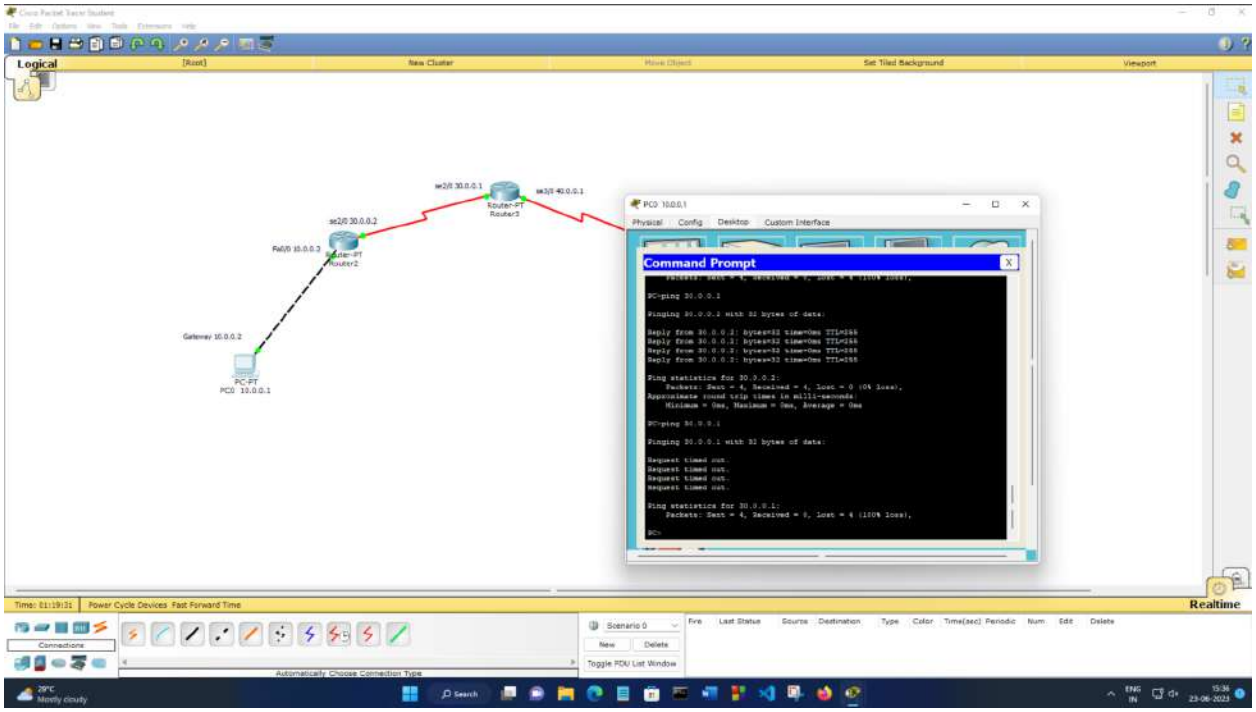
- Ping from PC0 10.0.0.1 to PC 1 20.0.0.1

PC> ping 20.0.0.1

Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable

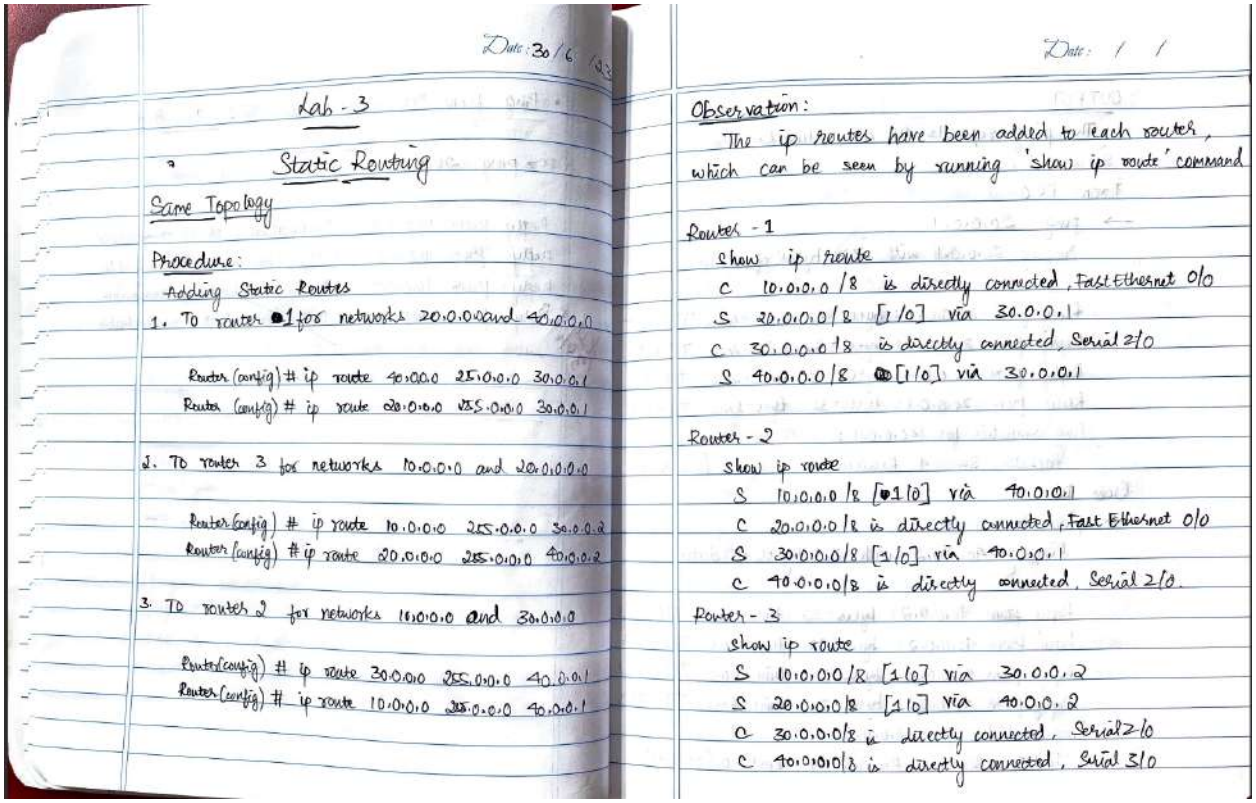
OUTPUT




Logical
Command Prompt
Realtime

PROGRAM 3

Configure default route, static route to the Router



Date: 30/6/23

Lab - 3

Static Routing

Same Topology

Procedure:

Adding Static Routes

1. To router 1 for networks 20.0.0.0 and 40.0.0.0

Router (config)# ip route 40.0.0.0 255.0.0.0 30.0.0.1
Router (config)# ip route 20.0.0.0 255.0.0.0 30.0.0.1

2. To router 3 for networks 10.0.0.0 and 20.0.0.0

Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.2
Router(config)# ip route 20.0.0.0 255.0.0.0 40.0.0.2

3. To router 2 for networks 10.0.0.0 and 30.0.0.0

Router(config)# ip route 30.0.0.0 255.0.0.0 40.0.0.1
Router(config)# ip route 10.0.0.0 255.0.0.0 40.0.0.1

Date: / /

Observation:

The ip routes have been added to each router, which can be seen by running 'show ip route' command

Router - 1

show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 is directly connected, Serial 2/0
S 40.0.0.0/8 [1/0] via 30.0.0.1

Router - 2

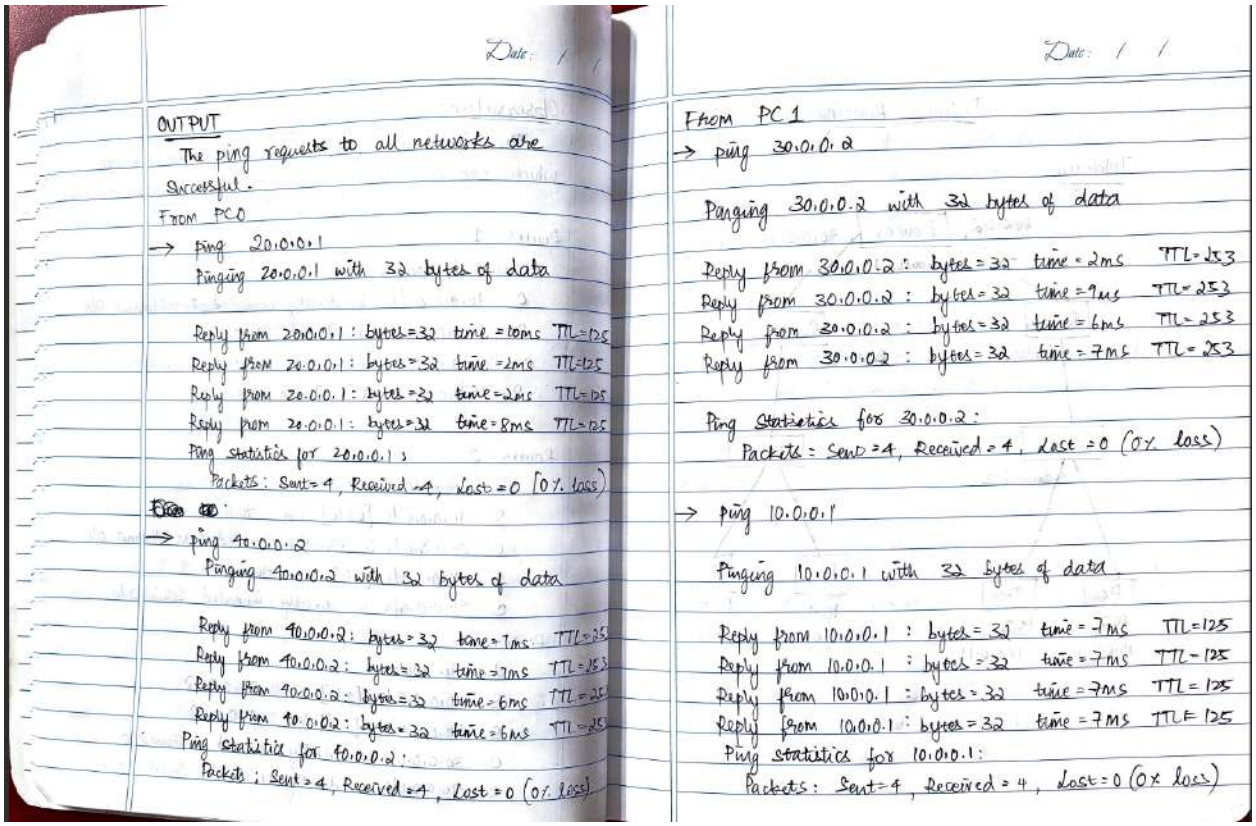
show ip route

S 10.0.0.0/8 [1/0] via 40.0.0.1
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
S 30.0.0.0/8 [1/0] via 40.0.0.1
C 40.0.0.0/8 is directly connected, Serial 2/0

Router - 3

show ip route

S 10.0.0.0/8 [1/0] via 30.0.0.2
S 20.0.0.0/8 [1/0] via 40.0.0.2
C 30.0.0.0/8 is directly connected, Serial 2/0
C 40.0.0.0/8 is directly connected, Serial 3/0



Date: / /

OUTPUT

The ping requests to all networks are successful.

From PC 0

→ ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=2ms TTL=125
Reply from 20.0.0.1: bytes=32 time=2ms TTL=125
Reply from 20.0.0.1: bytes=32 time=8ms TTL=125

Ping statistics for 20.0.0.1:

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

→ 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=253
Reply from 40.0.0.2: bytes=32 time=7ms TTL=253
Reply from 40.0.0.2: bytes=32 time=6ms TTL=253
Reply from 40.0.0.2: bytes=32 time=6ms TTL=253

Ping statistics for 40.0.0.2:

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

Date: / /

From PC 1

→ ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data

Reply from 30.0.0.2: bytes=32 time=2ms TTL=253
Reply from 30.0.0.2: bytes=32 time=9ms TTL=253
Reply from 30.0.0.2: bytes=32 time=6ms TTL=253
Reply from 30.0.0.2: bytes=32 time=7ms TTL=253

Ping statistics for 30.0.0.2:

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

→ ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125

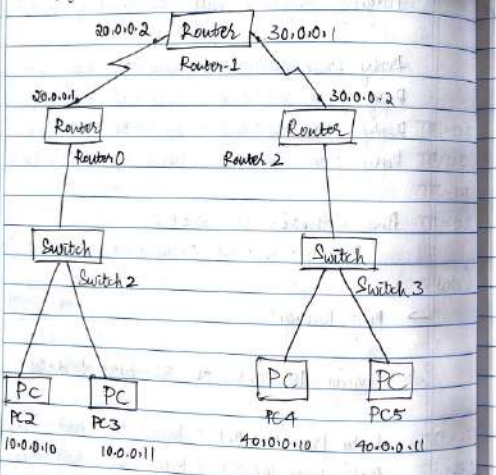
Ping statistics for 10.0.0.1:

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

Date: / /

Default Routing

Topology



Date: / /

Procedure

- Configuring Default route to Router 0 and Router 2

Router 0

```
Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2
```

Router 2

```
Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1
```

- configuring two static routes to Router 1

Router 1

```
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
```

Observation

The default routes to router 0 and router 2 and the static routes to router 1 have been added.

