VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Shashank C (1BM22CS254)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Academic Year 2024-25 (odd)

Scanned with OKEN Scanner

B.M.S. College of Engineering

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Computer Network (23CS5PCCON)" carried out by **Shashank C (1BM22CS254),** who is Bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Prof. Rekha G S Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Scanned with OKEN Scanner

Index-Cycle-I

Sl. No.	Date	Experiment Title	Page No.
1	04/10/2024	Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.	1
2	18/10/2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	5
3	25/10/2024	Configure default route, static route to the router	9
4	08/11/2024	Configure DHCP within a LAN and outside LAN.	16
5	22/11/2024	Configure RIP routing Protocol in Routers	21
6	22/11/2024	Configure OSPF routing protocol	24
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	31
8	08/11/2024	Configure Web Server, DNS within a LAN.	34
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	38
10	20/12/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	41
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	44
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	47

Github Link:https://github.com/ShashankCS254/CN-LAB.git

Scanned with OKEN Scanner

Index-Cycle-II

Sl. No.	Date	Experiment Title	Page No.
1	15/11/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	50
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	53
3	20/12/2024	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.	55
4	20/12/2024	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.	58

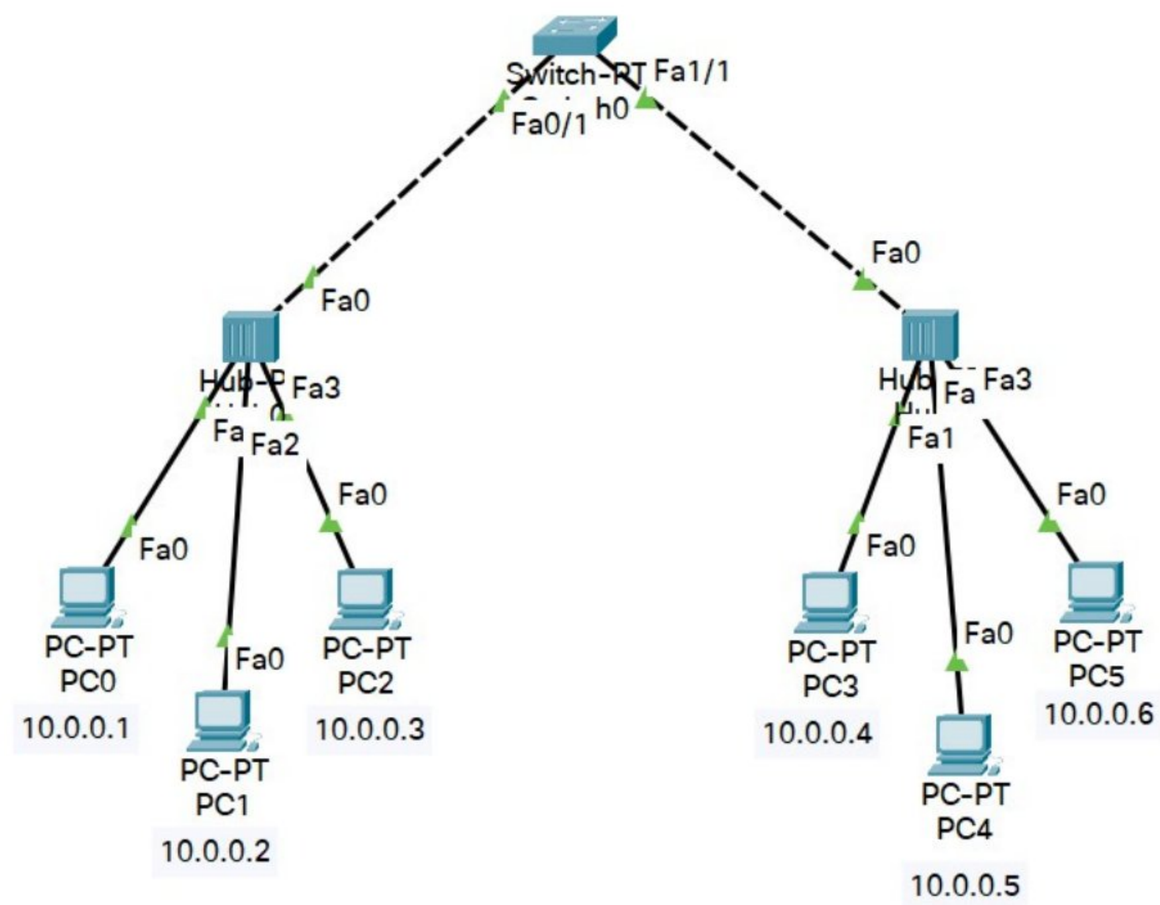
Scanned with OKEN Scanner

Cycle-I

Program 1

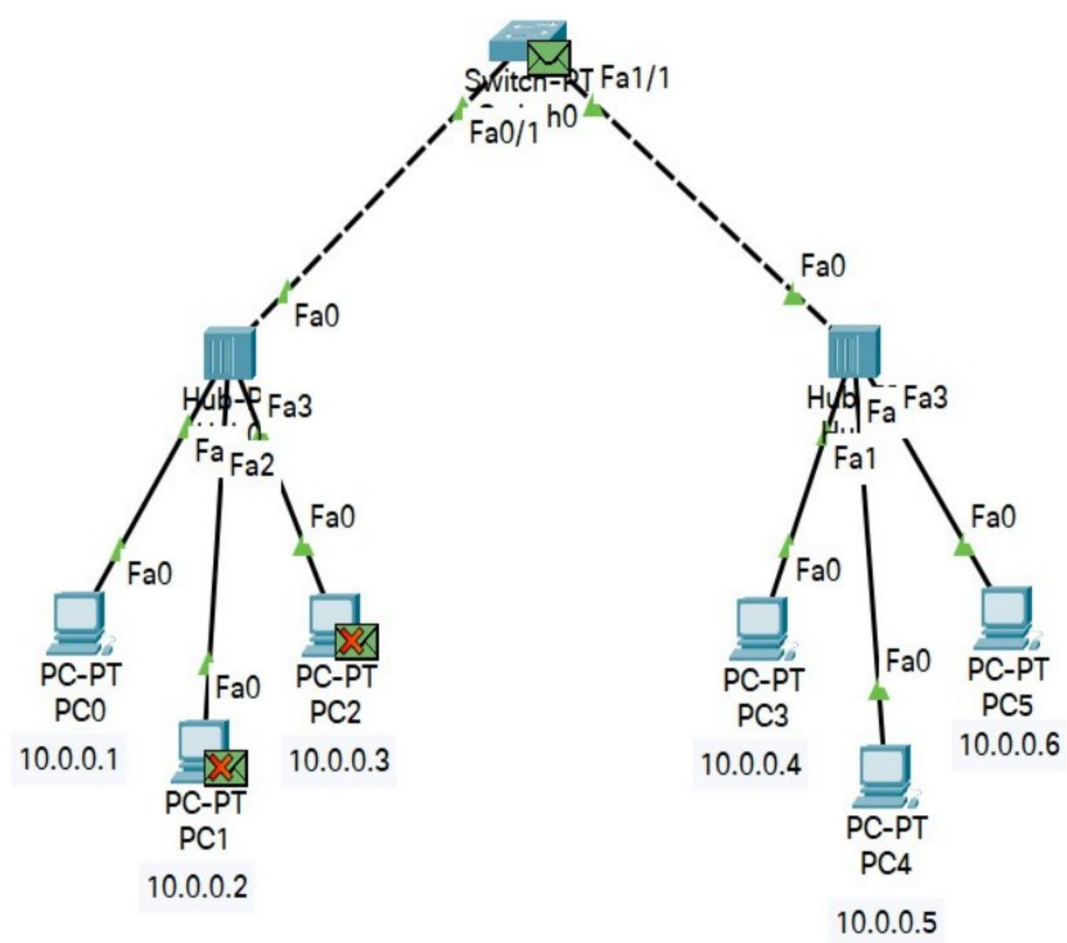
i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.

ii. Procedure along with the topology



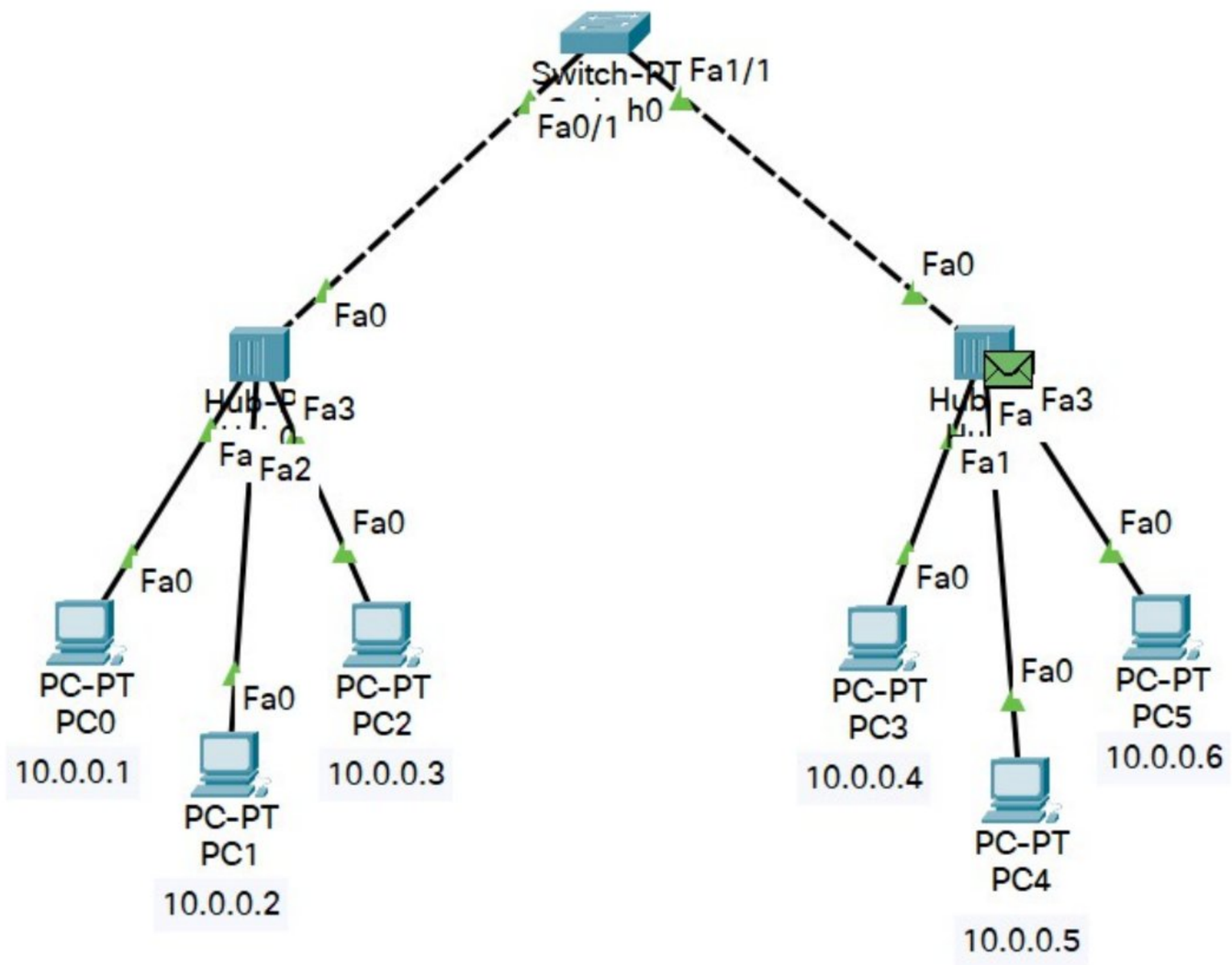
iii. Screen shots/ output

Hub behaviour at sending end

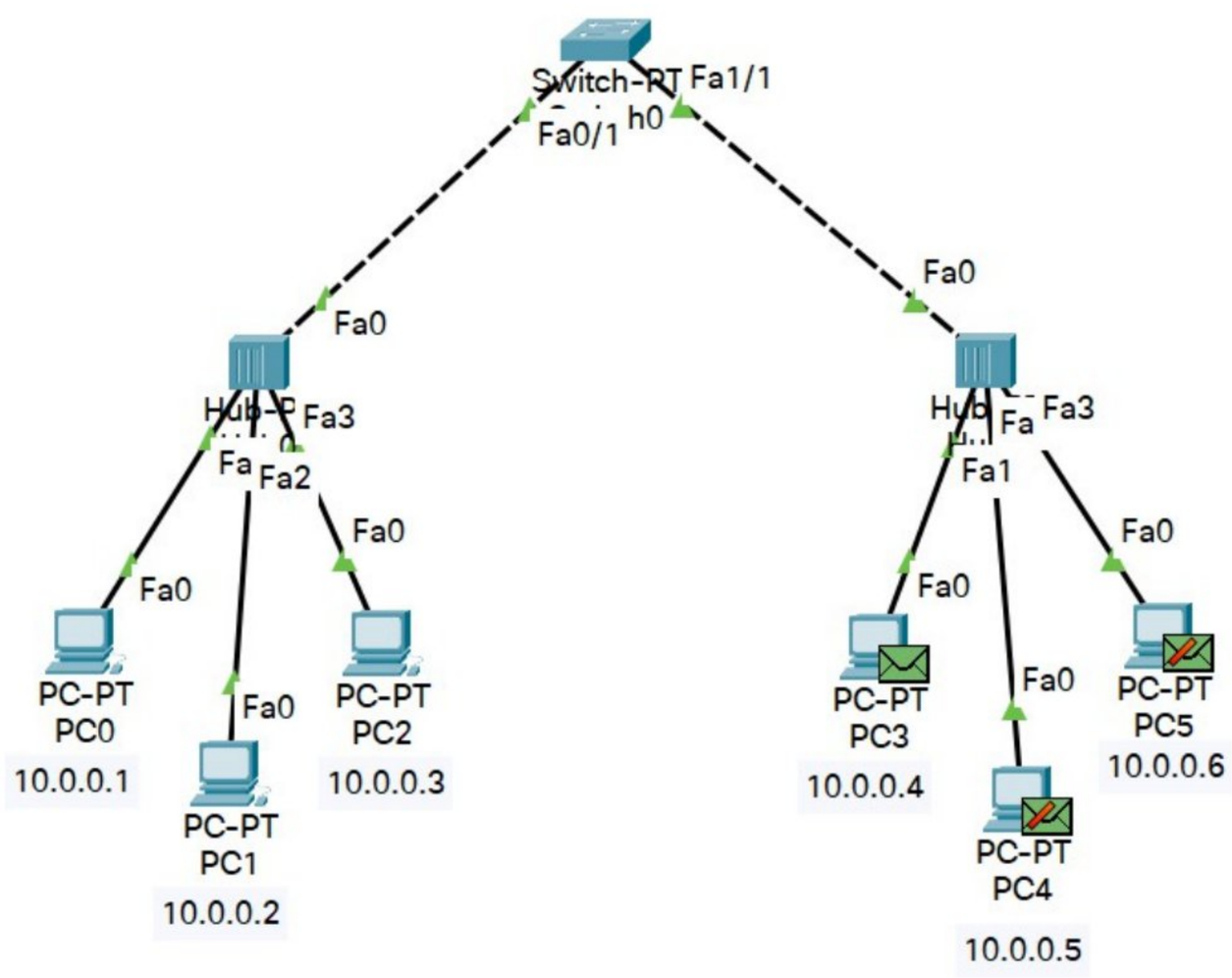


Switch behaviour

Scanned with OKEN Scanner

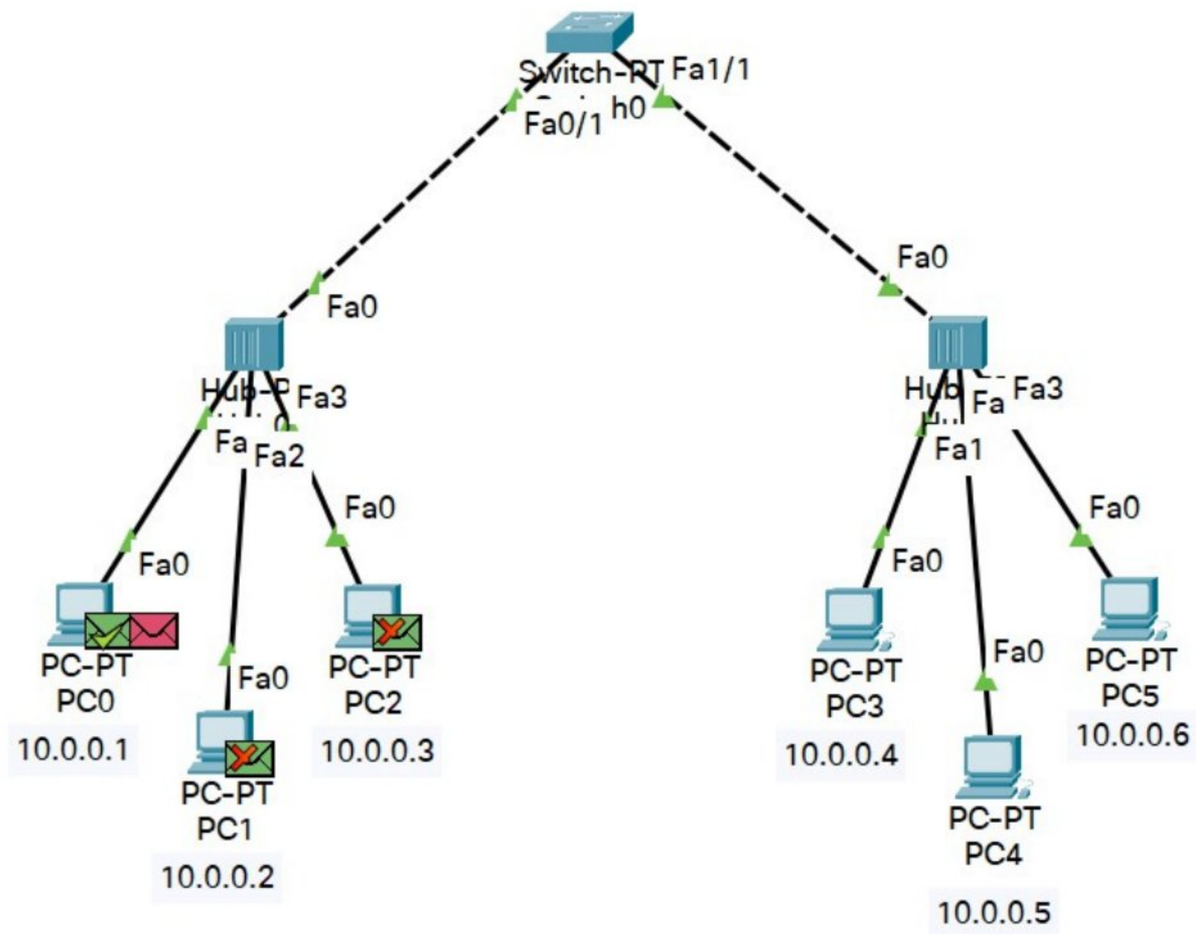


Hub behaviour at receiving end

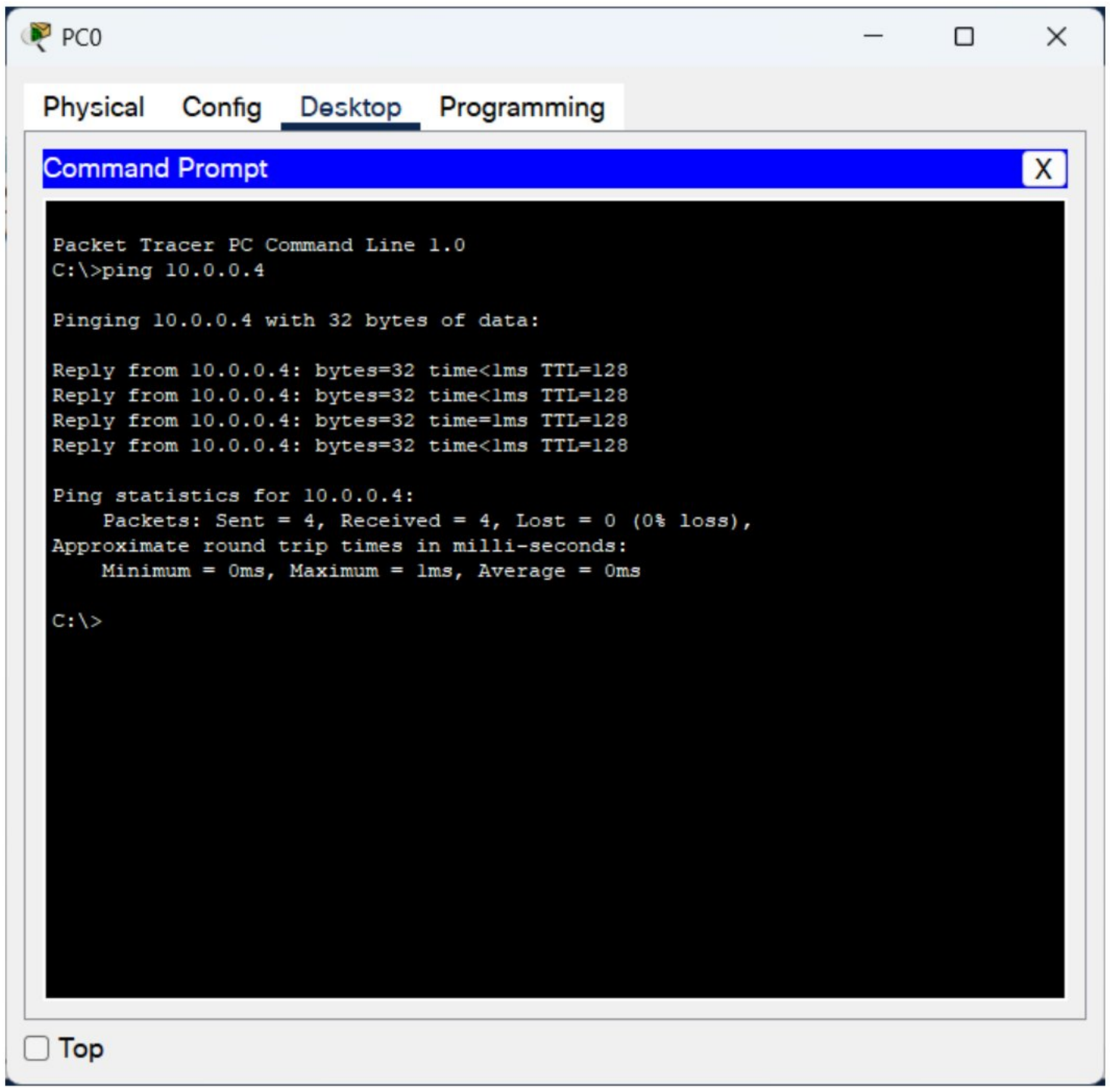


Scanned with OKEN Scanner

Hub behaviour when back to sender



Ping command to connectivity



Scanned with OKEN Scanner

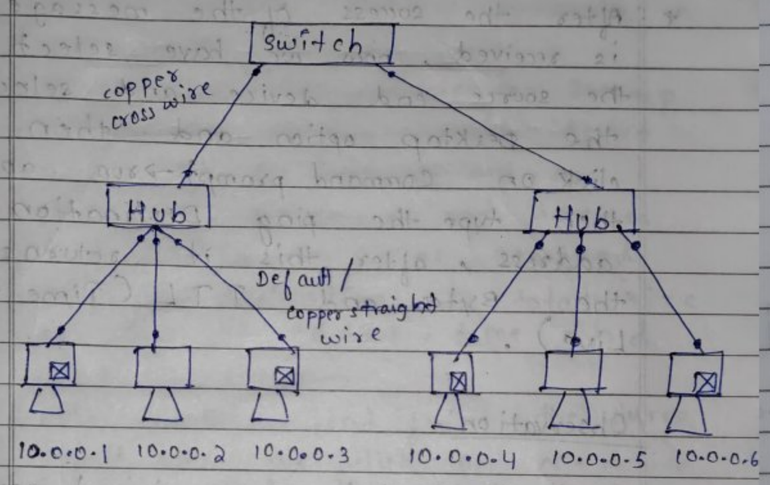
iv. Observation

4/10/24

DATE: PAGE:

LAB - 2

Create a topology involving multiple Hubs and a switch connecting them to simulate a simple PDU



Implementation:-

* At the first the 6 PC's or end-devices of PC-PT PC0 to PC-PT PC5 are placed.
* The first three end devices are connected to first Hub and the next three end devices are connected to second Hub. The Hubs are connected to the switch.
* The connection between the Hub and end device is done using the copper straight wire and the connection between Hub and switch is done using copper cross wire.
* After all the connections it represents the star topology.

DATE: PAGE:

* A message that is, a simple PDU is dropped from the PC-PT PC0 (source) to PC-PT PC2 (destination) and the same procedure is followed for PC-PT PC3 to PC-PT PC5.
* After the success of the message is received, now we have select the source end device and select the Desktop option and then click on Command prompt → run and then type the ping Destination address, after this it returns the Bytes and TTL (Time to Live).

Observation:-

* It is found that green signal appears at the bottom end is seen which indicates the successful connection.
* The status was successful and type of layer was ICMP for both the messages which was sent and the messages are received by the destination end devices.
* The hub broadcast the data to all the connected devices in the network and switch identify the MAC or the data header to transmit the data only for the device that requested it.
* So here the switch checks whether the message is sent between two hubs

DATE: PAGE:

or not and if sent it sends the message to a green destination hub-end device and else it becomes incompatible and does not send the message from the hubs from which it received and does its operation in the network.
* The command prompt in the Desktop option in the source end device where the 4 data packets are sent from the source end devices and 3 or 4 replies as received from the destination and total size of the ip address, time and TTL is shown.
* The message sent from the source and when we click on N-device, we can see the different layers of the network.

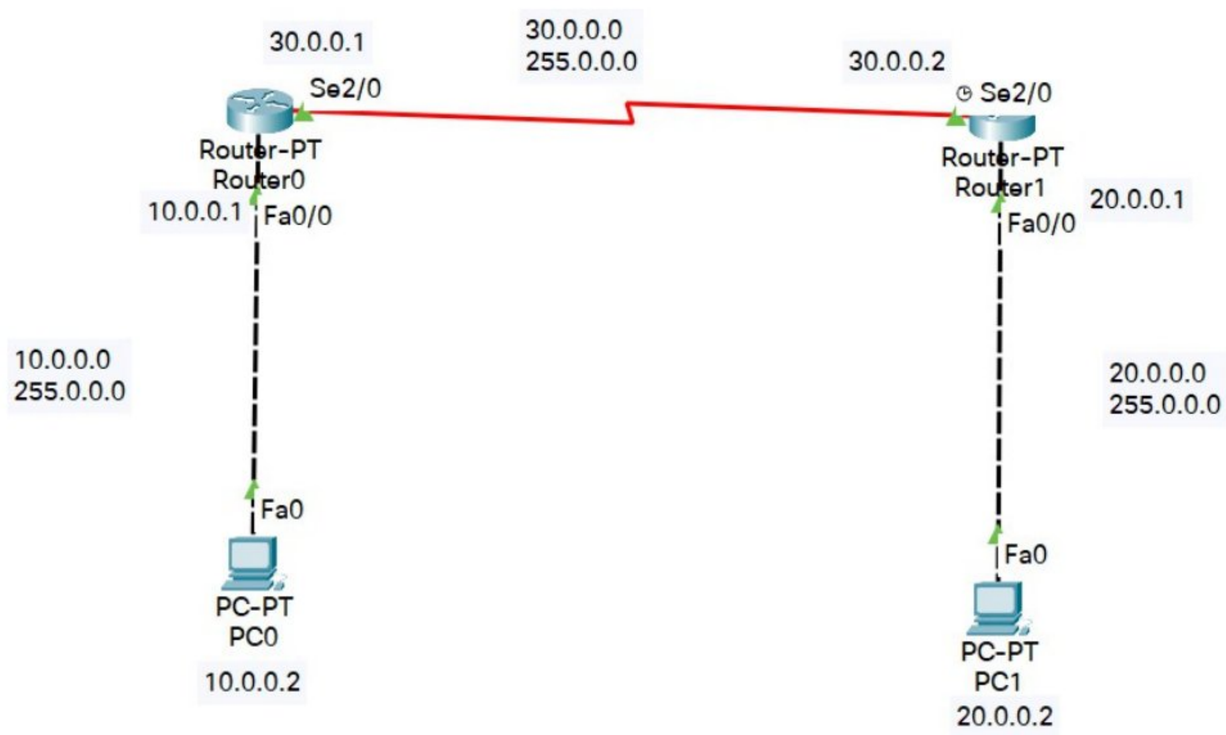
4/10/24

Scanned with OKEN Scanner

Program 2

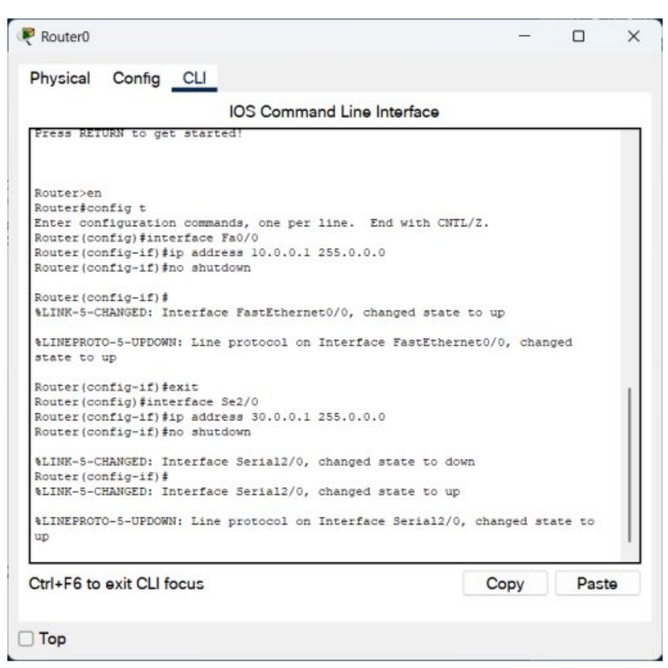
i. Configure IP address to routers in packet tracer.Explore the following messages:ping responses,destination unreachable,request timed out,reply.

ii. Procedure along with the topology



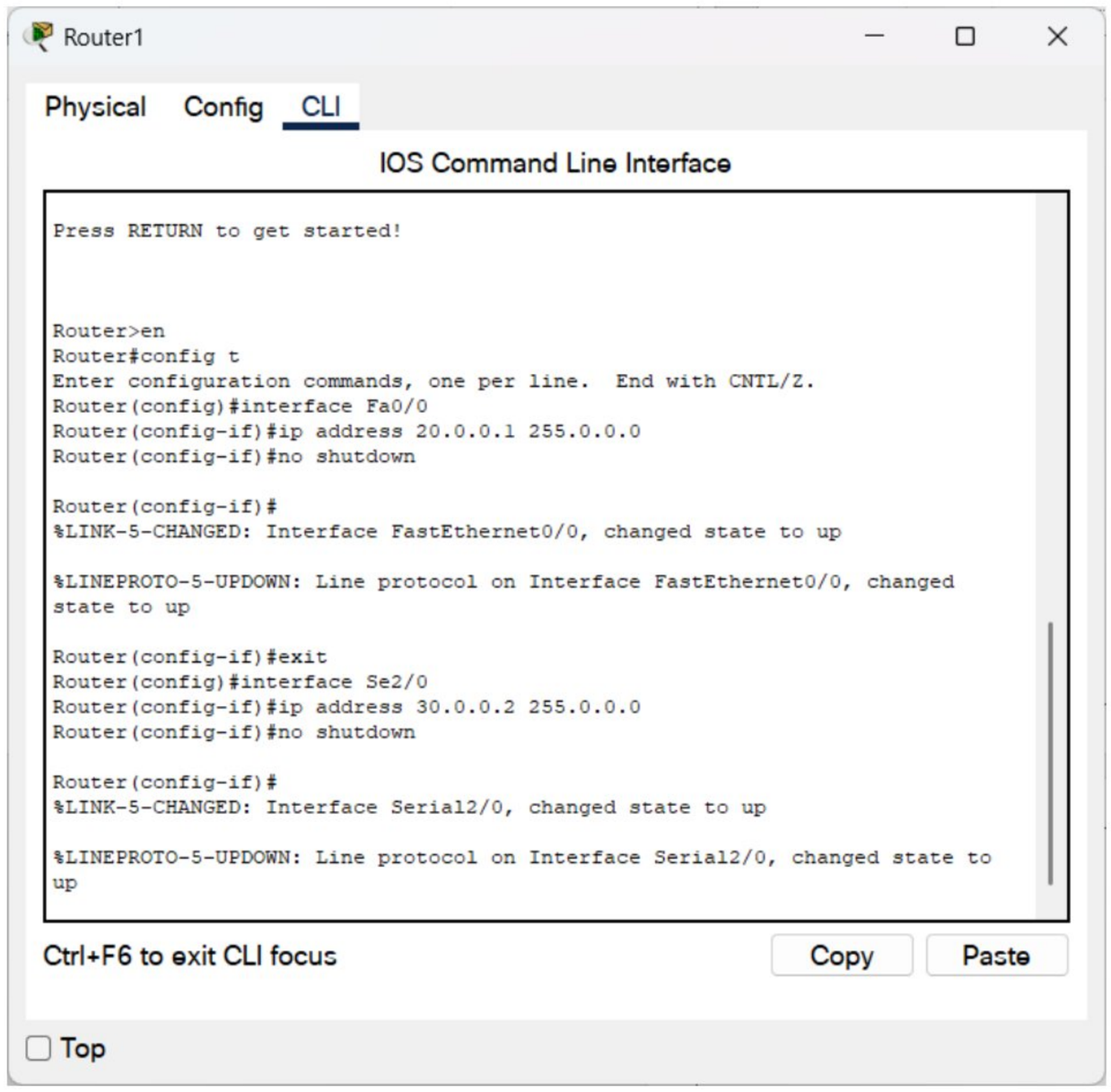
iii. Screen shots/ output

Router0 configuration



Scanned with OKEN Scanner

Router1 configuration



Ip route command in Router0

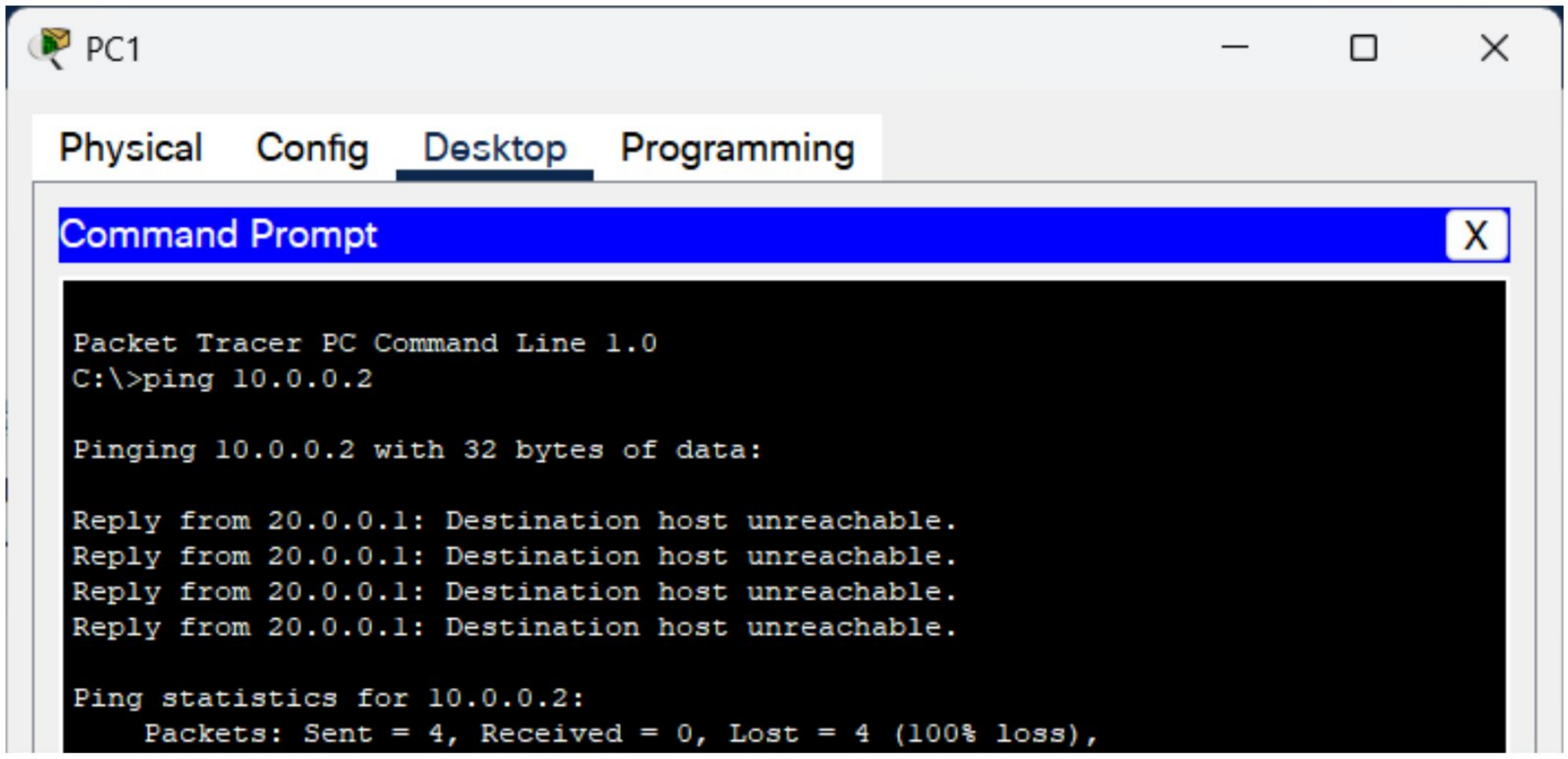
```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.2
Router(config)#
```

