VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS (23CS5PCCON)

Submitted by

SAKSHI KALLUR (1BM22CS345)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Sep 2024-Jan 2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**COMPUTER NETWORKS**" carried out by **SAKSHI KALLUR (1BM22CS345),** 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 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Computer Networks Lab - (23CS5PCCON)** work prescribed for the said degree.

Spoorthi D M

Assistant Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

CYCLE 1

INDEX

CYCLE 2

Github Link:

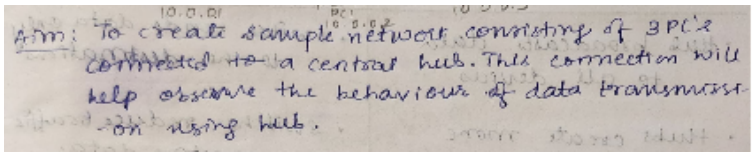
https://github.com/Sakshi-Kallur/CN-LAB.git

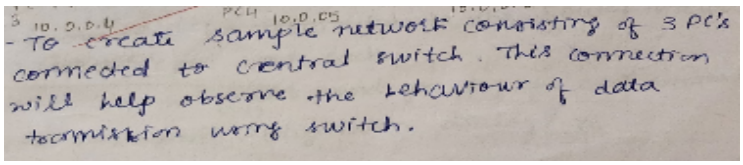
CYCLE 1

Program 1

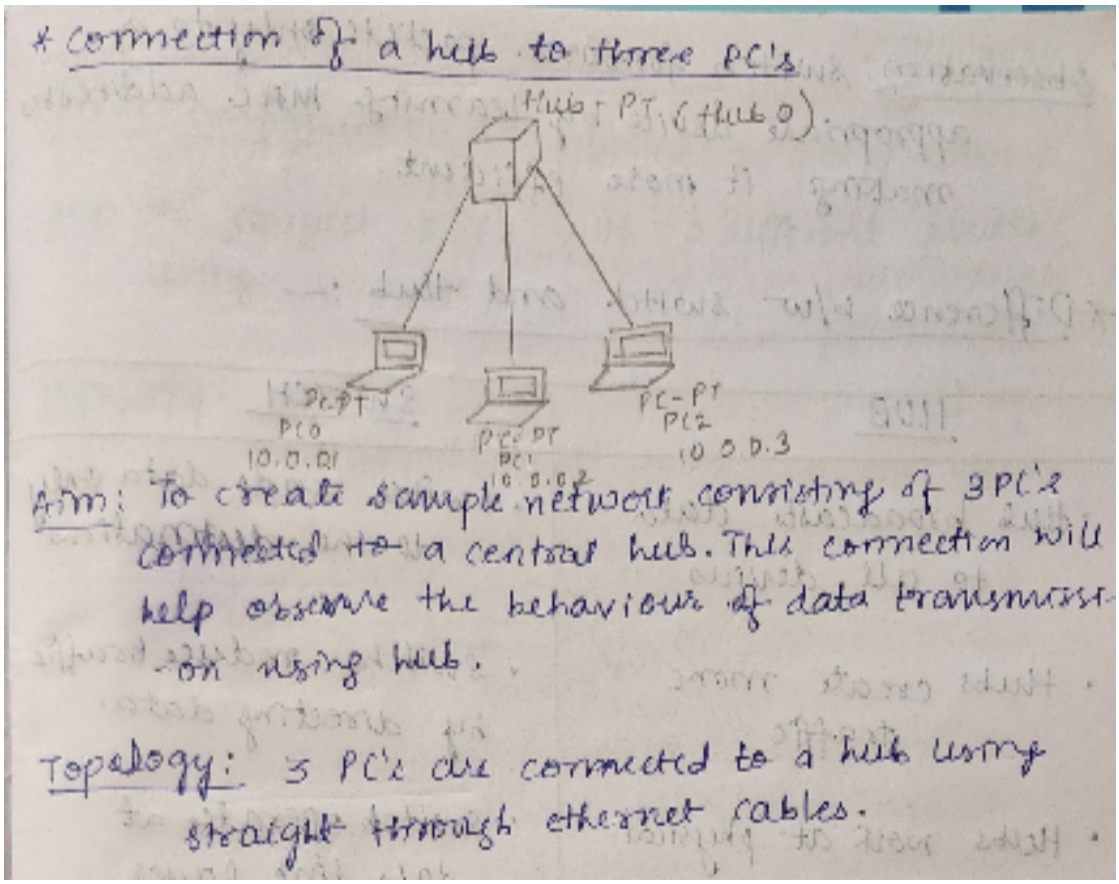
Question:

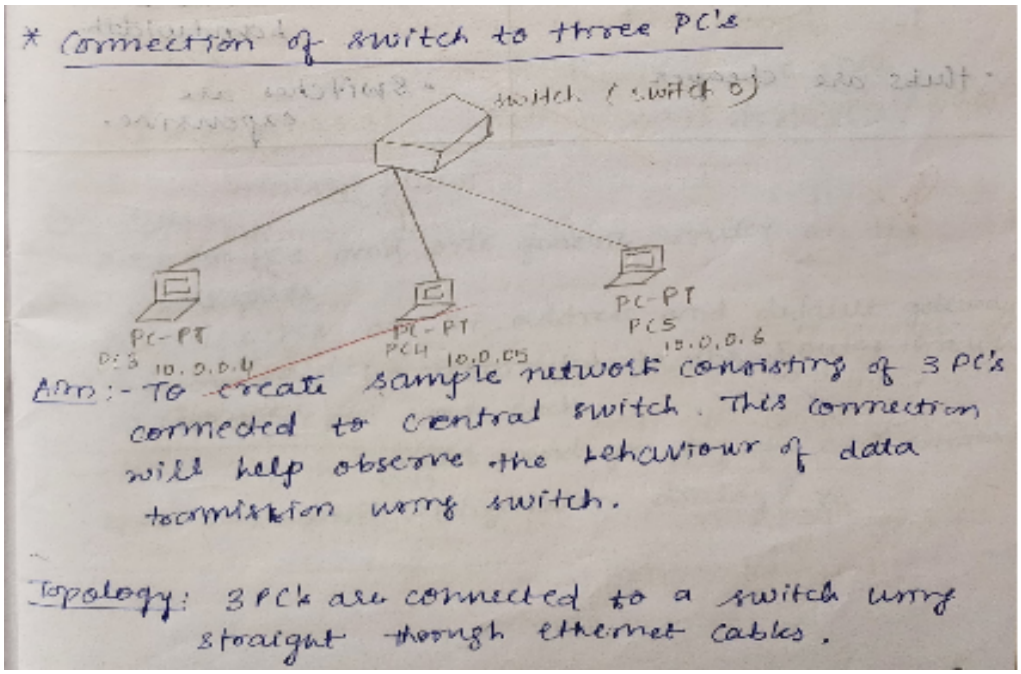
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

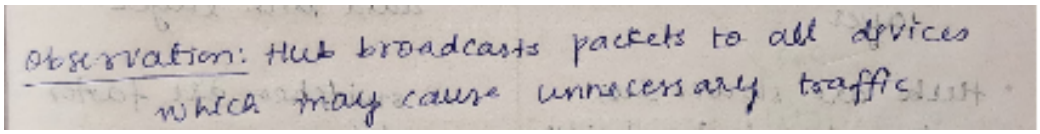

Aim: To create sample network consisting of 3 PC's connected to a central hub. This connection will help observe the behaviour of data transmission using hub.

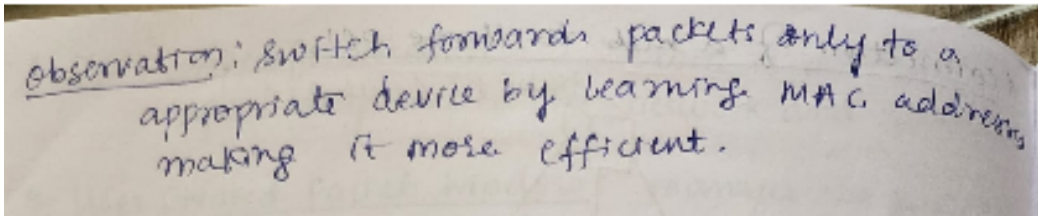

To create sample network consisting of 3 PC's connected to central switch. This connection will help observe the behaviour of data transmission using switch.