Router 0

```
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
```

Date: / /

Router 2

```
show ip route
C   30.0.0.0/8 is directly connected, Serial 2/0
C   40.0.0.0/8 is directly connected, FastEthernet
S*  0.0.0.0/0 [1/0] via 30.0.0.1
```

Router 1

```
Show ip route
S   10.0.0.0/8 [1/0] via 20.0.0.1
C   20.0.0.0/8 is directly connected, Serial 2/0
C   30.0.0.0/8 is directly connected, Serial 3/0
S   40.0.0.0/8 [1/0] via 30.0.0.2
```

OUTPUT

Ping requests : From PC5

```
→ ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data

Reply from 10.0.0.11: bytes=32 time=10ms TTL=125
Reply from 10.0.0.11: bytes=32 time=6ms TTL=125
Reply from 10.0.0.11: bytes=32 time=8ms TTL=125
Reply from 10.0.0.11: bytes=32 time=5ms TTL=125
```

Date: / /

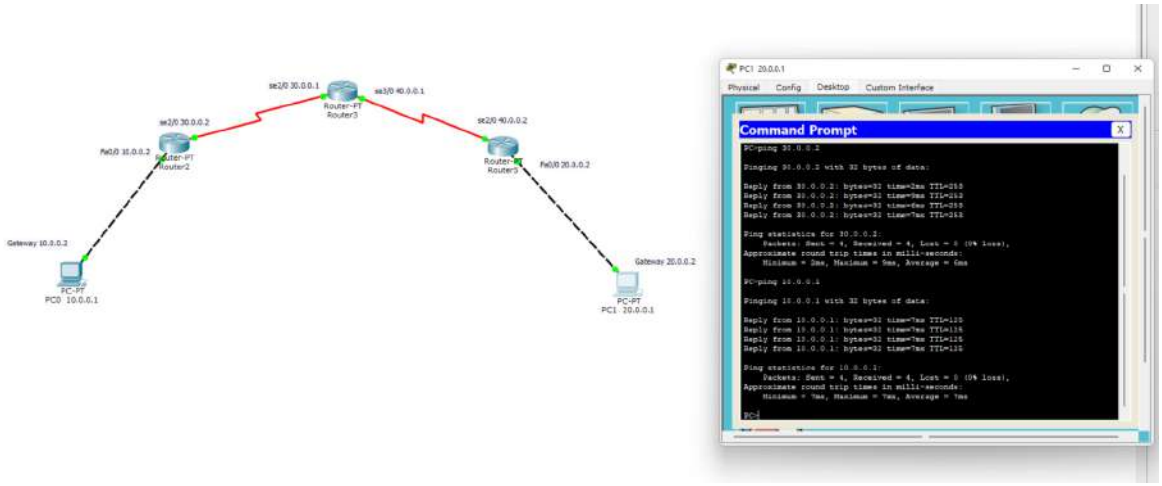
```
Ping statistics for 10.0.0.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
```

From PC2

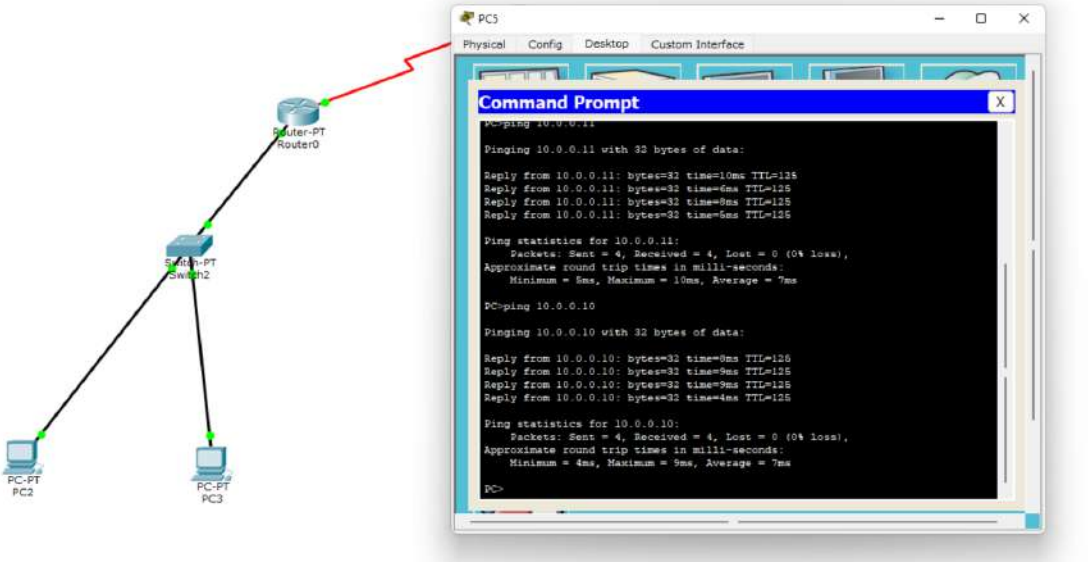
```
→ ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data

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
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)
```

OUTPUT

STATIC ROUTING



DEFAULT ROUTING



PROGRAM 4

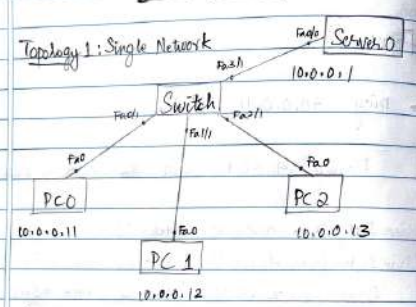
Configure DHCP within a LAN and outside LAN.

Date: / /

Lab-9

DHCP Protocol

Topology 1 : Single Network



Procedure :

- Configure the FastEthernet0 interface of server0 with an ip address
  Fa0/0 : 10.0.0.1
- Configure the DHCP service of the server
  Pool Name : serverPool
  DNS Server : 0.0.0.0
  Default Gateway : 0.0.0.0
  Start IP address : 10 0 0 11
  Subnet Mask : 255 0 0 0

Date: / /

- For each PC, change the IP setting from static to DHCP

Observation

- All PCs connected to the server through the switch are assigned an IP address dynamically

Output

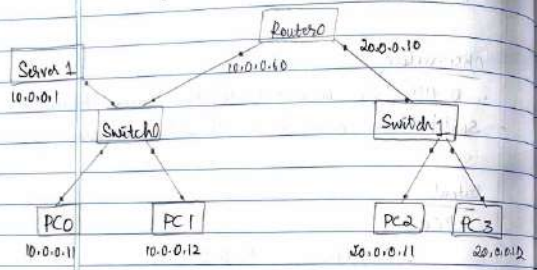
PC0
IP address 10.0.0.11

PC1
IP address 10.0.0.12

PC2
IP address 10.0.0.13

Date: / /

Topology - 2



Procedure

- Connect another network through a router to the previous topology
- Add as helper address to the router, the ip address of server.
  ip helper-address 10.0.0.1
- Add as gateway to the server, the FastEthernet0 port address of the router
  Gateway : 10.0.0.60

Date: / /

- Add another pool to the DHCP service
  Pool Name : serverPool 1
  Default gateway : 10.0.0.60
  Start IP address : 20 0 0 11
- Change IP configuration to DHCP in all PCs.