Ip route command in Router1

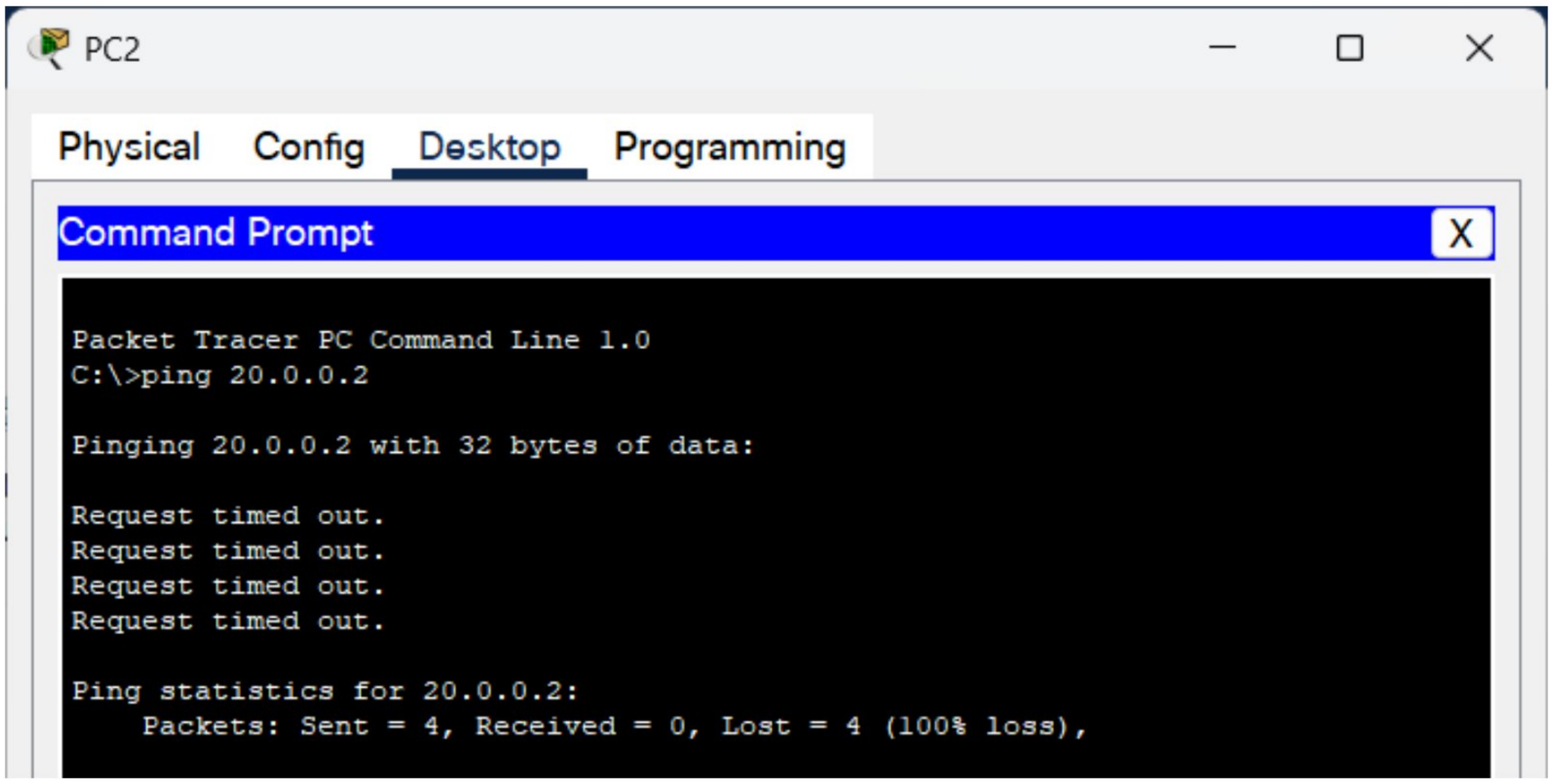
```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#
```

Destination host Unreachable (Before establishing network Fully)

Scanned with OKEN Scanner

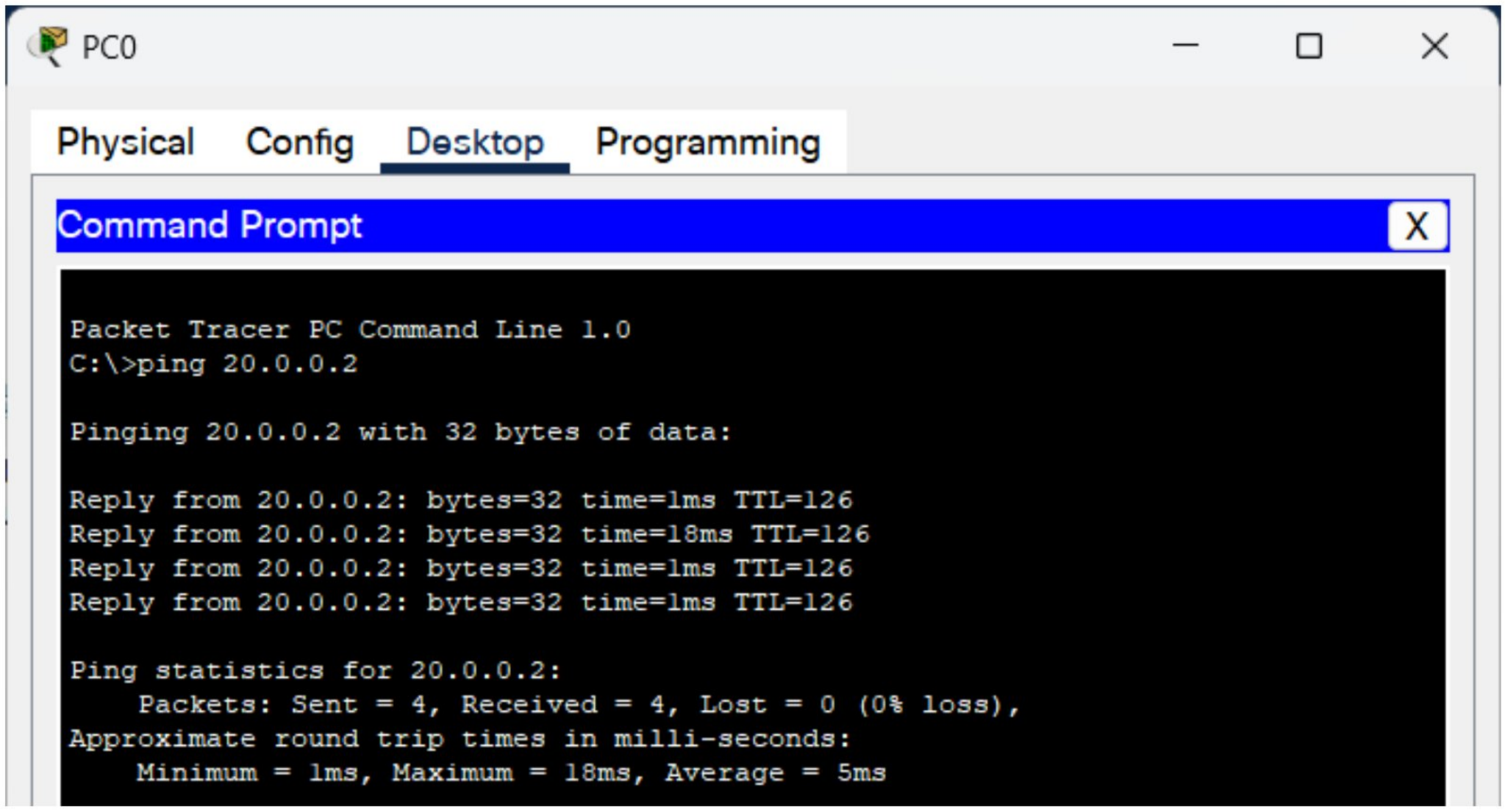


Request Timed Out

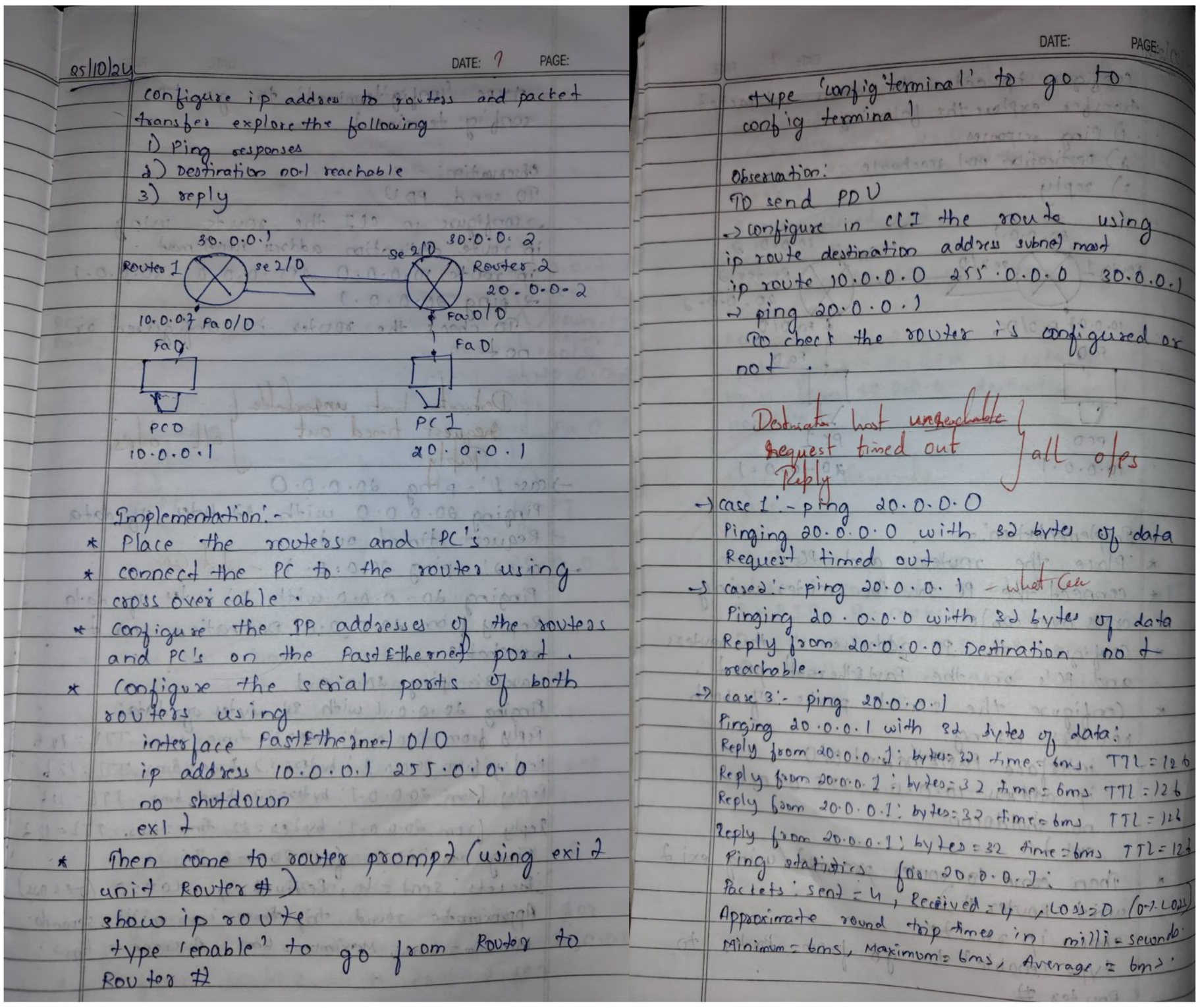


Reply from Destination

Scanned with OKEN Scanner



iv. Observation



05/10/24 DATE: PAGE:

configure ip address to routers and packet transfer explore the following
1) Ping responses
2) Destination not reachable
3) reply

Implementation:-
* Place the routers and PC's
* connect the PC to the router using cross over cable.
* Configure the IP addresses of the routers and PC's on the FastEthernet port.
* Configure the serial ports of both routers using
  interface FastEthernet 0/0
  ip address 10.0.0.1 255.0.0.0
  no shutdown
  exit
* Then come to router prompt (using exit until Router #)
  show ip route
  type 'enable' to go from Router to Router #

DATE: PAGE:

type 'config terminal' to go to config terminal

Observation:
To send PDU
→ configure in CLI the route using
ip route destination address subnet mask
ip route 10.0.0.0 255.0.0.0 30.0.0.1
→ ping 20.0.0.1
To check the router is configured or not.

Destination host unreachable
Request timed out
Reply
} all o/ps

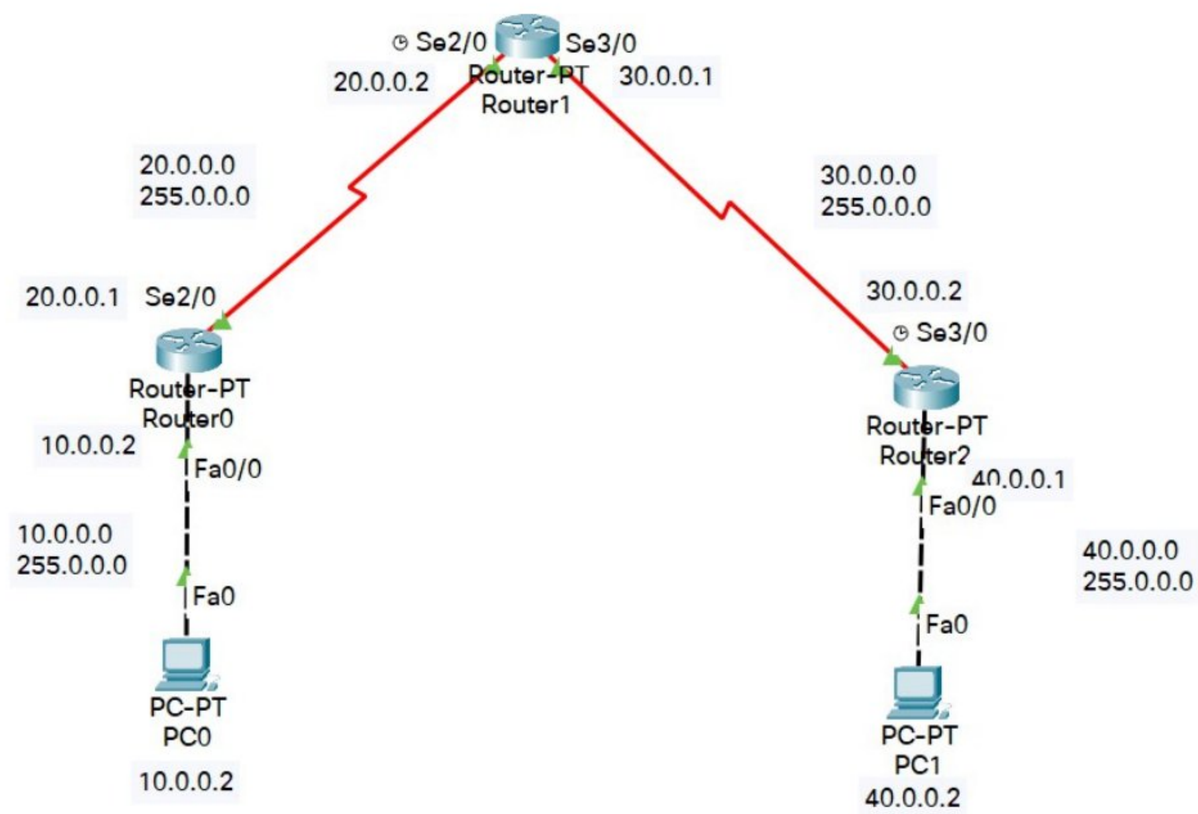
→ case 1:- ping 20.0.0.0
Pinging 20.0.0.0 with 32 bytes of data
Request timed out
→ case 2:- ping 20.0.0.1 - what can
Pinging 20.0.0.0 with 32 bytes of data
Reply from 20.0.0.0 Destination not reachable.
→ case 3:- ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=6ms TTL=126
Reply from 20.0.0.1: bytes=32 time=6ms TTL=126
Reply from 20.0.0.1: bytes=32 time=6ms TTL=126
Reply from 20.0.0.1: bytes=32 time=6ms TTL=126
Ping statistics for 20.0.0.1:
Packets: sent = 4, Received = 4, Loss = 0 (0% Loss)
Approximate round trip times in milli-seconds
Minimum = 6ms, Maximum = 6ms, Average = 6ms.

Scanned with OKEN Scanner

Program 3

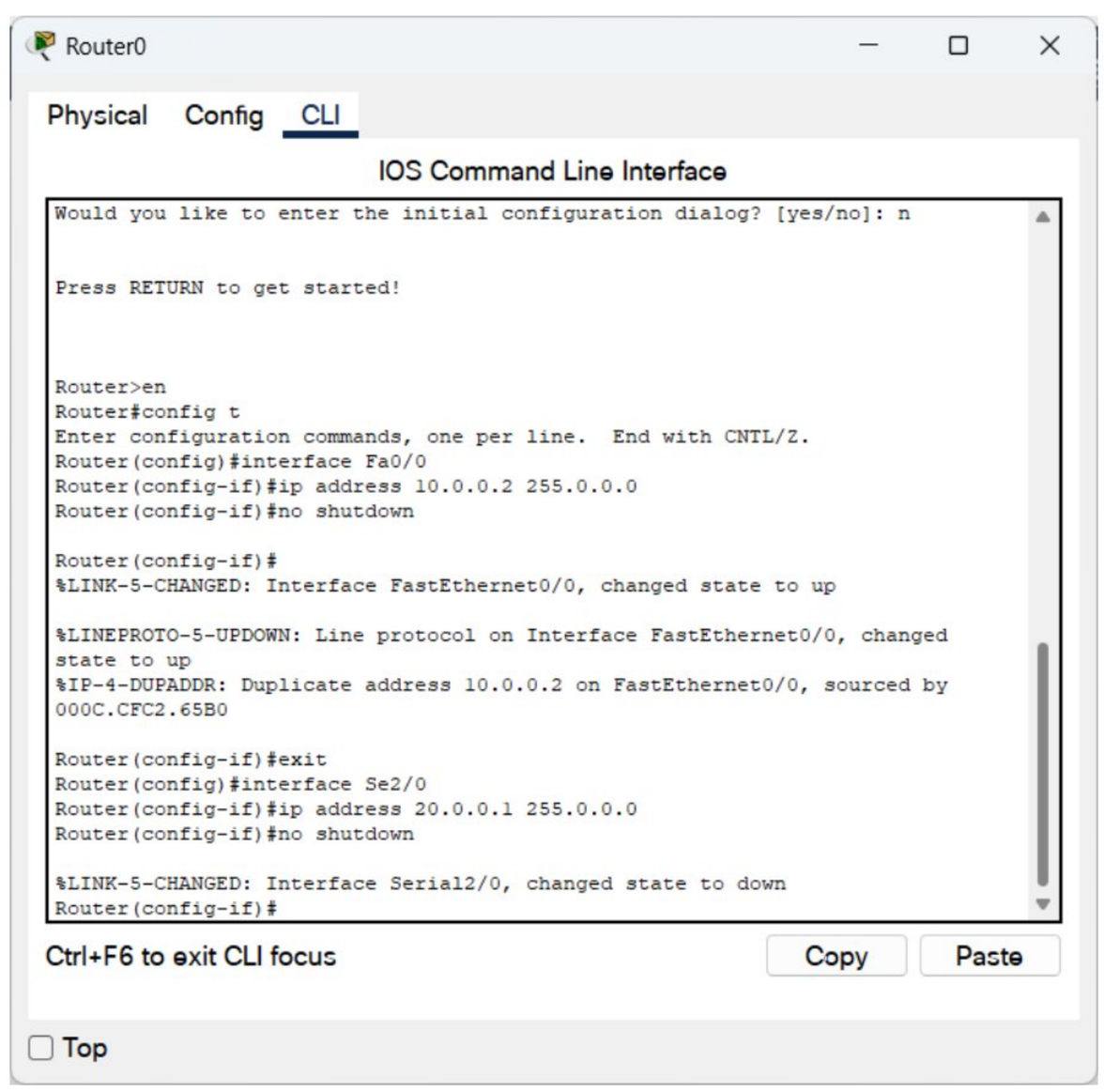
i. Configure default route, static route to the router

ii. Procedure along with the topology



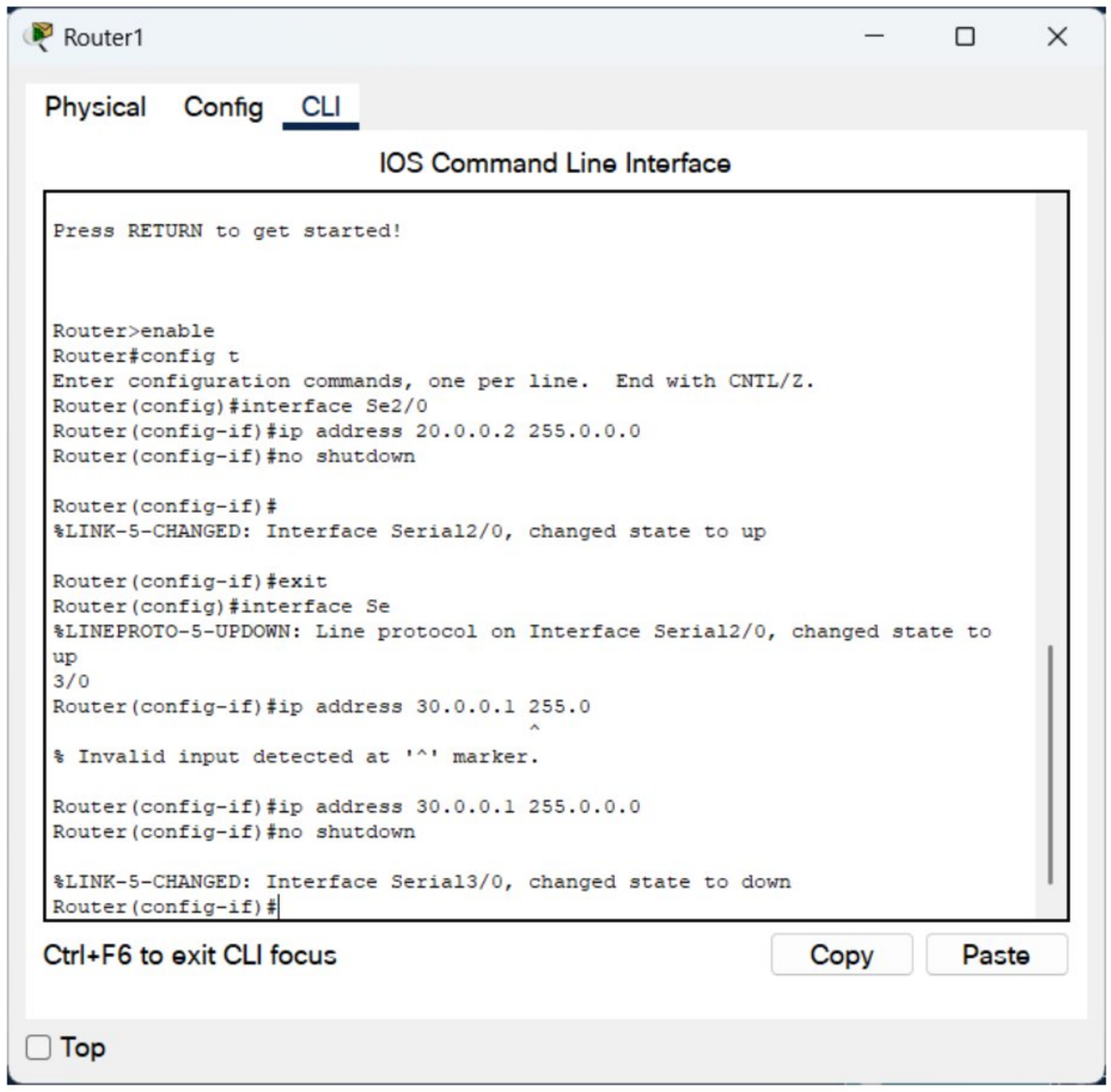
iii. Screen shots/ output

Router0 configuration

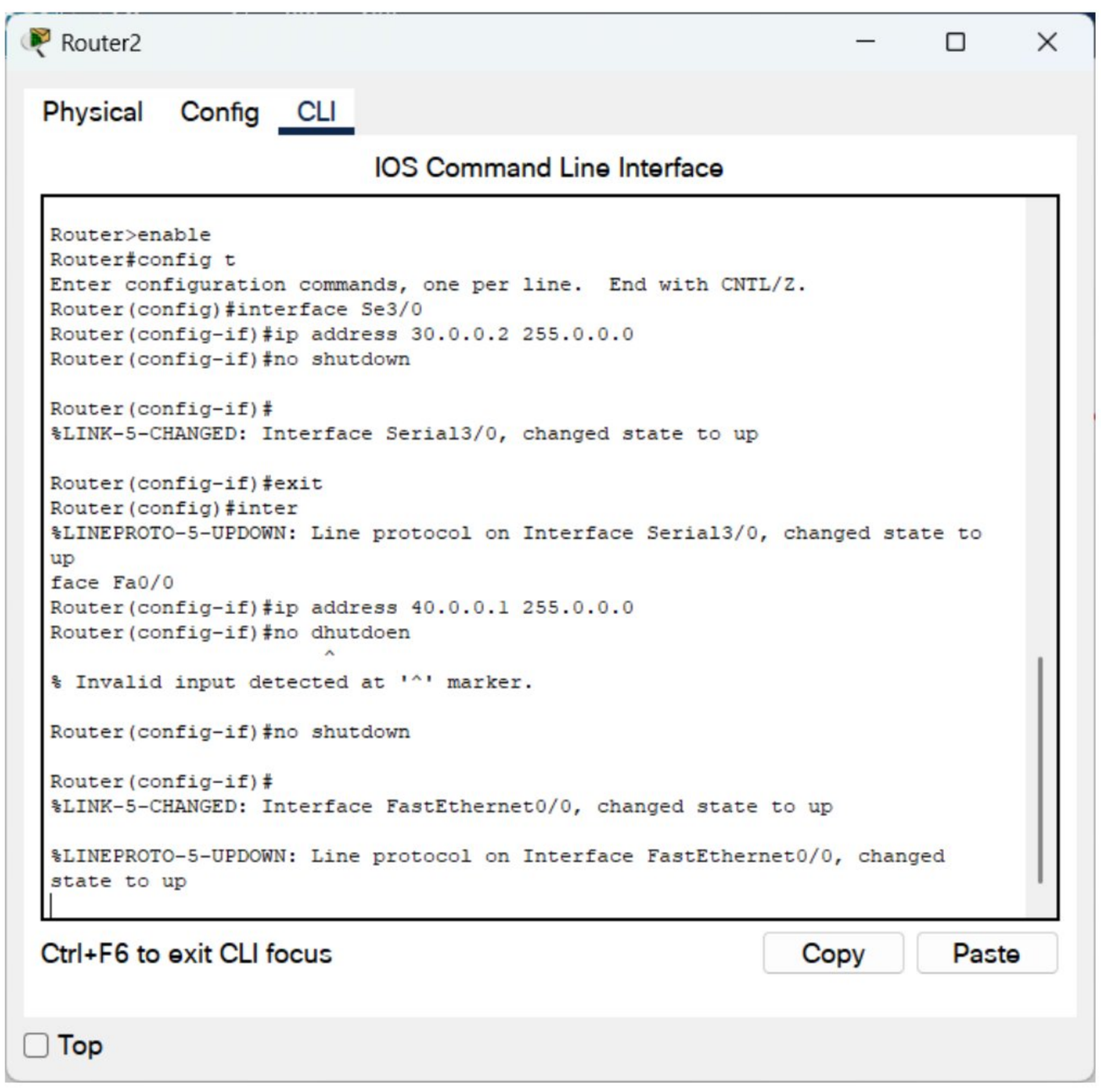


Scanned with OKEN Scanner

Router1 configuration



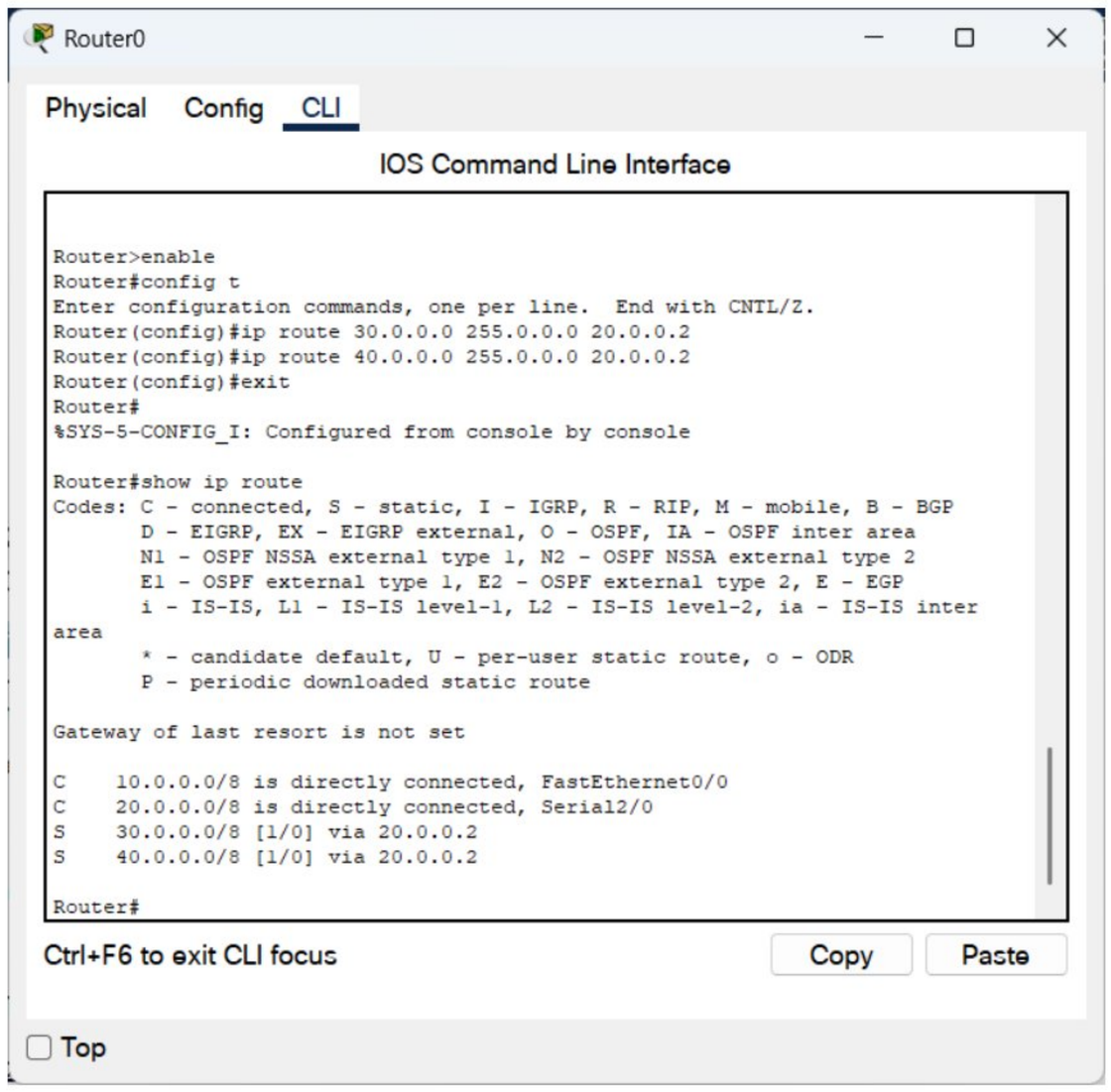
Router2 configuration



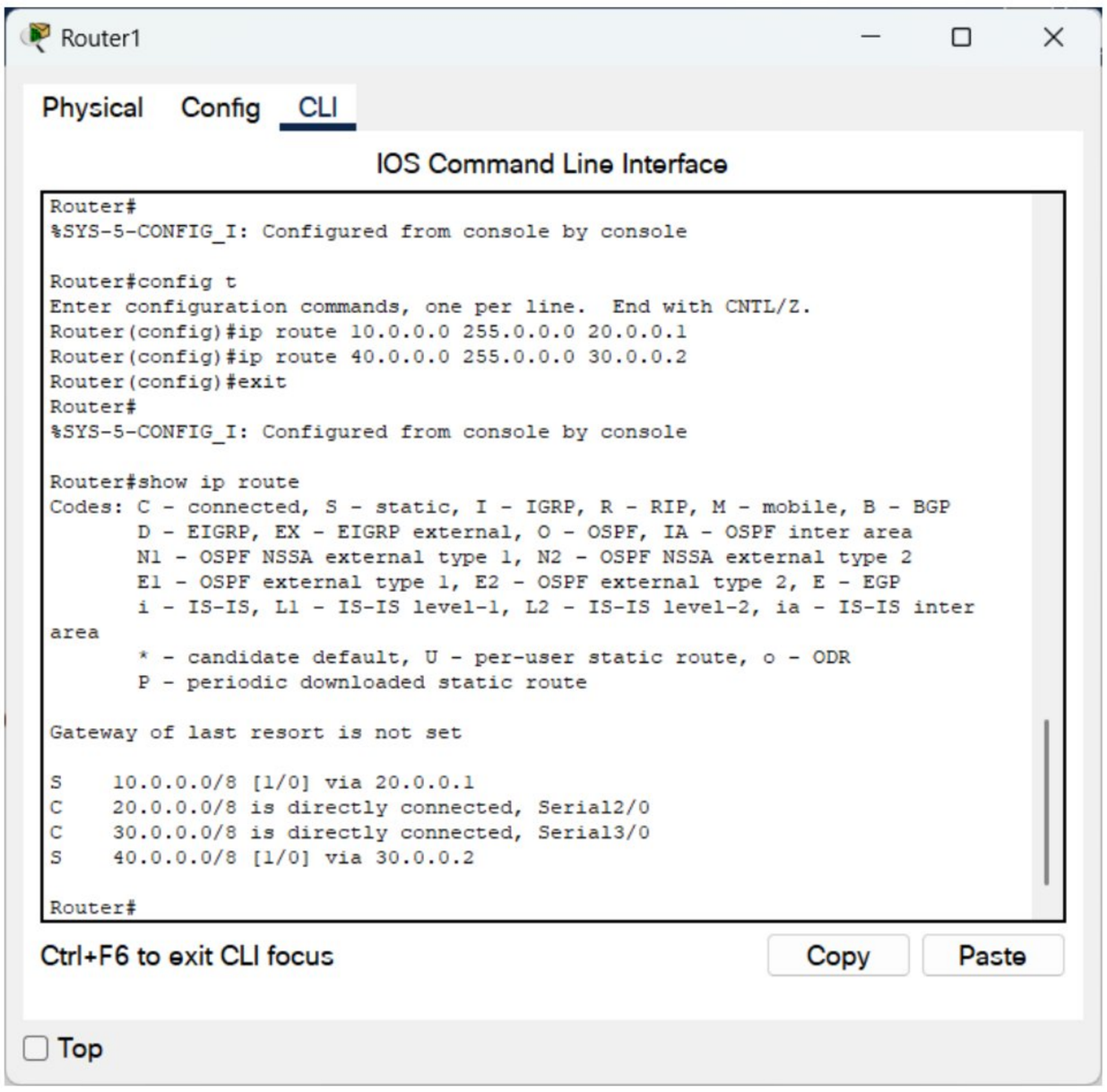
Scanned with OKEN Scanner

Static Routing:

Router0

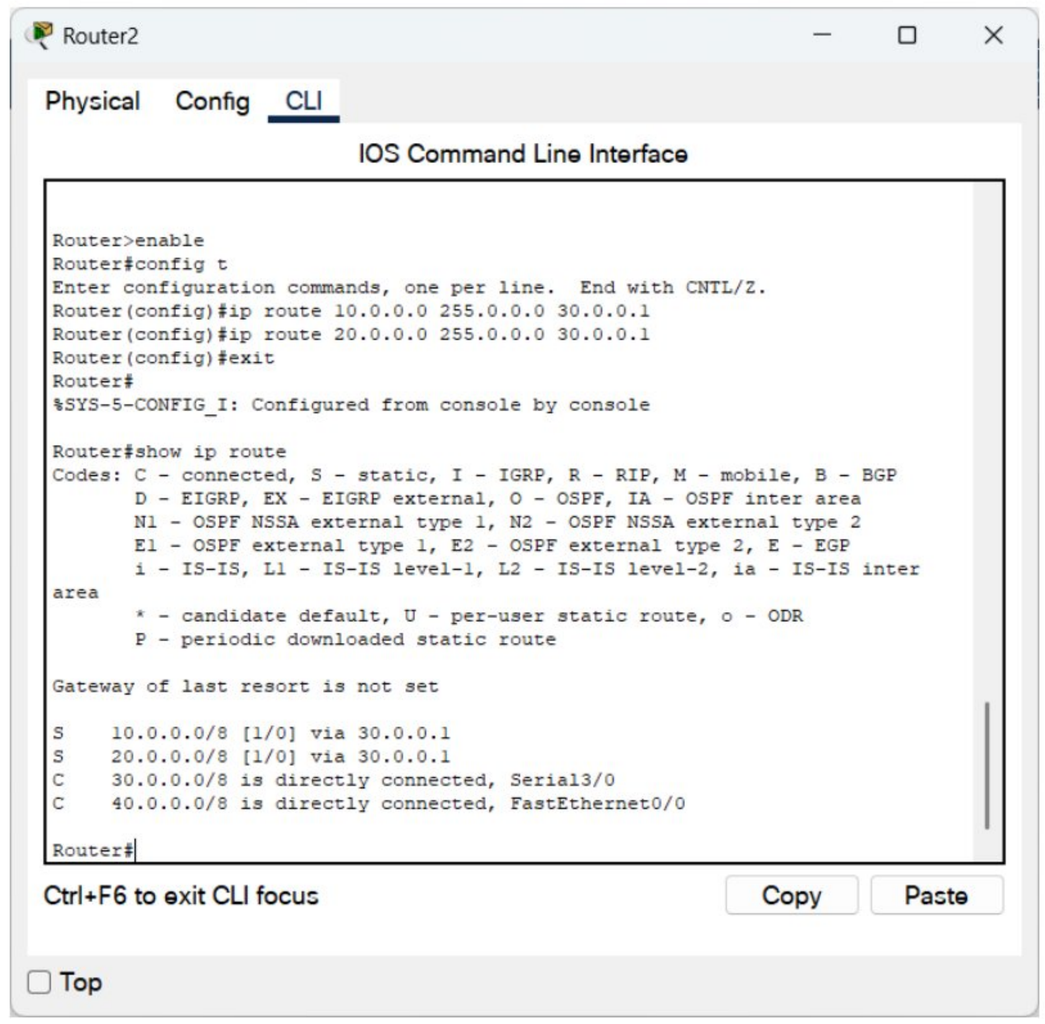


Router1



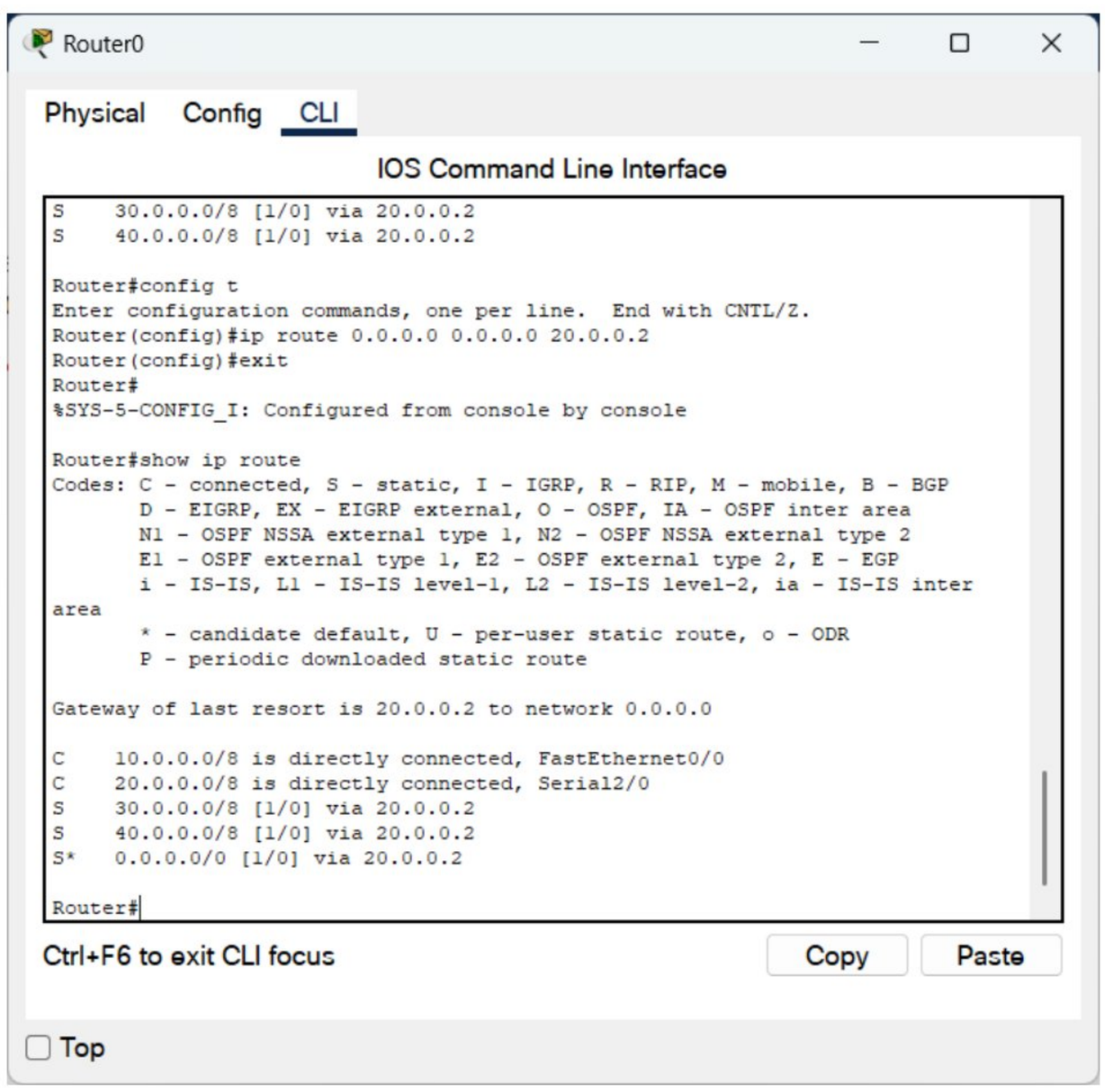
Scanned with OKEN Scanner

Router2



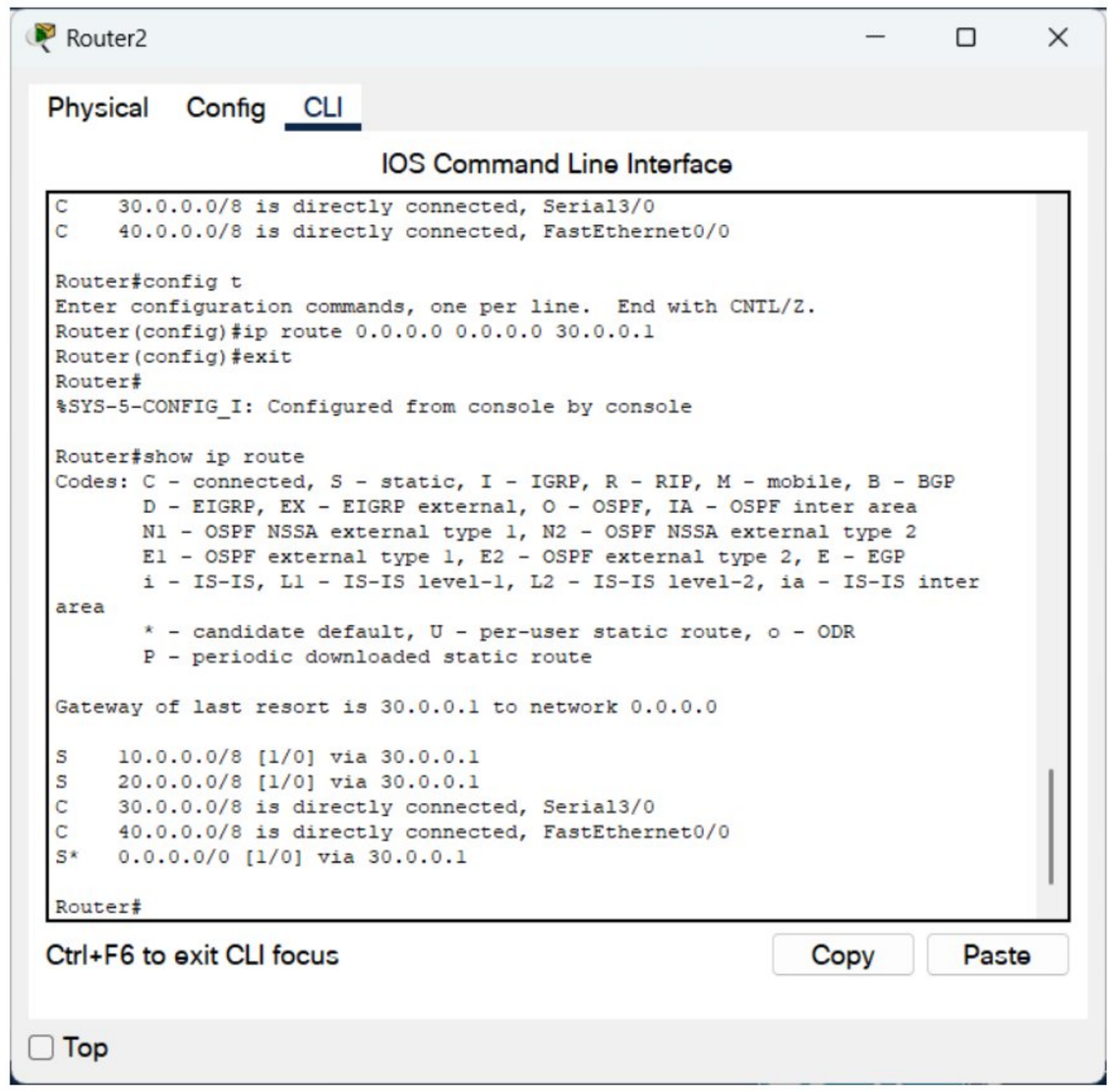
Default Routing:

Route0



Scanned with OKEN Scanner

Router2



Pinging:

```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=21ms TTL=125
Reply from 40.0.0.2: bytes=32 time=17ms TTL=125
Reply from 40.0.0.2: bytes=32 time=25ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 25ms, Average = 16ms
```


Scanned with OKEN Scanner

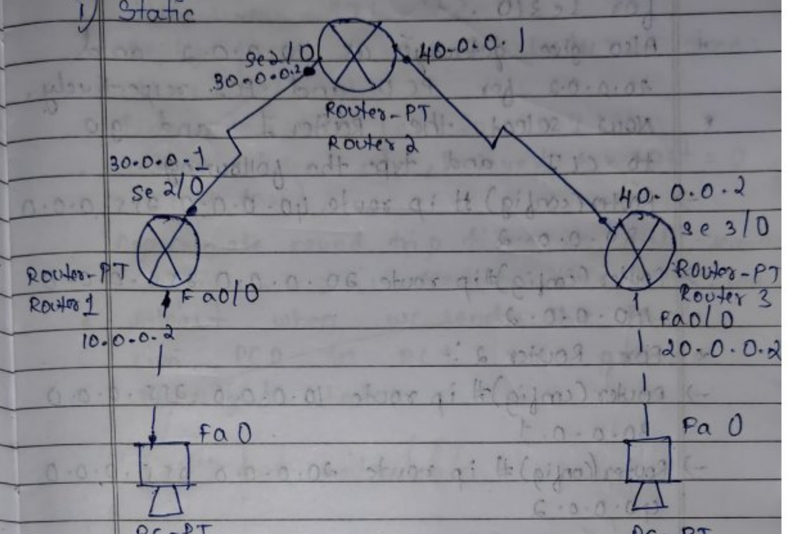
iv. Observation

25/10/24

DATE: PAGE:

Configure default route, static route to the router.

i) Static



Implementation:-

* Place the Generic routers i.e, three routers which creates 4 network.
* Also place end-devices and configure them.
* connect the Router and PC using copper cross wire and Router to Router using Serial DCE
* configure the PC0 with 10.0.0.1 & PC1 with 20.0.0.1
* configure Router 1 with 10.0.0.2 for fastEthernet 0/0 and Serial 2/0 is 30.0.0.1
* configure Router 2 with 30.0.0.2 for se2/0 ~~fastEthernet~~ and 40.0.0.1 for se 3/0.
* configure Router 3 with 40.0.0.2 for se3/0.
* Also give gateways as 10.0.0.2 and 20.0.0.2 for PC0 and PC1 respectively.
* Now select the Router 1 and go to CLI and type the following:-

-> Router(config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2

-> Router(config)# ip route 20.0.0.0 255.0.0.0 40.0.0.2

* For Router 2:-

-> Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.1

-> Router(config)# ip route 20.0.0.0 255.0.0.0 40.0.0.2

* For Router 3:

-> 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 20.0.0.1

Observation:-

* Now after all the ip route commands for the router now select the PC0 and go to command Prompt and type

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=125

Reply from 20.0.0.1: bytes=32 time=19ms TTL=125

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

Reply from 20.0.0.1: bytes=32 time=12ms TTL=125

Ping statistics for 20.0.0.1:

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

Approximate round trip times in milli-seconds:

Minimum = 2ms, Maximum = 19ms, Average = 8ms

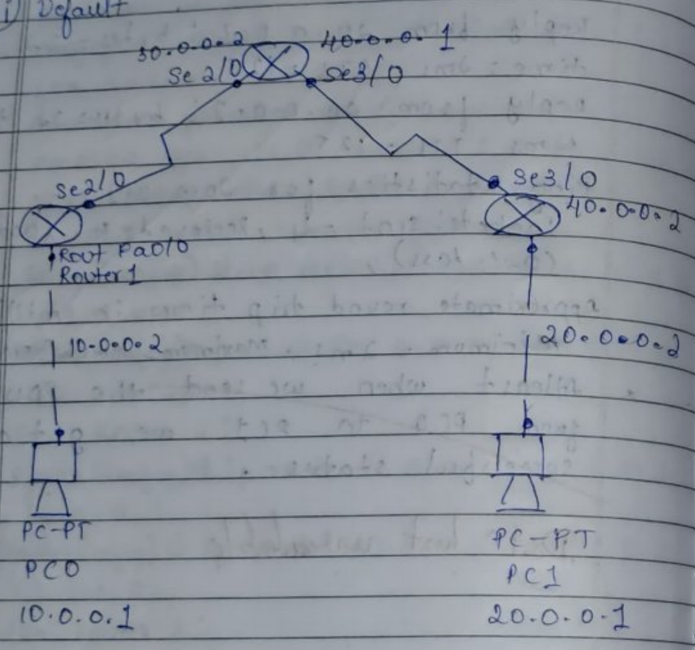
* Atleast when we send the PDU's from PC0 to PC1 we get an successful status.

Destn host unreachable

Scanned with OKEN Scanner

DATE: PAGE:

(ii) Default



Implementation :-

* Place the generic routers, i.e, three routers which creates 4 network.
* Also place the end-devices and configure them.
* Connect the Router and PC using copper cross wire and Router to Router using serial DCE.
* configure PC0 and PC1 as 10.0.0.1 and 20.0.0.1
* configure Router 1 with 10.0.0.2 for Fastethernet 0/0 and se2/0 i.e 30.0.0.1
* III'ly Router 2 with 30.0.0.2 for se2/0 and 40.0.0.1 for se3/0 and Router 3 as 40.0.0.2 for se3/0.

DATE: PAGE:

* Also give gateways as 10.0.0.2 and 20.0.0.2 for PC0 and PC1 respectively.
* Now select the Router 1 and go to CLI and type the following :-

→ Router (Config)# ip route 0.0.0.0 0.0.0.0 30.0.0.2

* For Router 2

→ Router (Config)# ip route 10.0.0.0 255.0.0.0 30.0.0.1

→ Router (Config)# ip route 20.0.0.0 255.0.0.0 40.0.0.2

* For Router 3

→ Router (Config)# ip route 0.0.0.0 0.0.0.0 40.0.0.1

Observation:-

* After all the router configurations now select PC0 and go to command prompt and type :-

PC> ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out
Reply from 20.0.0.1: bytes=32 time=16ms TTL=125
Reply from 20.0.0.1: bytes=32 time=16ms TTL=125
Reply from 20.0.0.1: bytes=32 time=11ms TTL=125
Ping statistics for 20.0.0.1:
Packets: sent = 4, Received = 3, Lost = 1 (25% Loss),
Approximate round trip times in milli-seconds:
Minimum = 11ms, Maximum = 16ms, Average = 14ms

* Atlast when we send the PDU from PC0 to PC1 we get an successful status.
* To verify the default connection the

DATE: PAGE:

Output should be as follows:
Gateway of last resort is 30.0.0.2 to network 0.0.0.0
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 30.0.0.0/8 is directly connected, Serial2/0
S* 0.0.0.0/0 [1/0] via 30.0.0.2

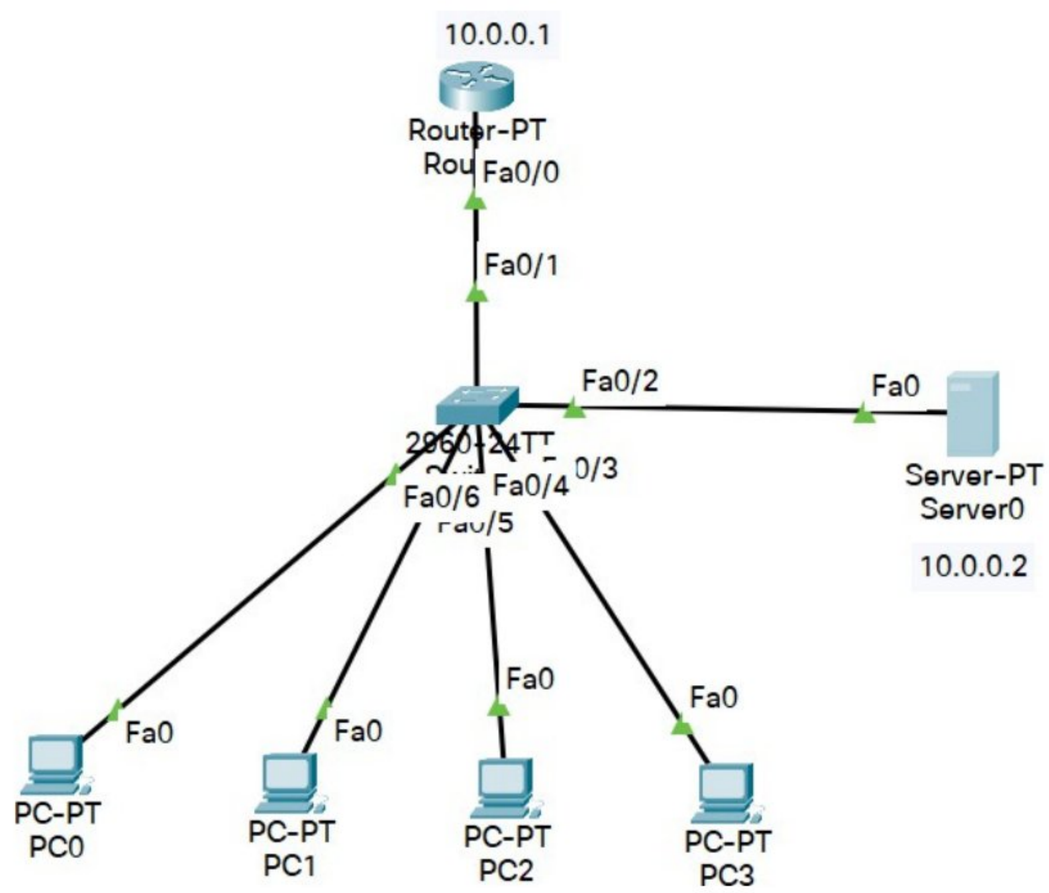
23/10/22

Scanned with OKEN Scanner

Program 4

i. Configure DHCP within a LAN and outside LAN.

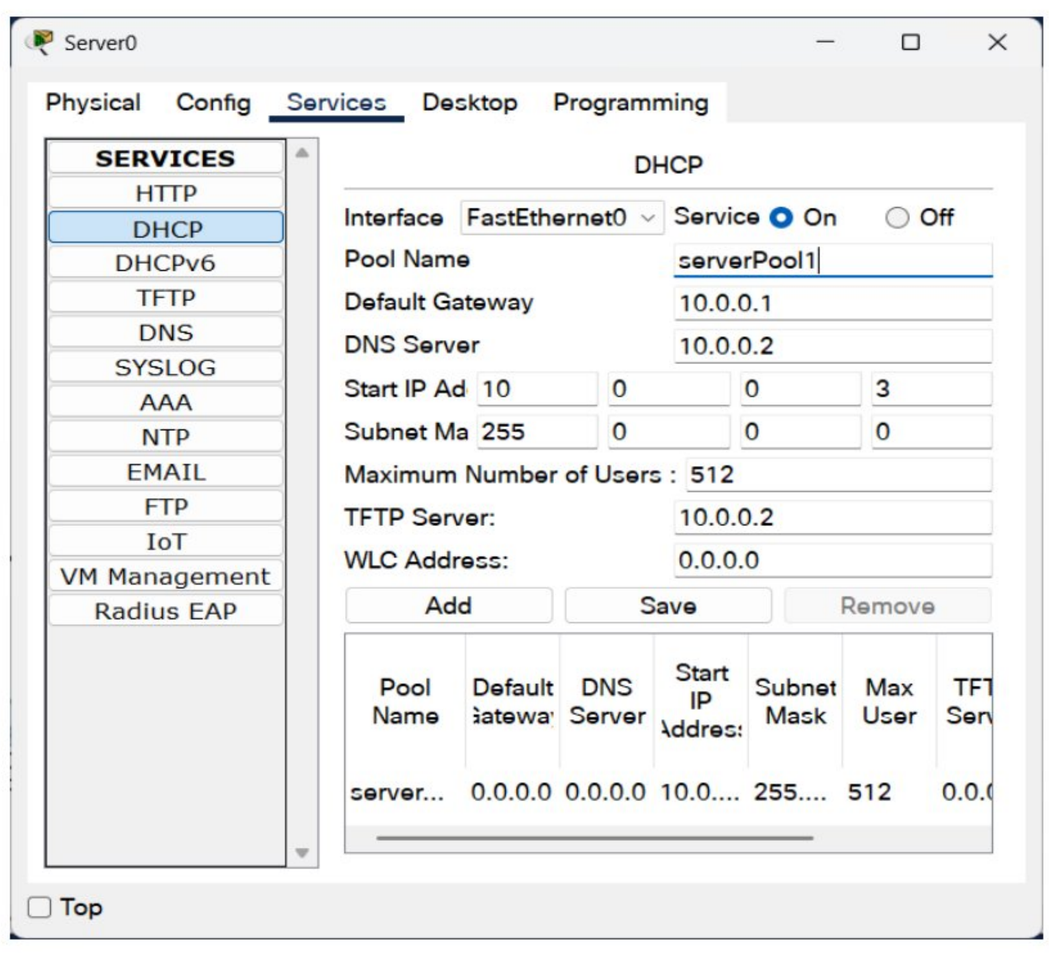
ii. Procedure along with the topology



iii. Screen shots/ output

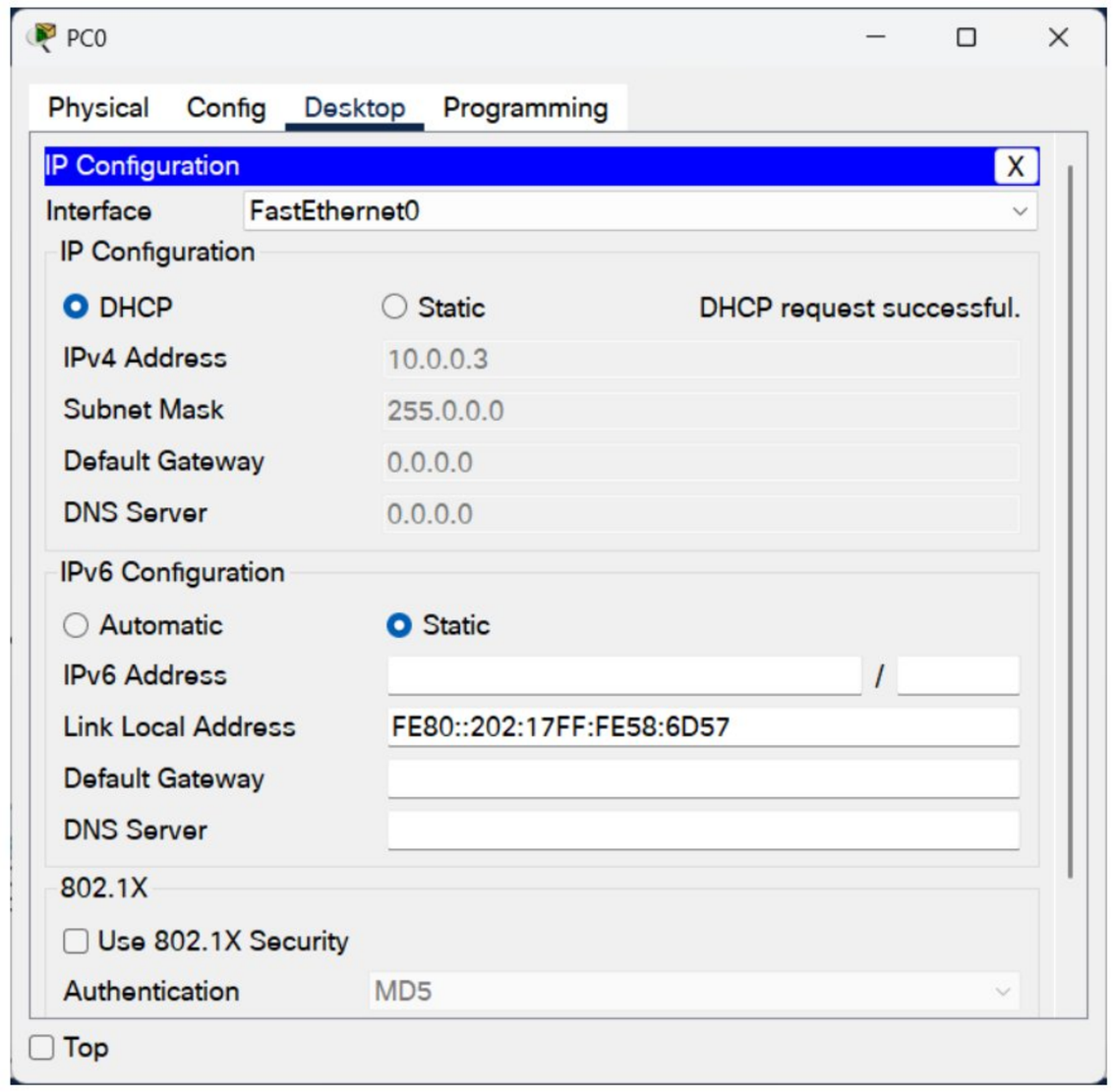
DHCP Within LAN

DHCP Configuration

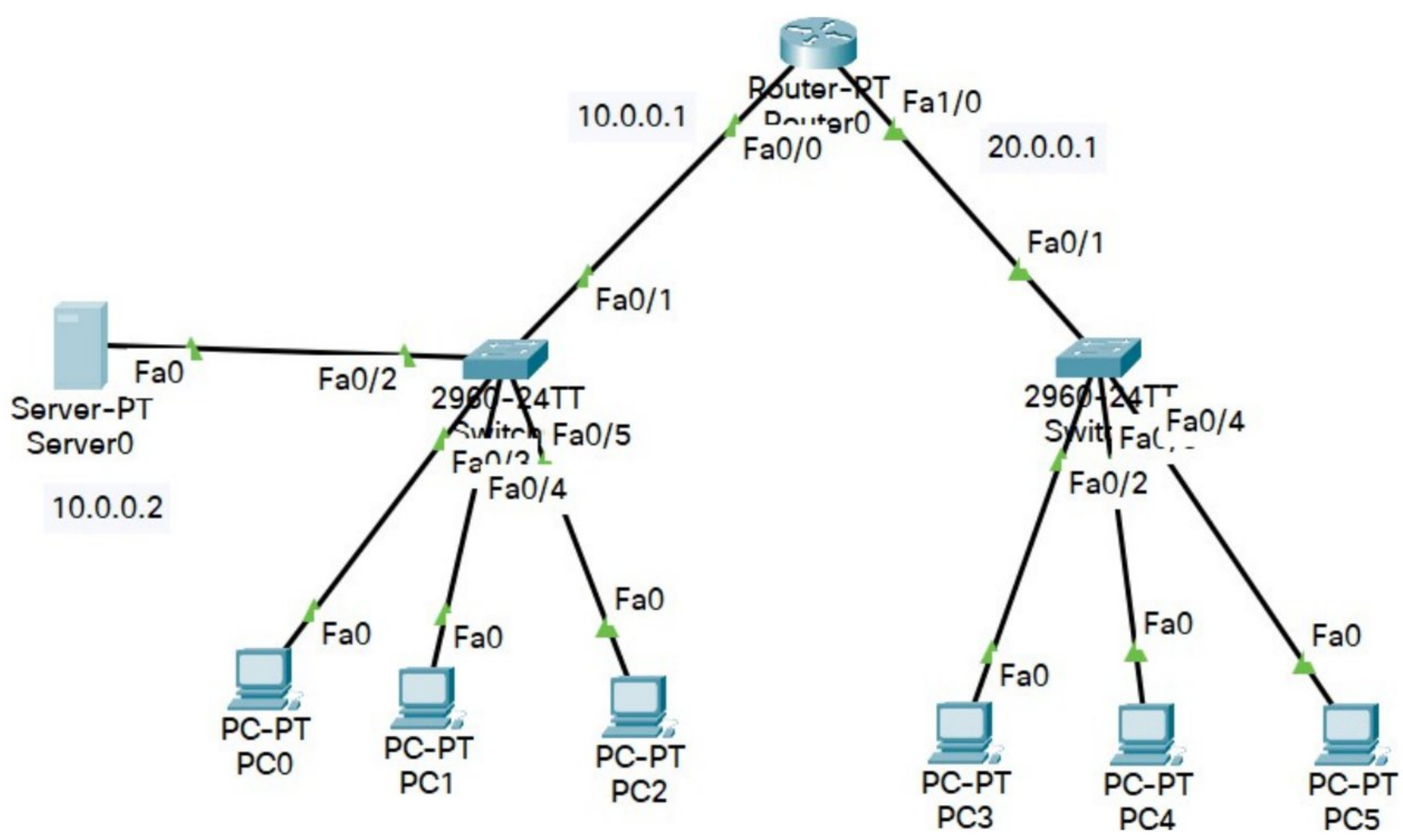


Scanned with OKEN Scanner

PC settings

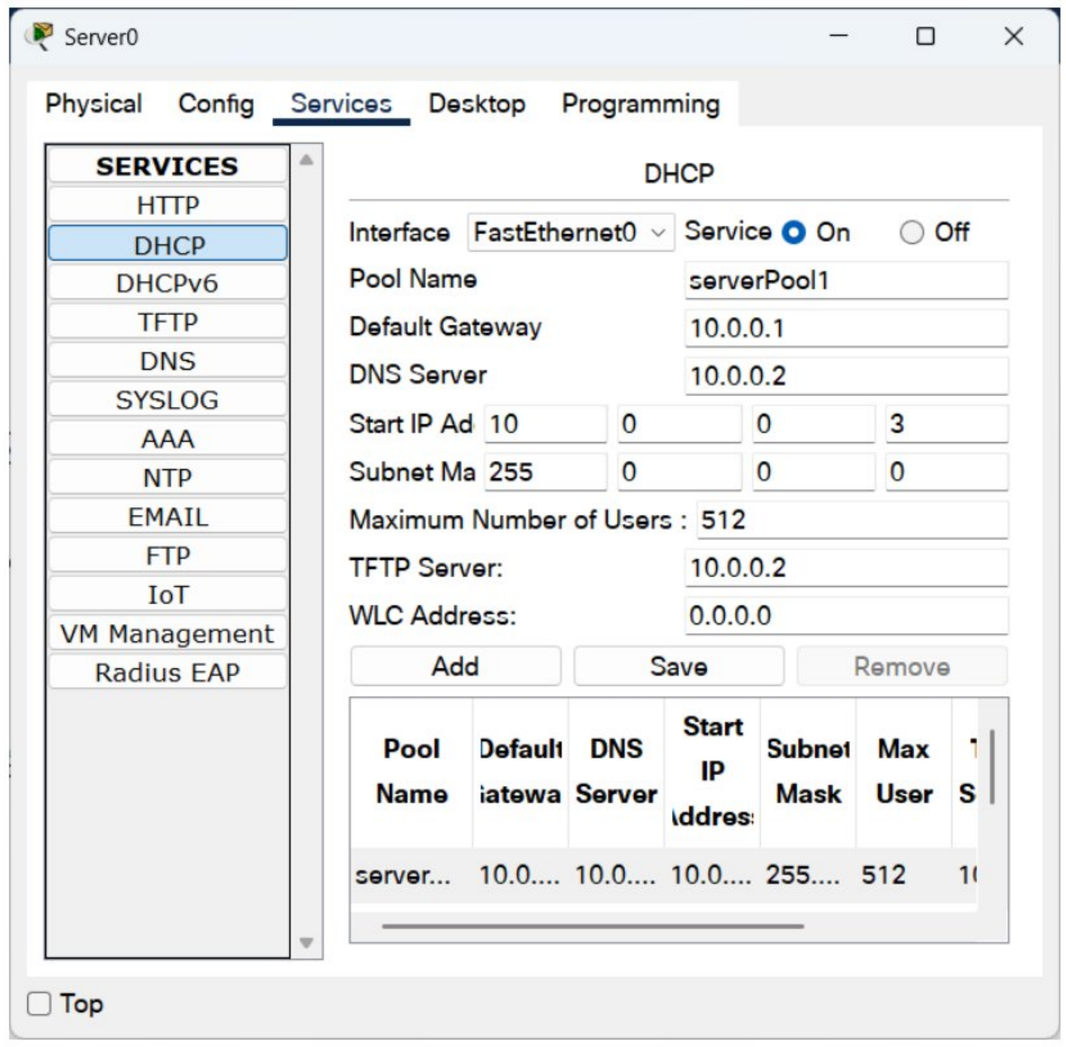


DHCP outside LAN:

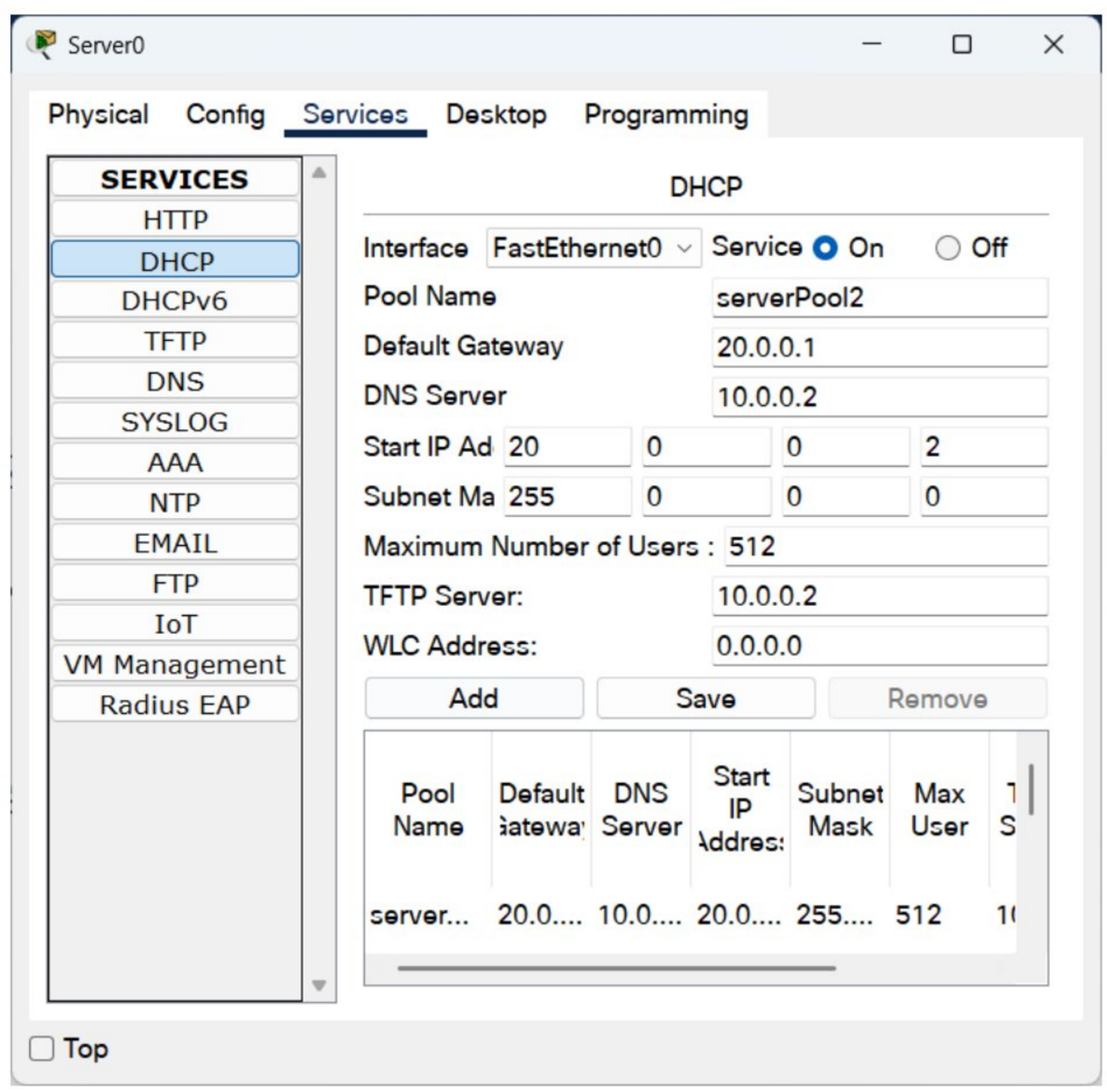


Scanned with OKEN Scanner

DHCP configuration for inside LAN



DHCP configuration for outside LAN

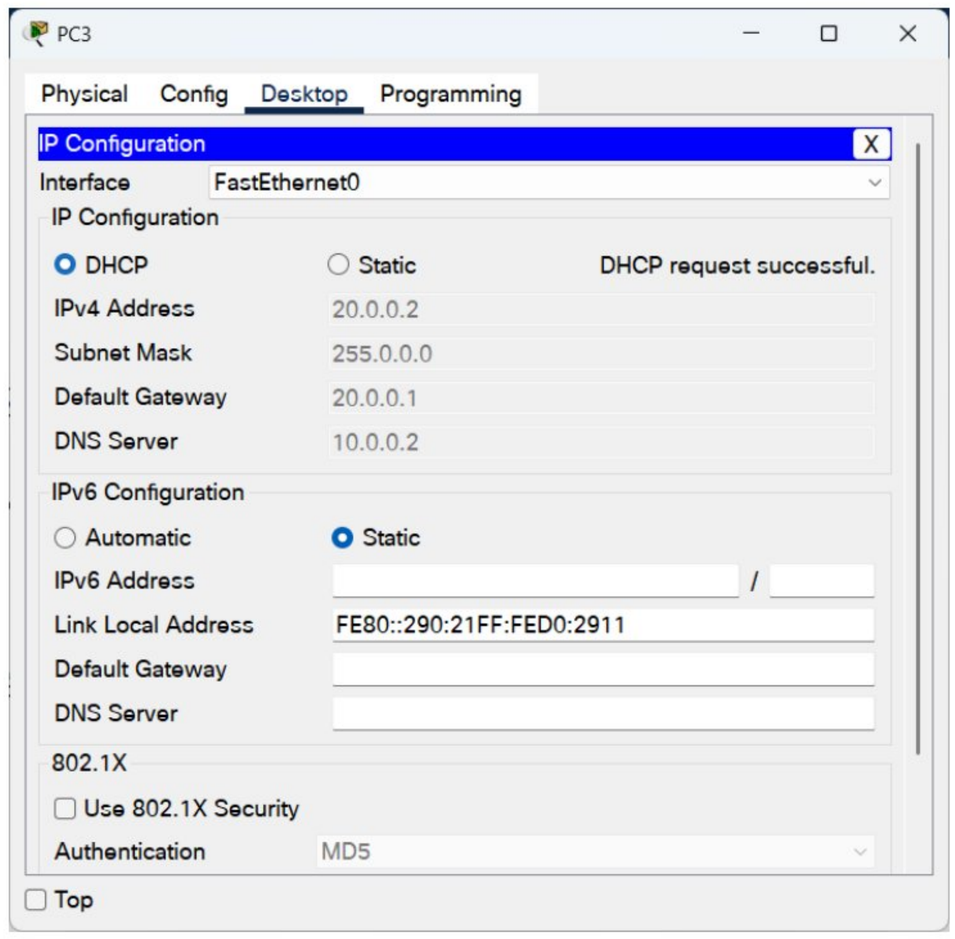


Scanned with OKEN Scanner

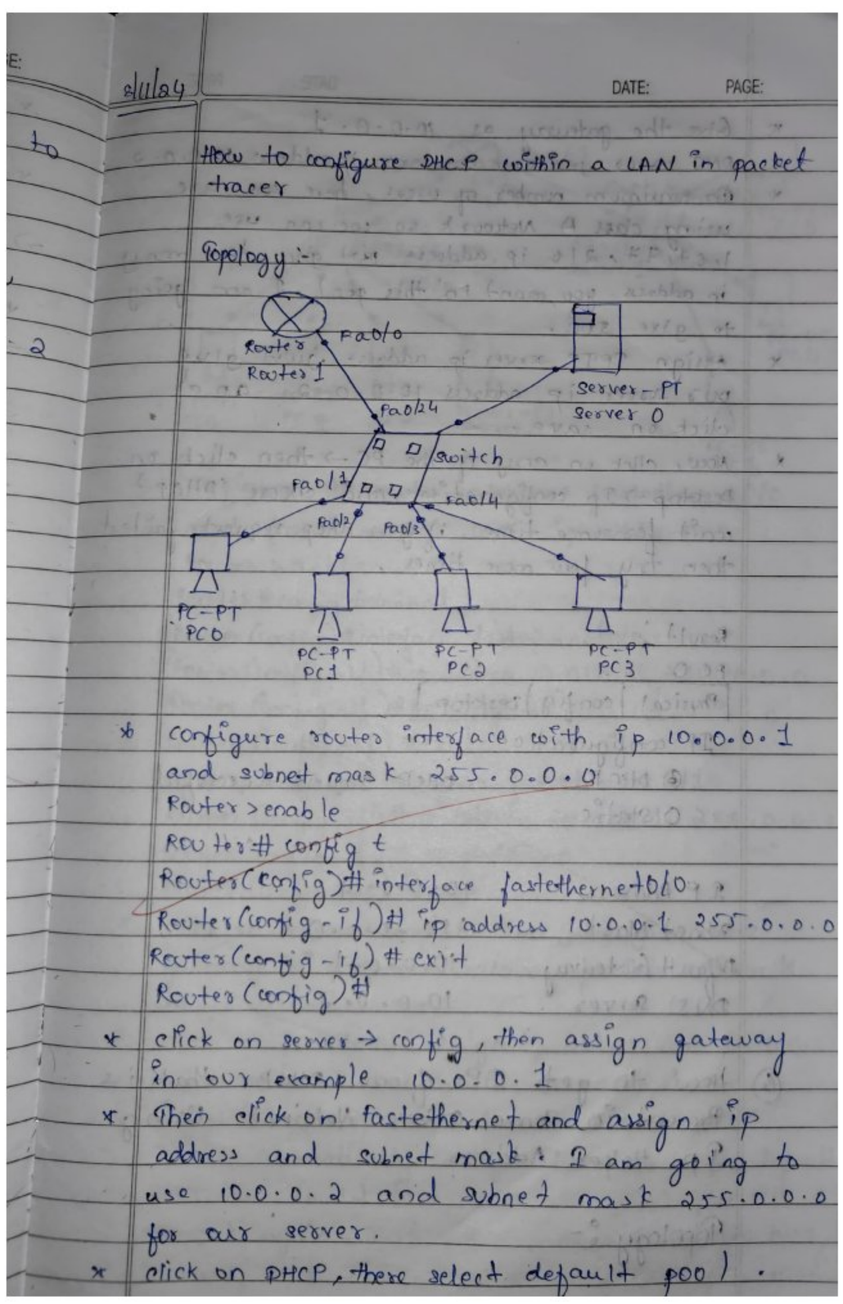
Ip helper command in Router

```
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#
```

PC setting in another network



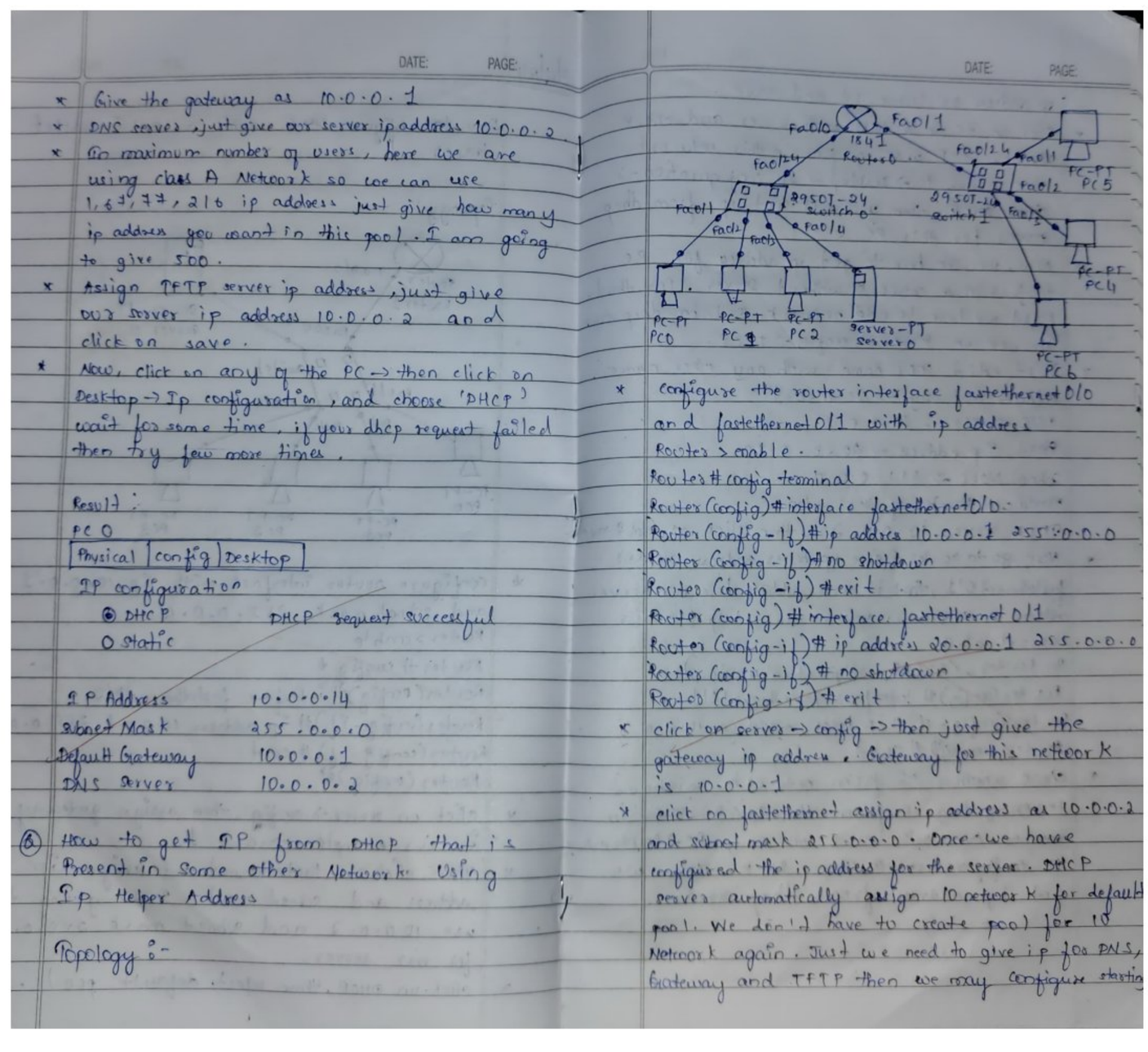
iv. Observation



How to configure DHCP within a LAN in packet tracer

Topology :-

* configure router interface with ip 10.0.0.1 and subnet mask 255.0.0.0
Router > enable
Router # config t
Router(config)# interface fastethernet0/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if) # exit
Router(config)#
* click on server → config, then assign gateway in our example 10.0.0.1
* Then click on fastethernet and assign ip address and subnet mask. I am going to use 10.0.0.2 and subnet mask 255.0.0.0 for our server.
* click on DHCP, there select default pool.



* Give the gateway as 10.0.0.1
* DNS server, just give our server ip address 10.0.0.2
* In maximum number of users, here we are using class A Network so we can use 1, 6^4, 77, 216 ip address just give how many ip address you want in this pool. I am going to give 500.
* Assign TFTP server ip address, just give our server ip address 10.0.0.2 and click on save.
* Now, click on any of the PC → then click on Desktop → Ip configuration, and choose 'DHCP' wait for some time, if your dhcp request failed then try few more times.

Result :
PC 0
Physical | Config | Desktop
IP configuration
DHCP — DHCP request successful
Static

IP Address 10.0.0.14
Subnet Mask 255.0.0.0
Default Gateway 10.0.0.1
DNS Server 10.0.0.2

(a) How to get IP from DHCP that is present in some other Network using Ip Helper Address

Topology :-

Fa0/0 Fa0/1 1841 Router 0 Fa0/24 Fa0/1 2950T-24 switch 0 Fa0/4 Fa0/2 Fa0/3 Fa0/24 Fa0/1 2950T-24 switch 1 Fa0/2 Fa0/3 PC-PT PC5 PC-PT PC4 PC-PT PC6 PC-PT PC0 PC-PT PC1 PC-PT PC2 Server-PT Server 0

* configure the router interface fastethernet0/0 and fastethernet0/1 with ip address
Router > enable.
Router # config terminal
Router(config)# interface fastethernet0/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if) # exit
Router(config)# interface fastethernet 0/1
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if) # no shutdown
Router(config-if) # exit
* click on server → config → then just give the gateway ip address. Gateway for this network is 10.0.0.1
* click on fastethernet assign ip address as 10.0.0.2 and subnet mask 255.0.0.0. Once we have configured the ip address for the server. DHCP server automatically assign 10 network for default pool. We don't have to create pool for 10 Network again. Just we need to give ip for DNS, Gateway and TFTP then we may configure starting

Scanned with OKEN Scanner

DATE: PAGE:

ip address or leave it and save.

* click on PC in a LAN with server and check whether DHCP working fine in this network, click on any PC -> Desktop -> IP configuration -> choose DHCP, then you will get ip from dhcp server for this PC.

* Now, we see how to get ip address for PC that is in a network without server. For that, first we have to add network pool in dhcp server. So, click on server -> config -> DHCP.

* Just edit POOL name with any other name.
Default Gateway -> 20.0.0.1
DNS server -> 10.0.0.2
Start IP address -> 20.0.0.10
Subnet Mask -> 255.0.0.0
Maximum Number of Users -> 100
TFTP server 10.0.0.2, then click on add & save

* Now go to router & give ip helper address under fastethernet 0/1, that is service address here our server address is 10.0.0.2, Now we can get ip for this network also.
In Router, (Global configuration mode)
Router(config)# interface fastethernet 0/1
Router(config-if)# ip helper-address 10.0.0.2
Router(config-if)# exit.

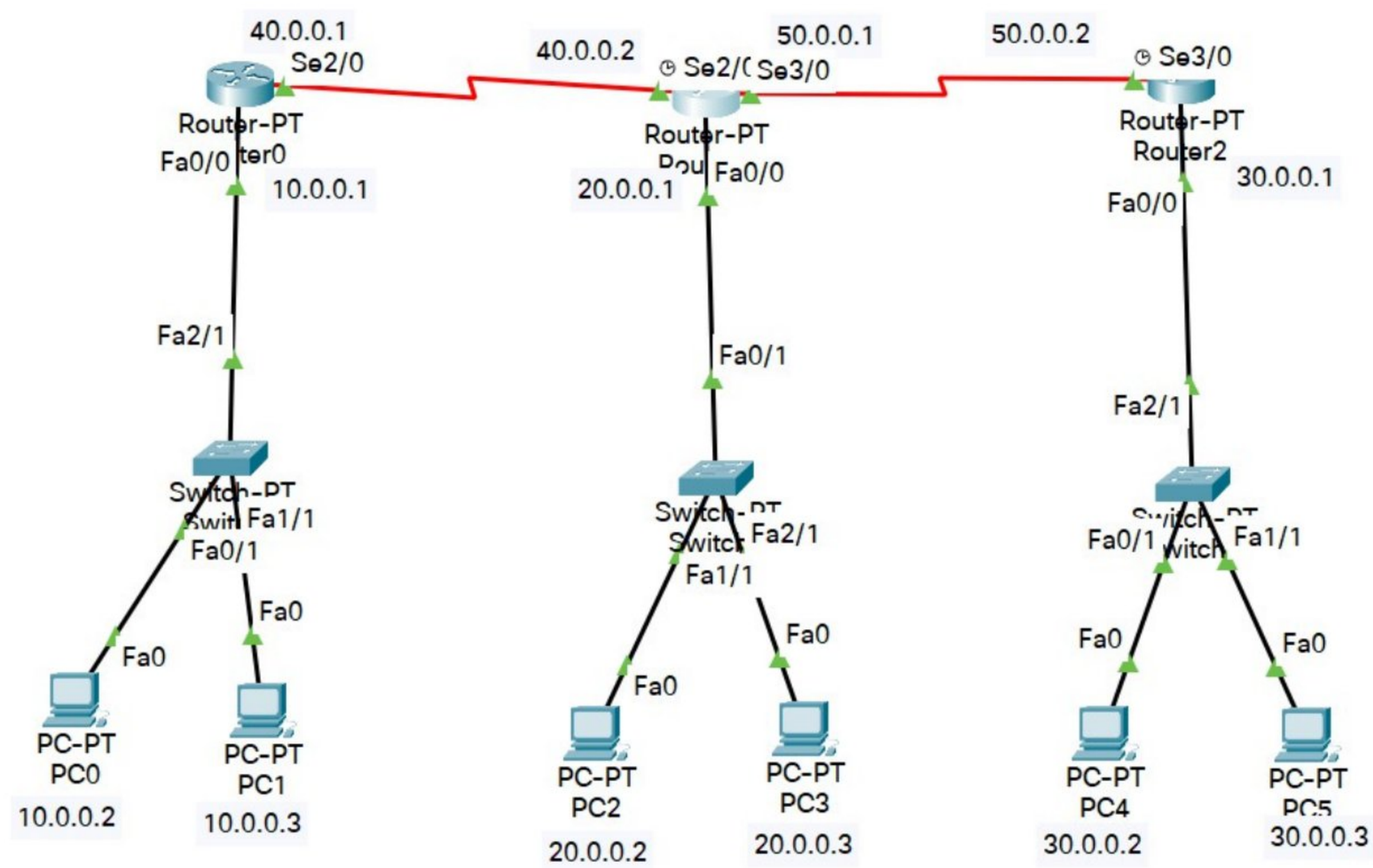
* check whether PC from network without server getting ip from the DHCP server in another Network. Click on any PC -> Desktop -> IP configuration -> choose DHCP. Now we got ip address from dhcp server.

Scanned with OKEN Scanner

Program 5

i. Configure RIP routing Protocol in Routers

ii. Procedure along with the topology



iii. Screen shots/ output

Router0

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#end
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

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    40.0.0.0/8 is directly connected, Serial2/0
```


Scanned with OKEN Scanner

Router1

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#end
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 40.0.0.1, 00:00:08, Serial2/0
C    20.0.0.0/8 is directly connected, FastEthernet0/0
R    30.0.0.0/8 [120/1] via 50.0.0.2, 00:00:10, Serial3/0
C    40.0.0.0/8 is directly connected, Serial2/0
C    50.0.0.0/8 is directly connected, Serial3/0
```

Router2

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#end
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 [120/2] via 50.0.0.1, 00:00:28, Serial3/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:28, Serial3/0
C    30.0.0.0/8 is directly connected, FastEthernet0/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:28, Serial3/0
C    50.0.0.0/8 is directly connected, Serial3/0
```

Pinging:

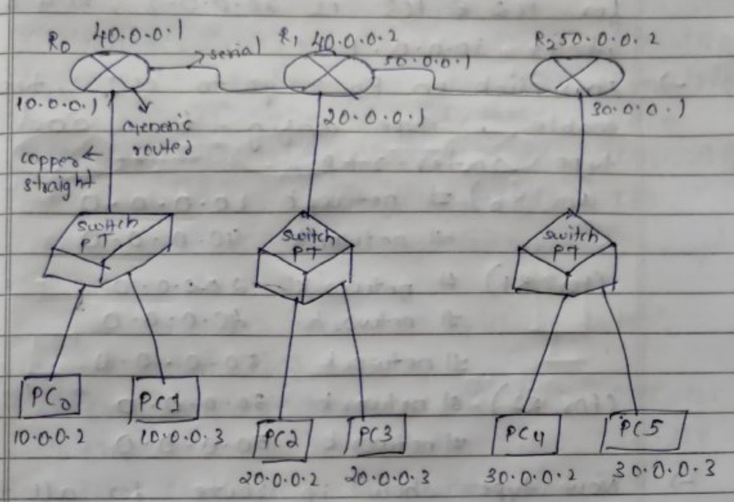
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 9ms, Average = 6ms
```


Scanned with OKEN Scanner

iv. Observation

11/24 DATE: PAGE:

Routing Information Protocol (RIP)



Implementation:-

-> Drag and drop the Routers (Generic) on the window and configure them

-> Also the switch-PT should be placed along with the PC's on the window.

-> Now the connection between Routers is serial DCE and between Routers and switch is copper straight and switch & PC's is automatic connection.

-> configure the PC's as 10.0.0.2, 10.0.0.3, 20.0.0.2, 20.0.0.3, 30.0.0.2, 30.0.0.3.

-> Now configure R0, for fastethernet 0/0 ip address is 10.0.0.1, for serial 2/0 is 40.0.0.1, for R1 fastethernet 0/0 is 20.0.0.1, serial 2/0 is 40.0.0.2, serial 3/0 is 50.0.0.1, for R2 fastethernet 0/0 is 30.0.0.1, serial 3/0 is 50.0.0.2

DATE: PAGE:

-> Now assign the gateways for all the PC's, for PC0 & PC1 it is 10.0.0.1, for PC2 & PC3 is 20.0.0.1, PC4 & PC5 it is 30.0.0.1

-> Now click on R0, go to CLI, type enable, type config-t, and type router rip,

(For R0) # network 10.0.0.0
network 40.0.0.0

(For R1) # network 20.0.0.0
network 40.0.0.0
network 50.0.0.0

(For R2) # network 30.0.0.0
network 50.0.0.0

-> Now type show ip route for all the the routers, Router# show ip route.

Observation:-

-> The output after show ip route is as follows:-

For R0:-

Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
R 20.0.0.0/8 [120/1] via 40.0.0.2, 00:00:13, serial 2/0
R 30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:13, serial 2/0
C 40.0.0.0/8 is directly connected, serial 2/0
R 50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:13, serial 2/0

For R1:-

Gateway of last resort is not set
R 10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:06, serial 2/0
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
R 30.0.0.0/8 [120/1] via 50.0.0.2, 00:00:09, serial 3/0
C 40.0.0.0/8 is directly connected, serial 2/0

DATE: PAGE:

C 50.0.0.0/8 is directly connected, serial 3/0

For R2:-

Gateway of last resort is not set
R 10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:24, serial 3/0
R 20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:24, serial 3/0
C 30.0.0.0/8 is directly connected, FastEthernet 0/0
R 40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:24, serial 3/0
C 50.0.0.0/8 is directly connected, serial 3/0

-> Now click on PC0, goto command prompt

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=6ms TTL=125
Reply from 30.0.0.2: bytes=32 time=2ms TTL=125
Reply from 30.0.0.2: bytes=32 time=2ms 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 = 2ms, Maximum = 6ms, Average = 3ms

-> Checking TTL

For the same topology as above

* Go to simulation mode
* click on PDU and send from 10.0.0.2 to 30.0.0.2
* Autocapture/play it until it succeeds.

-> click on the color code and observe the TTL

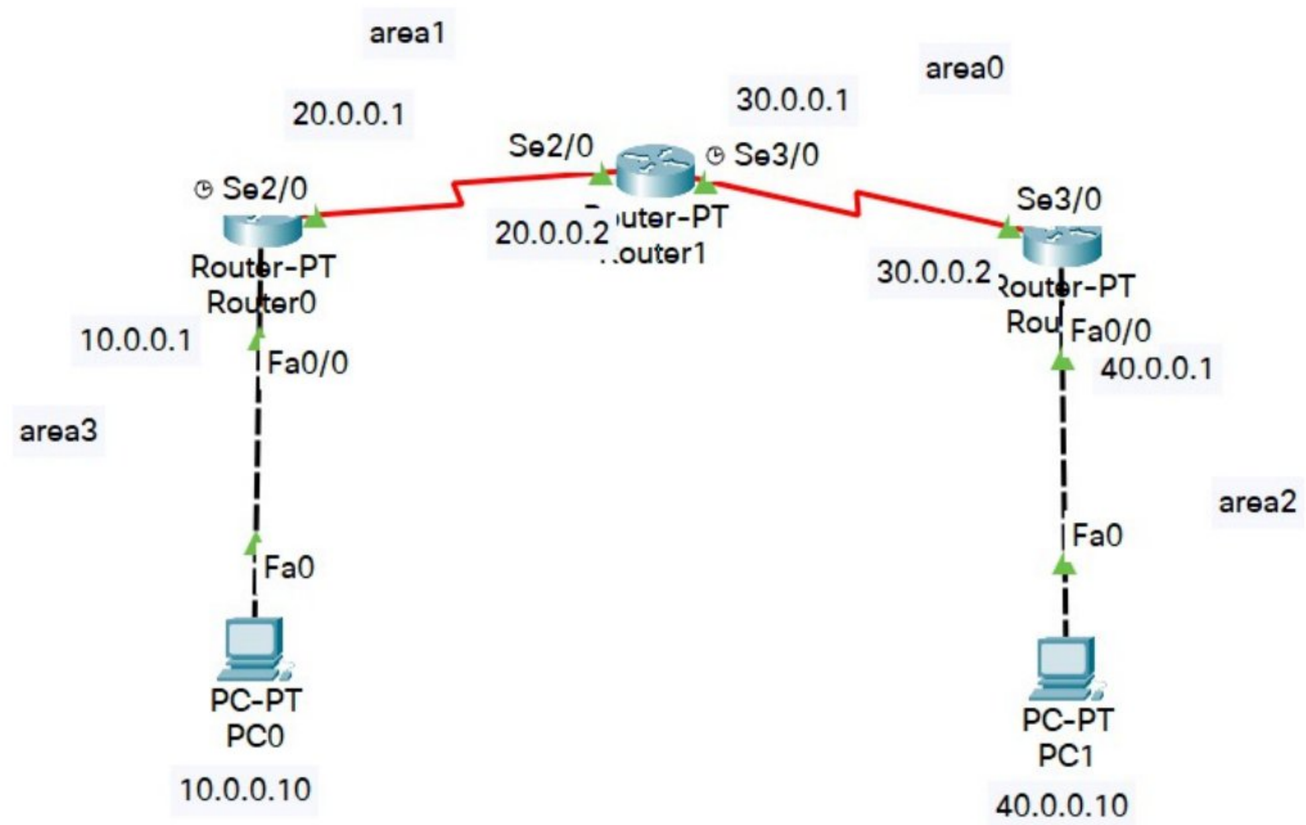
- If started at 255 before encountering a router
- When it reaches a router incom ip is 255 outgoing will be 254
- The same way for the other two routers & reaches the destination it is 252.

Scanned with OKEN Scanner

Program 6

i. Configure OSPF routing protocol

ii. Procedure along with the topology



iii. Screen shots/ output

Encapsulation:

Router0

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Fa0/0
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 Se2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#exit
Router(config)#
```


Scanned with OKEN Scanner

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to
up

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no dhutdown
                     ^
% Invalid input detected at '^' marker.

Router(config-if)#no shutdown
```

Router2

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
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
%IP-4-DUPADDR: Duplicate address 40.0.0.1 on FastEthernet0/0, sourced by
000D.BDDA.0123

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to
up
```


Scanned with OKEN Scanner

OSPF Routing Protocol

Router0

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 1.1.1.1
Router(config-router)#network 10.0.0.0 0.255.255.255 area 3
Router(config-router)#network 20.0.0.0 0.255.255.255 area 1
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#sho
00:27:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FULL, Loading Done
w 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
     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.0.0.0/8 is directly connected, Serial2/0
C       20.0.0.2/32 is directly connected, Serial2/0
O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:00:02, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:02, Serial2/0
```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
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 1
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

00:26:21: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on Serial3/0 from LOADING to FULL, Loading Done

00:27:18: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial2/0 from LOADING to FULL, Loading Done

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

     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.0.0.0/8 is directly connected, Serial2/0
C       20.0.0.1/32 is directly connected, Serial2/0
     30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial3/0
C       30.0.0.2/32 is directly connected, Serial3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:02:00, Serial3/0
```


Scanned with OKEN Scanner

Router2

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 3.3.3.3
Router(config-router)#network 40.0.0.0 0.255.255.255 area 2
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
00:26:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial3/0 from LOADING to FULL, Loading Done

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

O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:02:45, Serial3/0
     30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial3/0
C       30.0.0.1/32 is directly connected, Serial3/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
```

Configure Loopback address

Router0

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
up

Router(config-if)#ip address 172.16.1.252 255.255.0.0
Router(config-if)#no shutdown
```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
up

Router(config-if)#ip address 172.16.1.253 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#
```


Scanned with OKEN Scanner

Router2

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
up

Router(config-if)#ip address 172.16.1.254 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#
```

Create Virtual Link

Router0

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#area 1 virtual-link 2.2.2.2
Router(config-router)#
```

Router1

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
01:11:01: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

01:11:11: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

Router(config)#route
01:11:21: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0
r ospf 1
Router(config-router)#
01:11:31: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

Router(config-router)#area 1 v
01:11:41: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0
irtual-link 1.1.1.1
Router(config-router)#
01:11:56: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on OSPF_VL0 from LOADING to
FULL, Loading Done
```

Pinging