Observation writeup:

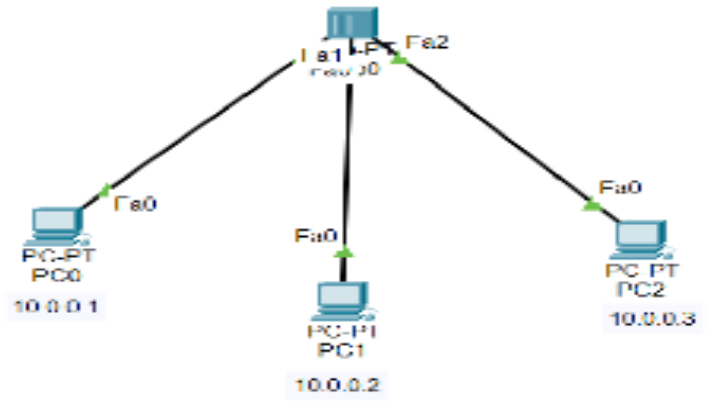


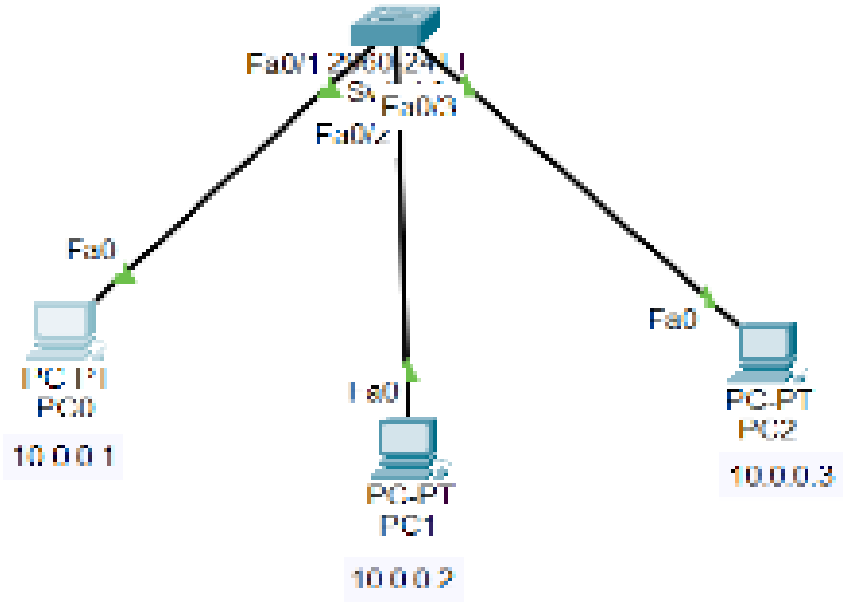

* Connection of switch to three PC's
switch (switch 0)
PC-PT
PC5
10.0.0.6
PC-PT
PC4
10.0.0.5
PC-PT
PC5
10.0.0.6
Aim:- To create sample network consisting of 3 PC's connected to central switch. This connection will help observe the behaviour of data transmission using switch.
Topology: 3 PC's are connected to a switch using straight through ethernet cables.


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


observation: switch forwards packets only to a appropriate device by learning MAC addresses making it more efficient.

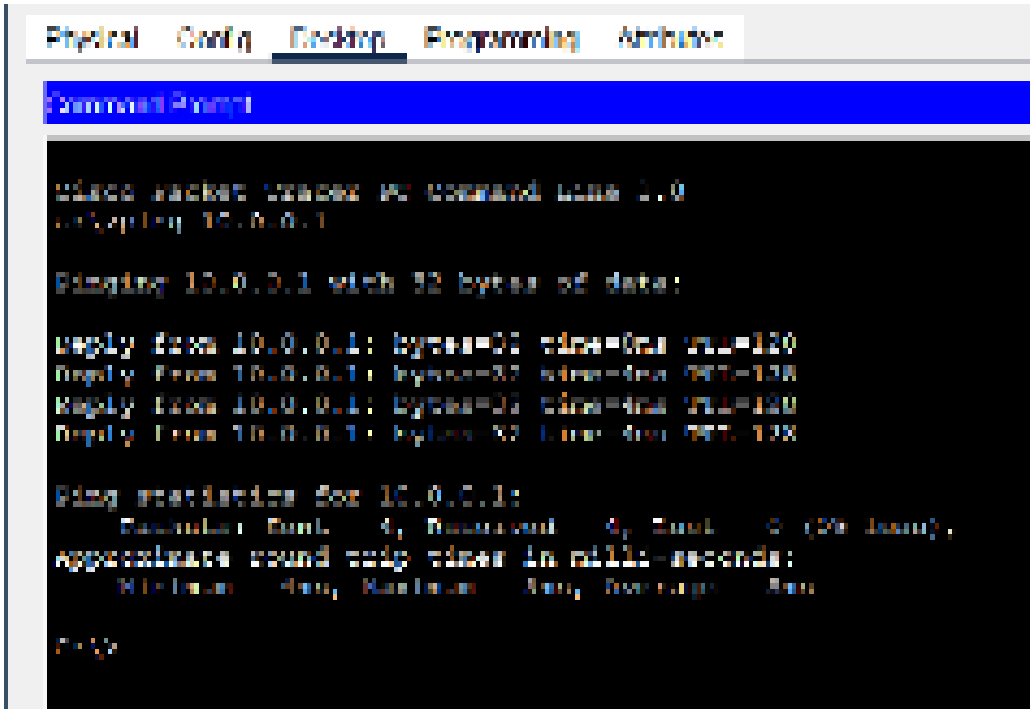
Topology:


Fa2
Fa0
Fa0
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3

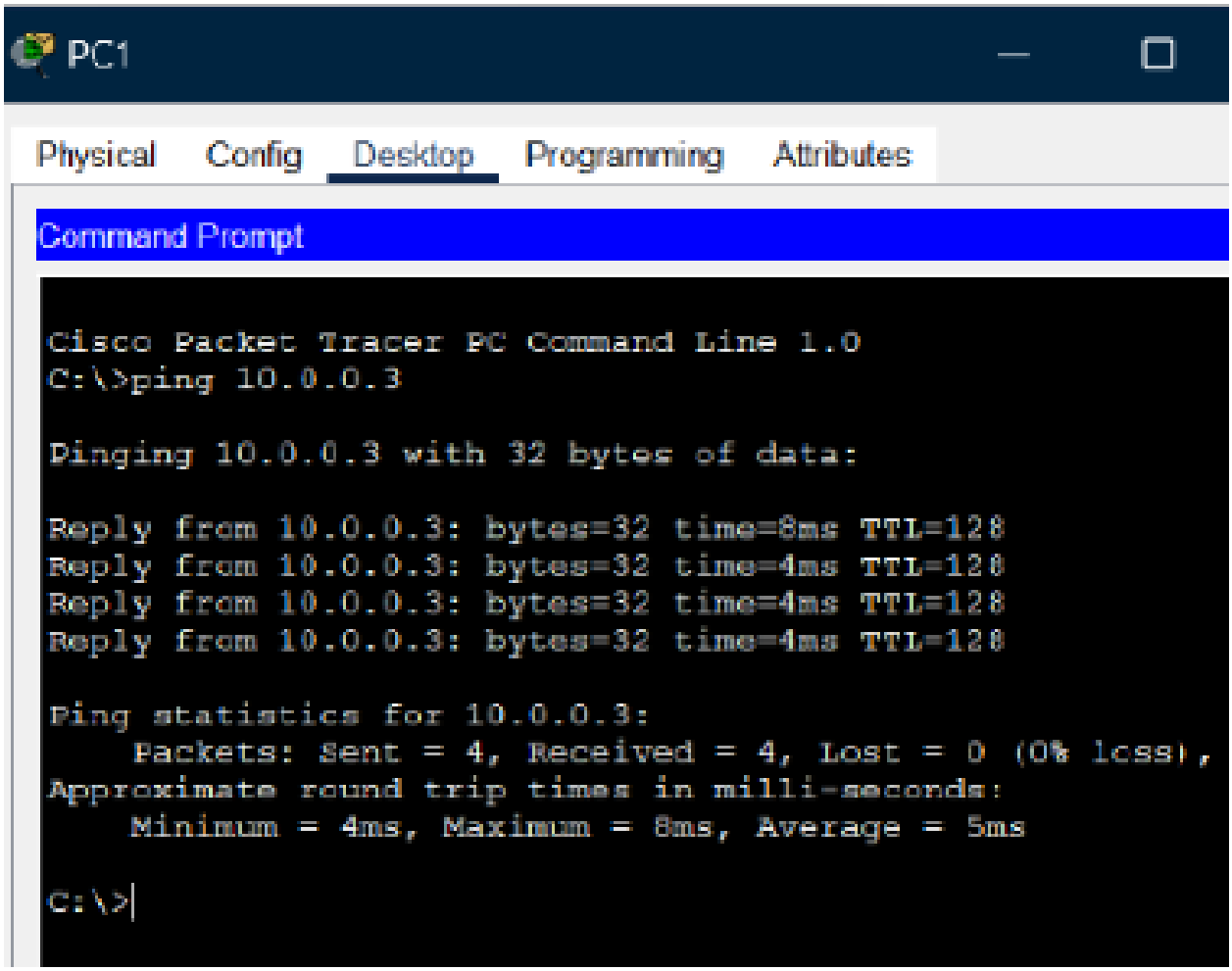


Output:

SWITCH:



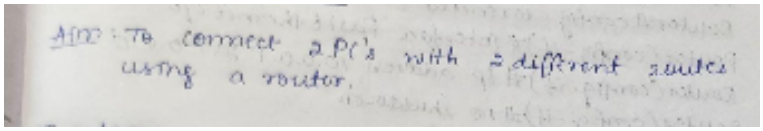
HUB:

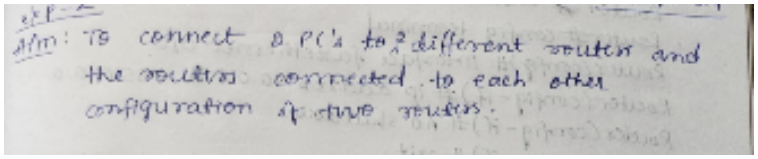

PC1
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3
Dinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=8ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 8ms, Average = 5ms
C:\>

Program 2

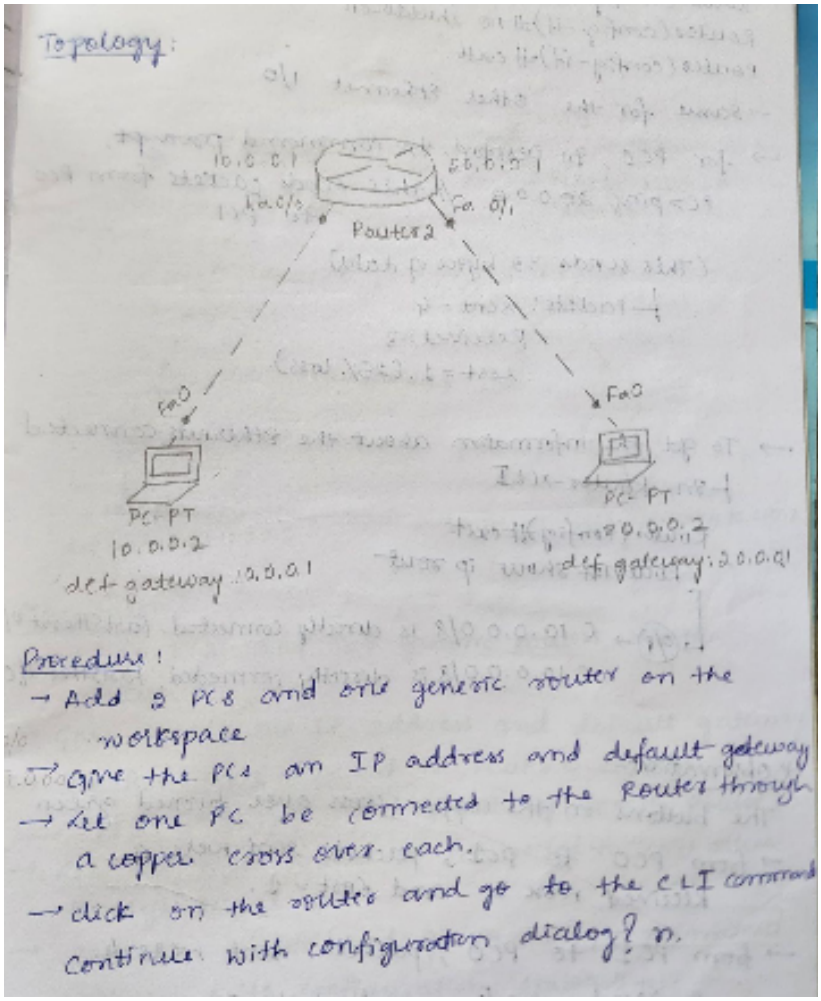
Question:

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





Observation writeup:



Router#enable
Router# config terminal
Router(config-if)# interface FastEthernet 0/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit.

→ same for the other Ethernet 1/0

→ for PC0, in Desktop, in command prompt,
PC> ping 20.0.0.2 // this sends packets from PC0 to PC1

(This sends 32 bytes of data)

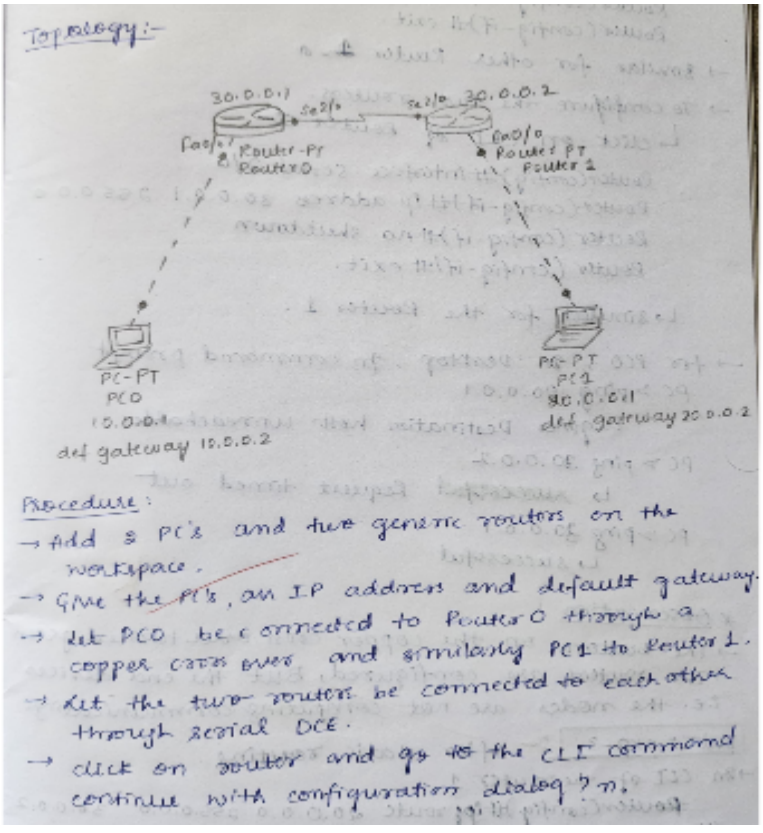
- Packets: Sent = 4
  Received = 3
  Lost = 1. (25% loss).