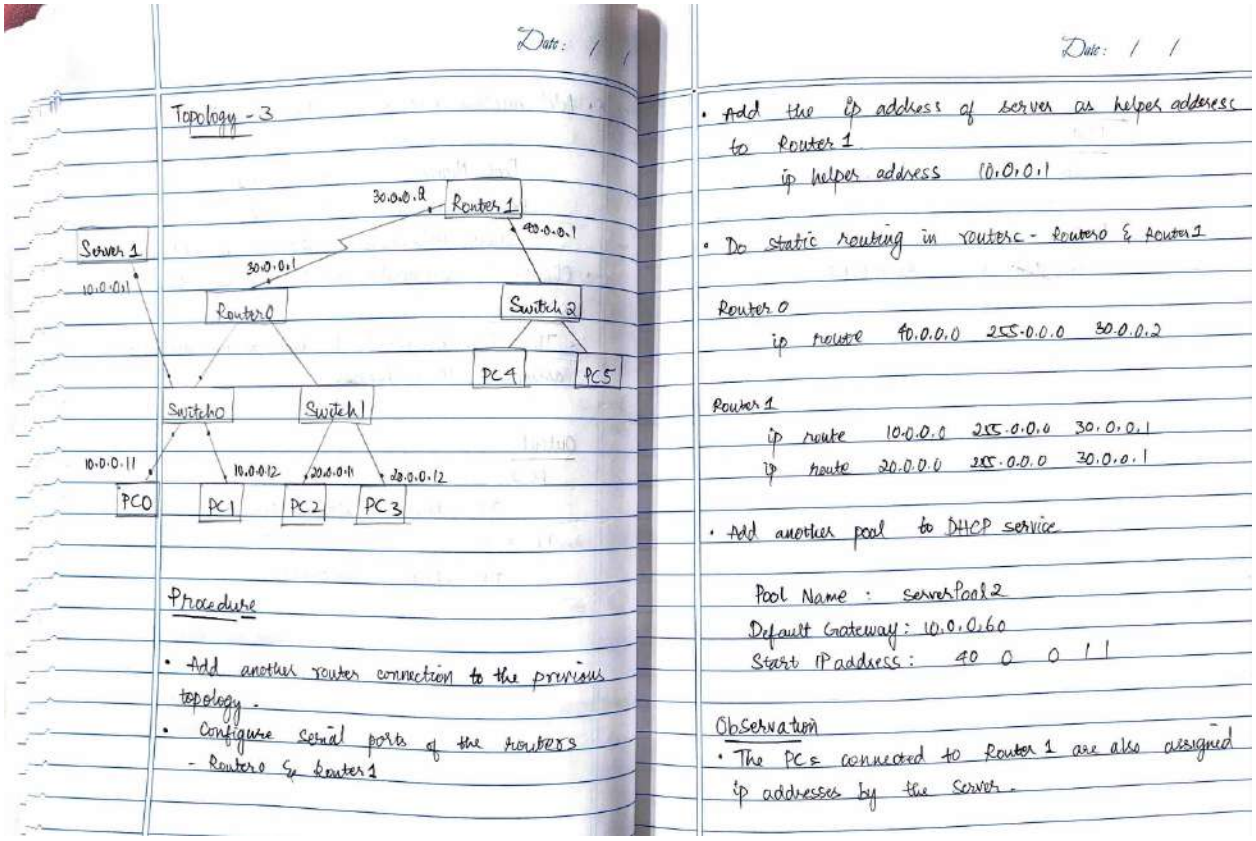
observation

- The PCs connected to the router are also assigned IP addresses

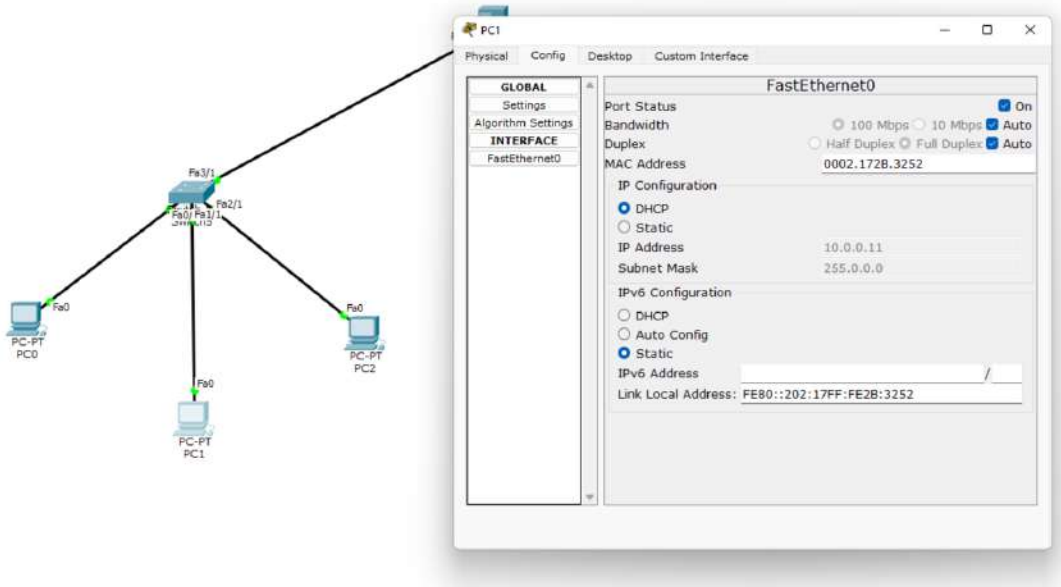
Output

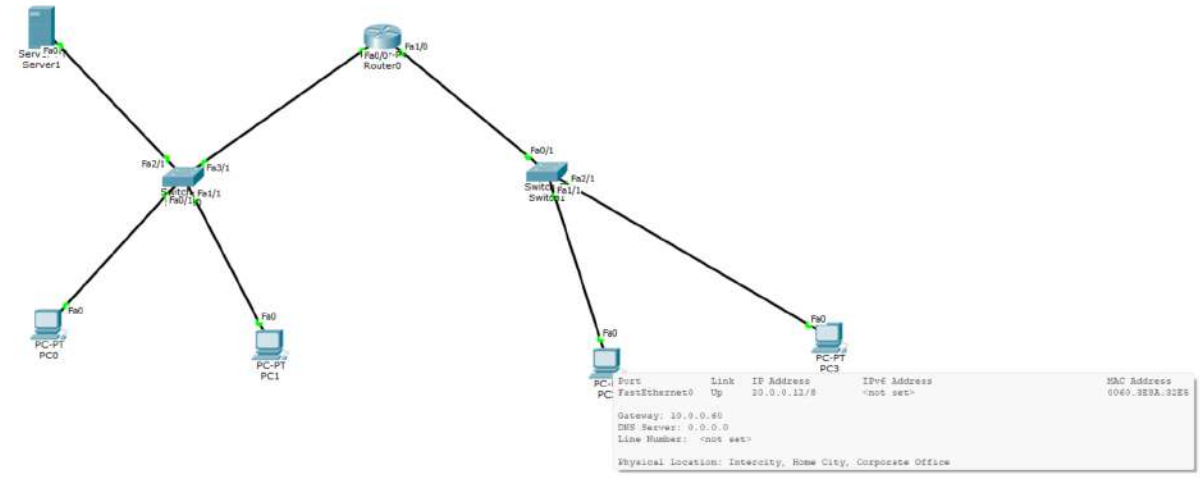
PC2
IP address 20.0.0.11

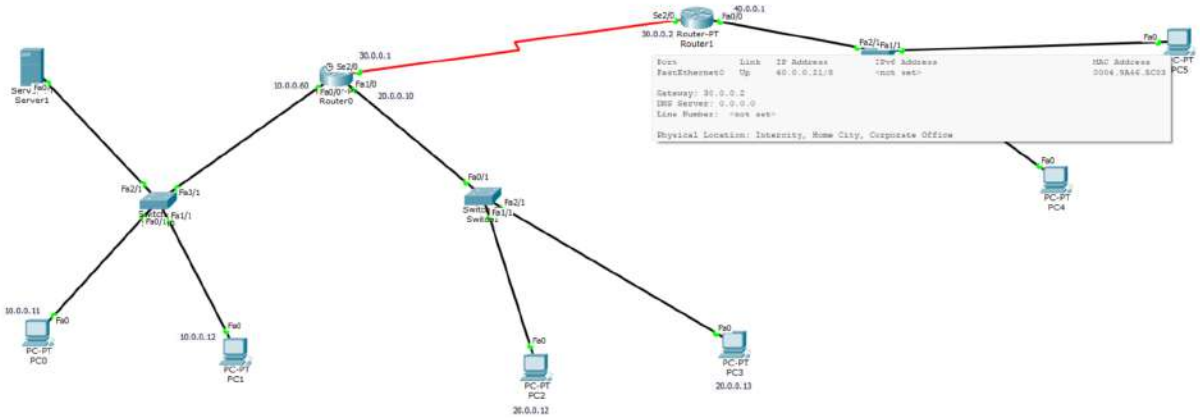
PC3
IP address 20.0.0.12


Date: / /
Topology - 3
Server 1
10.0.0.1
30.0.0.2
Router 1
40.0.0.1
30.0.0.1
Router 0
Switch 2
PC4
PC5
Switch0
Switch1
10.0.0.11
10.0.0.12
20.0.0.11
20.0.0.12
PC0
PC1
PC2
PC3
Procedure
• Add another router connection to the previous topology.
• Configure serial ports of the routers - Router 0 & Router 1
Date: / /
• Add the ip address of server as helper address to Router 1
ip helper address 10.0.0.1
• Do static routing in routers - Router 0 & Router 1
Router 0
ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router 1
ip route 10.0.0.0 255.0.0.0 30.0.0.1
ip route 20.0.0.0 255.0.0.0 30.0.0.1
• Add another pool to DHCP service
Pool Name : serverPool2
Default Gateway: 10.0.0.60
Start IP address: 40 0 0 11
Observation
• The PCs connected to Router 1 are also assigned ip addresses by the server.

OUTPUT:


Fa3/1
Fa2/1
Fa0
PC-PT
PC0
Fa0
PC-PT
PC2
Fa0
PC-PT
PC1
PC1
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
0002.172B.3252
IP Configuration
DHCP
Static
IP Address
10.0.0.11
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address:
FE80::202:17FF:FE2B:3252


Server1
Router0
Switch0
Switch1
PC0
PC1
PC3
Port Link IP Address IPv6 Address MAC Address
FastEthernet0 Up 20.0.0.12/8 <not set> 0060.3E9A.32E6
Gateway: 10.0.0.60
DNS Server: 0.0.0.0
Line Number: <not set>
Physical Location: Intercity, Home City, Corporate Office

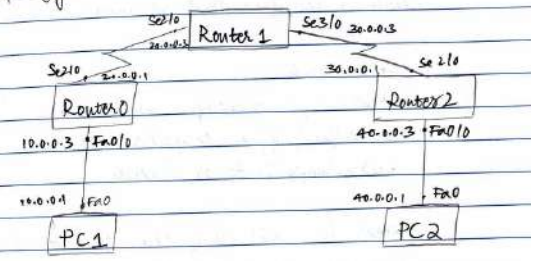

Server1
Router0
Router1
Switch0
Switch1
PC0
PC1
PC2
PC3
PC4
PC5
30.0.0.1
30.0.0.2
40.0.0.1
10.0.0.60
20.0.0.10
10.0.0.11
10.0.0.12
20.0.0.12
20.0.0.13
Port Link IP Address IPv6 Address MAC Address
FastEthernet0 Up 40.0.0.11/8 <not set> 0004.9A46.5C02
Gateway: 30.0.0.2
DNS Server: 0.0.0.0
Line Number: <not set>
Physical Location: Intercity, Home City, Corporate Office

PROGRAM 5

Configure RIP routing Protocol in Routers

Aim: To configure RIP routing protocol in router.

Topology:



Procedure

- Create a topology as shown as above.
- Configure the ip addresses of 2 PCs as 10.0.0.1 and 40.0.0.1 respectively and set gateways as 10.0.0.3 & 40.0.0.3
- Configure Router 0

```
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.3 255.0.0.0
Router (config-if) # no shutdown
```

```
Router (config) # ip interface serial 2/0
router (config-if) # ip address 20.0.0.1 255.0.0.0
router (config-if) # no shutdown
```

Similarly, configure ports of router 1 & router 2

- In Router 0

```
Router (config) # interface serial 2/0
Router (config-if) # encapsulation ppp
Router (config-if) # no shutdown
```

Repeat for serial 2/0 & 3/0 of router-1 and serial 2/0 port of Router 2.

- For routers 0 (serial 2/0) and router 1 (serial 3/0)

```
Router (config) # interface serial 2/0
Router (config-if) # clock rate 64000
Router (config-if) # no shutdown
router (config-if) # exit
```

- For all 3 routers, repeat step

```
Router 0
Router (config) # router rip
Router (config-router) # network 10.0.0.0
Router (config-router) # network 20.0.0.0
```

OBSERVATION

- Routes for every network can be seen in ip routes

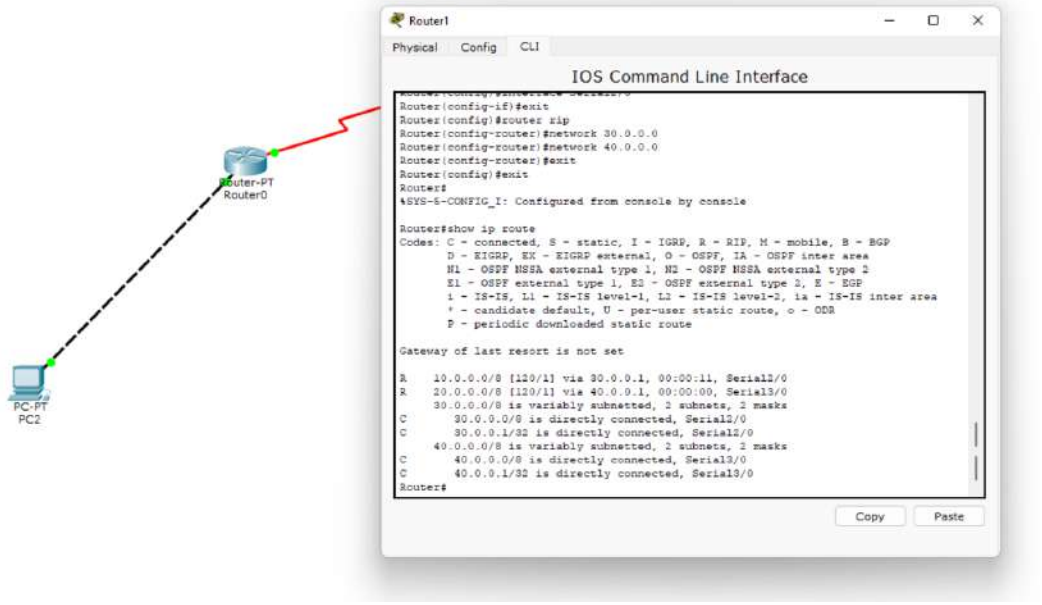
```
Routers 0
Router # show ip route
C.  10.0.0.0/8 is directly connected, Fast Ethernet 0/0
R   20.0.0.0/8 [120/2] via 30.0.0.2, serial 2/0
    30.0.0.0 /8 is variably subnetted, 2 subnets, 2 maske
C   30.0.0.0/8 is directly connected, Serial 2/0
C   30.0.0.2/32 is directly connected, Serial 2/0
R   40.0.0.0/8 [120/1] via 30.0.0.2, Serial 2/0
```

Result

```
PC 1
PC > ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data
```