```
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=24ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=20ms 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 = 18ms, Maximum = 24ms, Average = 20ms
```

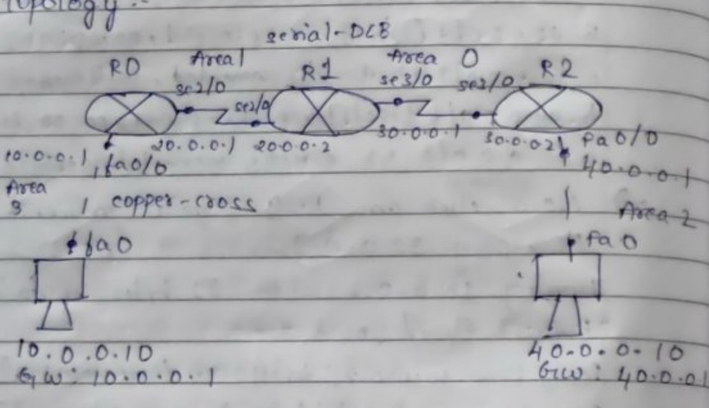

Scanned with OKEN Scanner

iv. Observation

29/11/24

How to configure OSPF Routing Protocol and connect Areas

Topology :-



Implementation :-

* Create a topology as shown above by drag & drop the generic routers and end-device
* Configure the end-devices with ip-address 10.0.0.10 , 40.0.0.10 .
* Between Router 0 & PC 0 the ip address is 10.0.0.1 and Between R0 & R1 it is 20.0.0.1 , & R1 & R0 it is 20.0.0.2 , & R1 & R2 it is 30.0.0.1 & R2 & R1 it is 30.0.0.2 , & R2 & PC1 it is 40.0.0.1
* Give the gateway for PC 0 & PC 1 as 10.0.0.1 & 40.0.0.1 respectively .
* In Router R1 ,

R1(config) # interface fastethernet 0/0
R1(conifg) # ip address 10.0.0.1 255.0.0.0
R1(config-if) # no shutdown
R1(config-if) # exit
R1(config) # interface serial 0/0
R1(config-if) # ip address 20.0.0.1 255.0.0.0
R1(config-if) # encapsulation ppp
R1(config-if) # clock rate 64000
R1(config-if) # no shutdown
R1(config-if) # exit

In Router R2 ,

R2(config) # interface serial 2/0
R2(config-if) # ip address 20.0.0.2 255.0.0.0
R2(config-if) # encapsulation ppp
R2(config-if) # no shutdown
R2(config-if) # exit
R2(config) # interface serial 3/0
R2(config-if) # ip address 30.0.0.1 255.0.0.0
R2(config-if) # encapsulation ppp
R2(config-if) # clock rate 64000
R2(config-if) # no shutdown
R2(config-if) # exit

In Router R3 ,

R3(config) # interface serial 3/0
R3(config-if) # ip address 30.0.0.2 255.0.0.0
R3(config-if) # encapsulation ppp
R3(config-if) # no shutdown
R3(config-if) # exit
R3(config) # interface fastethernet 0/0
R3(config-if) # ip address 40.0.0.1 255.0.0.0
R3(config-if) # no shutdown
R3(config-if) # exit.

* Now, Enable ip routing by configuring ospf protocols in all routers .

In Router R1 ,

R1(config) # router ospf 1
R1(config-router) # router-id 1.1.1.1
R1(config-router) # network 10.0.0.0 0.255.255.255 area 3

Scanned with OKEN Scanner

DATE: PAGE:

R1(config-router)#network 20.0.0.0 0.255.255.255 area 1
R1(config-router)#exit
In Router R2,
R2(config)#router ospf 1
R2(config-router)#router-id 2.2.2.2
R2(config-router)#network 20.0.0.0 0.255.255.255 area 1
R2(config-router)#network 30.0.0.0 0.255.255.255 area 0
R2(config-router)#exit
In Router R3,
R3(config)#router ospf 1
R3(config-router)#router-id 3.3.3.3
R3(config-router)#~~router~~ network 30.0.0.0 0.255.255.255 area 0
R3(config-router)#network 40.0.0.0 0.255.255.255 area 2
R3(config-router)#exit
* Now checking routing table of R1, R1#show ip route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, serial 2/0
C 20.0.0.2/32 is directly connected, serial 2/0
O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:02:08, serial 2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:21, serial 2/0
* Here R2, knows Area 0. Network 20.0.0.0 connected to R2 from R1, so R1 learns networks through this network
* There must be one interface, so configure loopback address to routers
R1(config)#interface loopback 0
R1(config-if)#ip add 172.16.1.252 255.255.0.0

DATE: PAGE:

R1(config-if)#no shutdown
R2(config-if)#interface loopback 0
R2(config-if)#ip add 172.16.1.253 255.255.0.0
R2(config-if)#no shutdown
R3(config)interface loopback 0
R3(config-if)#ip add 172.16.1.254 255.255.0.0
R3(config-if)#no shutdown
* Now check routing table of R3,
R3#show ip route
Gateway of last resort is not set
O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:01:38, serial 3/0
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:42:38, serial 3/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, serial 3/0
C 30.0.0.1/32 is directly connected, serial 3/0
C 40.0.0.0/8 is directly connected, FastEthernet 0/0
C 172.16.0.0/16 is directly connected, Loopback 0
* Create virtual link between R1, R2 by this we create a virtual link to connect area 2 to area 0
In Router R1,
R1(config)#router ospf 1
R1(config-router)#area 1 virtual-link 2.2.2.2
R1(config-router)
In Router R2,
R2(config-router)# area 1 virtual-link 1.1.1.1
R2(config-router)#exit
* R2 and R3 get updates about Area 2, and checking table of R3.
* check the connectivity between host 10.0.0.10 to

DATE: PAGE:

40.0.0.10, that typing PC> ping 40.0.0.10 is command prompt.

Observation:-
Pinging 40.0.0.10 with 32 bytes of data!
Reply from 40.0.0.10: bytes=32 time=12ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=5ms 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=4, Lost=0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum=5ms, Maximum=12ms, Average=8ms

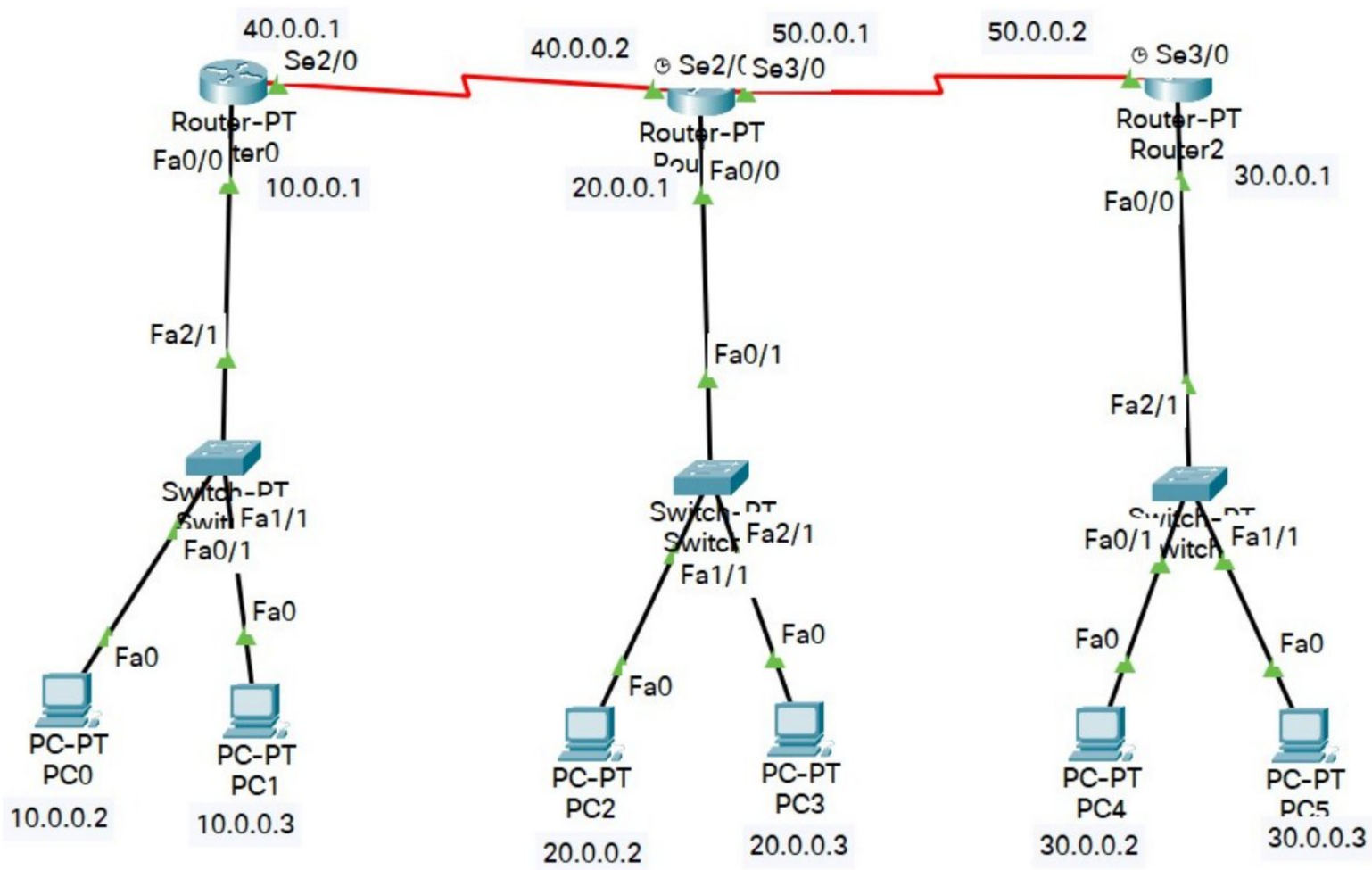
20/12/24

Scanned with OKEN Scanner

Program 7

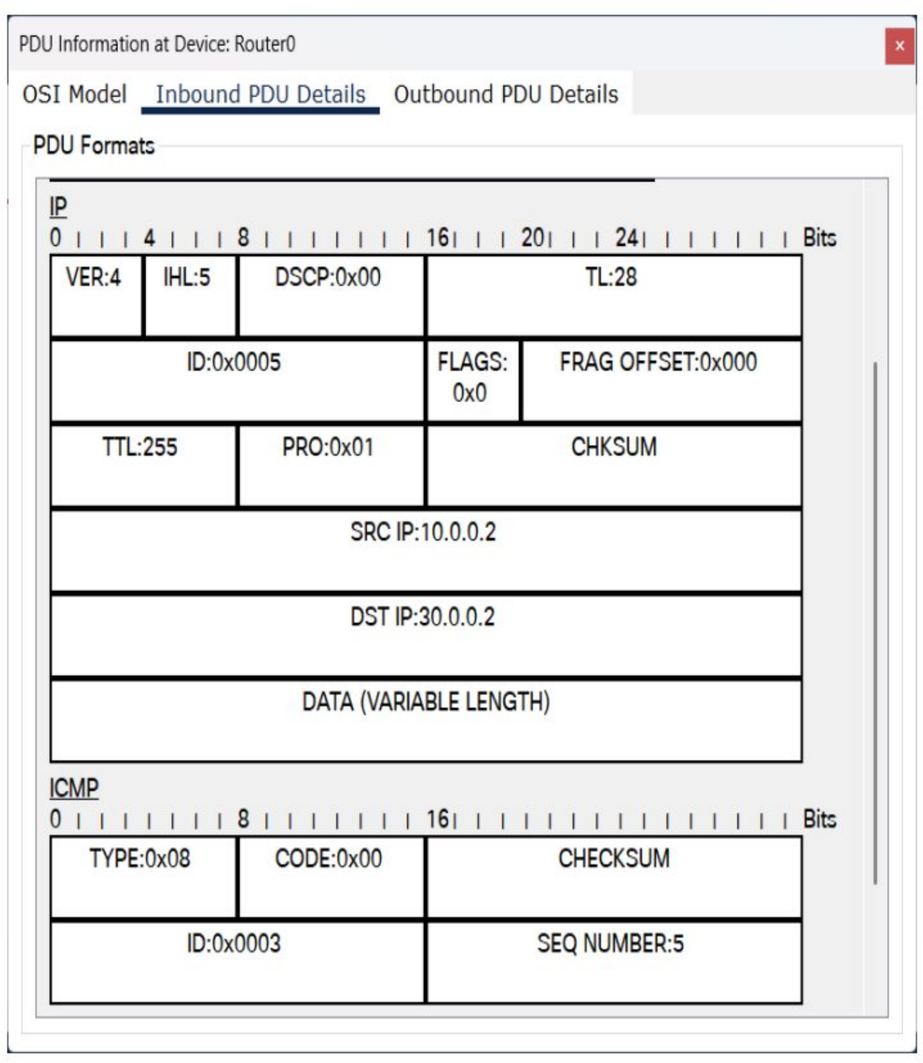
i. Demonstrate the TTL/ Life of a Packet

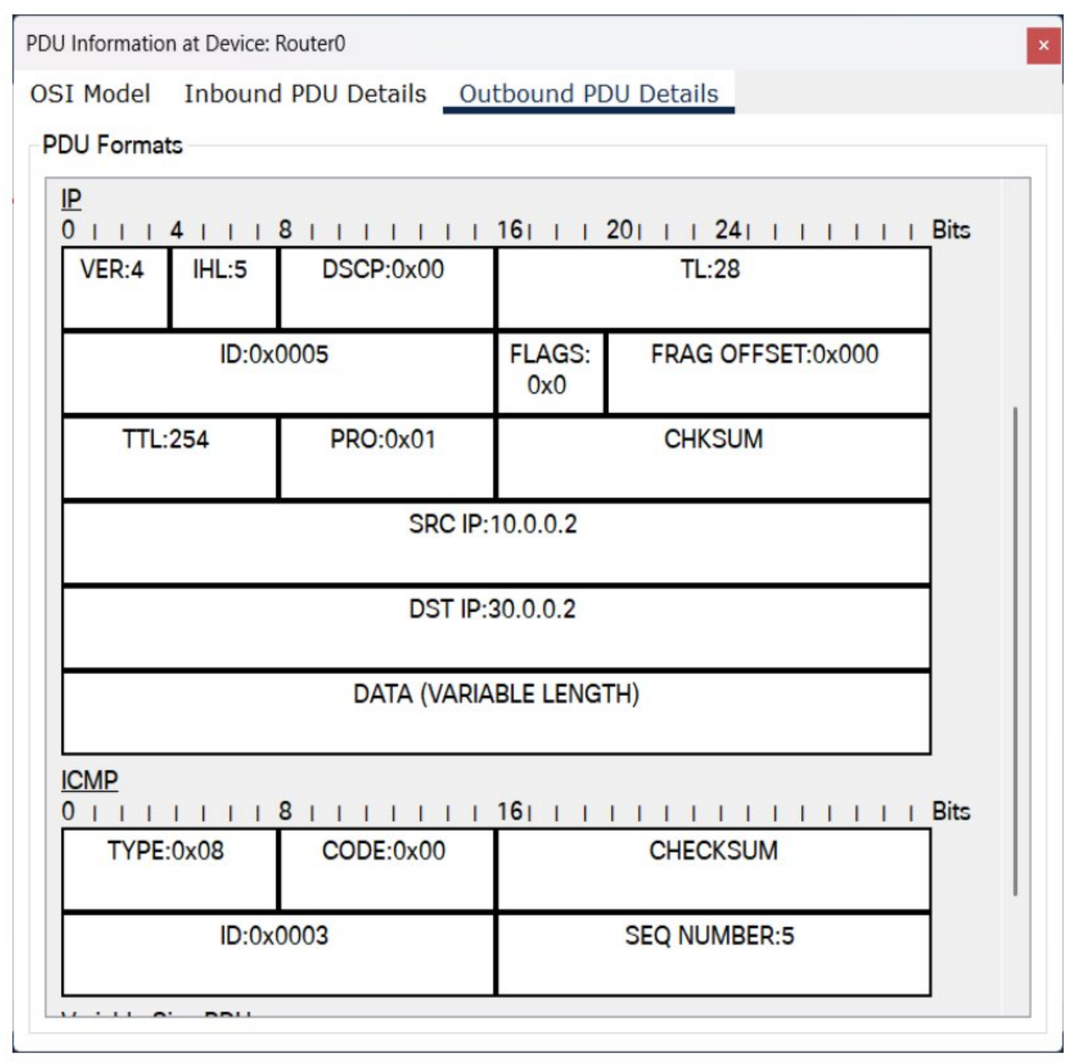
ii. Procedure along with the topology



iii. Screen shots/ output

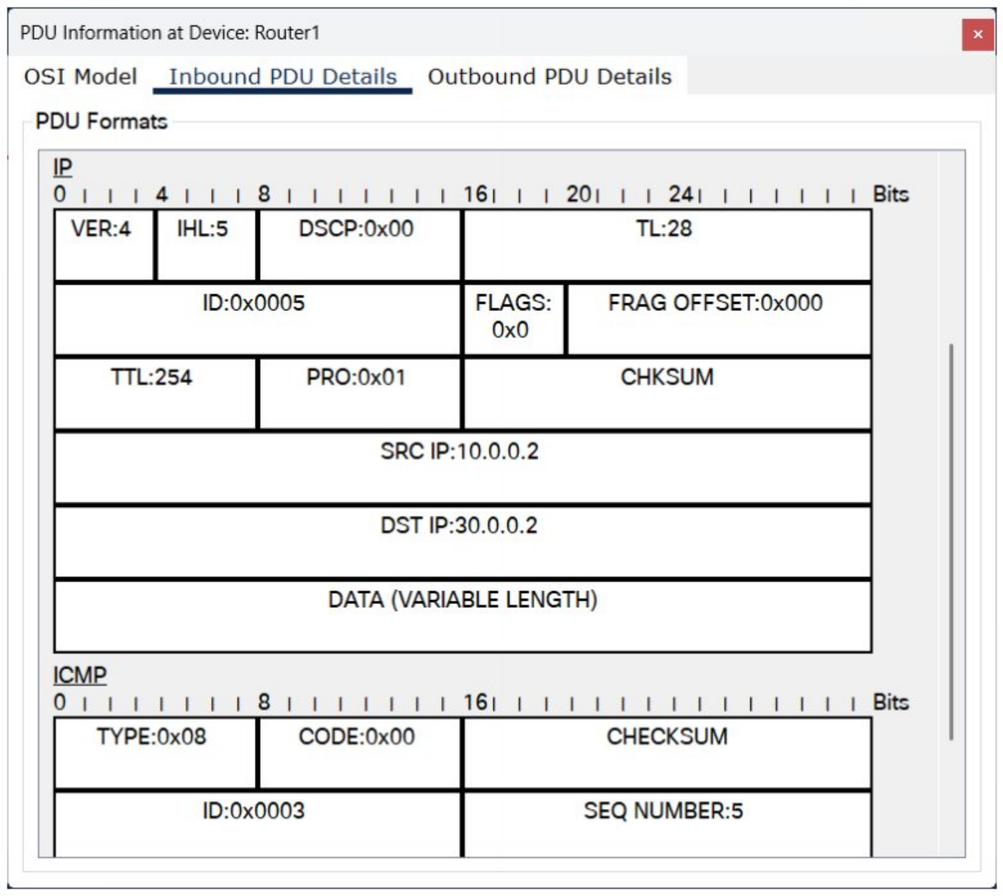
Packet at Router0

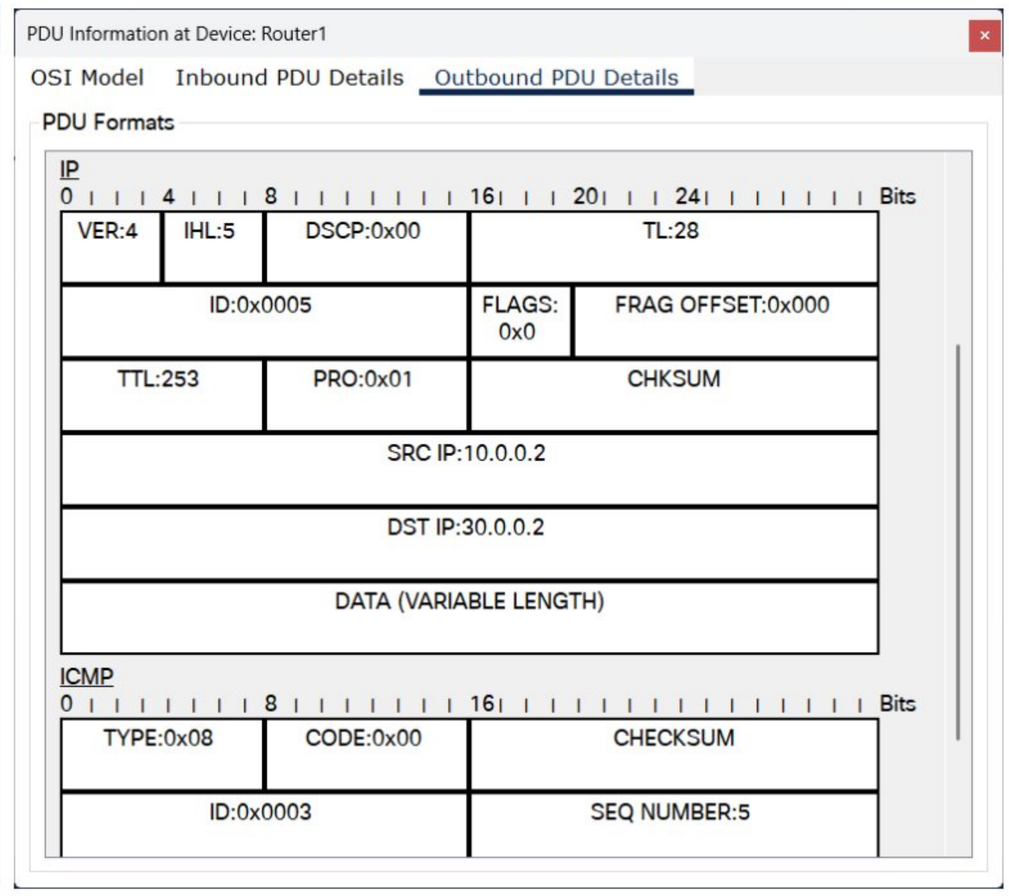




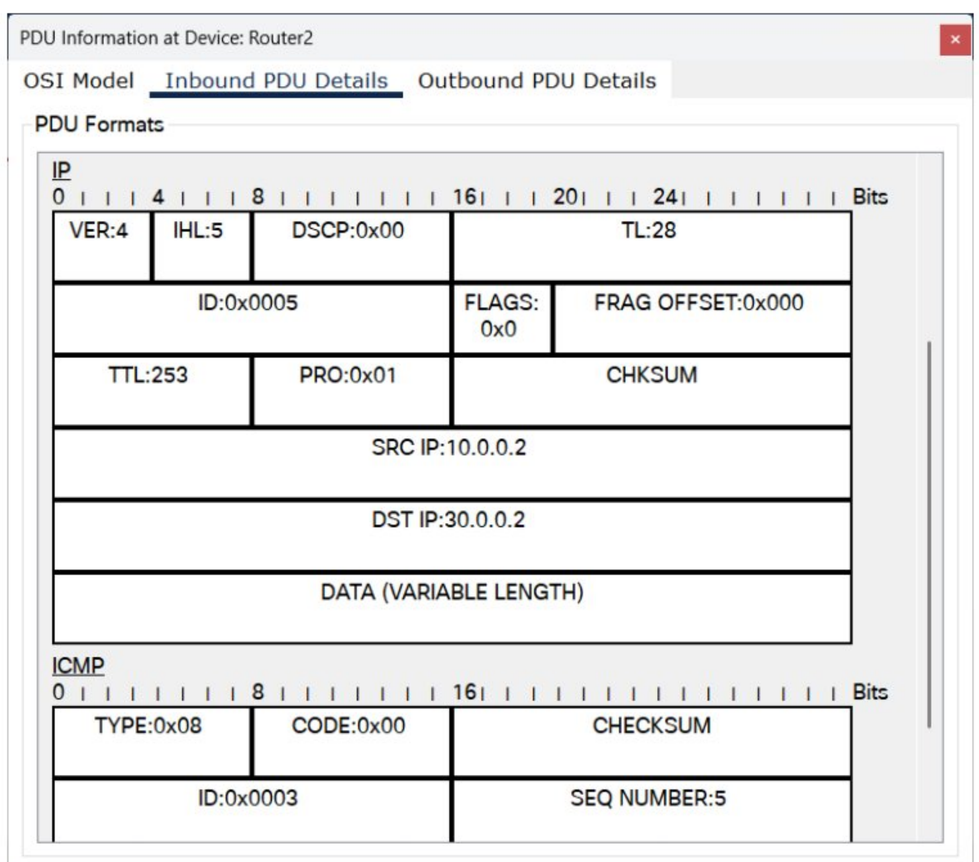
Scanned with OKEN Scanner

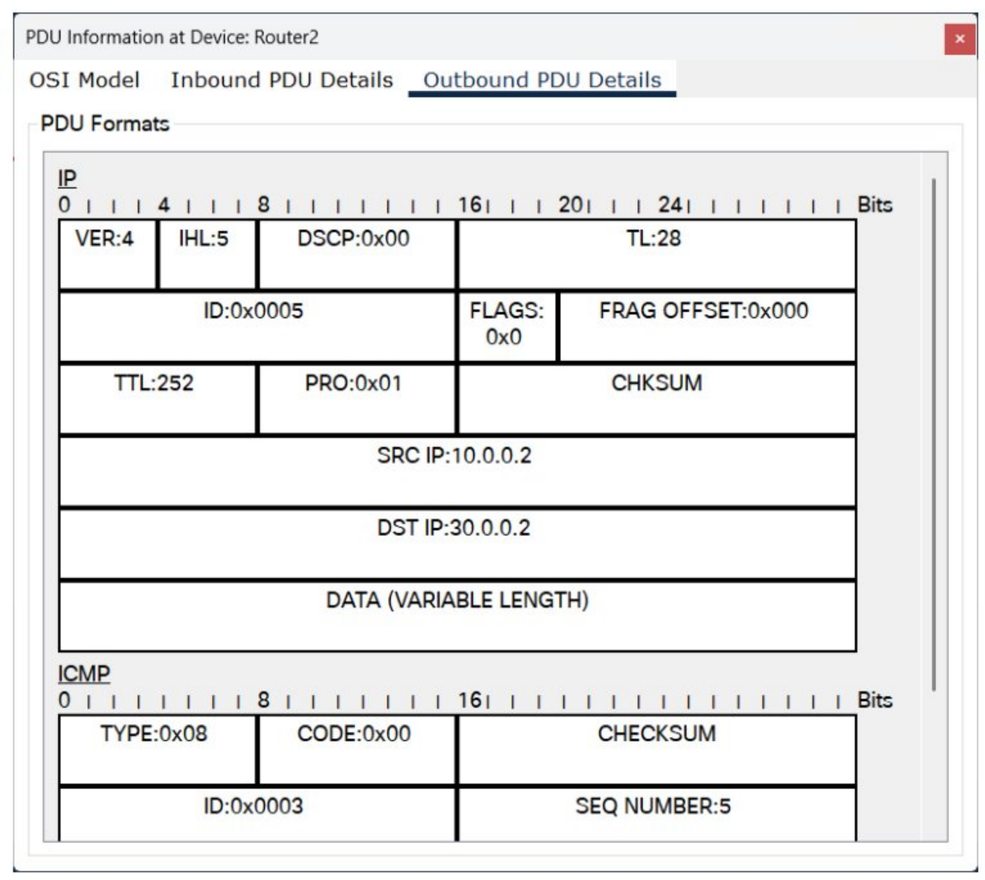
Packet at Router1





Packet at Router2





iv. Observation

Scanned with OKEN Scanner

DATE: PAGE:

-> Now assign the gateways for all the PC's, for PC0 & PC1 it is 10.0.0.1 for PC2 & PC3 is 20.0.0.1, PC4 & PC5 it is 30.0.0.1

-> Now click on R0, go to CLI, type enable, type config-t, and type router rip,

(For R0) # network 10.0.0.0
network 40.0.0.0

(For R1) # network 20.0.0.0
network 40.0.0.0
network 50.0.0.0

(For R2) # network 30.0.0.0
network 50.0.0.0

-> Now type show ip route for all the the routers, Router # show ip route.

Observation:-

-> The output after show ip route is as follows:-

For R0 :-

Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
R 20.0.0.0/8 [120/1] via 40.0.0.2, 00:00:13, serial2/0
R 30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:13, serial2/0
C 40.0.0.0/8 is directly connected, serial2/0
R 50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:13, serial2/0

For R1 :-

Gateway of last resort is not set
R 10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:06, serial2/0
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
R 30.0.0.0/8 [120/1] via 50.0.0.2, 00:00:09, serial3/0
C 40.0.0.0/8 is directly connected, serial2/0

DATE: PAGE:

C 50.0.0.0/8 is directly connected, serial3/0

For R2 :-

Gateway of last resort is not set
R 10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:24, serial3/0
R 20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:24, serial3/0
C 30.0.0.0/8 is directly connected, FastEthernet 0/0
R 40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:24, serial3/0
C 50.0.0.0/8 is directly connected, serial3/0

-> Now click on PC0, go to command prompt
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=6ms TTL=125
Reply from 30.0.0.2: bytes=32 time=2ms TTL=125
Reply from 30.0.0.2: bytes=32 time=2ms 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 = 2ms, Maximum = 6ms, Average = 3ms

-> Checking TTL

For the same topology as above
* Go to simulation mode
* click on PDU and send from 10.0.0.2 to 30.0.0.2
* Autocapture/play it until it succeeds.
-> click on the color code and observe the TTL

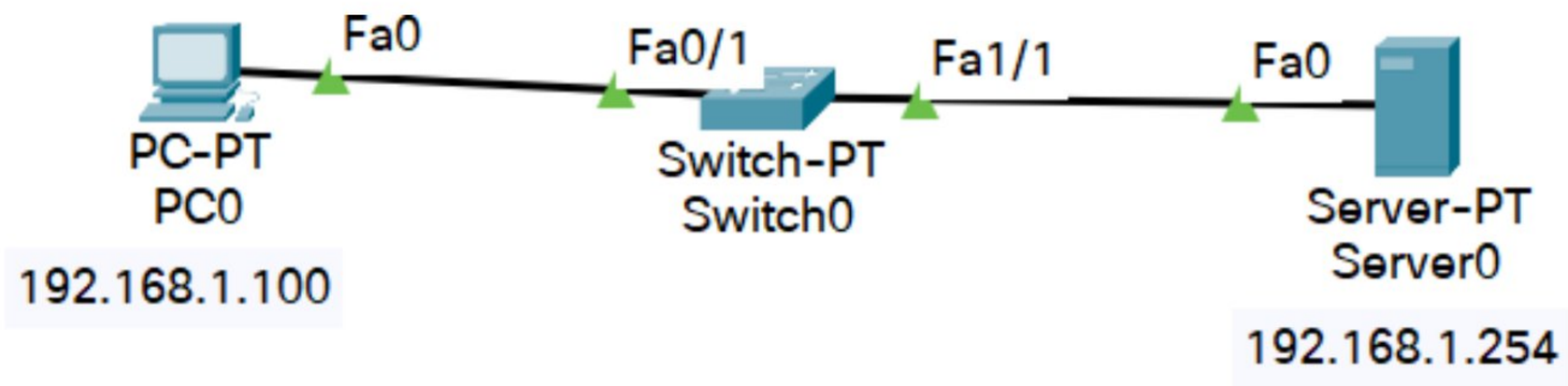
- If started at 255 before encountering a router
- When it reaches a router incom ip is 255 outgoing will be 254
- The same way for the other two routers & reaches the destination it is 252.

Scanned with OKEN Scanner

Program 8

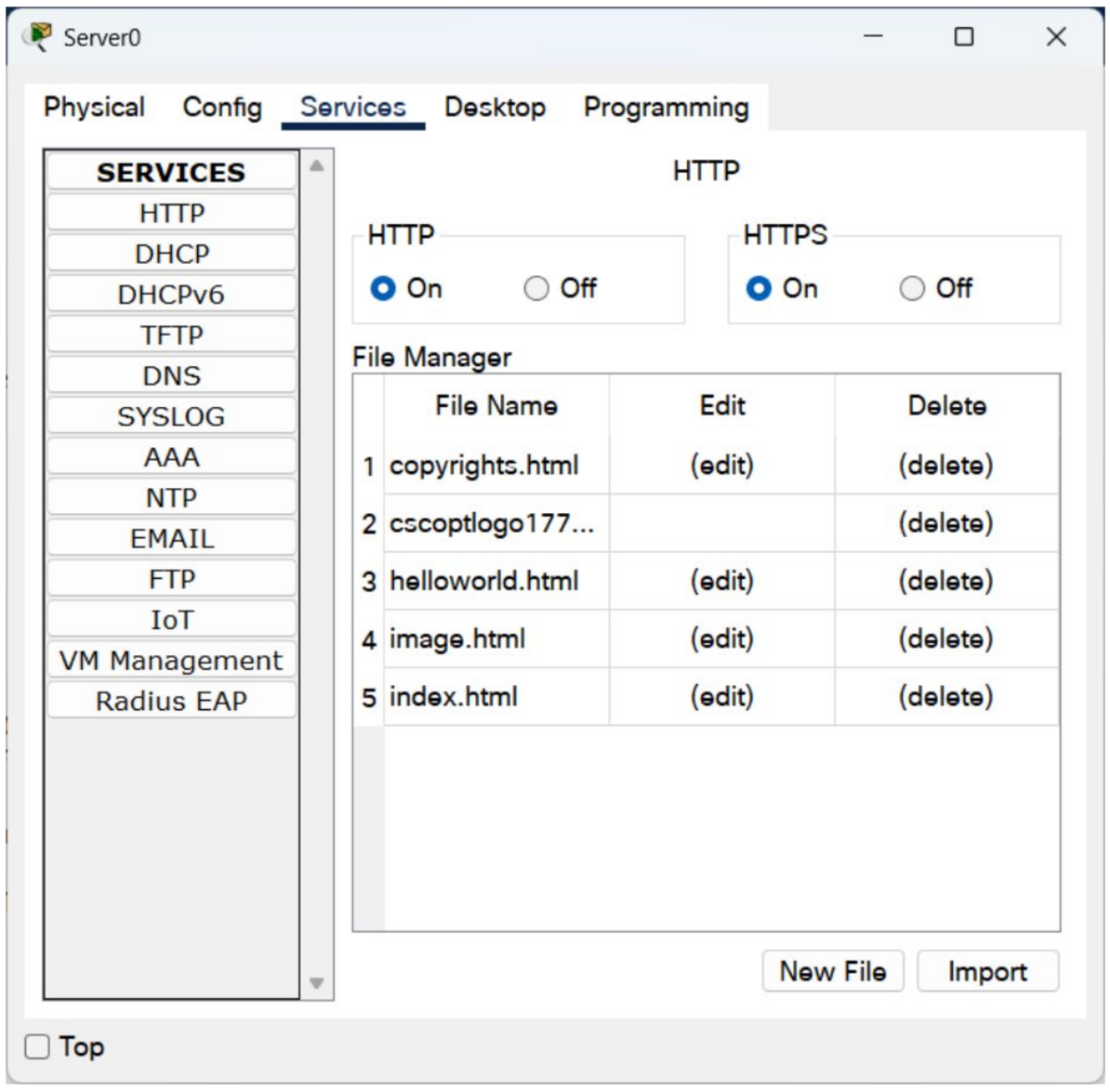
i. Configure Web Server, DNS within a LAN.

ii. Procedure along with the topology



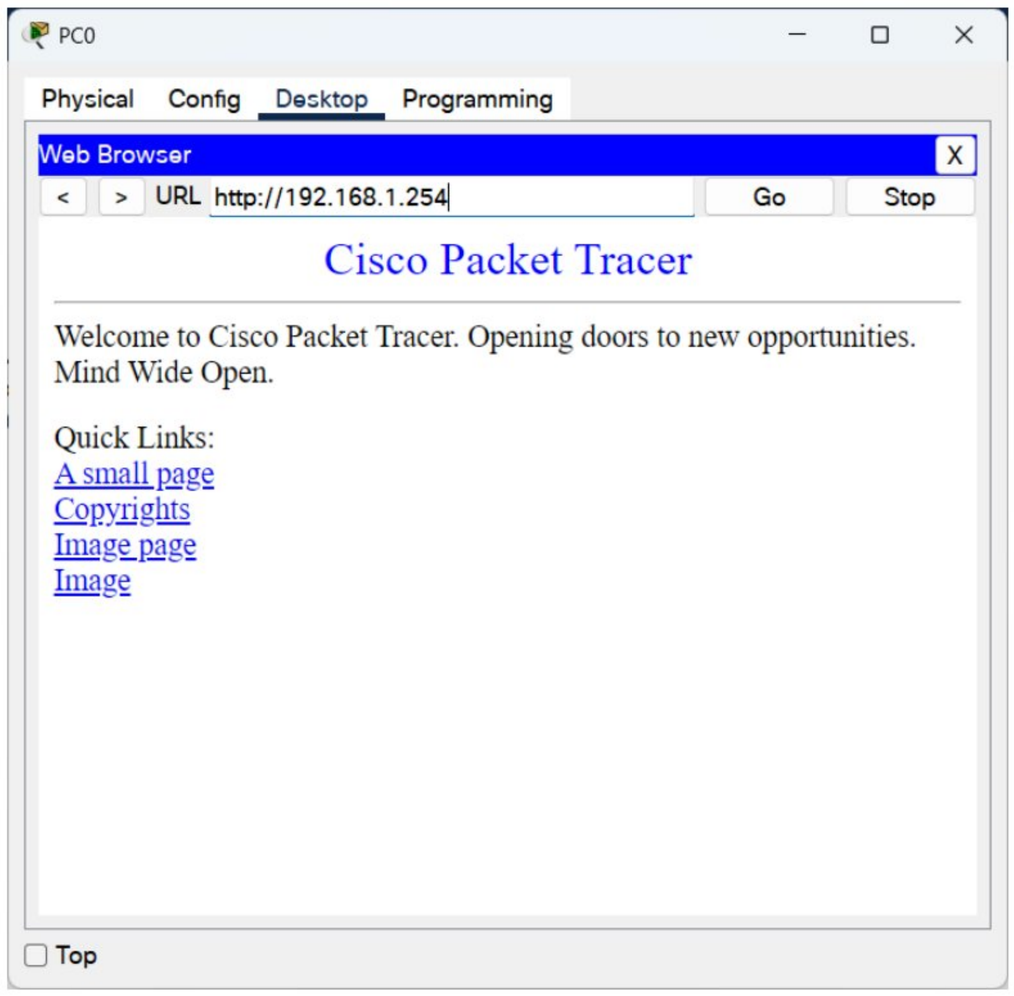
iii. Screen shots/ output

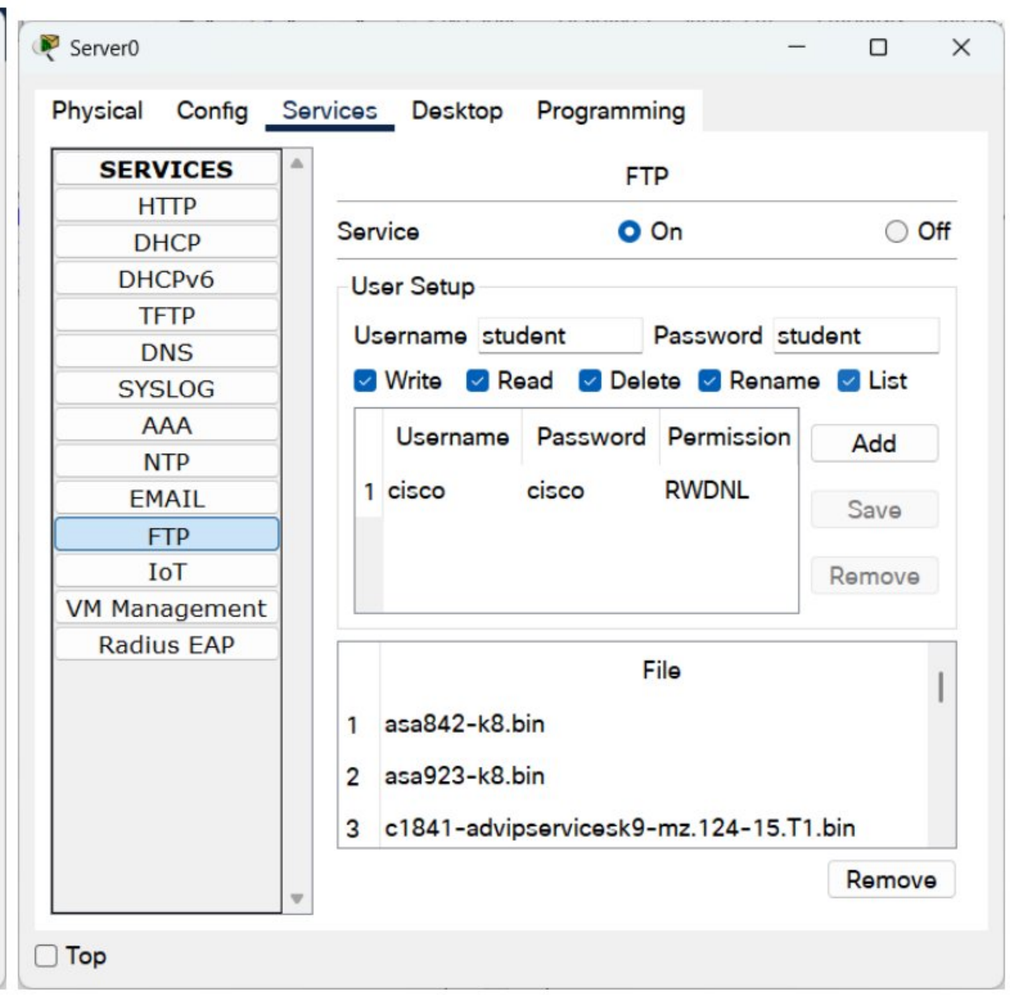
Server's services

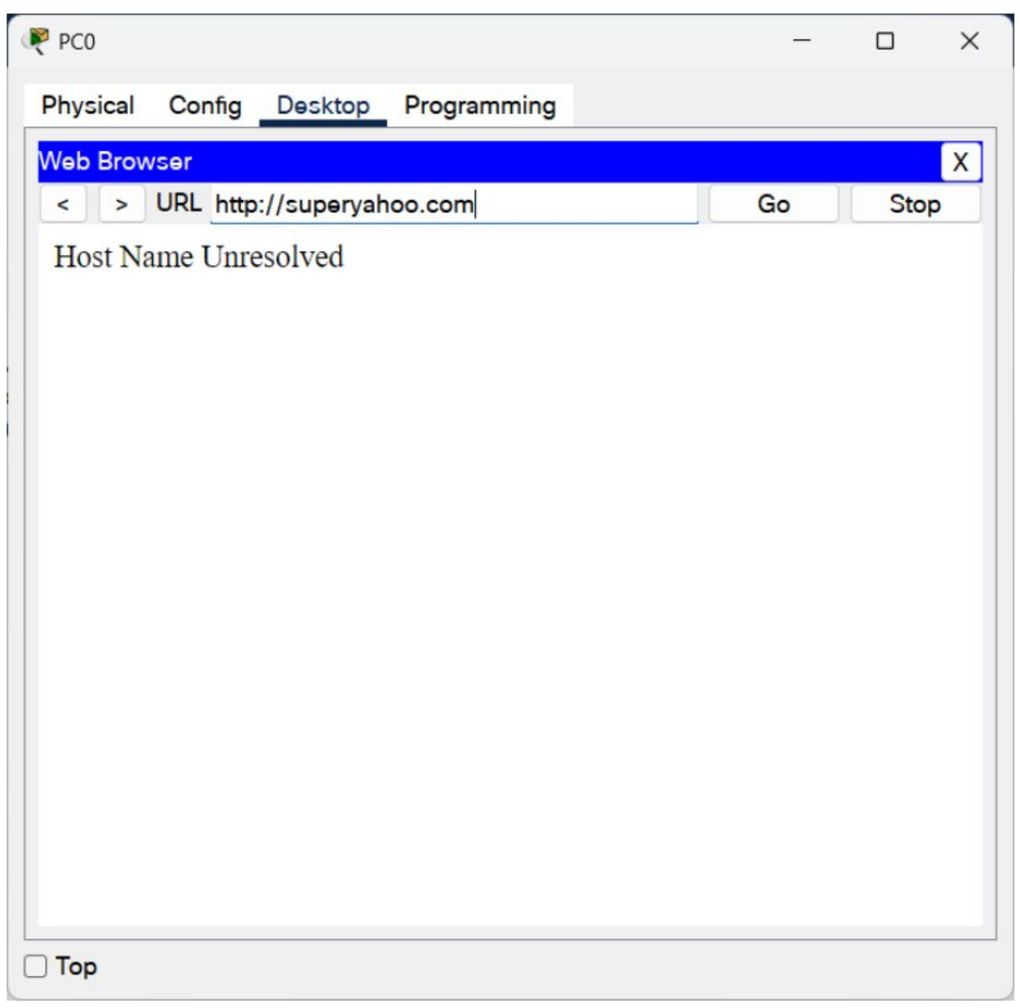


PC's Web Browser

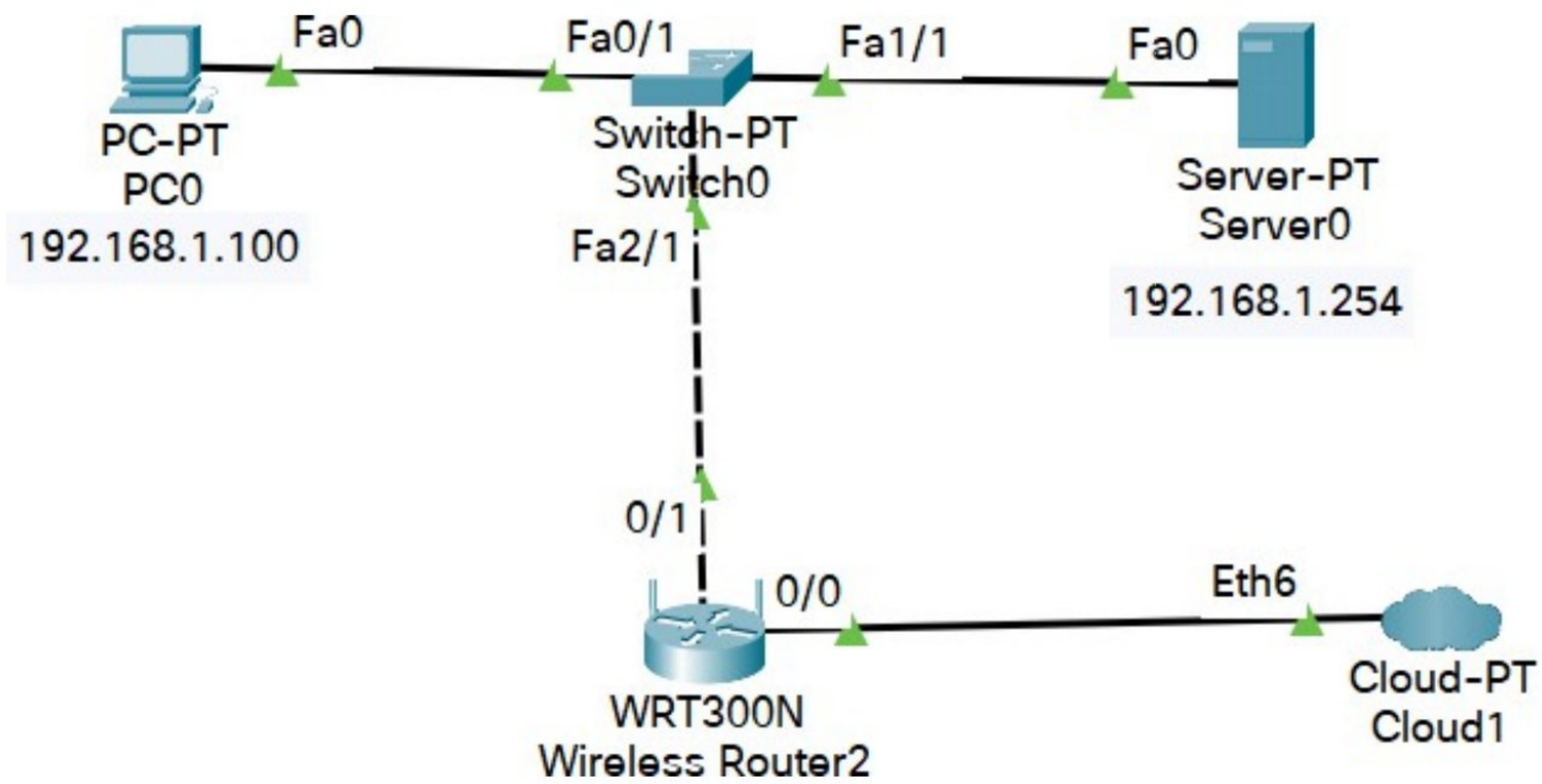
Scanned with OKEN Scanner

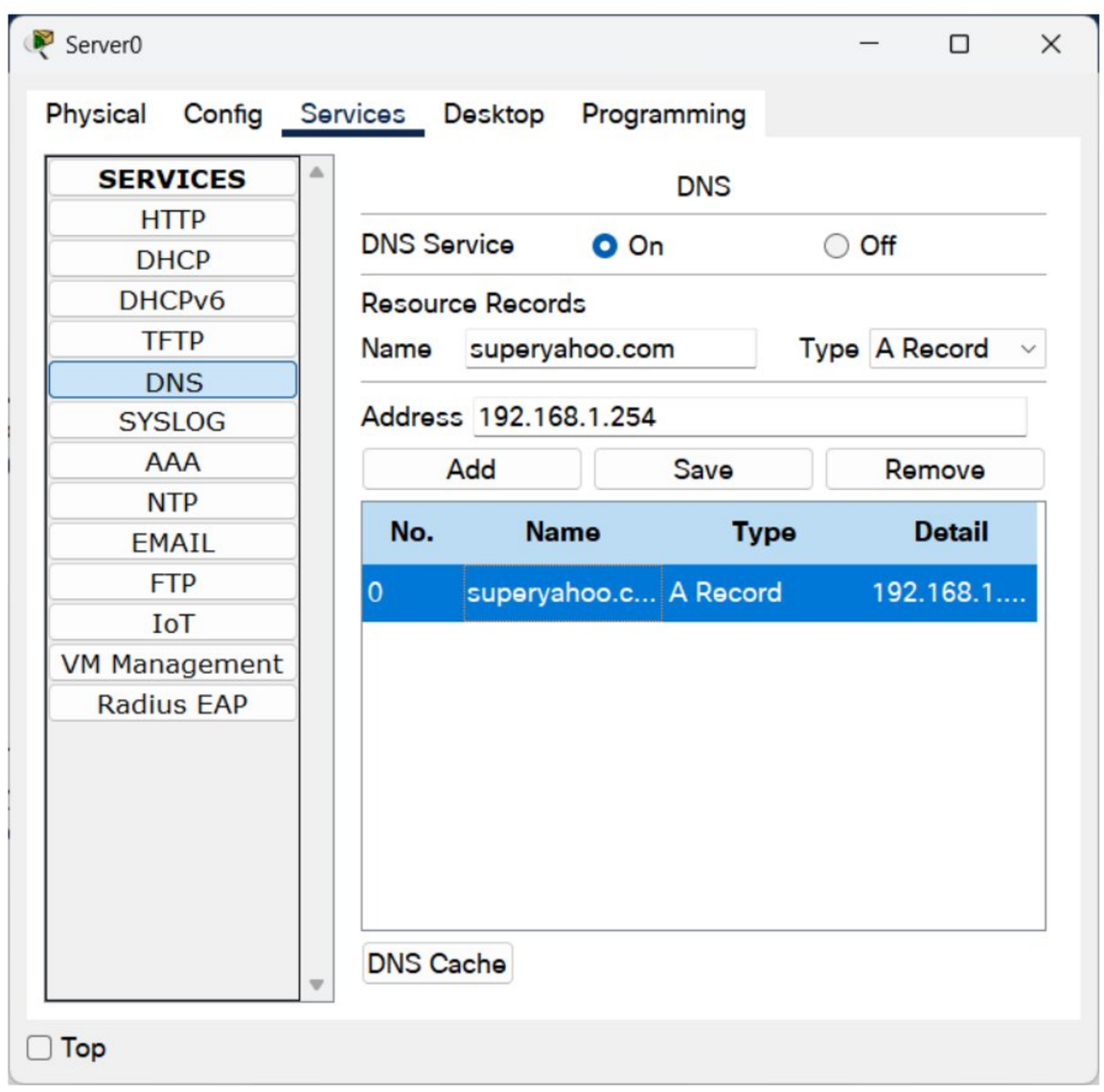






Scanned with OKEN Scanner





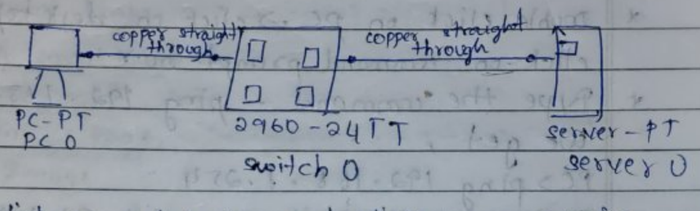
Scanned with OKEN Scanner

iv. Observation

DATE: PAGE:

(2) How to demonstrate WEB server and DNS using Packet Tracer

Topology:



* click on end devices and click on a generic pc and place it in logical work space.
* click on switches and then click a 2960 switch and place it.
* click on end devices and click on generic server and place it.
* connect connect PC to switch use a straight through cable, click on pc and click fast Ethernet, drag cable over to switch and click on fast Ethernet 01
* To connect the server to switch is also a straight through cable so select the cable and drag it from the server (fast Ethernet) to switch (fast Ethernet 02
* click on server label and type IP address 192.168.1.254, and click on PC label & type 192.168.1.100
* Now, go to server and double click and click on config tab, click on interface - fastEthernet on left hand side, make sure the port status is on.
* where it says static ip address type in the following 192.168.1.254. then place cursor on subnet mask box & will automatically appear.

DATE: PAGE:

* To set PC ip address click on PC, go to config tab, click on fastEthernet and make sure port is turned on under static ip address enter 192.168.1.100.
* Now ping from the pc to the server to make sure that we have a connection.
* Double click on pc -> click on desktop -> click on command prompt box.
* Type the command - ping 192.168.1.254

We get,