→ To get the information about the Ethernets connected in Router→CLI

Router(config)# exit
Router# show ip route

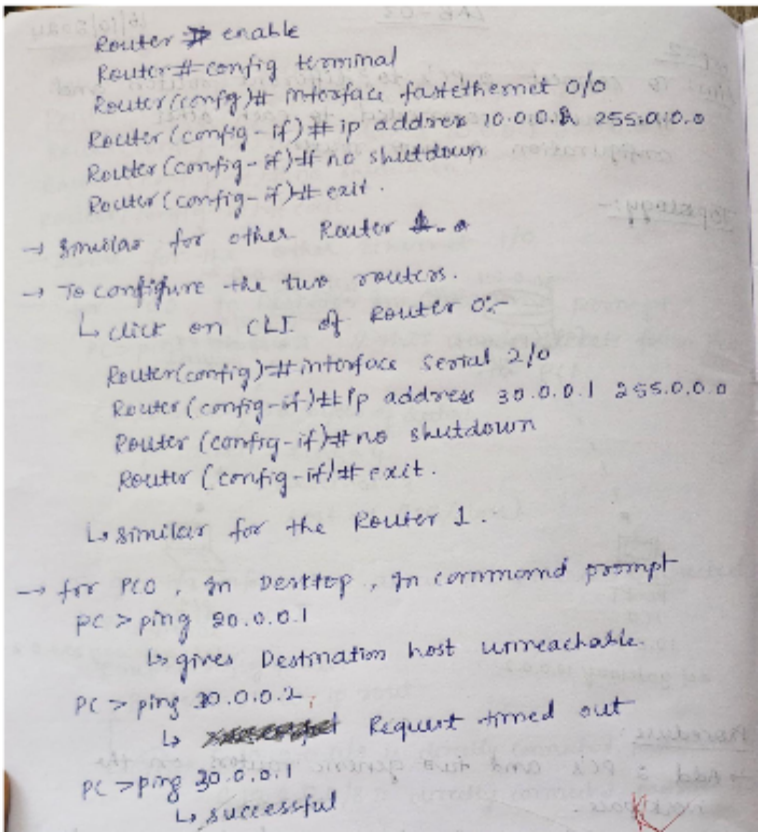
(C) → C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 10.0.0.0/8 is directly connected, FastEthernet 1/0

Topology:-



Procedure:

→ Add 2 PC's and two generic routers on the workspace.
→ Give the PC's, an IP address and default gateway.
→ Let PC0 be connected to Router 0 through a copper cross over and similarly PC1 to Router 1.
→ Let the two routers be connected to each other through serial DCE.
→ click on router and go to the CLI command continue with configuration dialog? n.

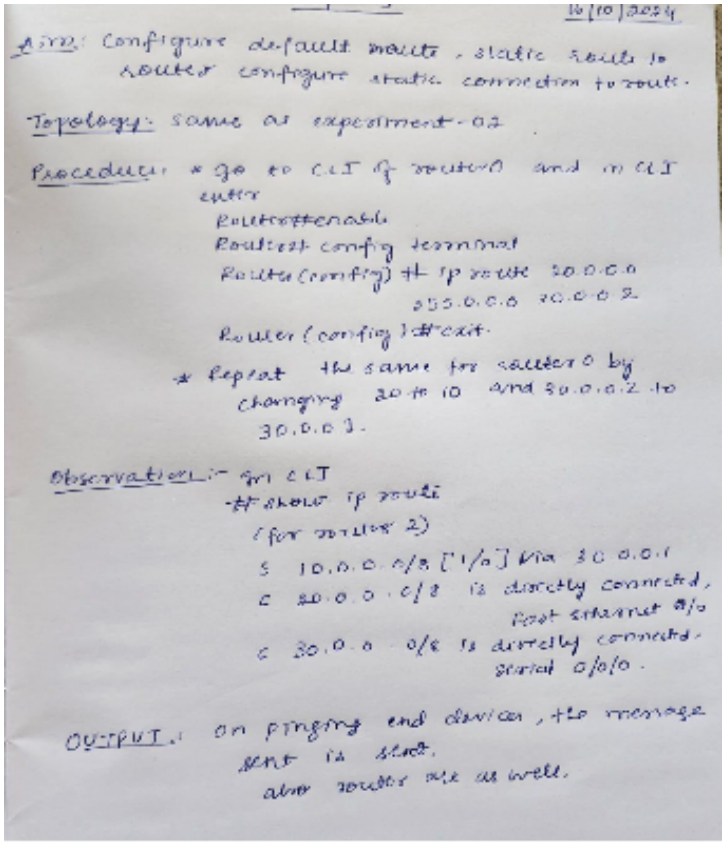


Router # enable
Router # config terminal
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # exit.

→ Similar for other Router A.

→ To configure the two routers.
  └ click on CLI of Router 0:-
    Router (config) # interface serial 2/0
    Router (config-if) # ip address 30.0.0.1 255.0.0.0
    Router (config-if) # no shutdown
    Router (config-if) # exit.
  └ Similar for the Router 1.

→ for PC0, in Desktop, in command prompt
  PC > ping 20.0.0.1
    └ gives Destination host unreachable.
  PC > ping 30.0.0.2
    └ Request timed out
  PC > ping 30.0.0.1
    └ successful



16/10/2024

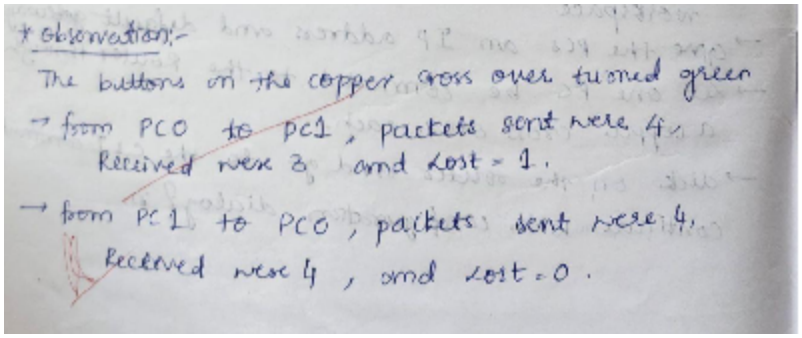
Aim: Configure default route, static route to Router configure static connection to route.

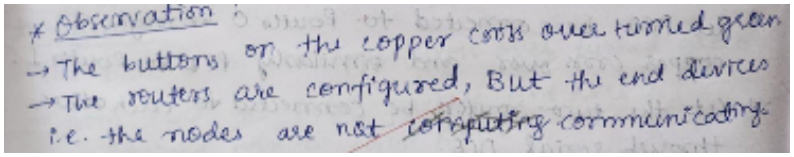
Topology: same as experiment-02

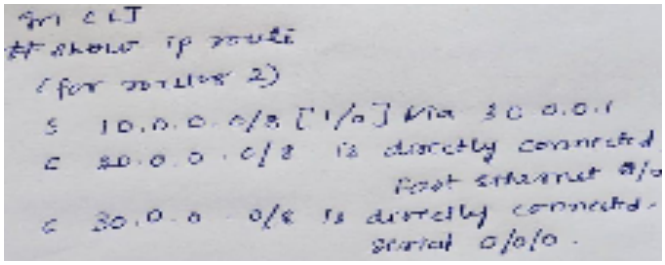
Procedure: * go to CLI of router0 and in CLI enter
  Router>enable
  Router# config terminal
  Router (config) # ip route 20.0.0.0 255.0.0.0 20.0.0.2
  Router (config) # exit.
* Repeat the same for router0 by changing 20 to 10 and 30.0.0.2 to 30.0.0.1.

Observation:- in CLI
  # show ip route
  (for router 2)
  S 10.0.0.0/8 [1/0] via 30.0.0.1
  C 20.0.0.0/8 is directly connected, FastEthernet 0/0
  C 30.0.0.0/8 is directly connected, Serial 0/0/0.

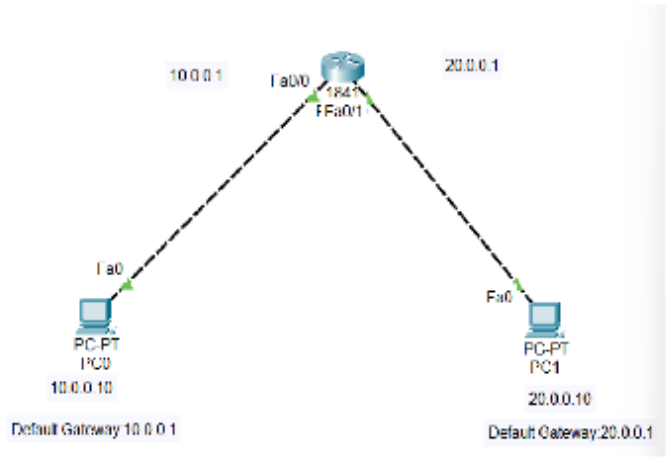
OUTPUT: on pinging end devices, the message sent is sent.
  also router pkt as well.


* Observation:-
The buttons on the copper cross over turned green
→ from PC0 to PC1, packets sent were 4,
Received were 3 and Lost = 1.
→ from PC1 to PC0, packets sent were 4,
Received were 4, and Lost = 0.