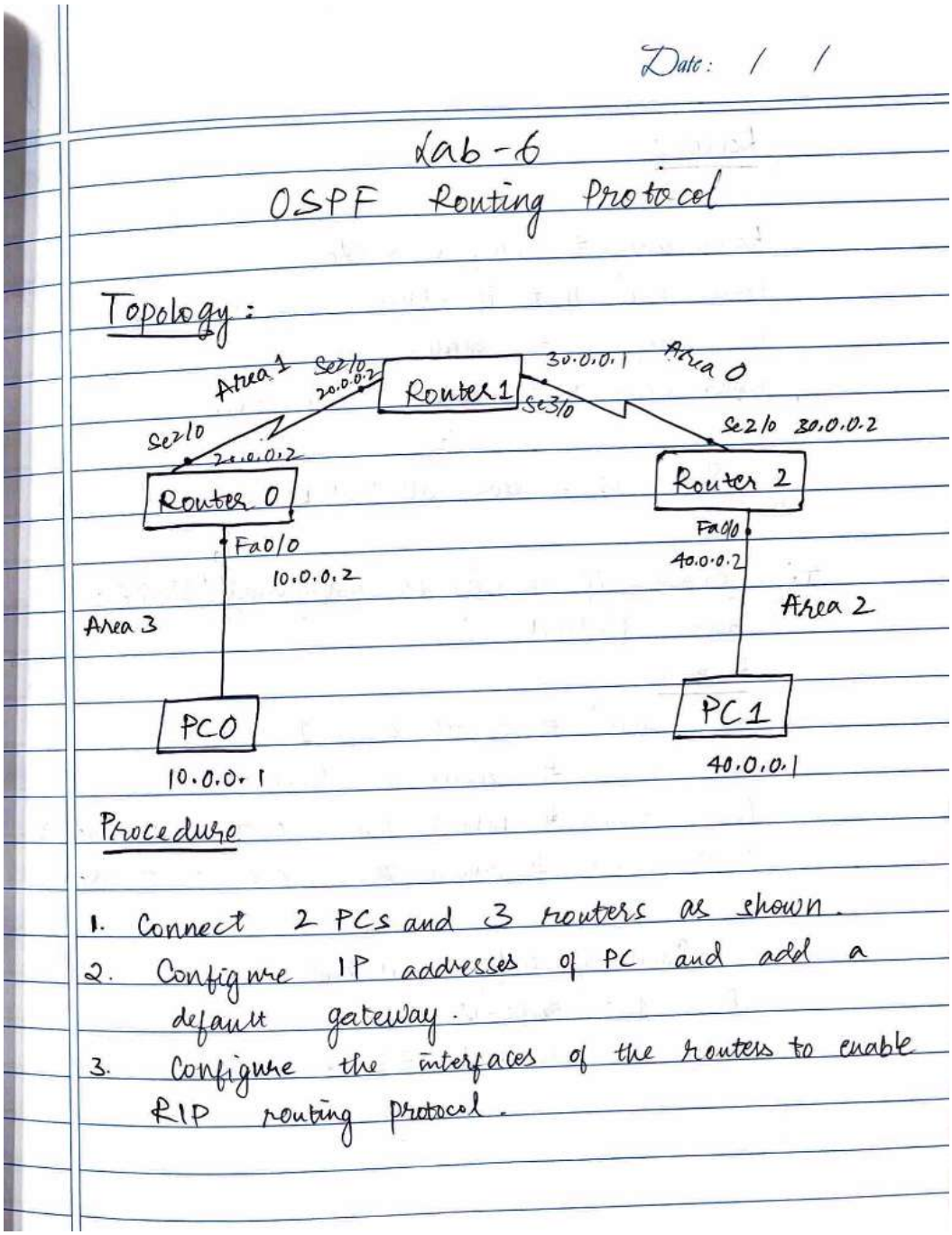
```
Reply from 40.0.0.1 : bytes=32 time=12ms TTL=
Reply from 40.0.0.1 : bytes=32 time=16ms TTL=12
Reply from 40.0.0.1 : bytes=32 time=12ms TTL=12
Reply from 40.0.0.1 : bytes=32 time=16ms TTL=12

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

OUTPUT:


Router-PT
Router0
PC-PT
PC2
Router1
Physical
Config
CLI
IOS Command Line Interface
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#exit
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 not set
R 10.0.0.0/8 [120/1] via 30.0.0.1, 00:00:11, Serial2/0
R 20.0.0.0/8 [120/1] via 40.0.0.1, 00:00:00, Serial3/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.1/32 is directly connected, Serial2/0
40.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 40.0.0.0/8 is directly connected, Serial3/0
C 40.0.0.1/32 is directly connected, Serial3/0
Router#
Copy
Paste

PROGRAM 6

Configure OSPF routing protocol



Router 0

Router-config # interface se2/0
Router-config-if # ip address 20.0.0.2 255.0.0.0
Router-config-if # encapsulation ppp
Router-config-if # clock rate 64000

Similarly configure all the interfaces.

4. Enable ip routing by configuring OSPF routing protocol

Router 0
(router-config) # router ospf 1
(router-config) # router-id 1.1.1.1
(config-router) # network 10.0.0.0 0.255.255.255 area 3
(config-router) # network 20.0.0.0 0.255.255.255 area 1

Similarly configure all routers
Router 1 : router-id 2.2.2.2
Router 2 : router-id 3.3.3.3

5. Configure loopback addresses to each router

Router 0
(router-config) # interface se2/0
(router-config-if) # interface loopback 0
(routes-config-if) # ip add 172.16.1.252 255.255.0.0
(router-config-if) # no shut

Do the same for
Router 1 : ip add 172.16.1.253
Router 2 : ip add 172.16.1.254

6. Create a virtual link between R0 and R2 to connect area 0 and area 3.

Router 0
(config)# router ospf 1
(config-router) # area 1 virtual link 2.2.2.2

Router 2
(config) # router ospf 1
(config-router) # area 1 virtual link 1.1.1.1

OBSERVATION

To connect routers between multiple networks we have to setup a loopback address and a virtual link between the routers for ospf routing to work. We have connect all the areas to the backbone area 0. Now the two end devices can be pinged.

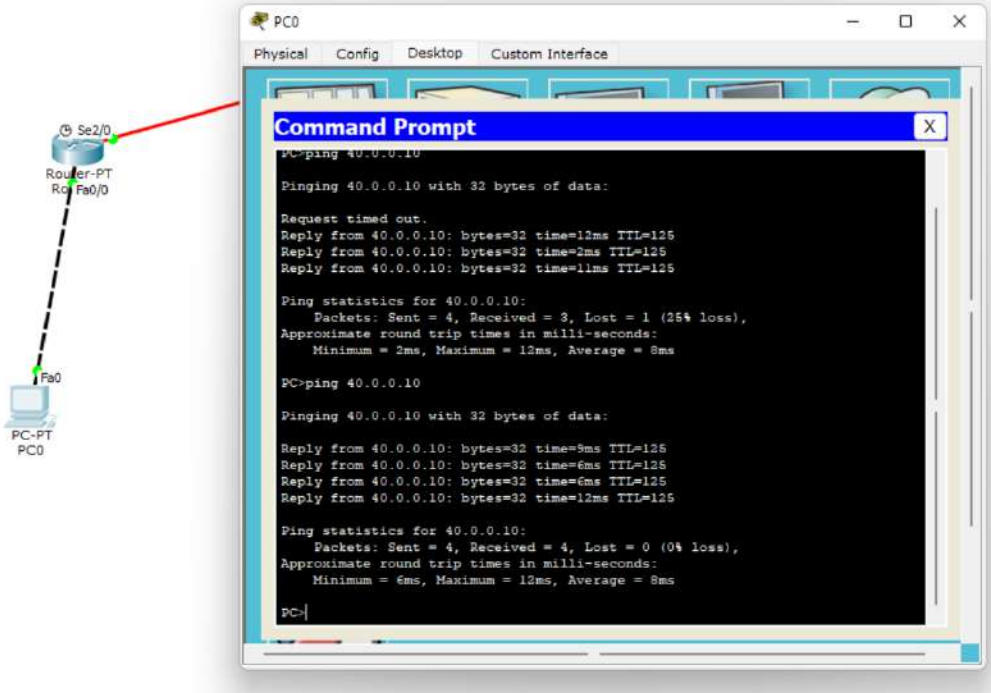
RESULT

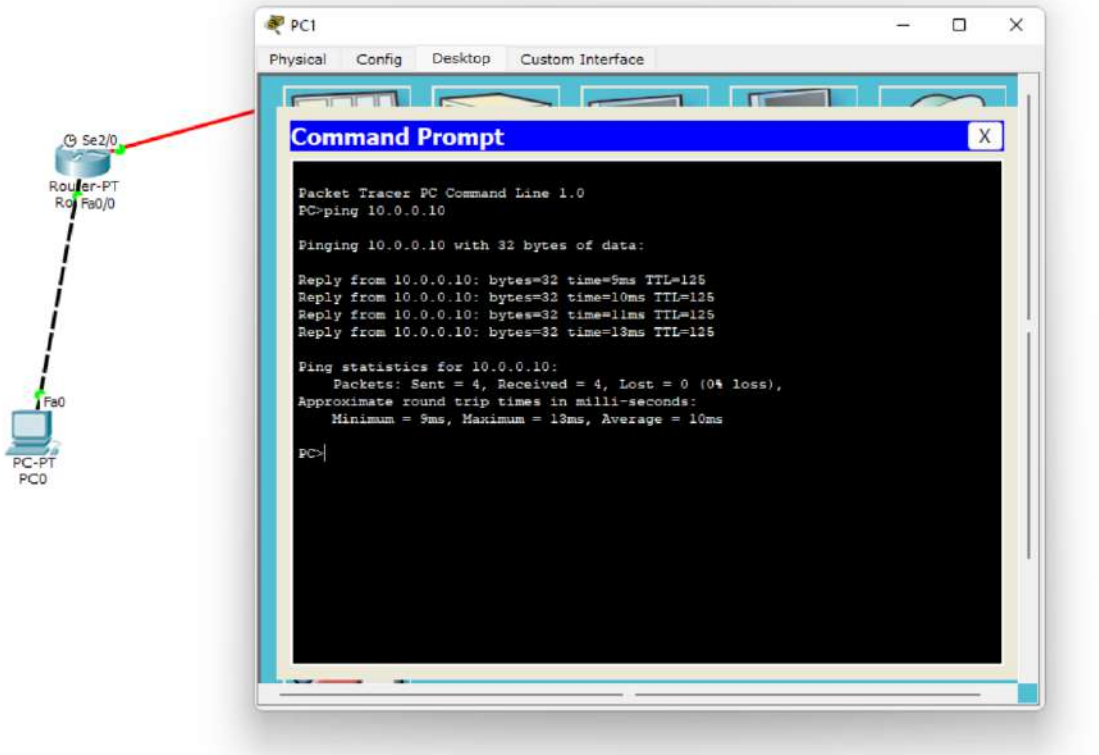
PC0

ping 40.0.0.1

Reply from 40.0.0.1 bytes = 32 time = 1ms TTL = 125
Reply from 40.0.0.1 bytes = 32 time = 1ms TTL = 125
Reply from 40.0.0.1 bytes = 32 time = TTL = 125
Reply from 40.0.0.1 bytes = 32 time = TTL = 125

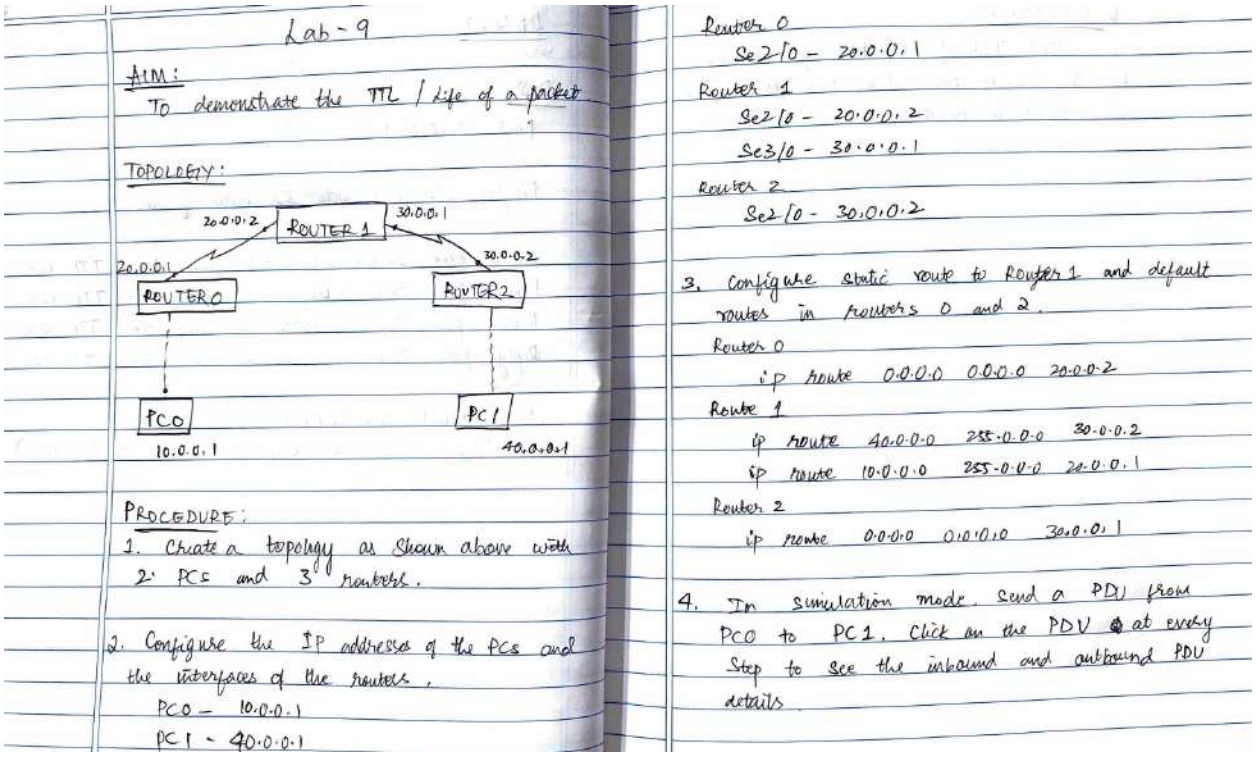
OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Se2/0
Router-PT
Fa0/0
Fa0
PC-PT
PC0
PC>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=12ms TTL=125
Reply from 40.0.0.10: bytes=32 time=2ms TTL=125
Reply from 40.0.0.10: bytes=32 time=11ms 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 = 12ms, Average = 8ms
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=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=12ms 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 = 12ms, Average = 8ms
PC>


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Se2/0
Router-PT
Fa0/0
Fa0
PC-PT
PC0
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.10
Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time=9ms TTL=125
Reply from 10.0.0.10: bytes=32 time=10ms TTL=125
Reply from 10.0.0.10: bytes=32 time=11ms TTL=125
Reply from 10.0.0.10: bytes=32 time=13ms TTL=125
Ping statistics for 10.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 9ms, Maximum = 13ms, Average = 10ms
PC>

PROGRAM 7

Demonstrate the TTL/ Life of a Packet



Lab - 9

AIM:
To demonstrate the TTL / life of a packet

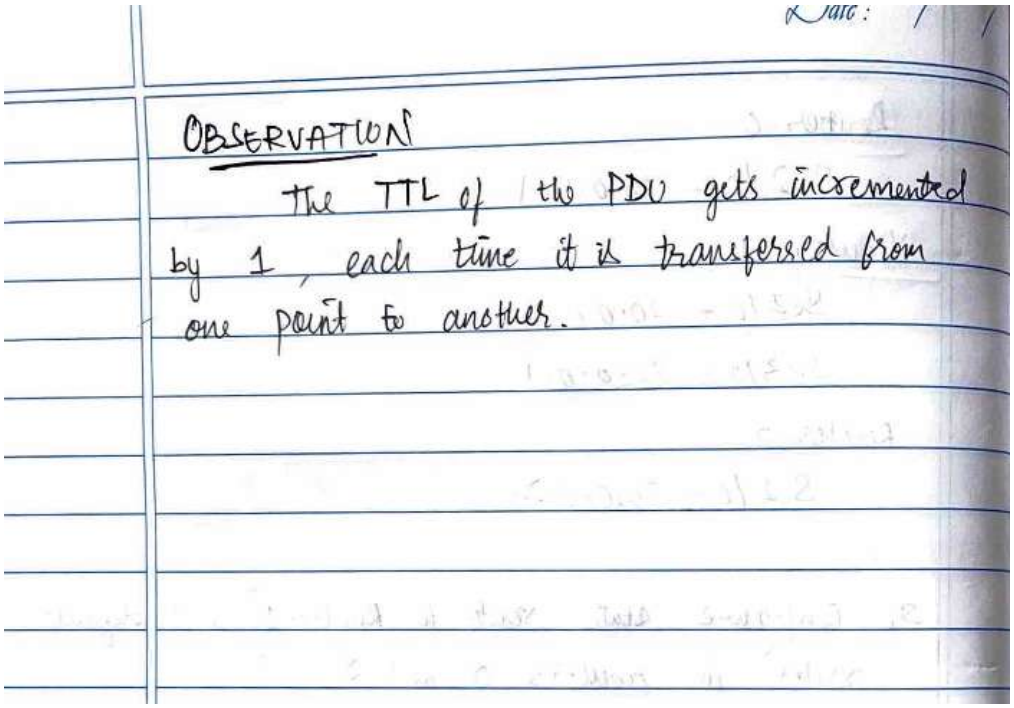
TOPOLOGY:

PROCEDURE:
1. Create a topology as shown above with 2 PCs and 3 routers.
2. Configure the IP addresses of the PCs and the interfaces of the routers.
PC0 - 10.0.0.1
PC1 - 40.0.0.1

Router 0
Se2/0 - 20.0.0.1
Router 1
Se2/0 - 20.0.0.2
Se3/0 - 30.0.0.1
Router 2
Se2/0 - 30.0.0.2

3. Configure static route to Router 1 and default routes in routers 0 and 2.
Router 0
ip route 0.0.0.0 0.0.0.0 20.0.0.2
Route 1
ip route 40.0.0.0 255.0.0.0 30.0.0.2
ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router 2
ip route 0.0.0.0 0.0.0.0 30.0.0.1

4. In simulation mode, send a PDU from PC0 to PC1. Click on the PDU at every step to see the inbound and outbound PDU details.

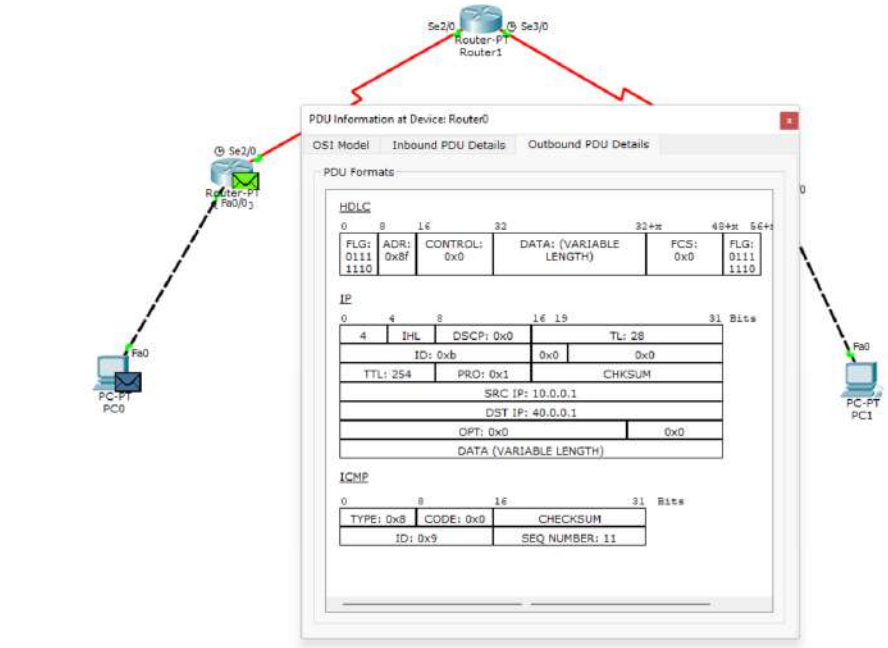


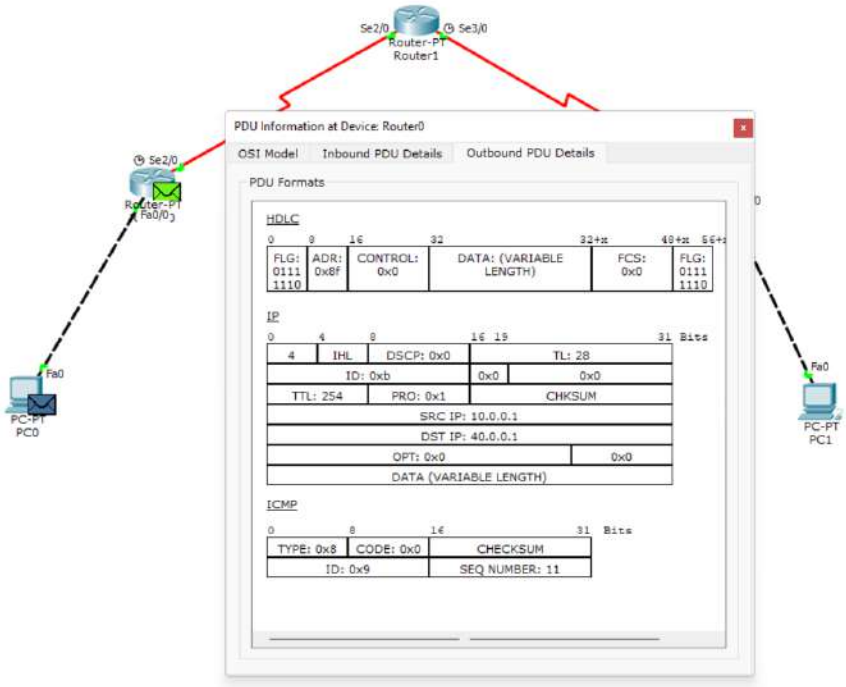
Date: / /

OBSERVATION

The TTL of the PDU gets incremented by 1, each time it is transfered from one point to another.

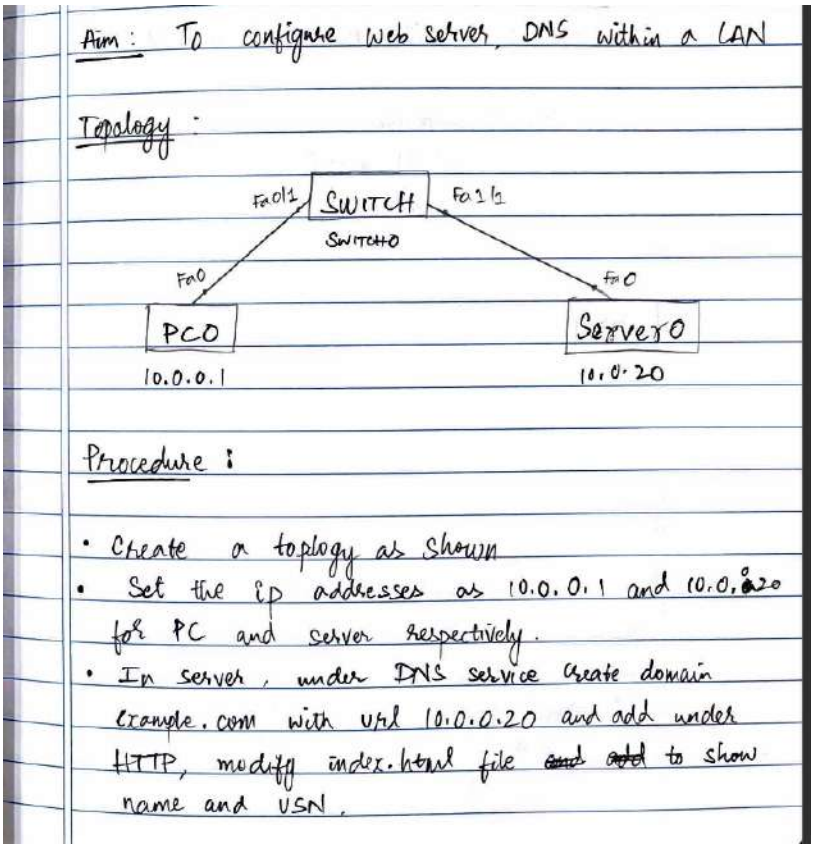
OUTPUT:


Se2/0
Se3/0
Router-PT
Router1
PDU Information at Device: Router0
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: 0xb
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x9
SEQ NUMBER: 11
Fa0
PC-PT
PC0
PC-PT
PC1


Se2/0
Se3/0
Router-PT
Router1
PDU Information at Device: Router0
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: 0xb
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x9
SEQ NUMBER: 11
Fa0
PC-PT
PC0
PC-PT
PC1

PROGRAM 8

Configure Web Server, DNS within a LAN.

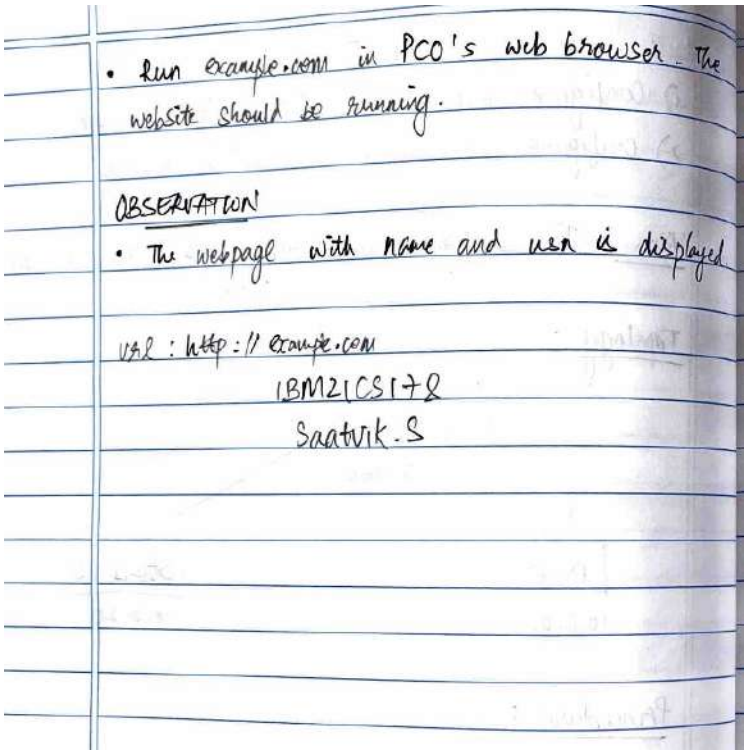


Aim: To configure web server, DNS within a LAN

Topology:

Procedure:

- Create a toplogy as shown
- Set the ip addresses as 10.0.0.1 and 10.0.0.20 for PC and server respectively.
- In server, under DNS service create domain example.com with url 10.0.0.20 and add under HTTP, modify index.html file ~~and add~~ to show name and USN.



- Run example.com in PC0's web browser. The website should be running.

OBSERVATION

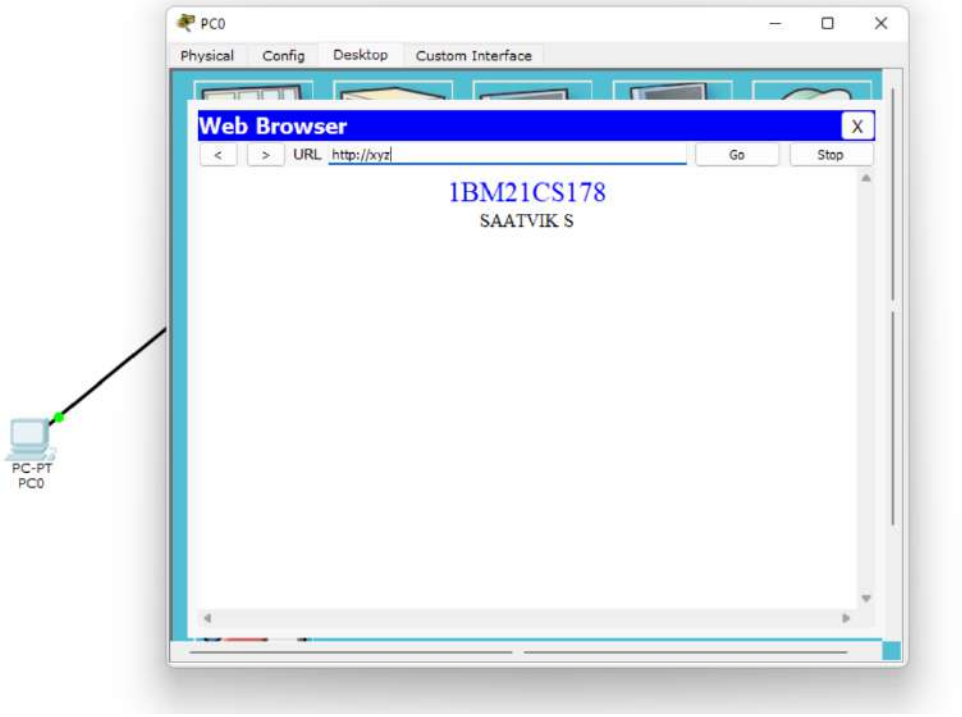
- The webpage with name and usn is displayed.

Url: http://example.com

1BM21CS178

Sraavik.S

OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL
http://xyz
Go
Stop
1BM21CS178
SAATVIK S
PC-PT
PC0

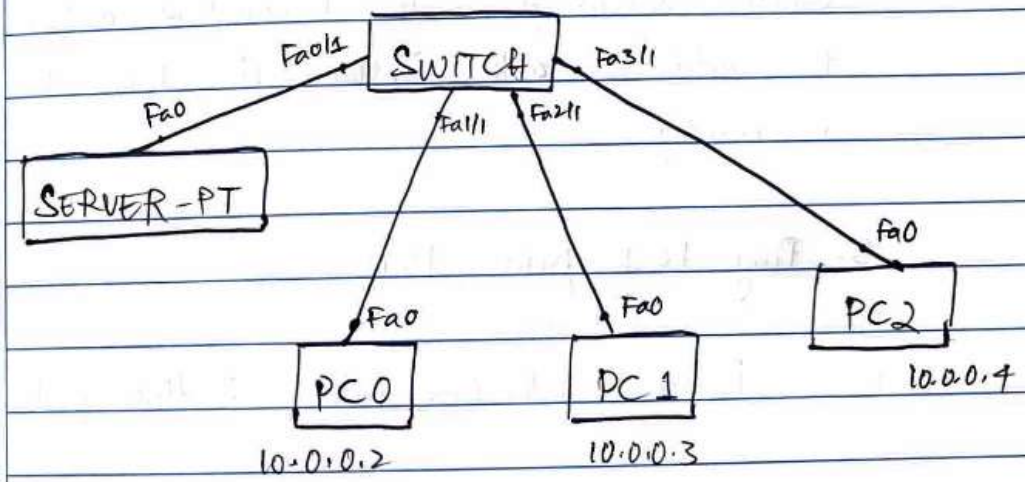
PROGRAM 9

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

AIM

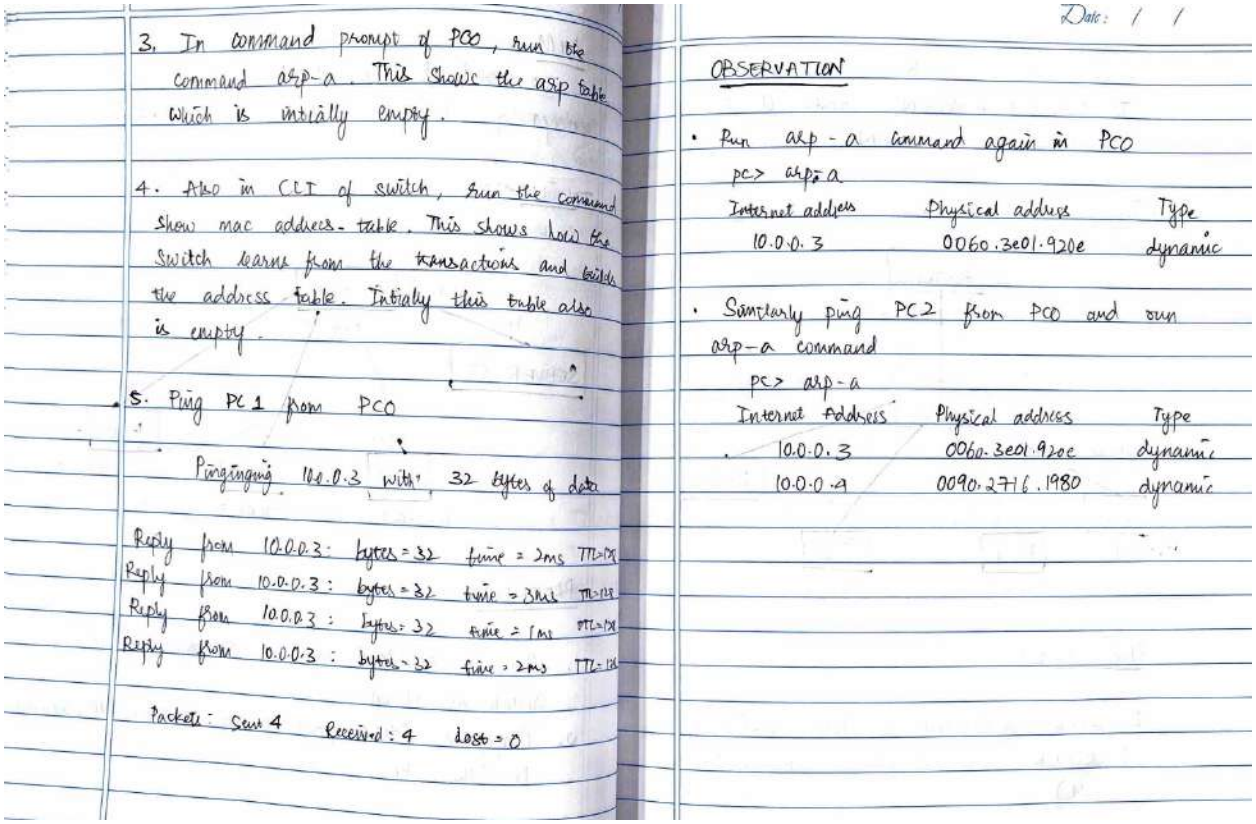
To construct simple LAN and understand the concept of Address Resolution Protocol (ARP)

TOPOLOGY:



PROCEDURE

1. Connect 3 PCs and a server using a switch as shown.
2. Configure DHCP protocol to assign IP addresses to the PCs



3. In command prompt of PC0, run the command arp-a. This shows the arp table which is initially empty.

4. Also in CLI of switch, run the command show mac address-table. This shows how the Switch learns from the transactions and builds the address table. Initialy this table also is empty.

5. Ping PC1 from PC0

Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3: bytes=32 time=2ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=2ms TTL=128

Packets: Sent 4 Received: 4 lost = 0

Date: / /

OBSERVATION

- Run arp -a command again in PC0

pc> arp -a

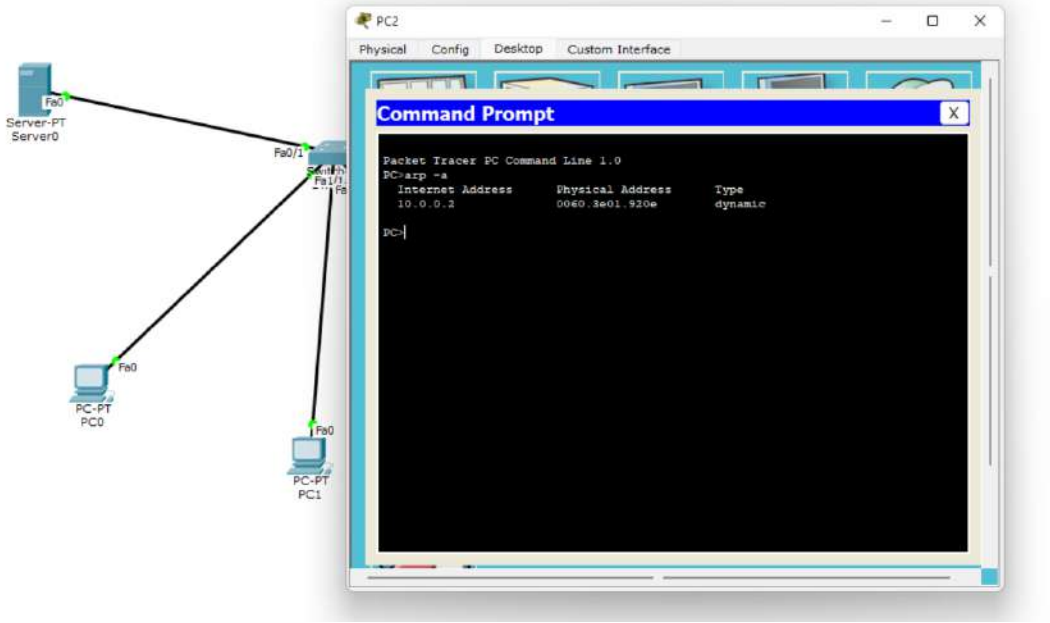
Internet address	Physical address	Type
10.0.0.3	0060.3e01.920e	dynamic

- Similarly ping PC2 from PC0 and run arp-a command

pc> arp -a

Internet address	Physical address	Type
10.0.0.3	0060.3e01.920e	dynamic
10.0.0.4	0090.2716.1980	dynamic

OUTPUT:



PROGRAM 10

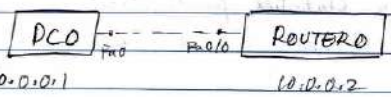
To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Lab - 11

AIM:

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

TOPOLOGY:



PROCEDURE

1. Create a topology using 1 PC and 1 router as shown.