```
PC> ping 192.168.1.254
Pinging 192.168.1.254 with 32 bytes of data:
Reply from 192.168.1.254: bytes=32 time=48ms TTL=128
Reply from 192.168.1.254: bytes=32 time=4ms TTL=128
Reply from 192.168.1.254: bytes=32 time=4ms TTL=128
Reply from 192.168.1.254: bytes=32 time=9ms TTL=128
Ping statistics for 192.168.1.254:
Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip time in milli-seconds:
Minimum=4ms, Maximum=178ms, Average=48ms
```

* Double click on the pc, to test our web server click on the desktop tab and then click on the web browser box
* In the virtual browser that appears type in the ip address of the server and click go.

Result:

DATE: PAGE:

192.168.1.100

[Physical | Config | Desktop | Software/services]

Web Browser

URL http://192.168.1.254 | Go | Stop

cisco Packet Tracer

Quick Links:

A small page

Copyright

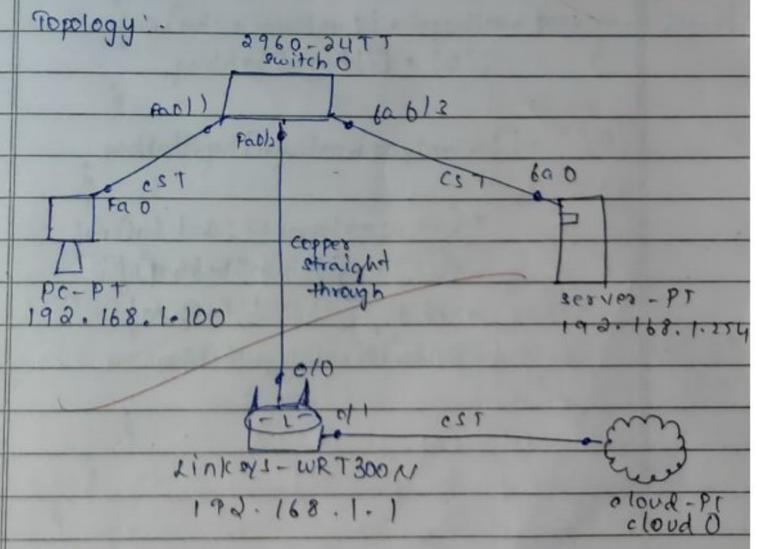
Image page

Image

* You we are connected to the FTP server.

Part-II

Topology:



* Create a LAN using a generic PC, server, 2960-24TT switch, WRT300N wireless device, a cloud using copper straight through

config PC -> 192.168.1.100 - fastethernet 0/3

server -> 192.168.1.254 - fastethernet 0/3

WRT300N -> 192.168.1.1 - Fastethernet 0/2

DATE: PAGE:

* switch to WRT300N -> Ethernet 1

WRT300N to cloud -> Ethernet 2

* Server -> Gateway -> 192.168.1.1

↳ Web Server -> http -> cisco packet tracer - html code

PC -> Gateway -> 192.168.1.1

* server -> dns -> dns service -> ON

↳ Resource Records -> Name - superyahoo.com

-> address - 192.168.1.254

-> add

* PC -> Desktop -> IP configuration -> DNS server - 192.168.1.254

↳ web browser - URL: superyahoo.com

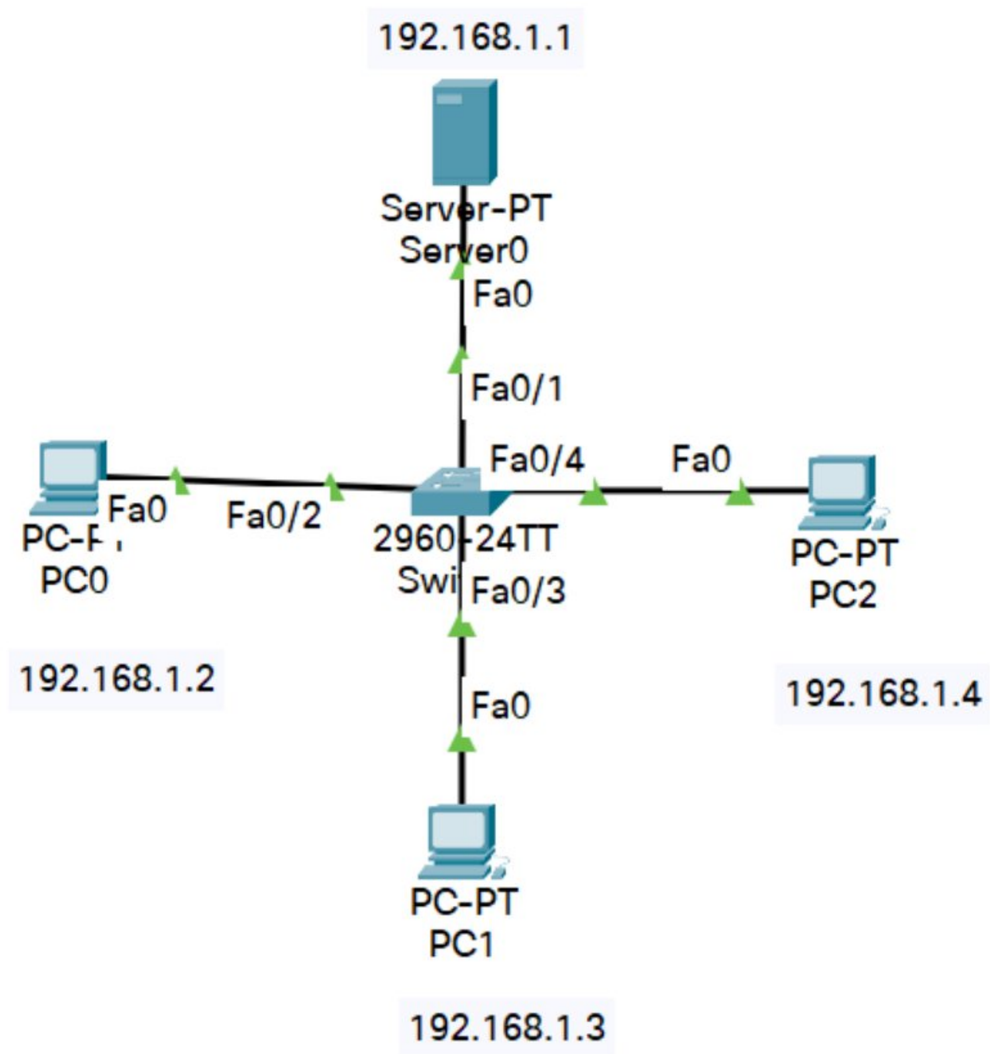
↳ cisco packet tracer web page displayed

Scanned with OKEN Scanner

Program 9

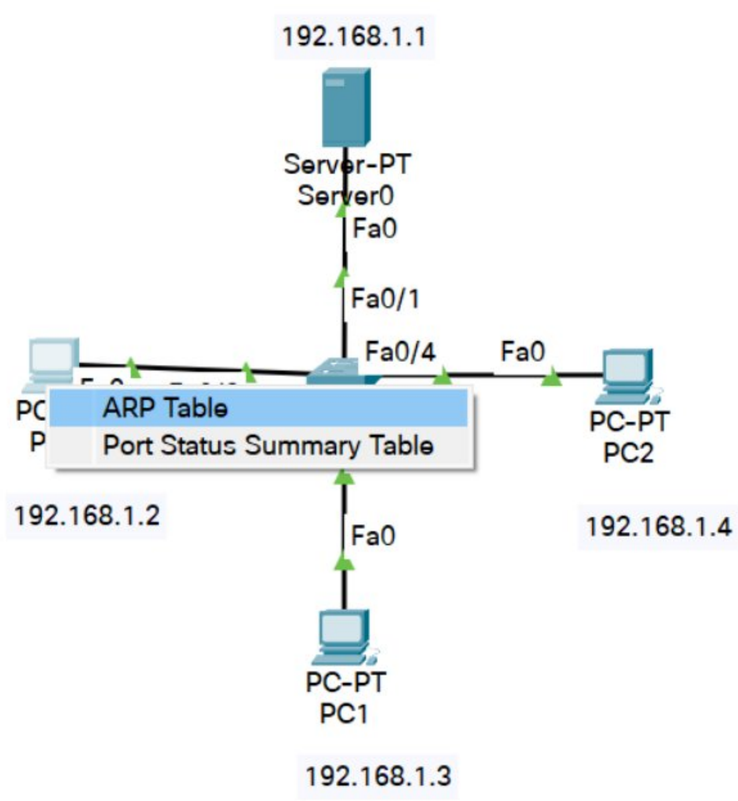
i. To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

ii. Procedure along with the topology



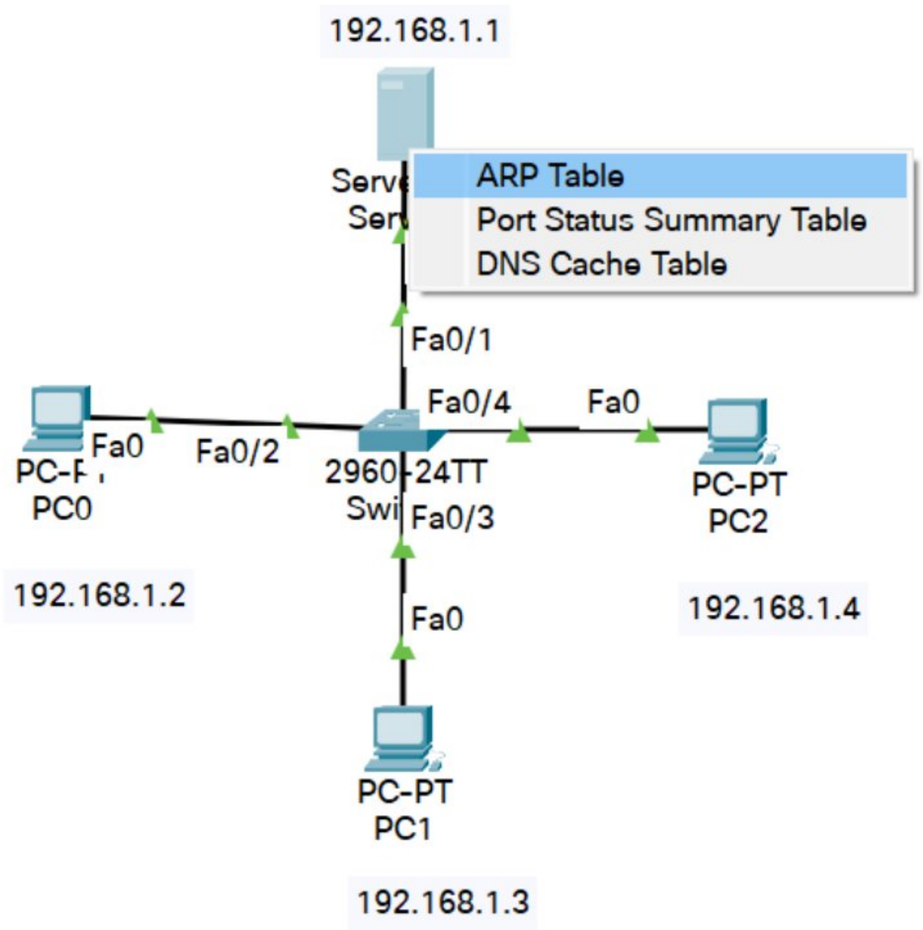
iii. Screen shots/ output

ARP Table of PC



ARP Table of Server

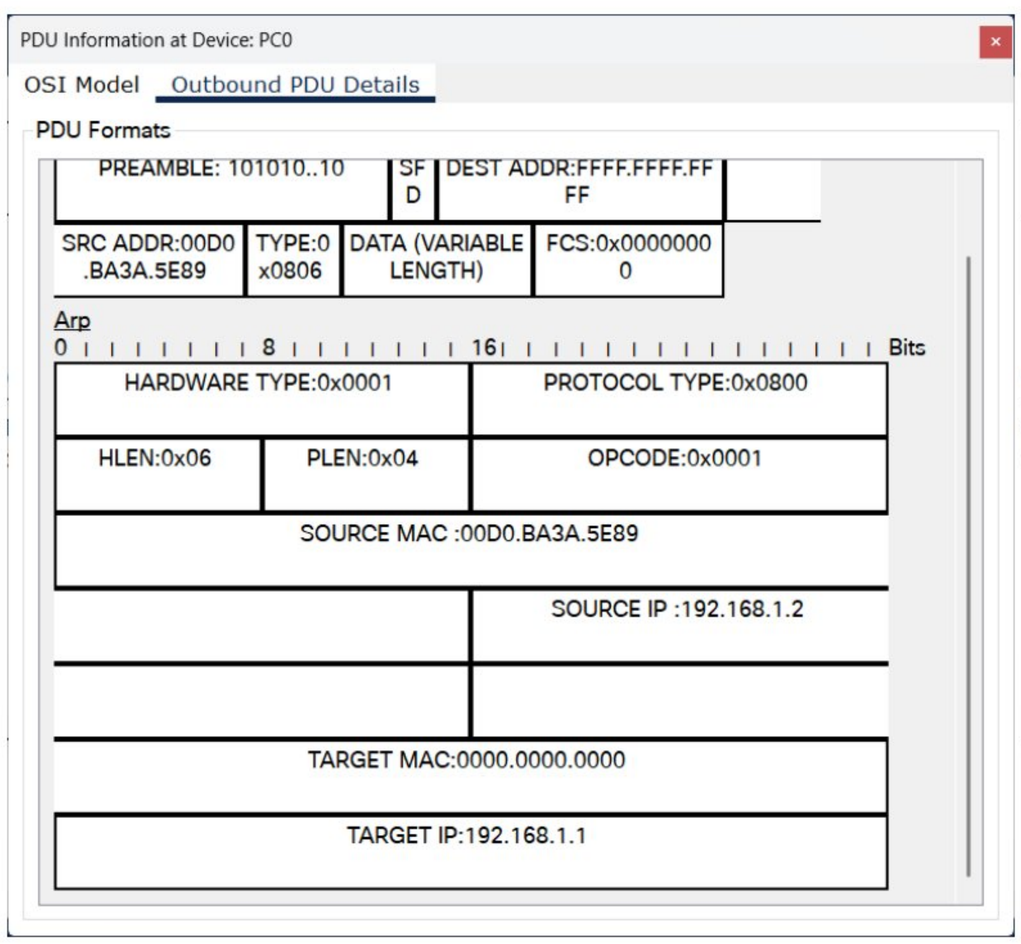
Scanned with OKEN Scanner

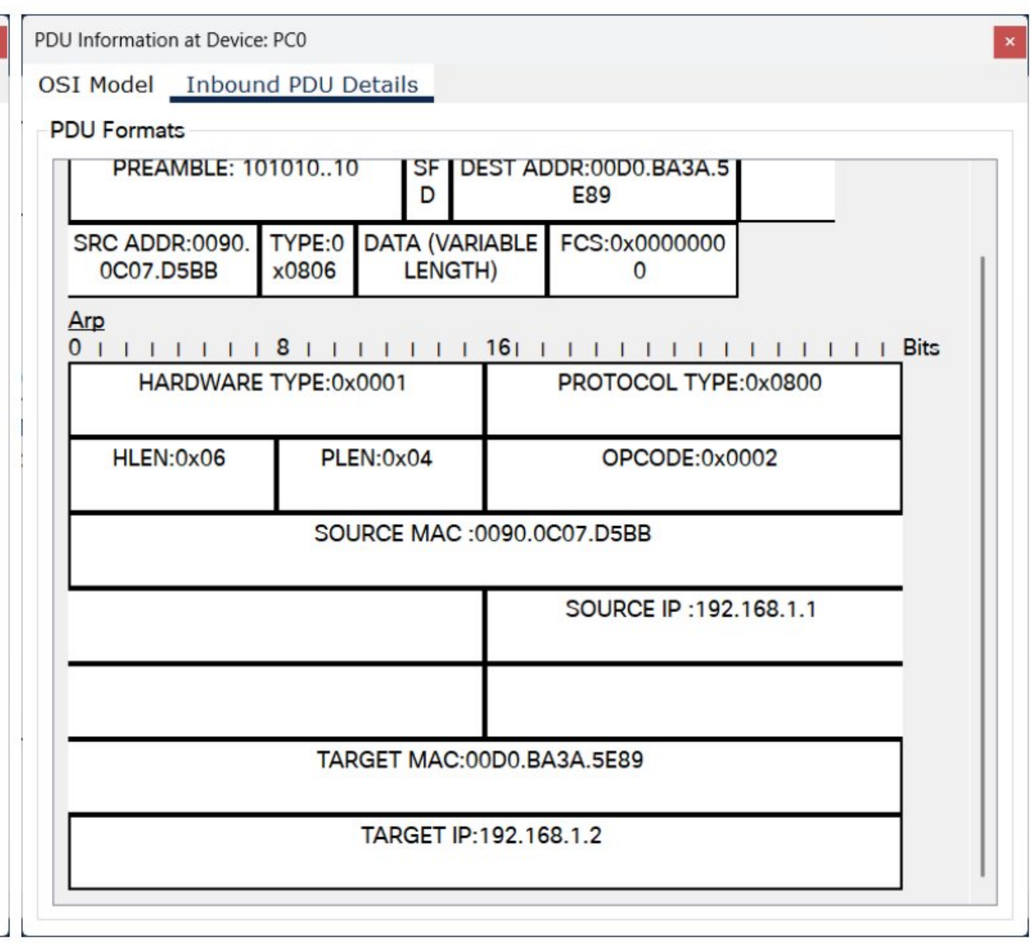


Command at PC

```
Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
C:\>
```

Pinging in Simulation Mode





Scanned with OKEN Scanner

iv. Observation

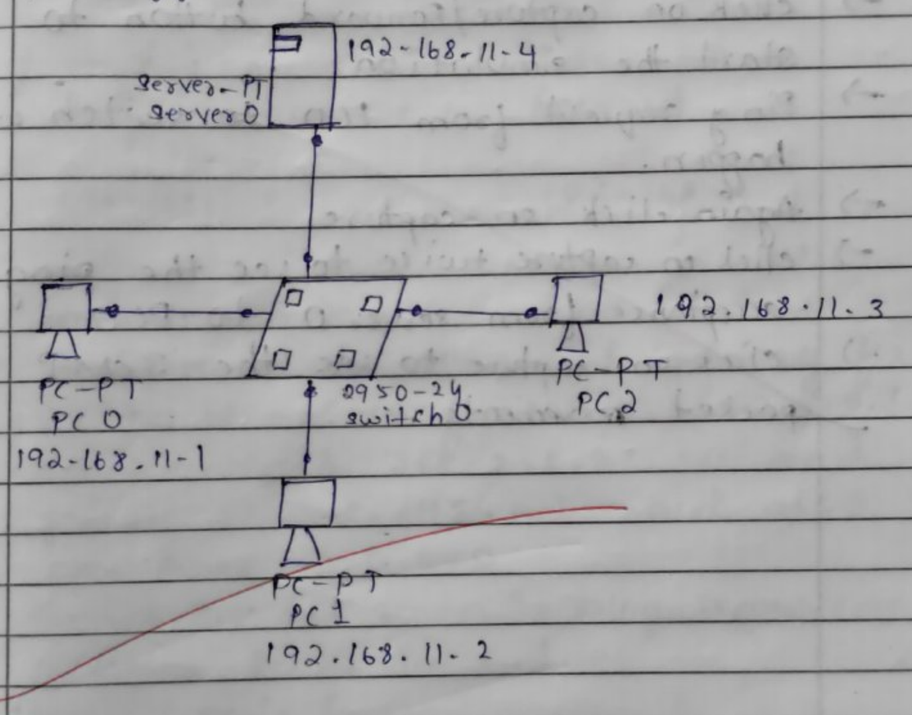
20/12/24

Date __/__/__
Page

ARP

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

Topology :-



Implementation:

-> configure the PC's with IP addresses 192.168.11.1, 11.2 and 11.3
-> configure server with IP address 192.168.11.4
-> Go to simulation panel, click on inspect and right click on PC0
-> We don't see any entries in ARP table
-> Now click on PC0 and go to command prompt and type arp -a
   => No entries found in ARP
-> Ping from PC0 to server
-> Ping 192.168.11.4

Date __/__/__
Page

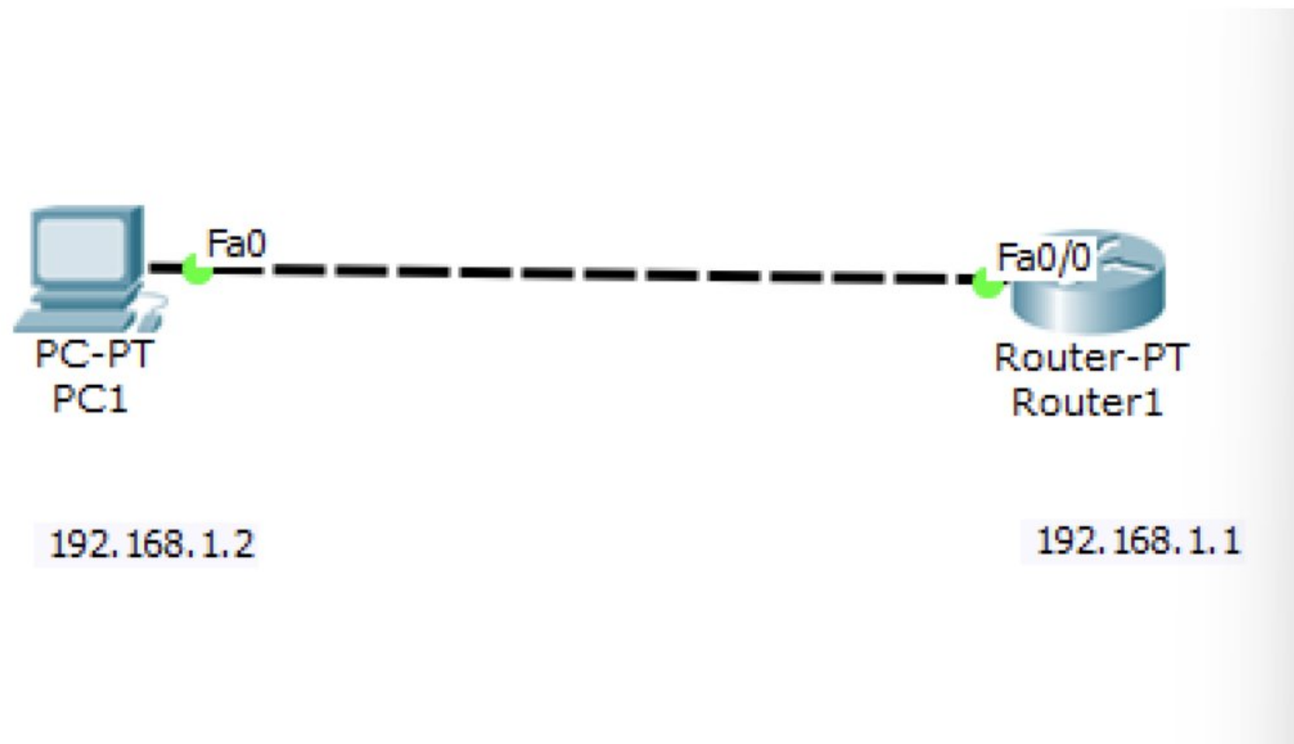
-> Two packets are created ICMP and ARP
-> Hover over the packets to check the type of packet.
-> click on ARP packet
-> Now you can see outbound PDU details
-> click on capture/Forward button to start the simulation.
-> Ping request from PC0 to switch will happen.
-> Again click on capture
-> click on capture twice to see the ping acceptance from server 0 to PC0
-> click on capture to see the ICMP packet movements.

Scanned with OKEN Scanner

Program 10

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

ii. Procedure along with the topology



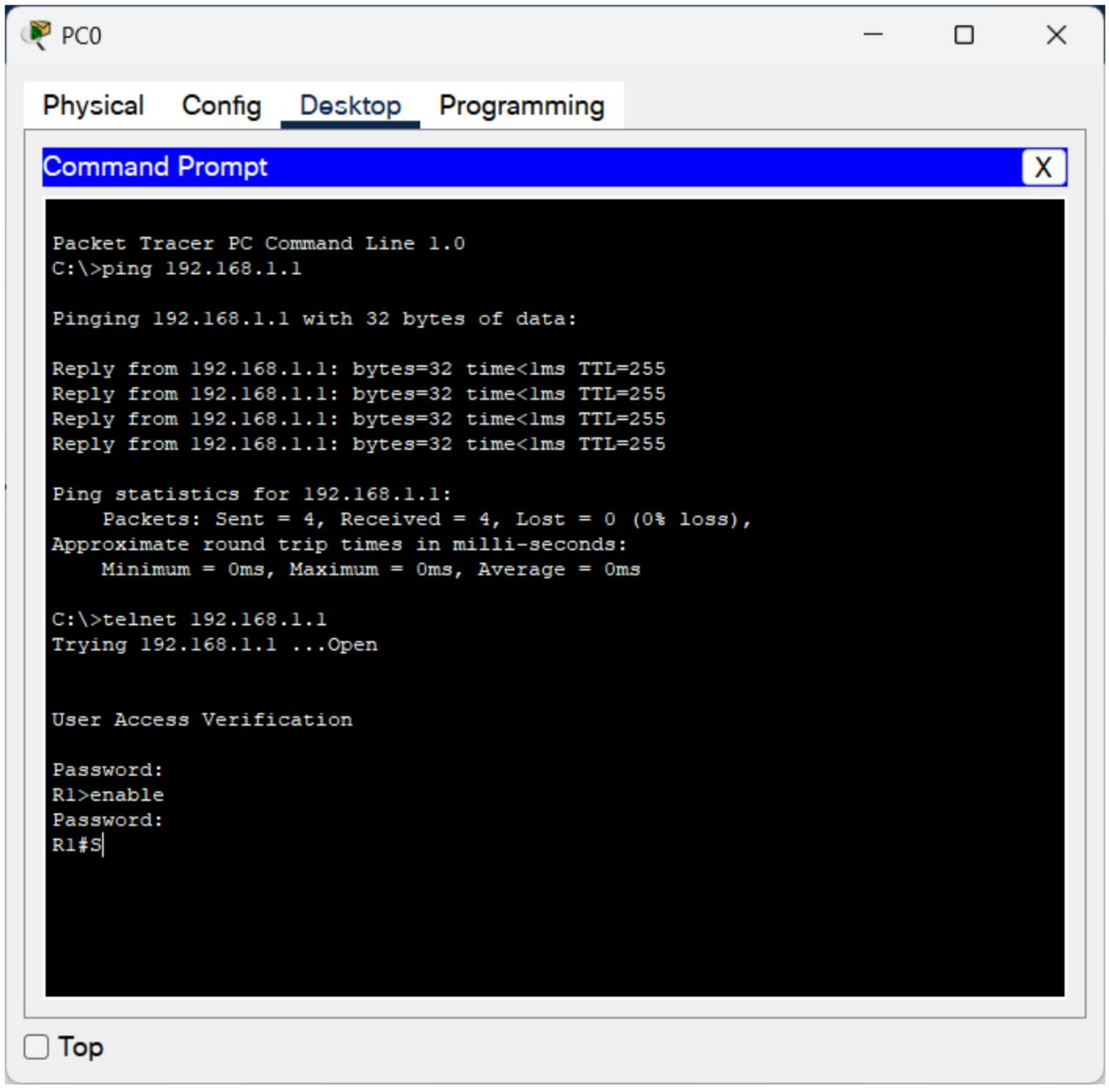
iii. Screen shots/ output

Router

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#enable secret hello
R1(config)#interface g0/0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up

R1(config-if)#line vty 0 5
R1(config-line)#login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password' is set
% Login disabled on line 7, until 'password' is set
R1(config-line)#password pass
R1(config-line)#end
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#wr
Building configuration...
[OK]
R1#
```