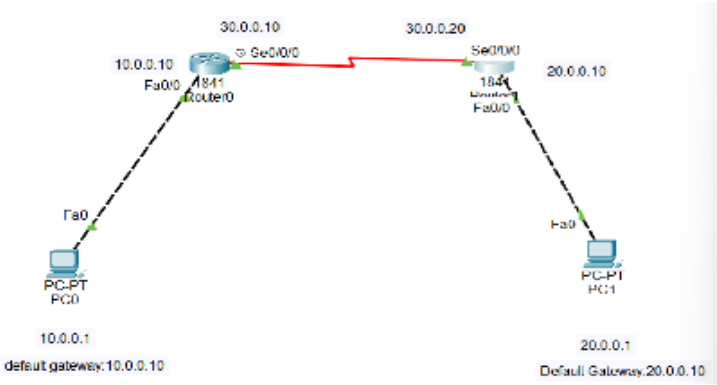

* Observation
→ The buttons on the copper cross over turned green
→ The routers are configured, But the end devices
i.e. the nodes are not computing communicating.


on pinging end devices, the message sent is sent.
also routers are as well.

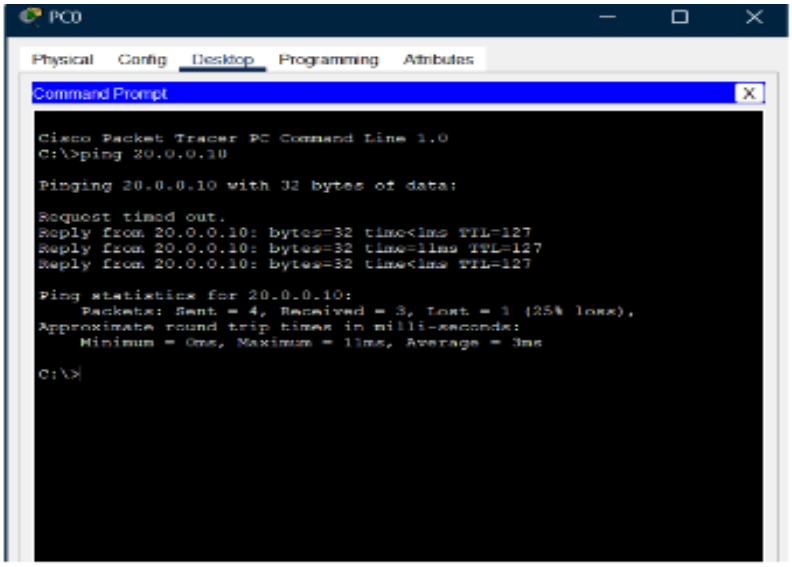
Topology:



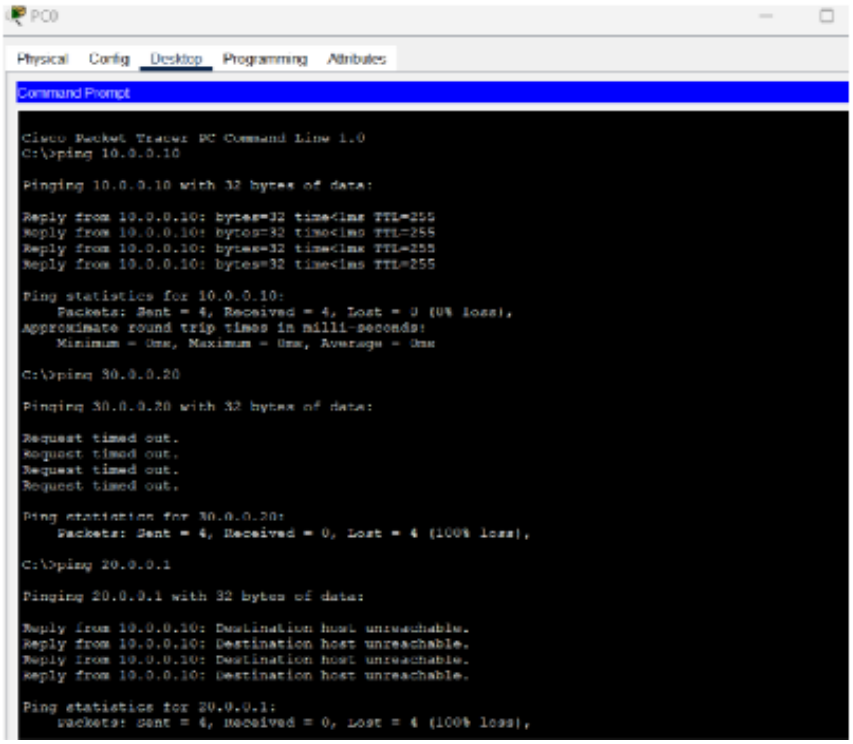
Topology:



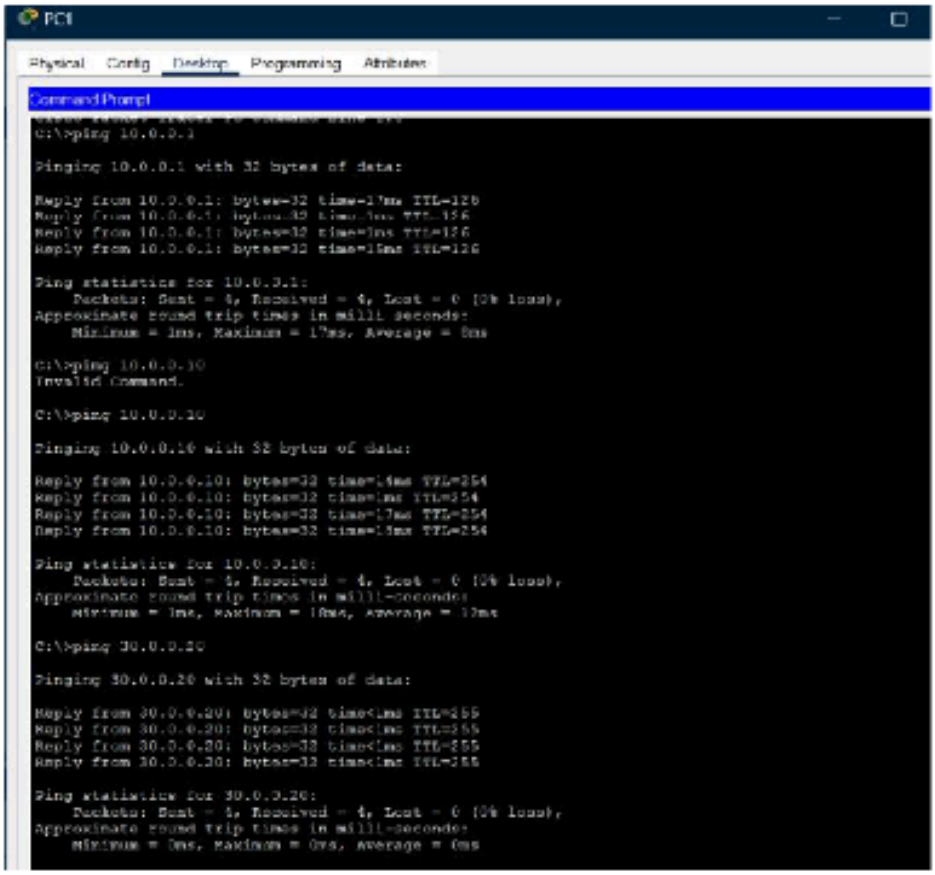
Output:



Output:(Before Static Routing):



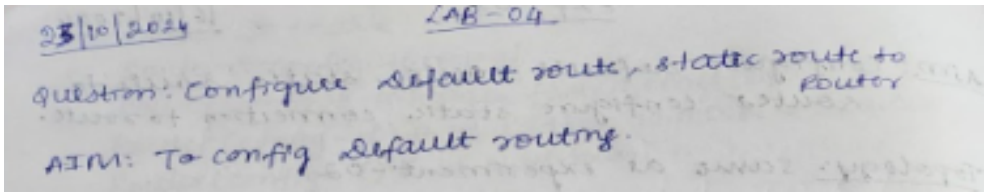
(After Static Routing):