2. Set the ip address and gateway as 10.0.0.1 and 10.0.0.2 for the PC.
3. In the router, go to CLI
   Router (config) # hostname R1
   R1 (config) # enable secret P1
   R1 (config) # interface fast ethernet 0/0
   R1 (config-if) # ip address 10.0.0.2 255.0.0.0
   R1 (config-if) # no shut.
   R1 (config-if) # line vty 05
   R1 (config-line) # login
   % login disabled on line 132, until 'password' is set.
   R1 (config-line) # password P0
   R1 (config-line) # exit.
   R1 # wr

4. PC > ping 10.0.0.2
   Pinging 10.0.0.2 with 32 bytes of data

   Reply from 10.0.0.2 bytes = 32 time = 0ms
   Reply from 10.0.0.2 bytes = 32 time = 0ms
   Reply from 10.0.0.2 bytes = 32 time = 0ms
   Reply from 10.0.0.2 bytes = 32 time = 0ms

   Ping statistics for 10.0.0.2
   Packets: Sent = 4 received = 4 lost = 0

5. Run command telnet 10.0.0.2

OBSERVATION

It can be seen that by creating a TELNET link between the PC and the router (server) we can give an authorization process such that PCs with login information are accepted.

RESULT

PC > telnet 10.0.0.2

trying 10.0.0.2 ..... open

User access verification

password : P0

R1 > enable

password : P1

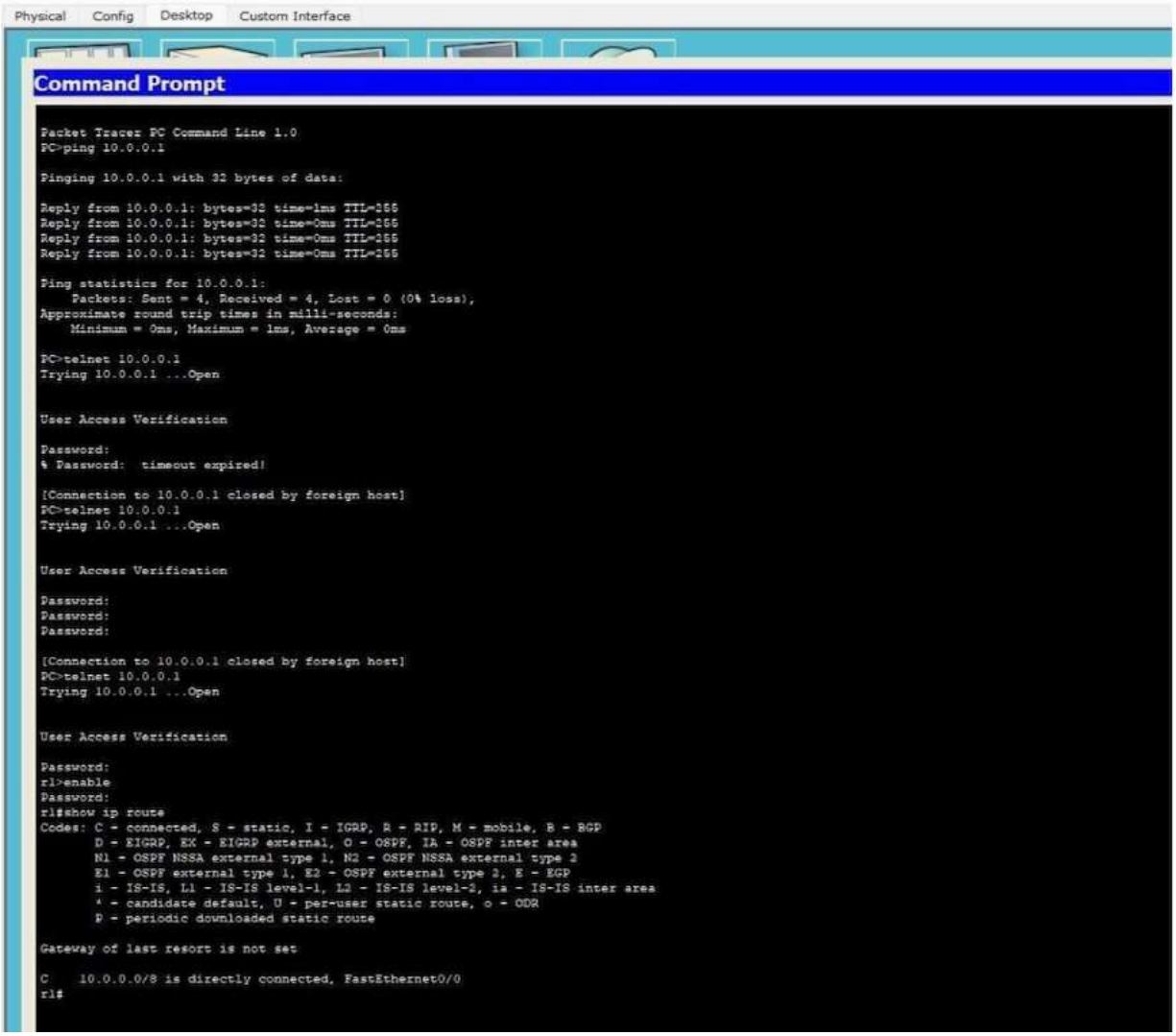
R1 # show ip route

nodes, C - connected

C 10.0.0.0/8 is directly connected

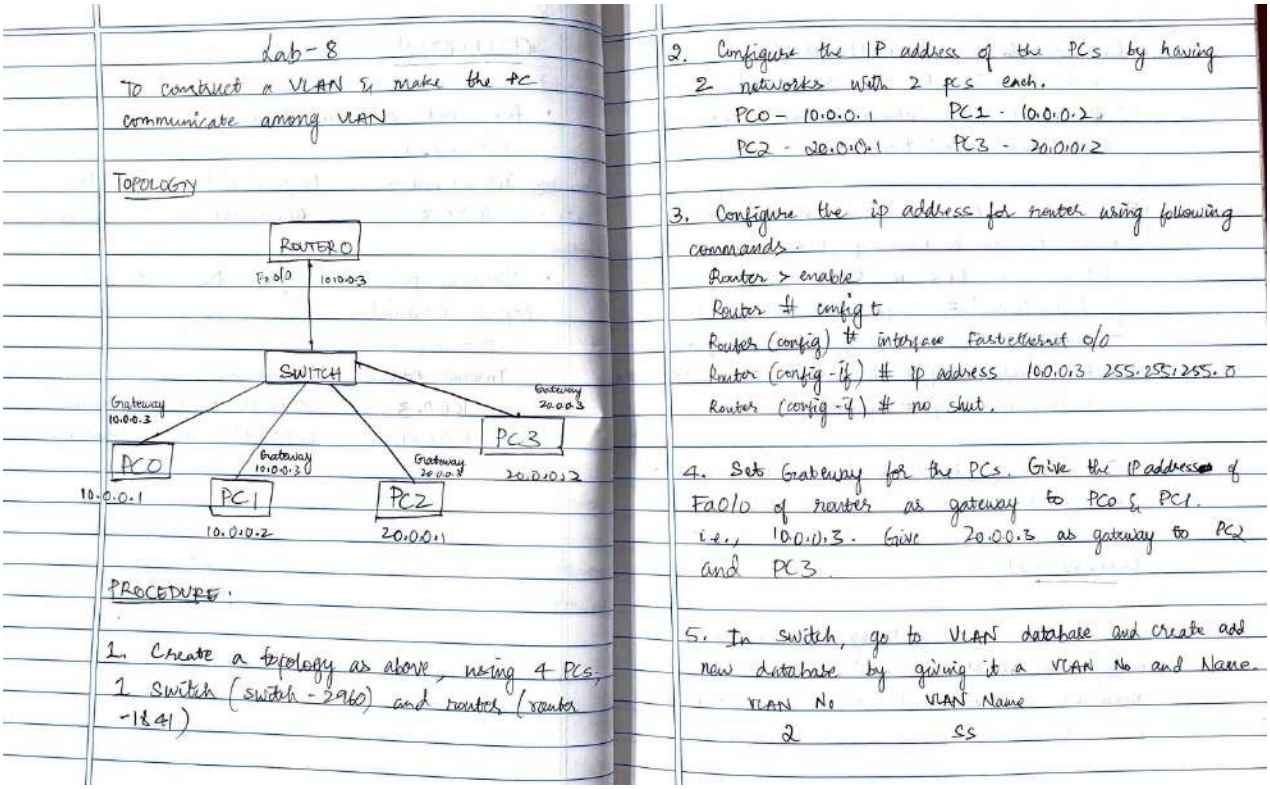
Thus using TELNET protocol we can access the router from the PC (directly connected)

OUTPUT:


Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
% Password: timeout expired!
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
Password:
Password:
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
r1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
r1#

PROGRAM 11

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



Lab - 8

To construct a VLAN & make the PC communicate among VLAN

TOPOLOGY

PROCEDURE:

1. Create a topology as above, using 4 PCs, 1 switch (switch - 2960) and router (router - 1841)

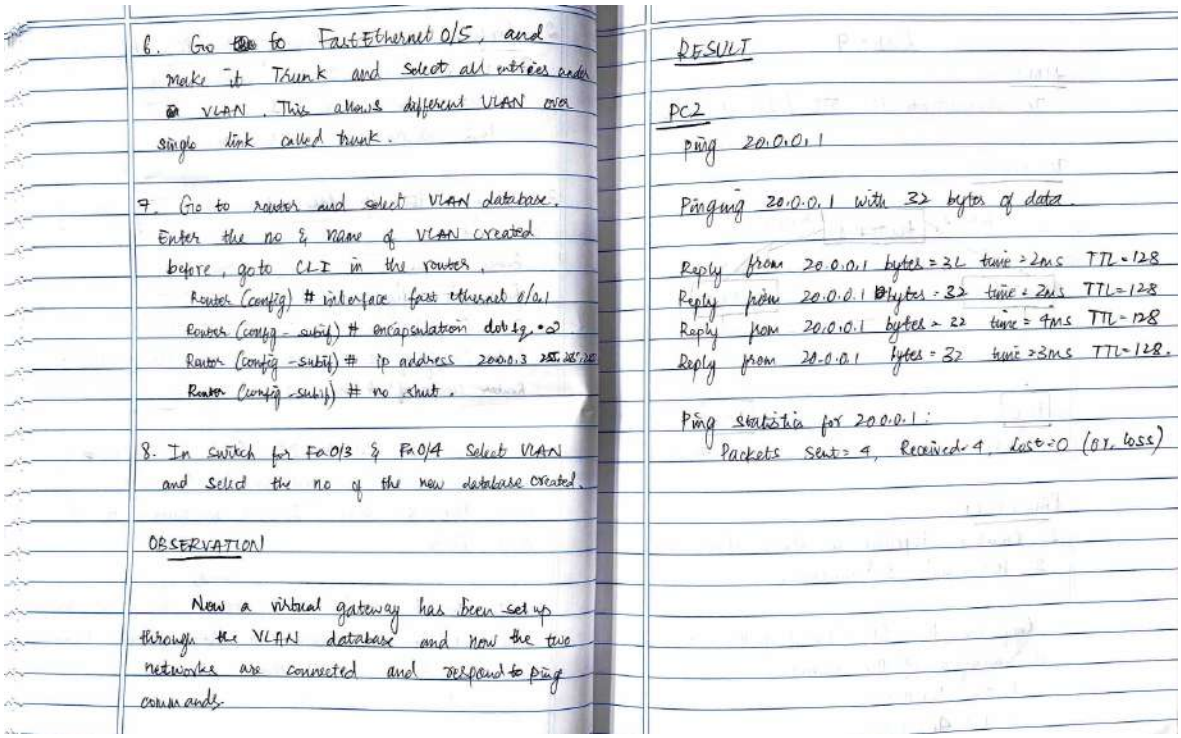
2. Configure the IP address of the PCs by having 2 networks with 2 pcs each.
   PC0 - 10.0.0.1 PC1 - 10.0.0.2
   PC2 - 20.0.0.1 PC3 - 20.0.0.2

3. Configure the ip address for router using following commands.
   Router > enable
   Router # config t
   Router (config) # interface FastEthernet 0/0
   Router (config-if) # ip address 10.0.0.3 255.255.255.0
   Router (config-if) # no shut.

4. Set Gateway for the PCs. Give the IP address of Fa0/0 of router as gateway to PC0 & PC1. i.e., 10.0.0.3. Give 20.0.0.3 as gateway to PC2 and PC3.

5. In switch, go to VLAN database and create add new database by giving it a VLAN No and Name.

VLAN No	VLAN Name
2	ss



6. Go to FastEthernet 0/5, and make it Trunk and select all entries under VLAN. This allows different VLAN via single link called trunk.

7. Go to router and select VLAN database. Enter the no & name of VLAN created before, go to CLI in the router.
   Router (config) # interface fast ethernet 0/0.1
   Router (config-subif) # encapsulation dot1q 2
   Router (config-subif) # ip address 20.0.0.3 255.255.255.0
   Router (config-subif) # no shut.

8. In switch for Fa0/3 & Fa0/4 Select VLAN and select the no of the new database created.

OBSERVATION

Now a virtual gateway has been set up through the VLAN database and now the two networks are connected and respond to ping commands.

RESULT

PC2

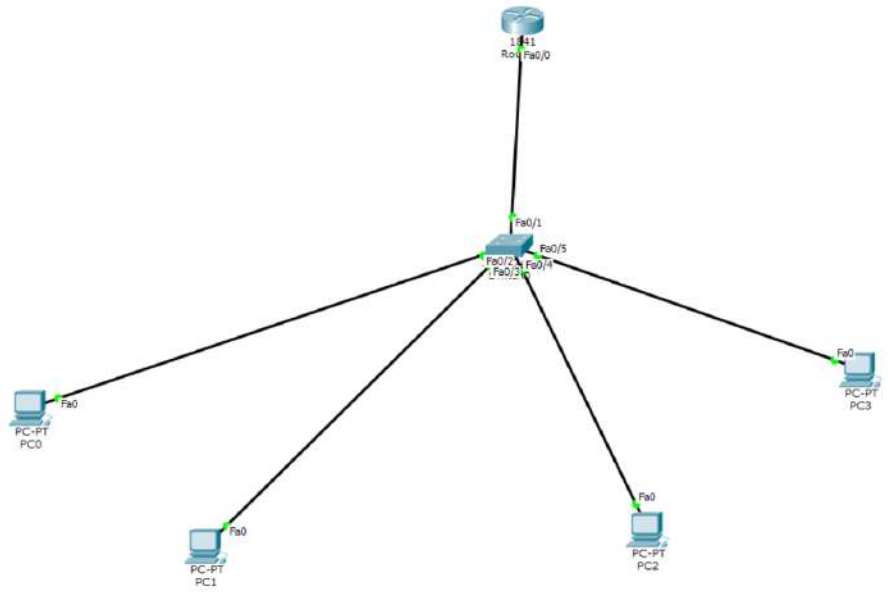
ping 20.0.0.1

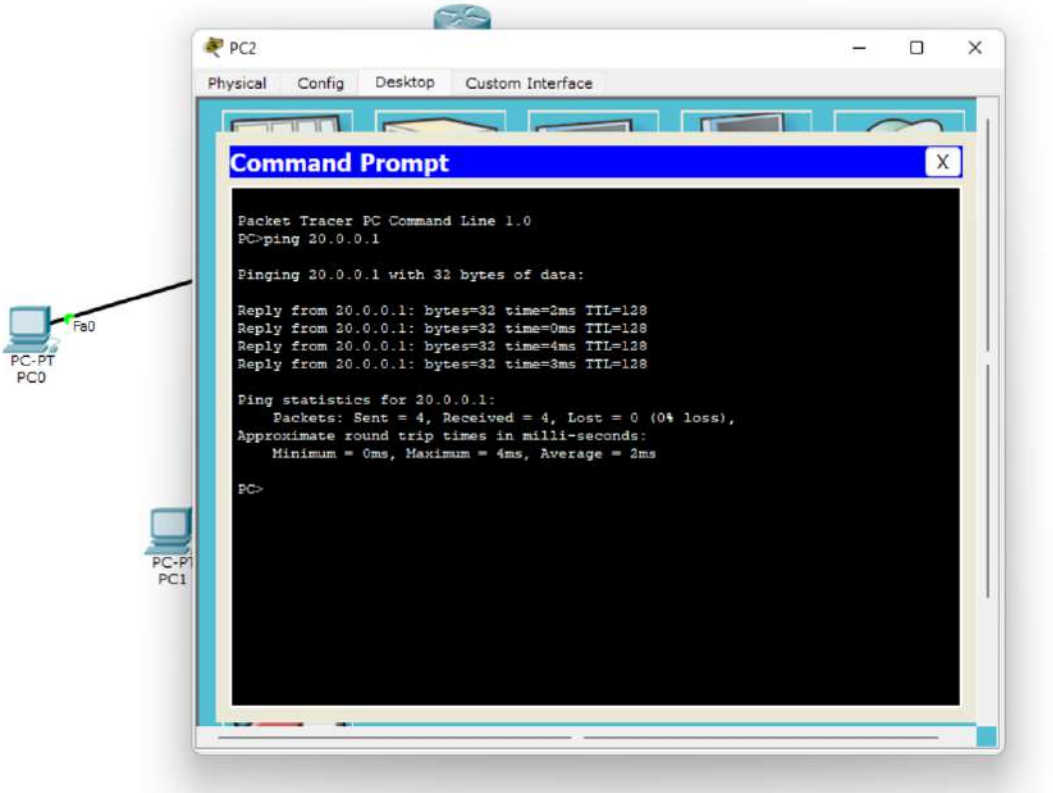
Pinging 20.0.0.1 with 32 bytes of data.

Reply from 20.0.0.1 bytes = 32 time = 2ms TTL = 128
Reply from 20.0.0.1 bytes = 32 time = 2ms TTL = 128
Reply from 20.0.0.1 bytes = 32 time = 4ms TTL = 128
Reply from 20.0.0.1 bytes = 32 time = 3ms TTL = 128.