Scanned with OKEN Scanner

PC



Scanned with OKEN Scanner

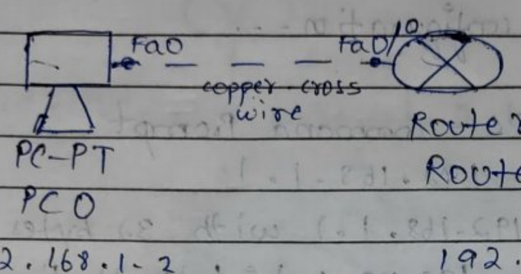
iv. Observation

20/12/24

Date __/__/__
Page _____

TELNET PROTOCOL

Topology :-



Implementation :-

→ Configure the IP address and gateway for PC0
→ Set the IP address for PC0 as 192.168.1.2, subnet mask as 255.255.255.0 and gateway as 192.168.1.1, and also DNS server 0.0.0.0
→ Connect PC0 and router0 using copper-cross wire
→ Configure Router in CLI

```
Router# config t
Router(Config)# hostname R1
R1(Config)#enable secret rp
R1(Config)#int fa0/0
R1(Config-if)#ip add 192.168.1.1 255.255.255.0
R1(Config-if)#no shut
R1(Config-if)# line vty 0 5
R1(Config-line)#login
# Login disabled on line 2, until 'password' is set
# Login disabled on line 3, until 'password' is set
# Login disabled on line 4, until 'password' is set
# Login disabled on line 5, until 'password' is set
# Login disabled on line 6, until 'password' is set
# Login disabled on line 7, until 'password' is set
```

Date __/__/__
Page _____

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

→ Go to PC- command Prompt

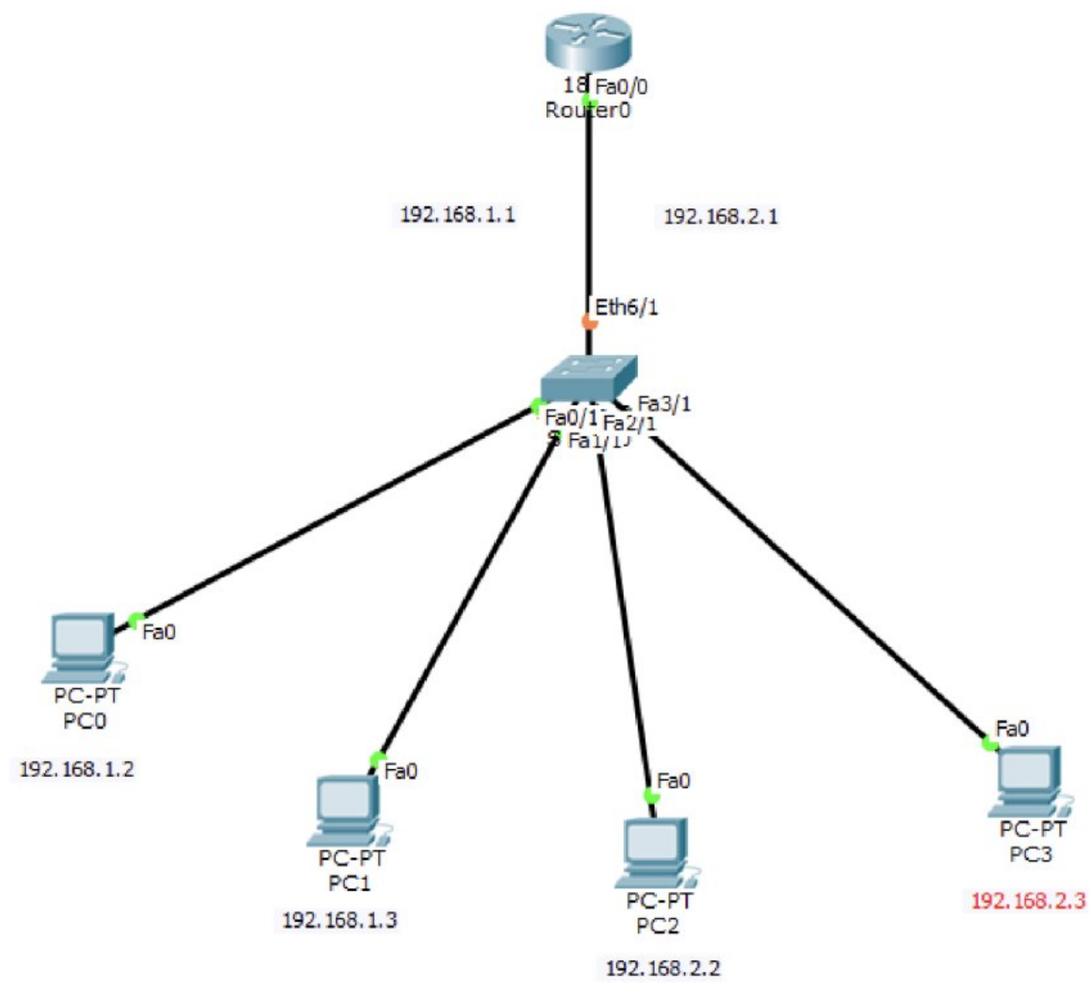
```
PC>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Ping statistics for 192.168.1.1:
  Packets: Sent=4, Received=4, Lost=0 (0% loss)
Approximate round trip times in milli-seconds:
  Minimum=0ms, Maximum=1ms, Average=0ms
PC>telnet 192.168.1.1
Trying 192.168.1.1 ... Open
User Access Verification
Password: rp
R1>en
Password: tp
R1#
```


Scanned with OKEN Scanner

Program 11

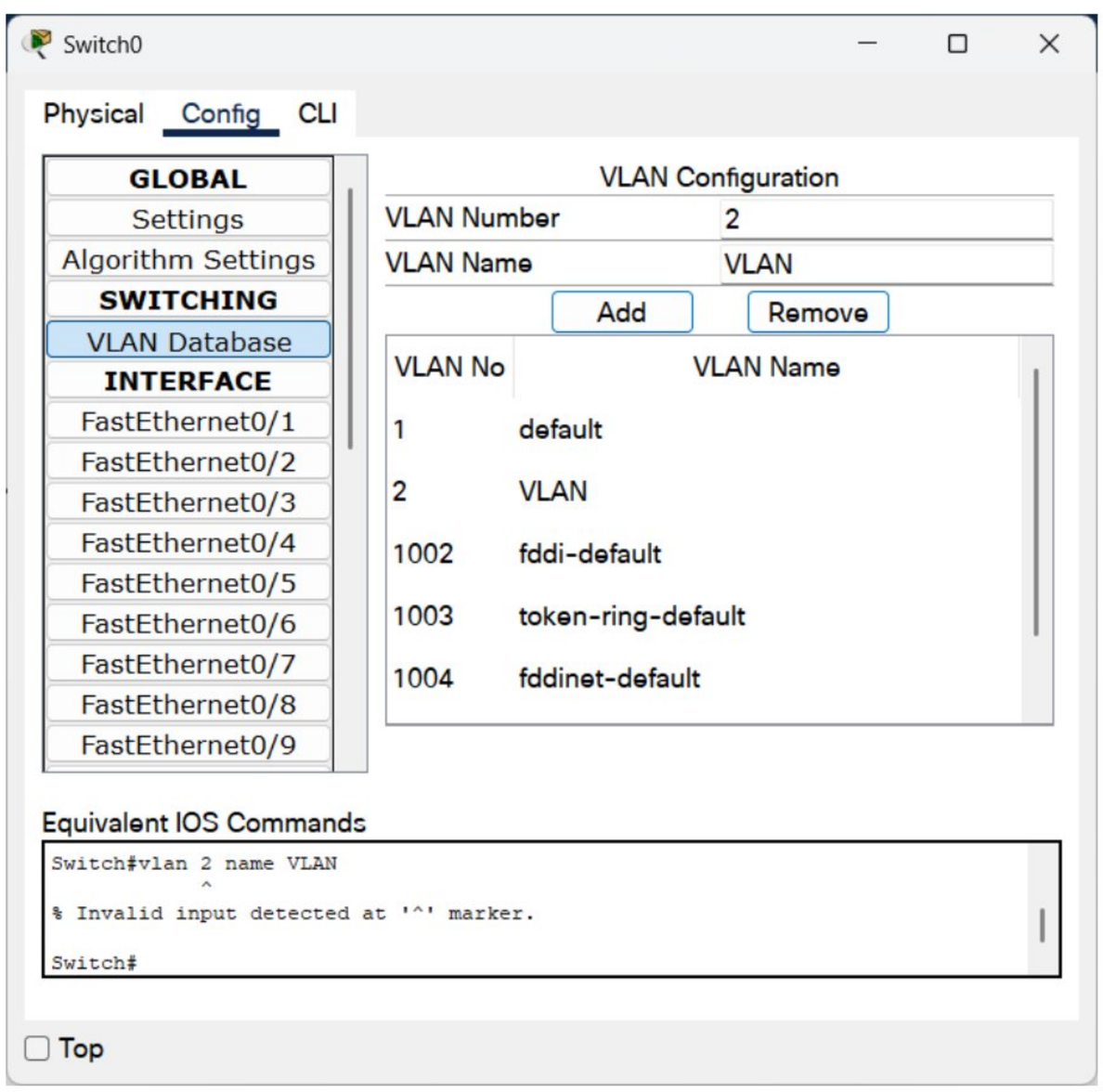
i. To construct a VLAN and make the PC's communicate among a VLAN

ii. Procedure along with the topology



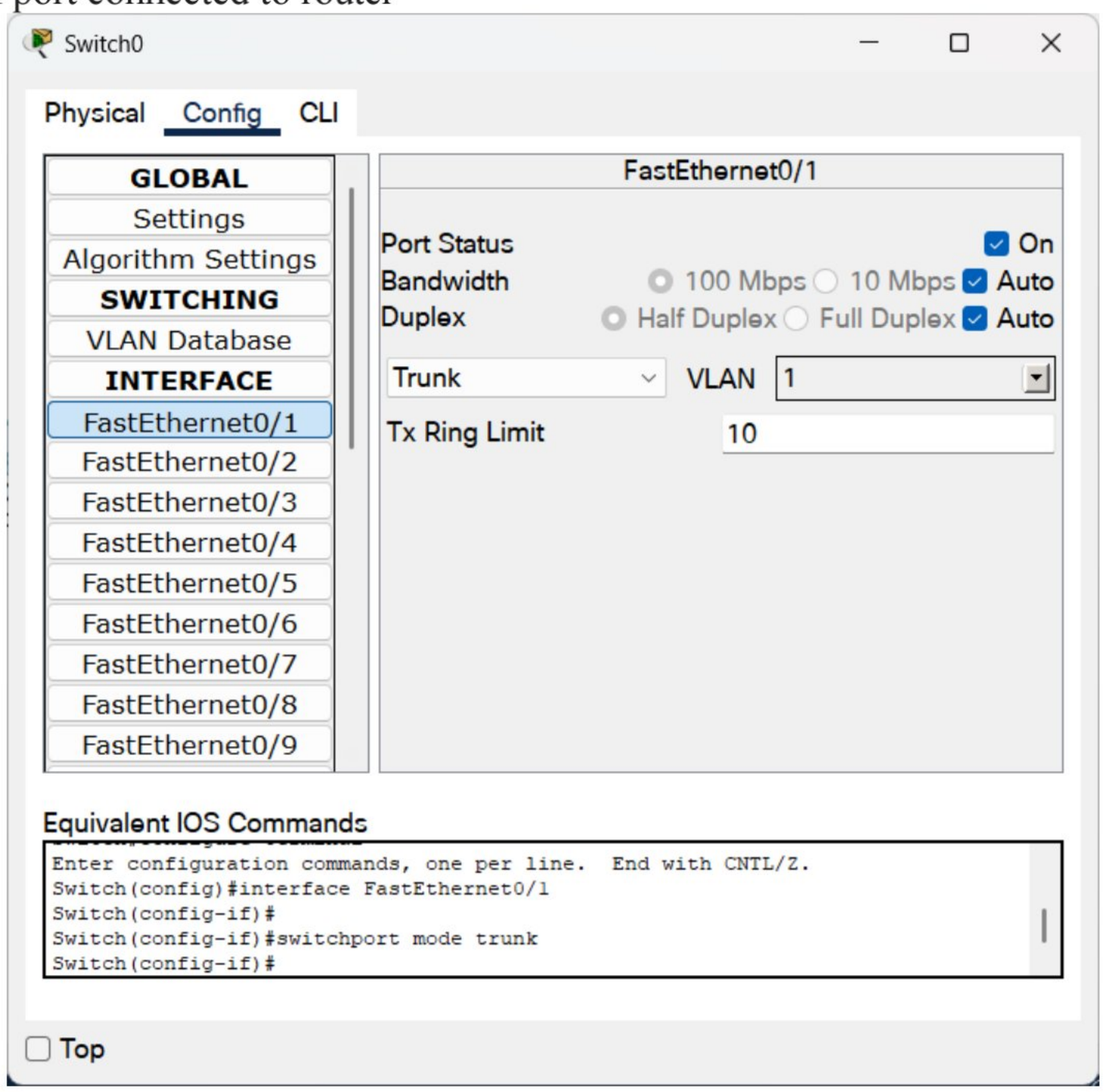
iii. Screen shots/ output

Switch Configuration

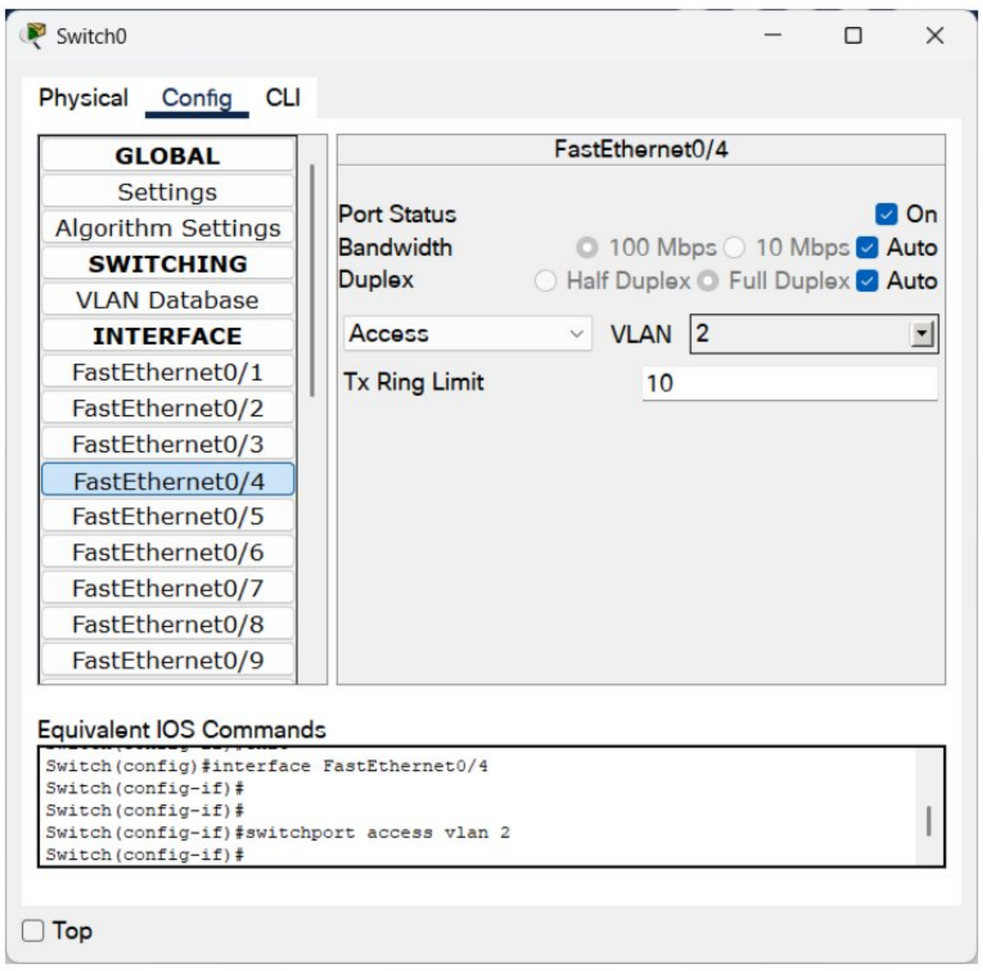


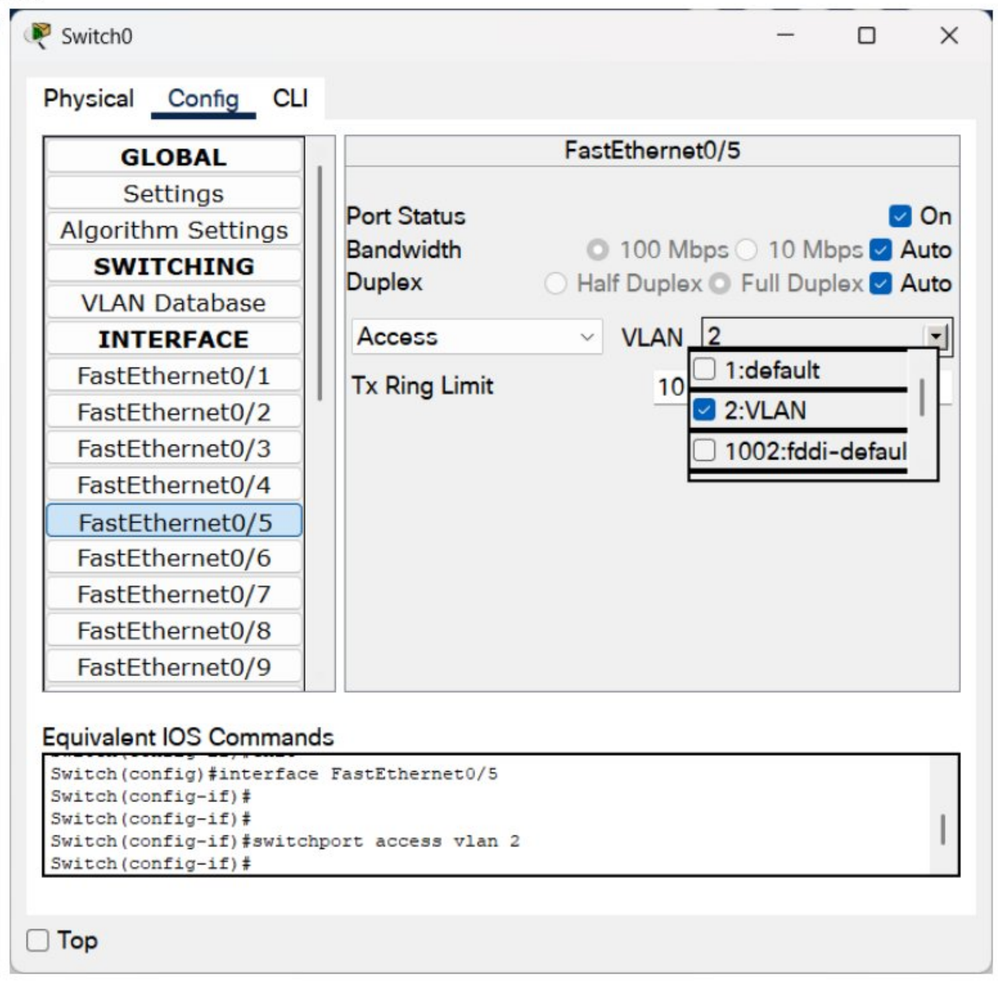
Scanned with OKEN Scanner

Configuration of port connected to router



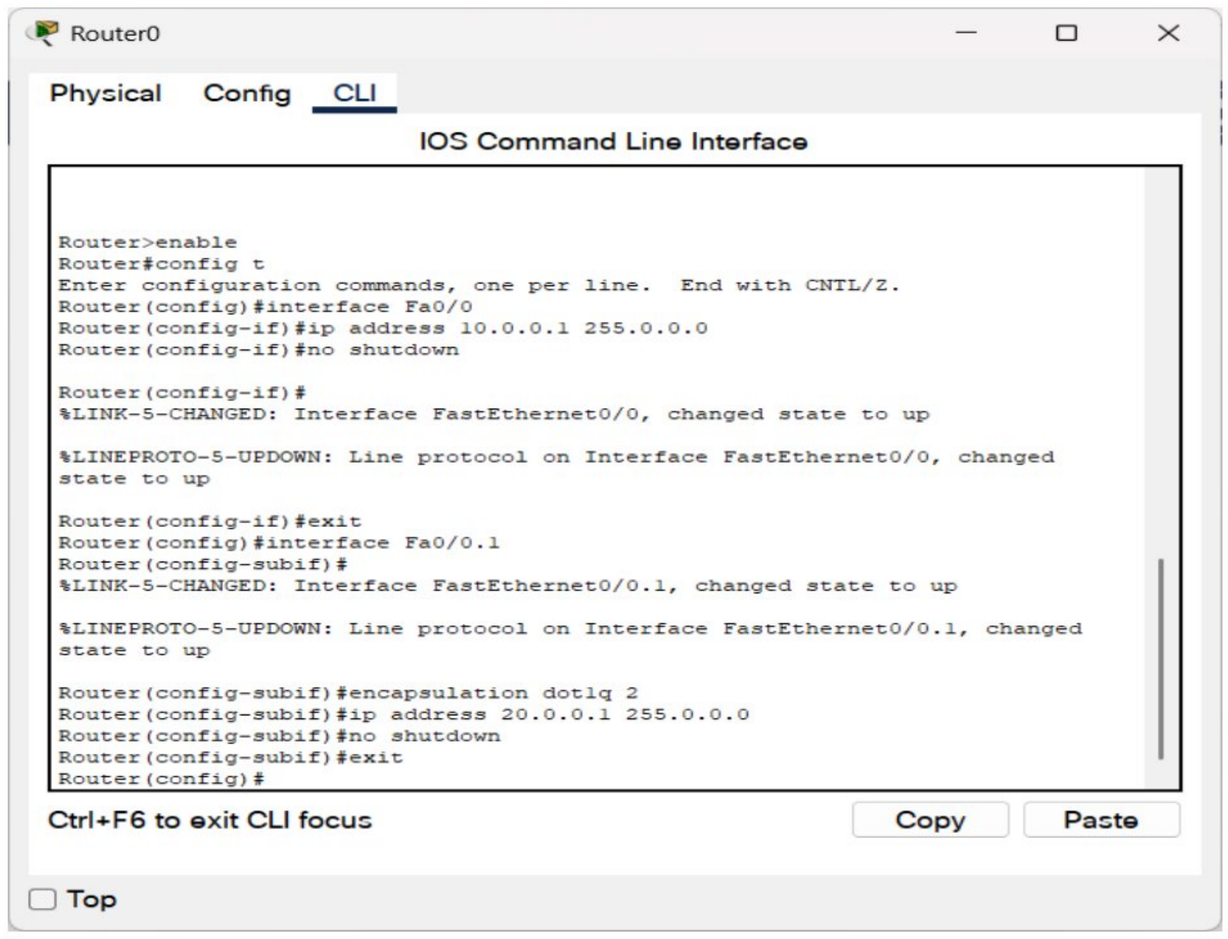
Configuration of ports connected to other networks



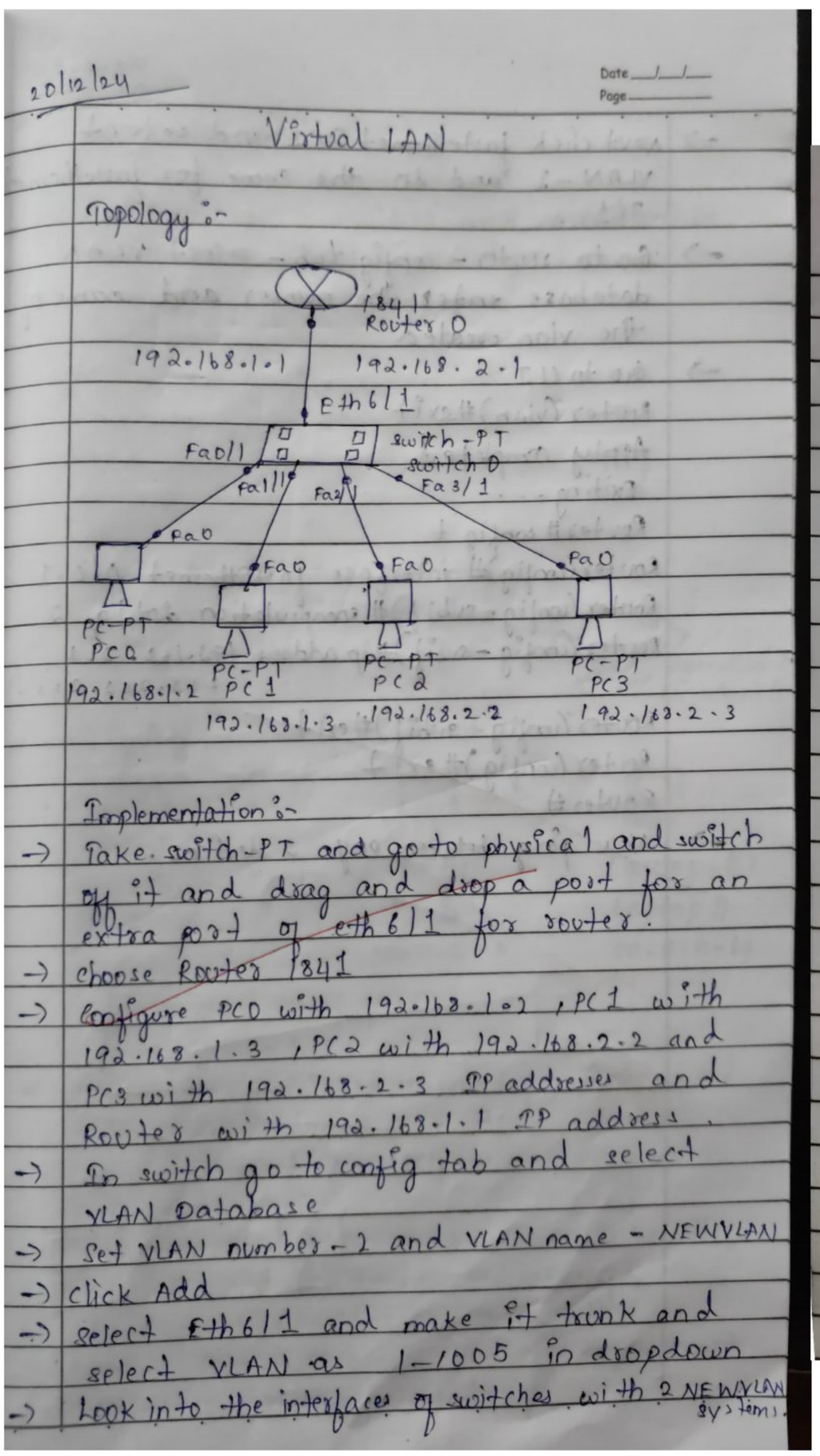


Scanned with OKEN Scanner

Configuartion of Router



iv. Observation



Implementation :-

-> Take switch-PT and go to physical and switch off it and drag and drop a port for an extra port of eth6/1 for router.

-> choose Router 1841

-> Configure PC0 with 192.168.1.2, PC1 with 192.168.1.3, PC2 with 192.168.2.2 and PC3 with 192.168.2.3 IP addresses and Router with 192.168.1.1 IP address.

-> In switch go to config tab and select VLAN Database

-> Set VLAN number - 2 and VLAN name - NEWVLAN

-> Click Add

-> Select Eth6/1 and make it trunk and select VLAN as 1-1005 in dropdown

-> Look into the interfaces of switches with 2 NEWVLAN systems.

-> Next click fastethernet 2/1 and select VLAN-2 and do the same for fastethernet 3/1.

-> Go to router - config tab - select VLAN database enter the number and name of the vlan created

-> Go to CLI

Router(vlan)#exit

Apply completed

Exiting...

Router# config t

Router(config)# interface fastEthernet 0/0.1

Router(config-subif)# encapsulation dot1q 2

Router(config-subif)#ip address 192.168.2.1 255.255.255.0

Router(config-subif)#exit

Router(config)#exit

Router#

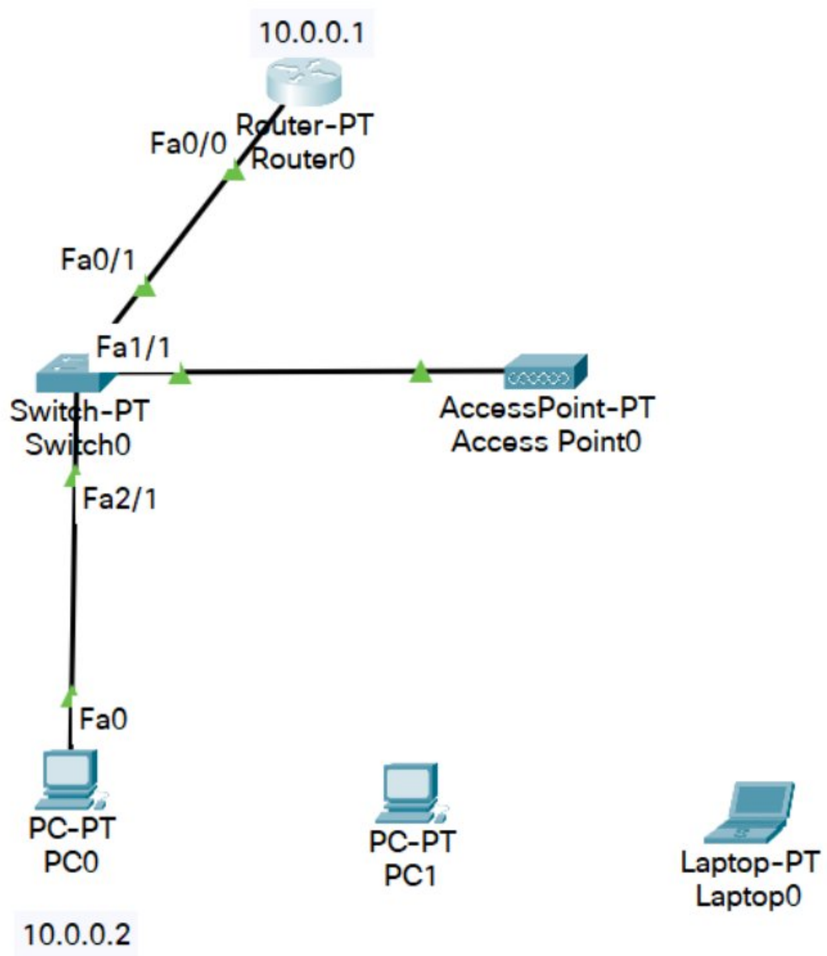
-> Now ping between any 2 devices.