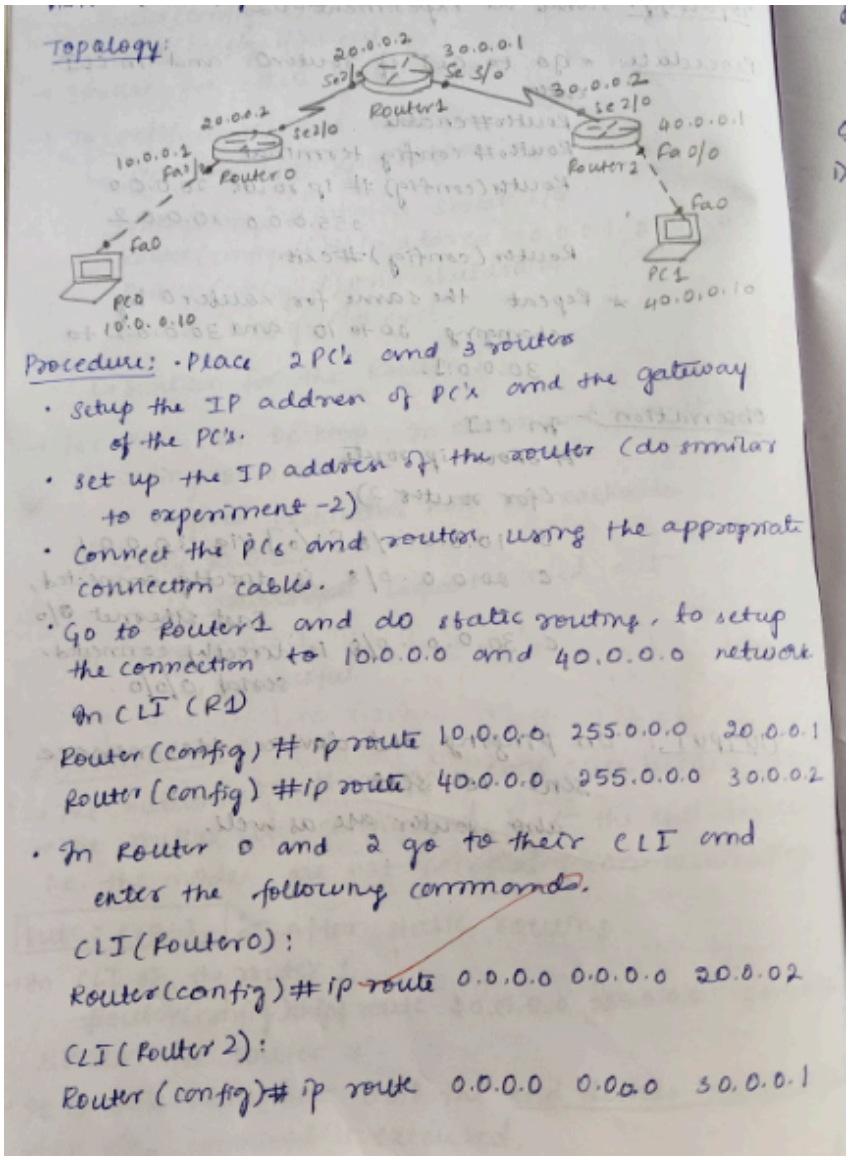
Program 3

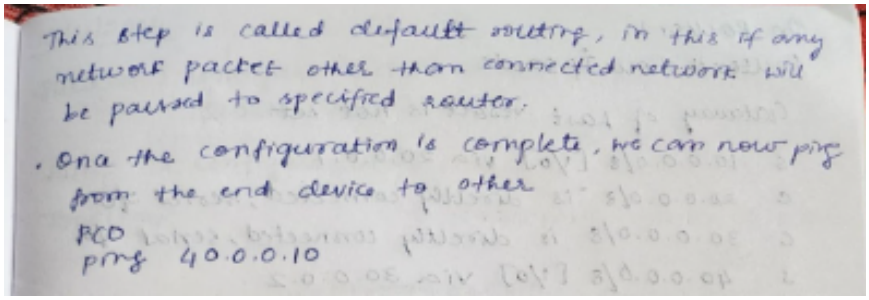
Question:

Configure default route, static route to the Router



Observation writeup:

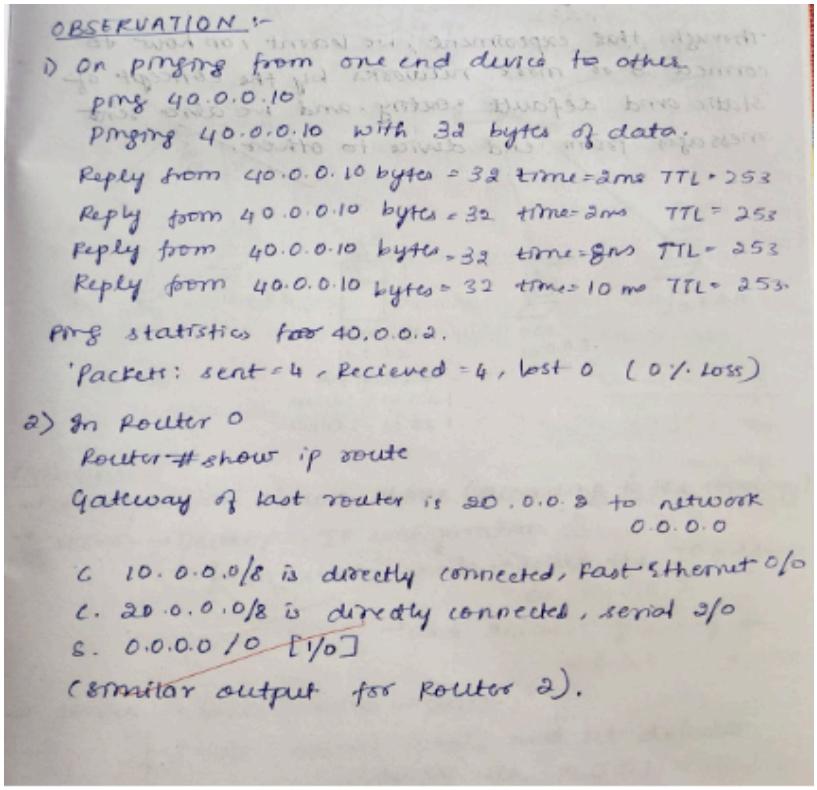




This step is called default routing, in this if any network packet other than connected network will be passed to specified router.

. Once the configuration is complete, we can now ping from the end device to other

PC0
ping 40.0.0.10



OBSERVATION :-

1) On pinging from one end device to other

ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data.

Reply from 40.0.0.10 bytes = 32 time=2ms TTL=253
Reply from 40.0.0.10 bytes = 32 time=2ms TTL=253
Reply from 40.0.0.10 bytes = 32 time=8ms TTL=253
Reply from 40.0.0.10 bytes = 32 time=10 ms TTL=253

Ping statistics for 40.0.0.2.

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

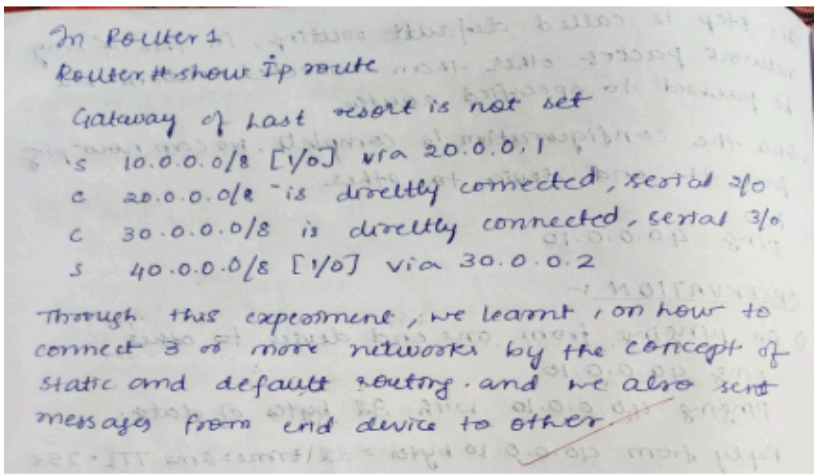
2) In Router 0

Router#show ip route

Gateway of last router is 20.0.0.2 to network 0.0.0.0

C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C. 20.0.0.0/8 is directly connected, serial 2/0
S. 0.0.0.0/0 [1/0]

(Similar output for Router 2).