Ping statistics for 20.0.0.1:
Packets Sent = 4, Received = 4, Lost = 0 (0% loss)

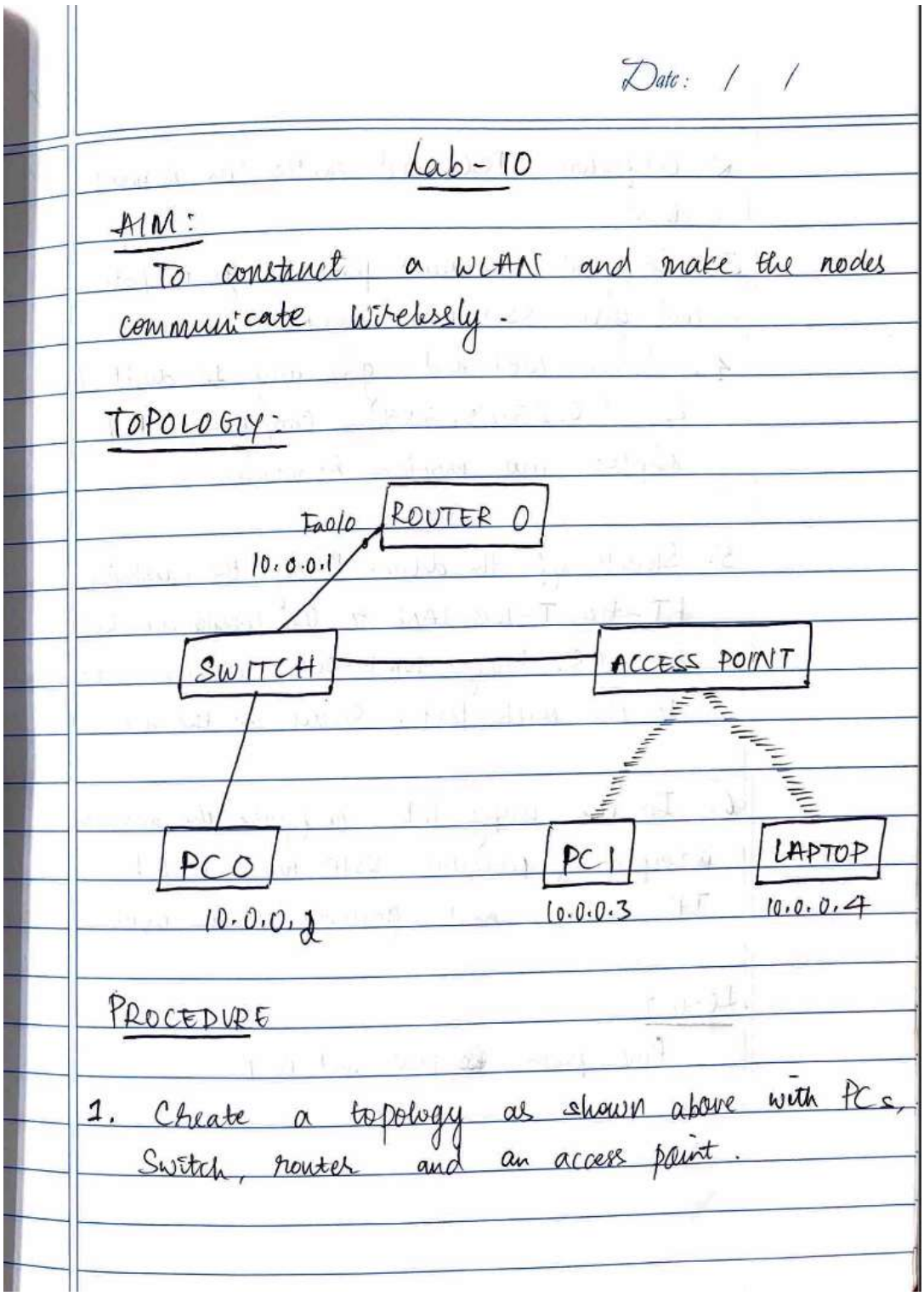
OUTPUT:


1841
Fa0/0
Fa0/1
Fa0/5
Fa0/4
Fa0/3
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2
Fa0
PC-PT
PC3


PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=2ms TTL=128
Reply from 20.0.0.1: bytes=32 time=0ms TTL=128
Reply from 20.0.0.1: bytes=32 time=4ms TTL=128
Reply from 20.0.0.1: bytes=32 time=3ms TTL=128
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 2ms
PC>
Fa0
PC-PT
PC0
PC-PT
PC1

PROGRAM 12

To construct a WLAN and make the nodes communicate wirelessly



Date: / /

2. Configure PC0 and Router as normally done.
3. Configure the access point 1, go to port 1 and give SSID name (any name)
4. Select WEP and give any 10 digit hex key (0123456789). Configure PC1 and laptop with wireless standards.
5. Switch off the device. Drag the existing PT-HOST-NM-1AN to the component listed in this. Drag WMP300N wireless interface to the empty port. Switch on the device.
6. In the config tab, configure the wireless interface. Configure SSID, WEP, WEP key, IP address and gateway to the device.

RESULT

Ping from PC0 and PC1

Date: / /

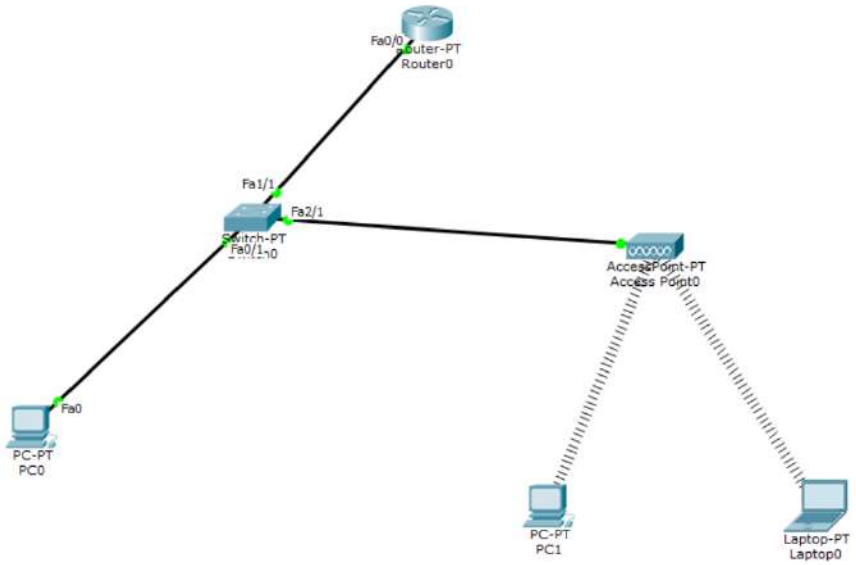
PC > ping 10.0.0.3

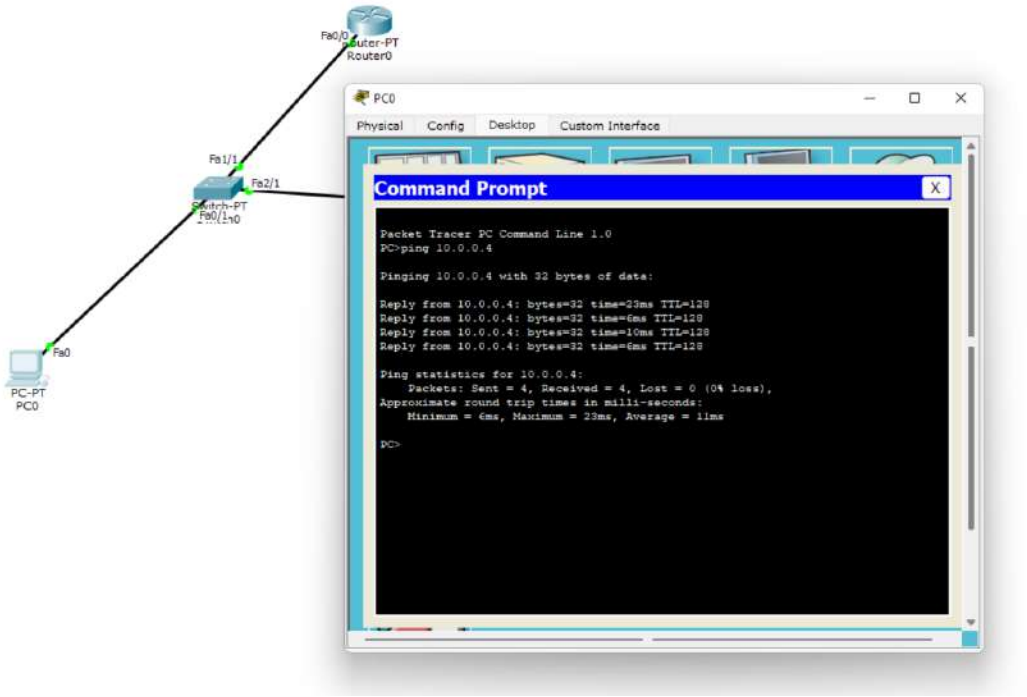
Pinging 10.0.0.3 with 32 bytes of data.

Reply from 10.0.0.3 bytes = 32 time = 23ms TTL = 120
Reply from 10.0.0.3 bytes = 32 time = 6ms TTL = 120
Reply from 10.0.0.3 bytes = 32 time = 10ms TTL = 120
Reply from 10.0.0.3 bytes = 32 time = 6ms TTL = 120

Ping statistics for 10.0.0.3:
Packets sent = 4, received = 4, lost = 0

OUTPUT:




Fa0/0
Router-PT
Router0
Fa1/1
Fa2/1
Switch-PT
Fa0/1
Fa0
PC-PT
PC0
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=23ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Reply from 10.0.0.4: bytes=32 time=10ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms 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 = 6ms, Maximum = 23ms, Average = 11ms
PC>

CYCLE 2

PROGRAM 13

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

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

#define N strlen(gen_poly)

char data[28];
char check_value[28];
char gen_poly[10];
int data_length, i, j;

void XOR() {
for (j = 1; j < N; j++)
check_value[j] = ((check_value[j] == gen_poly[j]) ? '0' : '1');
}
void crccrc() {
for (i = 0; i < N; i++)
check_value[i] = data[i];
i = 0; // Initialize i
do {
if (check_value[0] == '1')
XOR();
for (j = 0; j < N - 1; j++)
check_value[j] = check_value[j + 1];
check_value[j] = data[i++];
} while (i <= data_length + N - 1);
}

void receiver() {
printf("Enter the received data: ");
scanf("%s", data);
printf("\n-----------------------------\n");
printf("Data received: %s\n", data);
crccrc();
for (i = 0; (i < N - 1) && (check_value[i] != '1'); i++);
if (i < N - 1)
printf("Error detected\n\n");
else
printf("No error detected\n\n");
```

```
}

int main() {
printf("Enter data to be transmitted: ");
scanf("%s", data);
printf("\nEnter the Generating polynomial: ");
scanf("%s", gen_poly);
data_length = strlen(data);
for (i = data_length; i < data_length + N - 1; i++)
data[i] = '0';
printf("\n----------------------------------------");
printf("\nData padded with n-1 zeros : %s\n", data);
printf("\n----------------------------------------");
crccrc();
printf("\nCRC or Check value is : %s\n", check_value);
for (i = data_length; i < data_length + N - 1; i++)
data[i] = check_value[i - data_length];
printf("\n----------------------------------------");
printf("\nFinal data to be sent : %s\n", data);
printf("\n----------------------------------------\n");
receiver();
return 0;
}
```