Scanned with OKEN Scanner

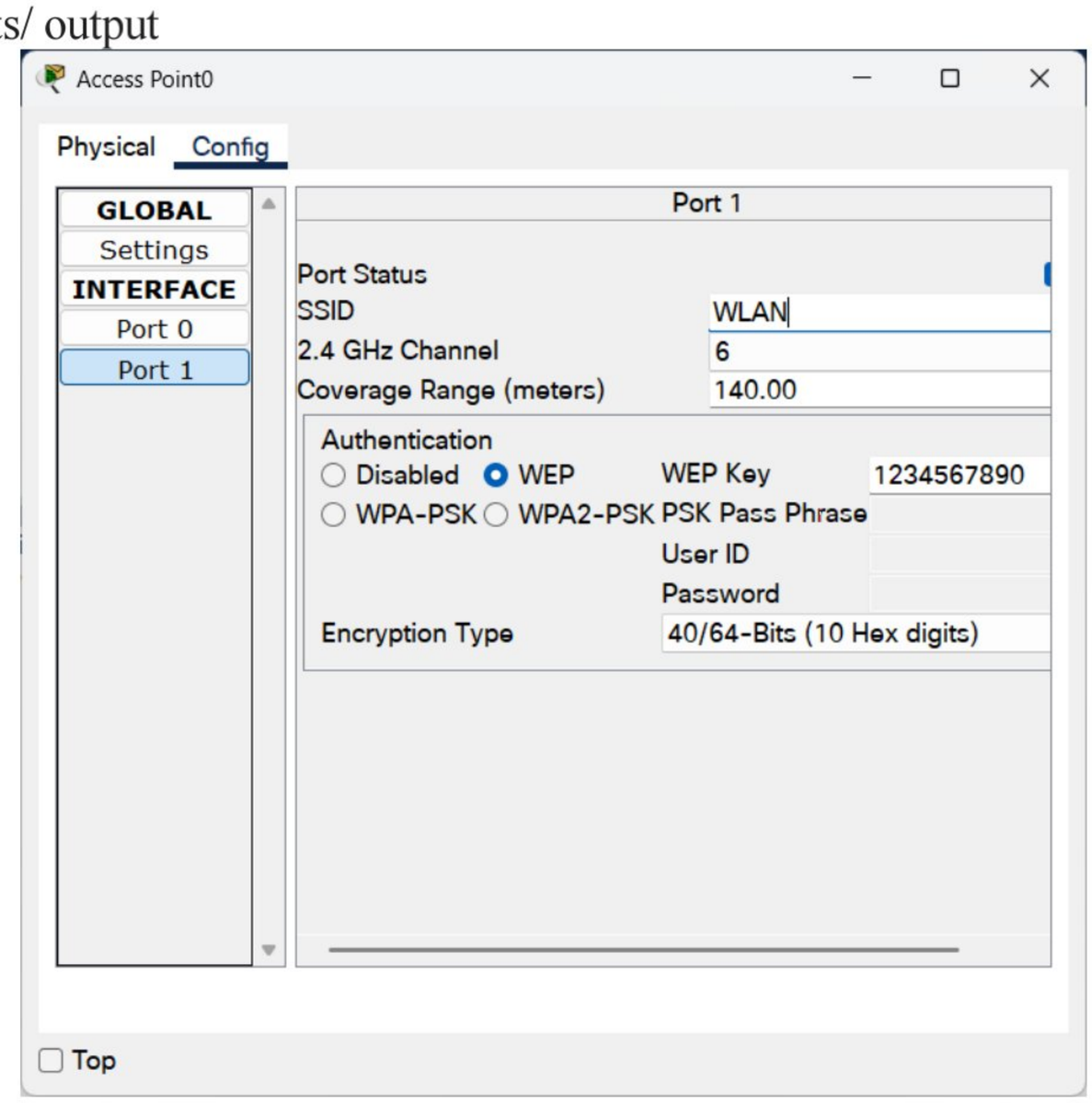
Program 12

i. To construct a WLAN and make the nodes communicate wirelessly.

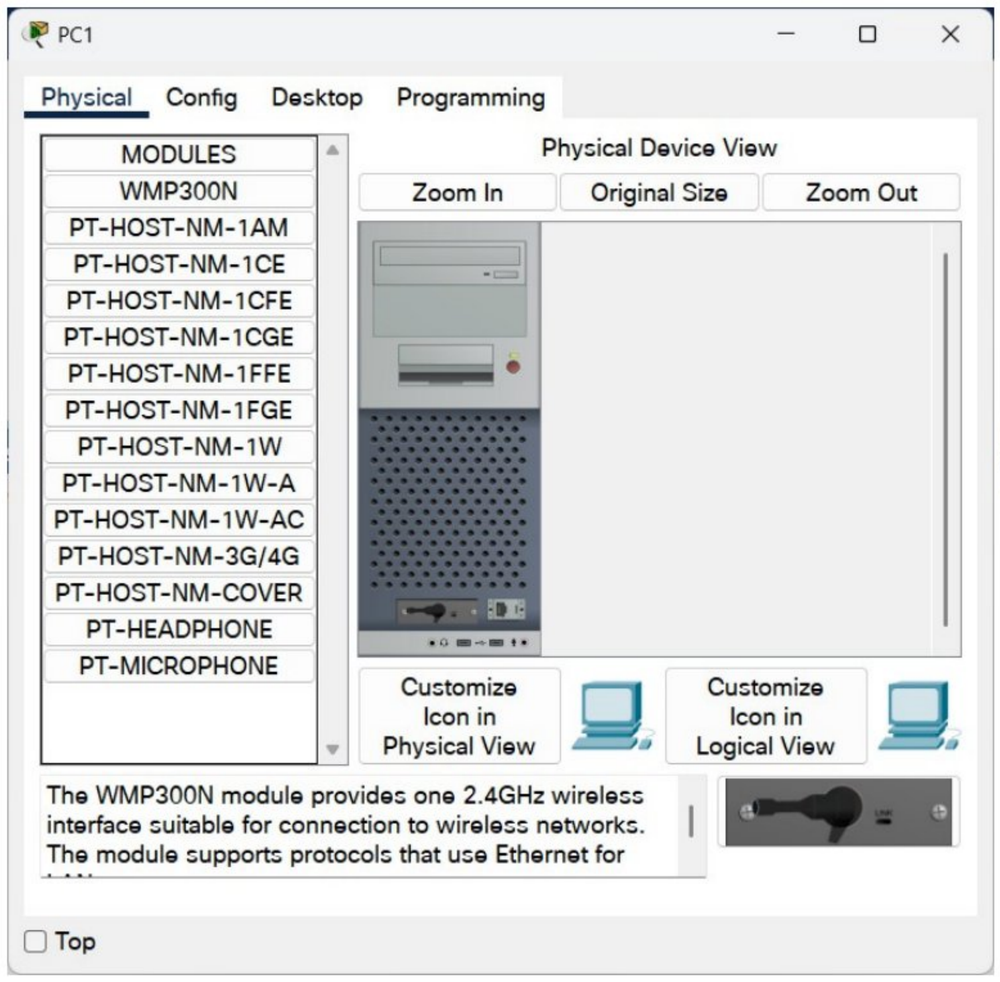
ii. Procedure along with the topology

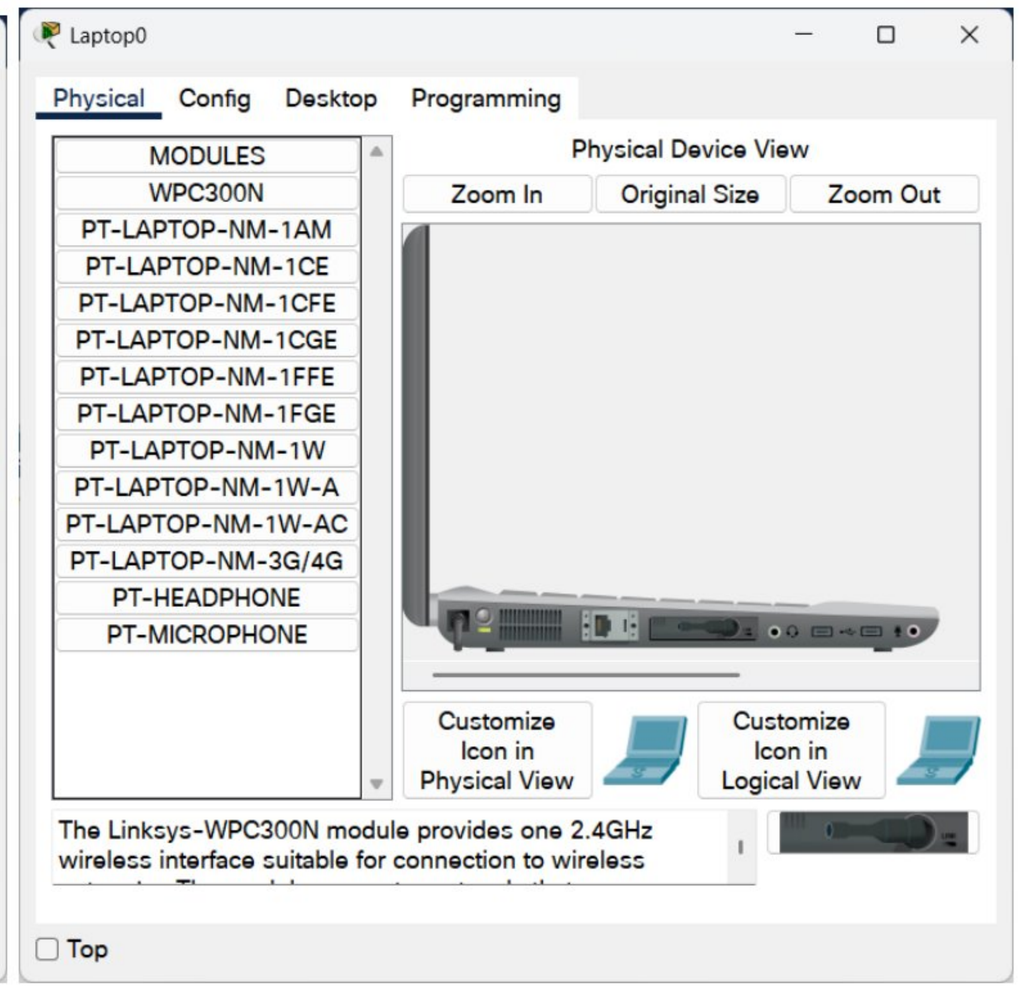


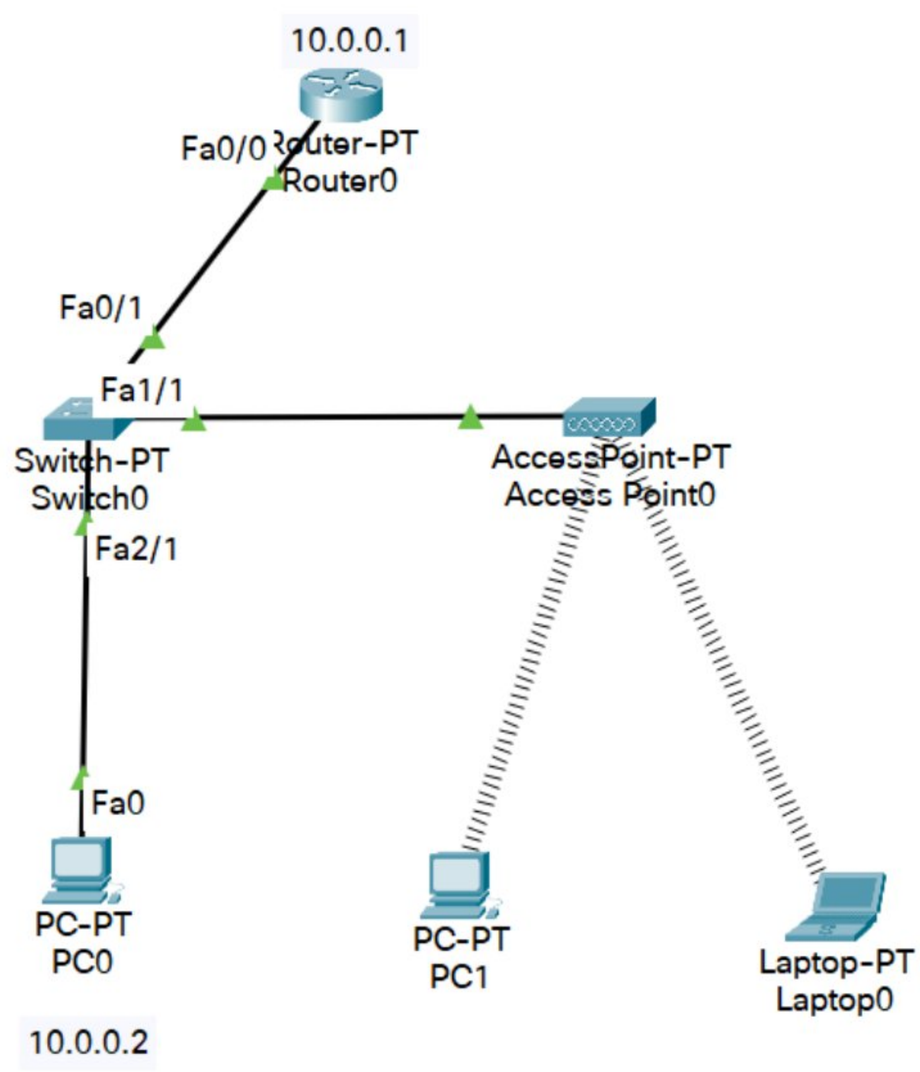
iii. Screen shots/ output



Scanned with OKEN Scanner

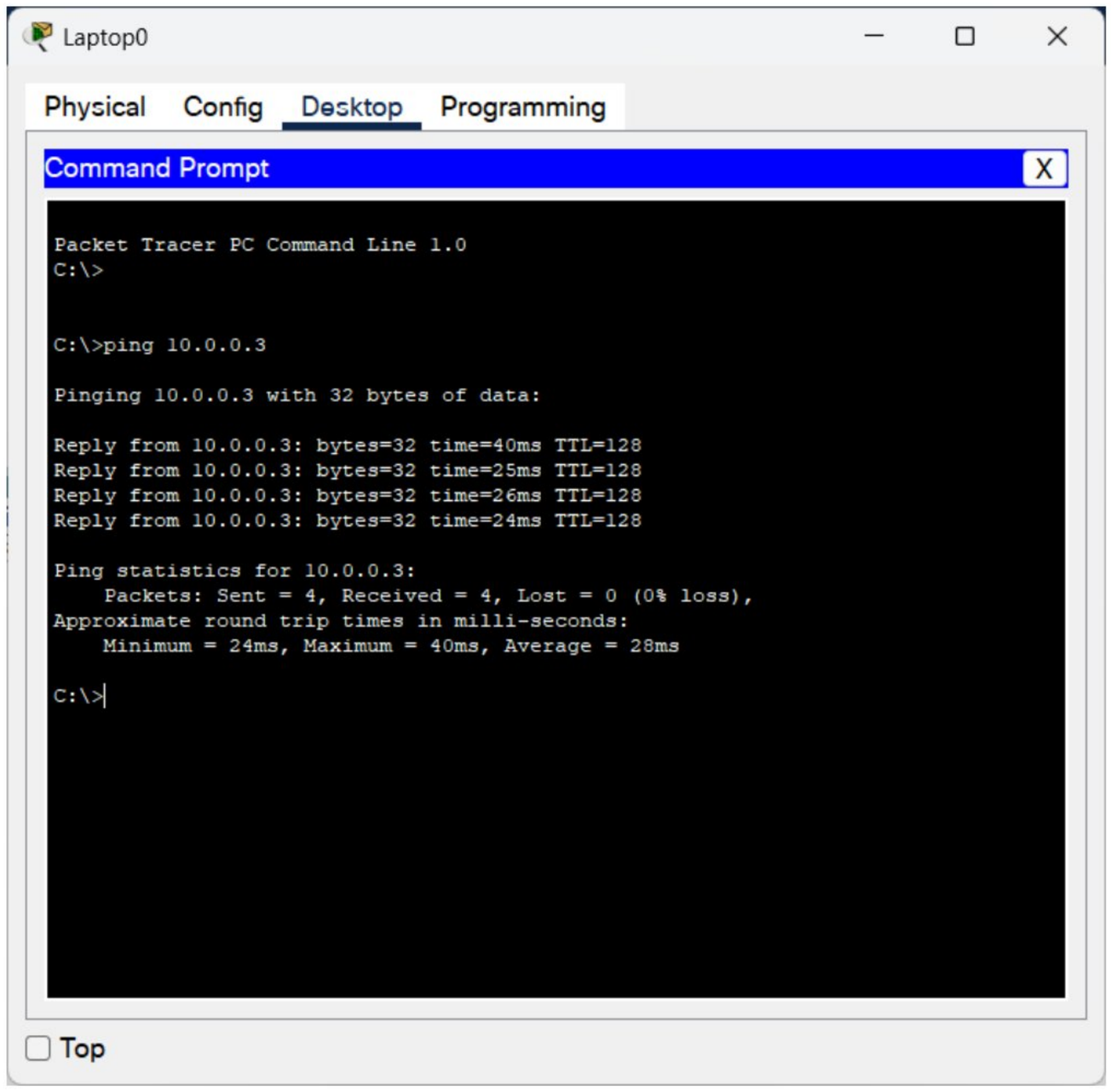




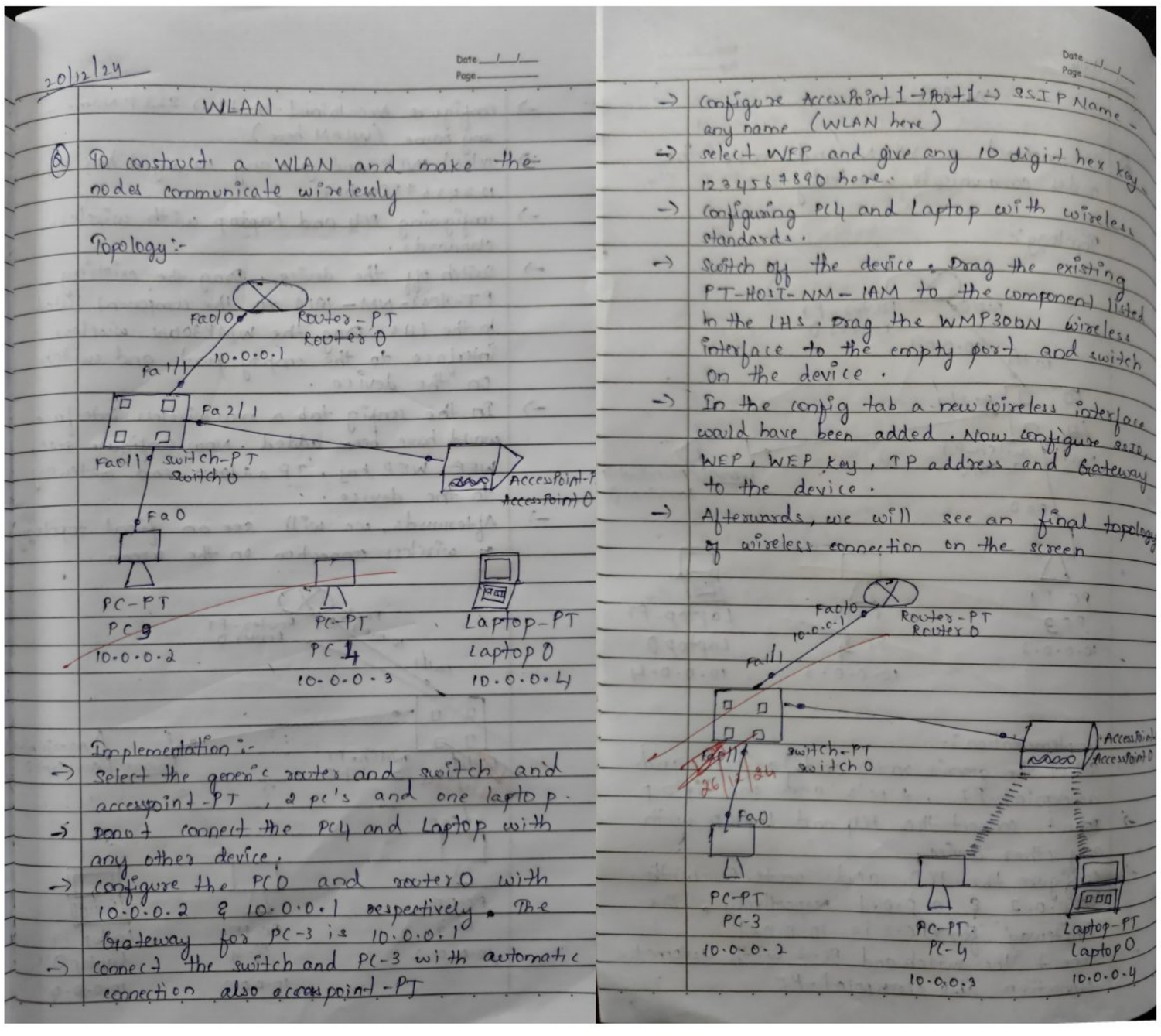


Scanned with OKEN Scanner

Ping:



iv. Observation



20/12/24

WLAN

Q) To construct a WLAN and make the nodes communicate wirelessly

Topology:-

Implementation:-

-> Select the generic router and switch and accesspoint-PT, 2 pc's and one laptop.
-> Donot connect the PC4 and Laptop with any other device.
-> Configure the PC0 and router0 with 10.0.0.2 & 10.0.0.1 respectively. The Gateway for PC-3 is 10.0.0.1
-> Connect the switch and PC-3 with automatic connection also accesspoint-PT
-> Configure Accesspoint 1 -> Port 1 -> SSID Name - any name (WLAN here)
-> select WEP and give any 10 digit hex key 1234567890 here.
-> Configuring PC4 and Laptop with wireless standards.
-> Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag the WMP300N wireless interface to the empty port and switch on the device.
-> In the config tab a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and Gateway to the device.
-> Afterwards, we will see an final topology of wireless connection on the screen

Scanned with OKEN Scanner

Cycle-II

Program 1

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

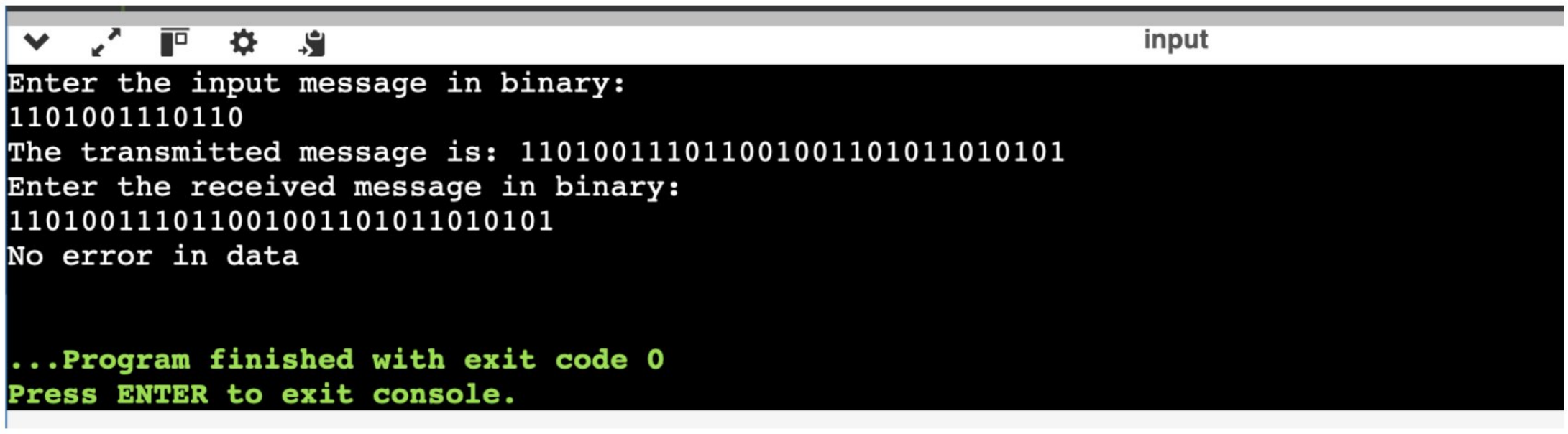
ii. Procedure

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

int crc(char *ip, char *op, char *poly, int mode)
   { strcpy(op, ip);

      if (mode) {
         for (int i = 1; i < strlen(poly); i++)
            { strcat(op, "0");
         }
      }

      // Perform XOR on the message with the selected polynomial
      for (int i = 0; i < strlen(ip); i++) {
         if (op[i] == '1') {
            for (int j = 0; j < strlen(poly); j++)
               { if (op[i + j] == poly[j])
                  op[i + j] = '0';
               else
                  op[i + j] = '1';
            }
         }
      }

      // Check for errors. Return 0 if error detected
      for (int i = 0; i < strlen(op); i++) {
         if (op[i] == '1')
            return 0;
      }

      return 1;
   }

int main() {
   char ip[50], op[50], recv[50];
   char poly[] = "10001000000100001";
```


Scanned with OKEN Scanner

```
printf("Enter the input message in binary: ");
scanf("%s", ip);
crc(ip, op, poly, 1);
printf("The transmitted message is: %s%s\n", ip, op + strlen(ip));

printf("Enter the received message in binary: ");

scanf("%s", recv);

if (crc(recv, op, poly, 0))
   { printf("No error in data\n");
} else {
   printf("Error in data transmission has occurred\n");
   }

   return 0;
}
```

iii. Screen shots/ output



Scanned with OKEN Scanner

iv. Observation

15/11/24

Q) Write a program for error detection using CRC – CCITT (16 bits)

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

void crc (char *input, char *output, char *poly,
        int mode)
{
    int inputlen = strlen (input);
    int polylen = strlen (poly);
    char paddedInput[100];
    strcpy (paddedInput, input);
    if (mode == 1) {
     for (int i = inputlen; i < inputlen + polylen - 1; i++){
        paddedInput[i] = '0';
     }
    paddedInput [inputlen + polylen - 1] = '\0';
    }
    for (int i=0; i < inputlen; i++) {
     if (paddedInput [i] == '1'){
      for (int j = 0; j < polylen; j++) {
        paddedInput [i+j] = (paddedInput [i+j] == poly[j])?
                '0' : '1';
      }
    }
    }
    strcpy (output, input);
    for (int i = inputlen; i < inputlen + polylen - 1; i++){
      output[i] = paddedInput[i];
    }
    output[i] = ~~paddedInput[i]~~; [inputlen + polylen - 1] = '\0';
}
int check - crc (char *received, char *poly)
{
    int receivedLen = strlen (received);
    int polyLen = strlen (poly);
    for (int i=0; i < receivedLen - polyLen + 1; i++){
      if (received [i] == '1') {
        for (int j = 0; j < polyLen; j++){
          received [i+j] = (received [i+j] == poly[j]) ? '0' : '1';
        }
      }
    }
    for (int i = receivedLen - polyLen + 1; i < receivedLen; i++){
      if (received [i] == '1') {
        return 0;
      }
    }
    return 1;
}
int main()
{
    char ip[50], op[50], recv [50];
    char poly [] = "1000100000010000 1";
    printf ("Enter the input message in binary:");
    scanf ("%s", ip);
    crc (ip, op, poly, 1);
    printf ("The transmitted message is: %s\n", op);
    printf ("Enter the received message in binary:");
    scanf ("%s, recv);
    if (check - crc (recv, poly)) {
      printf ("No error in data\n");
    } else {
      printf ("Error in data transmission has
            occured\n");
    }
    return 0;
}
```

Output :-

case 1 :-

Enter the input message in binary : 11111

The transmitted message is : 1111111100011110111110

Enter received message in binary: 1111111100011110111110

No error in data

case 2 :

Enter the input message in binary: 11111

The transmitted message is : 1111111100011110111110

Enter received message in binary : 1111111101011110111110

Error in data transmission has occurred.

Scanned with OKEN Scanner

Program 2

i. Write a program for congestion control using Leaky bucket algorithm

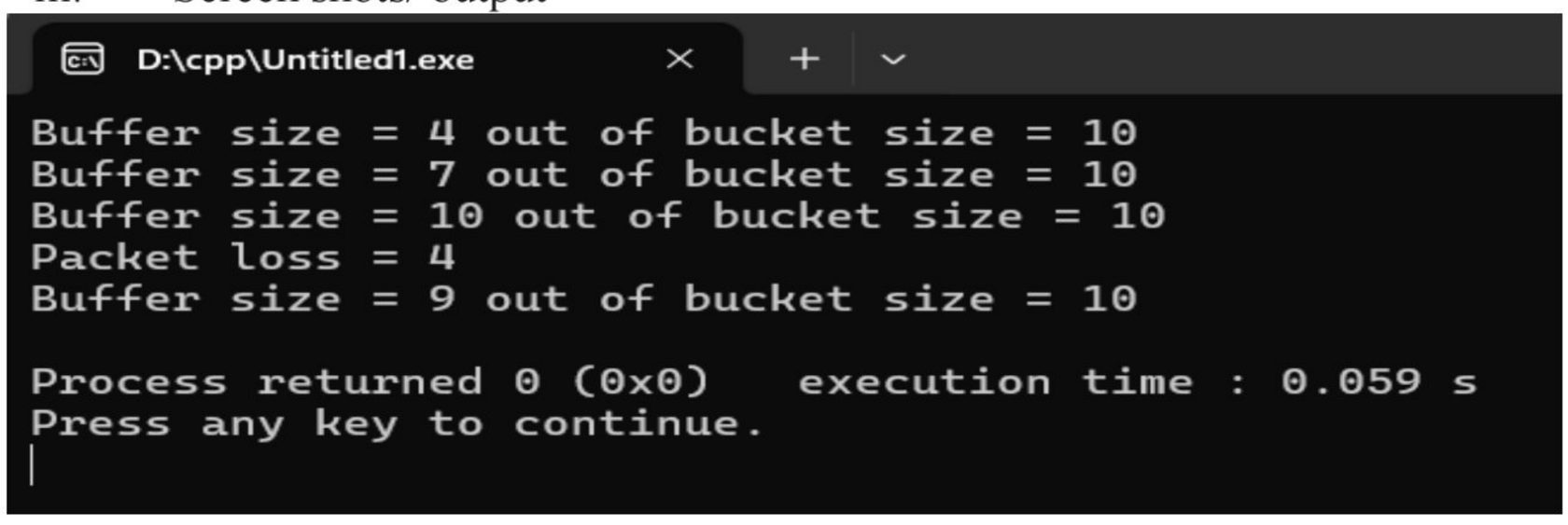
ii. Procedure

```
#include

<stdio.h>

int main() {
    int no_of_queries, storage, output_pkt_size;
    int input_pkt_size, bucket_size, size_left;
    storage = 0;
    no_of_queries = 4;
    bucket_size = 10;
    input_pkt_size = 4;
    output_pkt_size = 1;

    for (int i = 0; i < no_of_queries; i++)
        { size_left = bucket_size - storage;

        if (input_pkt_size <= size_left) {
            // Update storage
            storage += input_pkt_size;
        } else {
            printf("Packet loss = %d\n", input_pkt_size);
        }

        printf("Buffer size = %d out of bucket size = %d\n",
              storage, bucket_size);

        // Packets leaving the bucket
        storage -= output_pkt_size;

        // Ensure storage doesn't go negative
        if (storage < 0) {
            storage = 0;
        }
    }
```

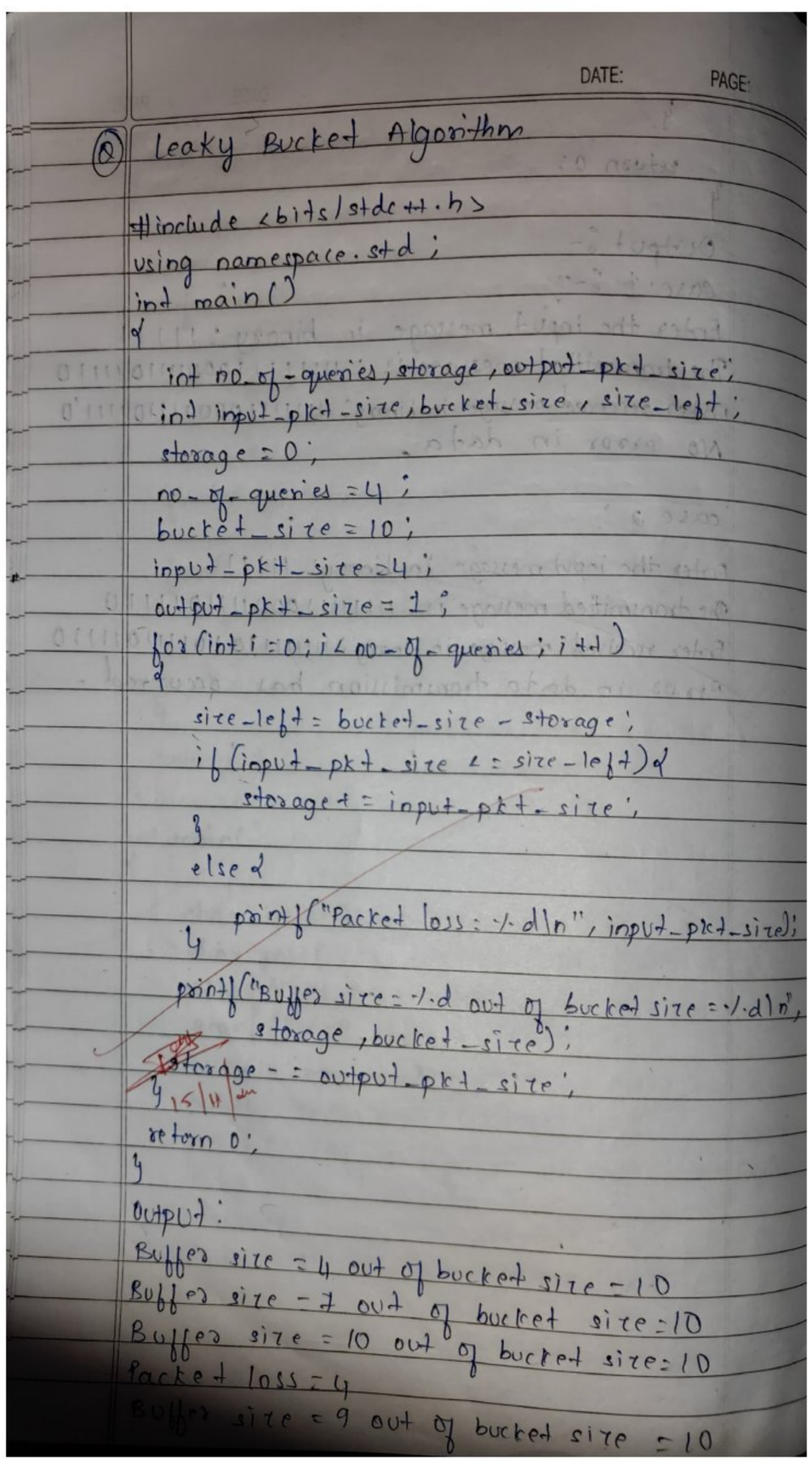

Scanned with OKEN Scanner

```
    return 0;
}
```

iii. Screen shots/ output



iv. Observation



DATE: PAGE:

Q Leaky Bucket Algorithm

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
  int no_of_queries, storage, output_pkt_size;
  int input_pkt_size, bucket_size, size_left;
  storage = 0;
  no_of_queries = 4;
  bucket_size = 10;
  input_pkt_size = 4;
  output_pkt_size = 1;
  for (int i = 0; i < no_of_queries; i++)
  {
    size_left = bucket_size - storage;
    if (input_pkt_size <= size_left){
      storage += input_pkt_size;
    }
    else{
      printf("Packet loss = %d\n", input_pkt_size);
    }
    printf("Buffer size = %d out of bucket size = %d\n",
            storage, bucket_size);
    storage -= output_pkt_size;
  }
  return 0;
}
```

Output:

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


Scanned with OKEN Scanner

Program 3

i. 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.

ii. Procedure

clientTCP.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()
```

serverTCP.py

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

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

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
```


Scanned with OKEN Scanner

connectionSocket.close()

iii. Screen shots/ output

Client

```
IDLE Shell 3.12.6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: E:\Engineering\5Sem\CN\Experiments\clientTCP.py ===========

Enter file name: serverTCP.py

From Server:

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()

>>>
```

Server

```
*IDLE Shell 3.12.6*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: E:\Engineering\5Sem\CN\Experiments\serverTCP.py ===========
The server is ready to receive

Sent contents of serverTCP.py
The server is ready to receive
```


Scanned with OKEN Scanner

iv. Observation

20/12/24

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

→ clientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\nEnter file name:")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('\n From Server:\n')
print(filecontents)
clientSocket.close()
```

→ serverTCP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while(1):
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('\n Sent contents of' + sentence)
    file.close()
    connectionSocket.close()
```

Output:-

The server is ready to receive
Sent contents of serverTCP.py
The server is ready to receive

Scanned with OKEN Scanner

Program 4

i. 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.

ii. Procedure

clientUDP.py

```
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()
```

serverUDP.py

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

    print ('\nSent contents of', end = ' ')
```


Scanned with OKEN Scanner

```
print (sentence)
#  for i in sentence:
    #  print (str(i), end = '')
file.close()
```

iii. Screen shots/ output

Client

```
IDLE Shell 3.12.6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Engineering\5Sem\CN\Experiments\clientUDP.py

Enter file name:  serverUDP.py

Reply from Server:

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()
```

Server

```
*IDLE Shell 3.12.6*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Engineering\5Sem\CN\Experiments\serverUDP.py
The server is ready to receive

Sent contents of  serverUDP.py
```


Scanned with OKEN Scanner

20/12/24

Date __/__/__
Page

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

→ clientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket (AF_INET, SOCK_DGRAM)
sentence = input ("\n Enter file name :")
clientSocket.sendto (bytes (sentence, "utf-8"), (serverName,
                                                 serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print ("\n Reply from server :\n")
print (filecontents.decode ("utf-8"))
clientSocket.close()
clientSocket.close()
```

→ ServerUDP.py

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

Date __
Page

```
    file.close()
```

Output :-

```
The server is ready to receive
Sent contents of server UDP.py
The server is ready to receive
```


Scanned with OKEN Scanner