In Router 1

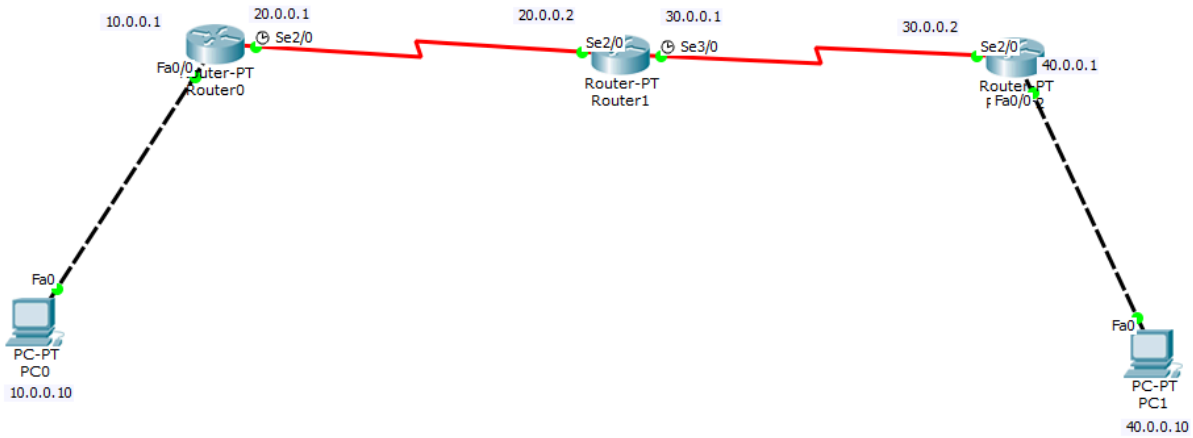
Router#show ip route

Gateway of last resort is not set

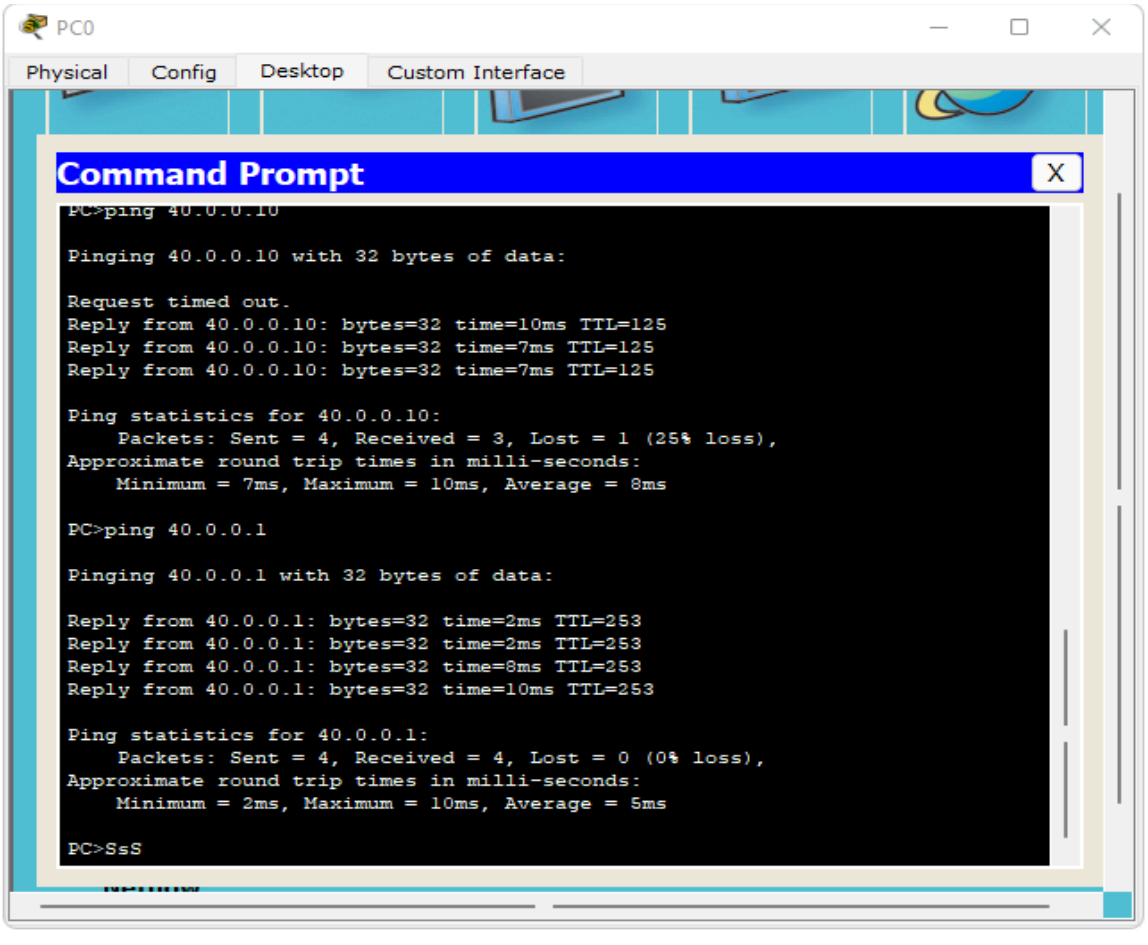
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, serial 2/0
C 30.0.0.0/8 is directly connected, serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

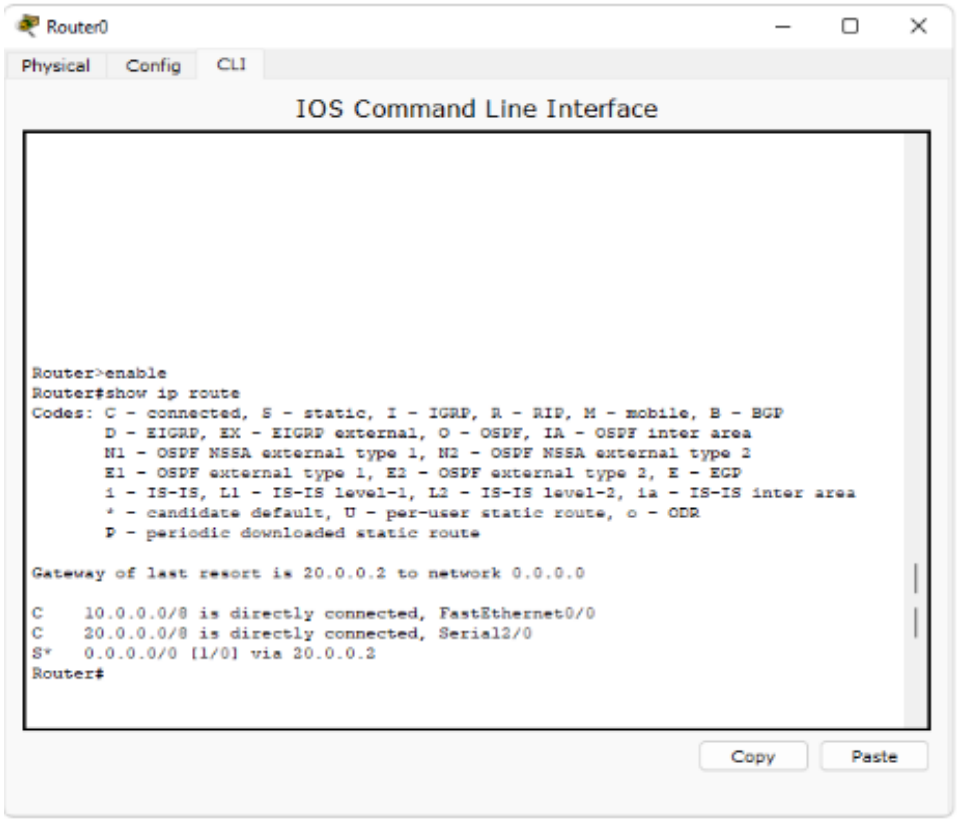
Through this experiment, we learnt, on how to connect 3 or more networks by the concept of static and default routing. and we also sent messages from end device to other