PROGRAM 14

Write a program for congestion control using Leaky bucket algorithm.

```
#include <stdio.h>
#include <stdlib.h>
struct packet
{
int time;
int size;
} p[50];

int main()
{
int i, n, m, k = 0;
int bsize, bfilled, outrate;
printf("Enter the number of packets:");
scanf("%d", &n);
printf("Enter packets in the order of their arrival time\n");
for (i = 0; i < n; i++)
{
printf("Enter the time and size:");
scanf("%d%d", &p[i].time, &p[i].size);
}
printf("Enter the bucket size:");
scanf("%d", &bsize);
printf("Enter the output rate:");
scanf("%d", &outrate);

m = p[n - 1].time;
i = 1;
k = 0;
bfilled = 0;
while (i <= m || bfilled != 0)
{
printf("\n\nAt time %d", i);

if (p[k].time == i)
{
if (bsize >= bfilled + p[k].size)
{
bfilled = bfilled + p[k].size;
printf("\n%dbyte packet is inserted", p[k].size);
```

```
k = k + 1;
}
else
{
printf("\n%dbyte packet is discarded", p[k].size);
k = k + 1;
}
}

if (bfilled == 0)
{
printf("\nNo packets to transmitte");
}
else if (bfilled >= outrate)
{
bfilled = bfilled - outrate;
printf("\n%dbytes transfered", outrate);
}

else
{
printf("\n%dbytes transfered", bfilled);
bfilled = 0;
}
printf("\nPackets in the bucket %d byte", bfilled);
i++;
}
return 0;
}
```

PROGRAM 15

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.

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)
while 1:
print ("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,"r")
l=file.read(1024)
connectionSocket.send(l.encode())
print ('\nSent contents of ' + sentence)
file.close()
connectionSocket.close()
```

CLIENT.PY

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

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

PROGRAM 16

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.

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()



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

sentence = input("\nEnter file name: ")

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

filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
print (filecontents.decode("utf-8"))
# for i in filecontents:
# print(str(i), end = '')
clientSocket.close()
clientSocket.close()
```