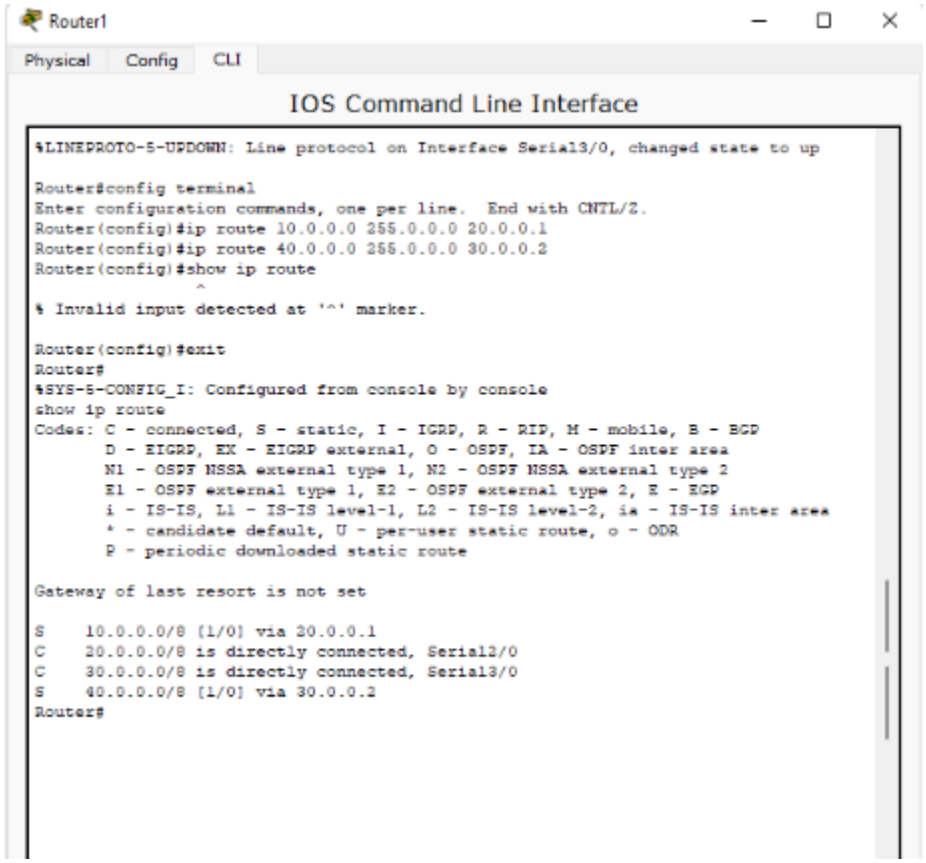
Topology:


10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
Se2/0
40.0.0.1
Router-PT
Fa0/0
Fa0
PC-PT
PC0
10.0.0.10
Fa0
PC-PT
PC1
40.0.0.10

iii. Output:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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=10ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 10ms, Average = 8ms
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=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=10ms TTL=253
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 10ms, Average = 5ms
PC>SsS

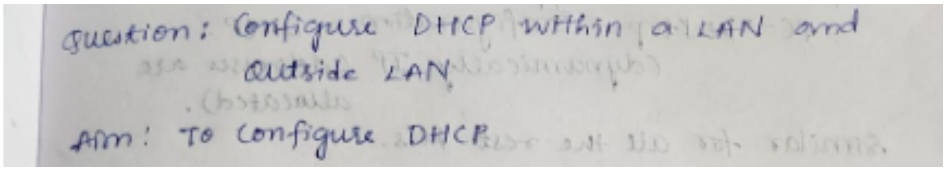

Router0
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is 20.0.0.2 to network 0.0.0.0
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2
Router#
Copy
Paste


Router1
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
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
Router(config)#show ip route
^
% Invalid input detected at '^' marker.
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
Router#

Program 4

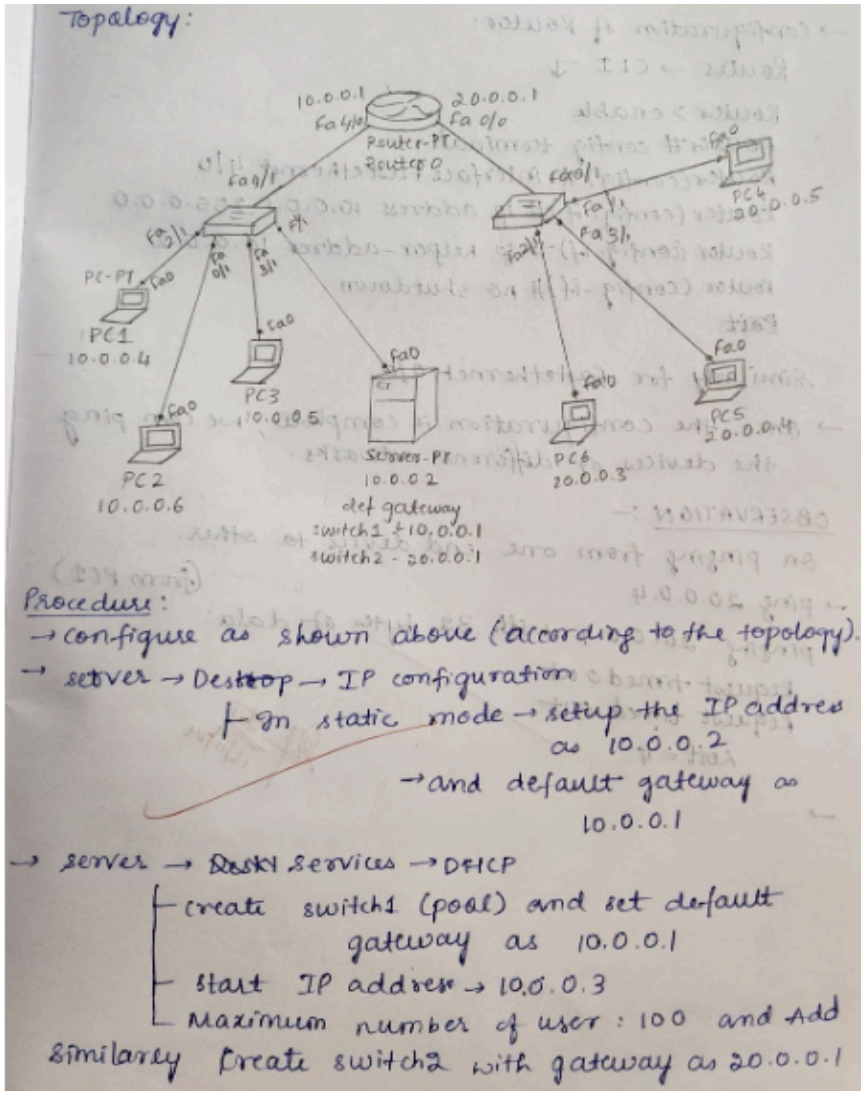
Question:

Configure DHCP within a LAN and outside LAN


Question: Configure DHCP within a LAN and Outside LAN

Aim: To Configure DHCP

Observation writeup:



→ setup the IP addresses of the PC's by DHCP

PC → Desktop → IP Configuration → DHCP

(dynamically IP addresses are allocated).

Similar for all the rest PCs.

→ Configuration of Router:

Router → CLI ↴

```
Router > enable
Router# config terminal
Router(config)# interface fastethernet 4/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# ip helper-address 10.0.0.2
Router(config-if)# no shutdown
Exit
```

Similarly for fastethernet 0/0.

→ Once the configuration is complete, we can ping the devices of different networks.

OBSERVATION :-

On pinging from one end device to other.

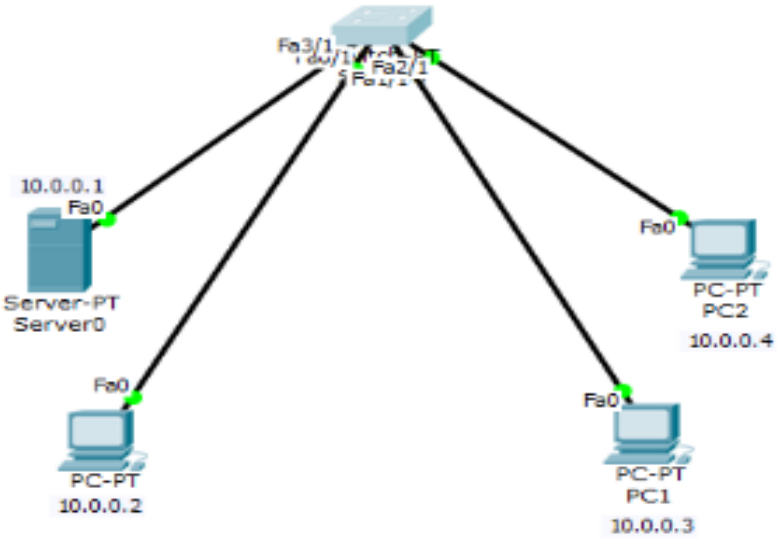
→ ping 20.0.0.4 (from PC1)

```
pinging 20.0.0.4 with 32 bytes of data:
Request timed out
Request timed out
Lost = 4
```

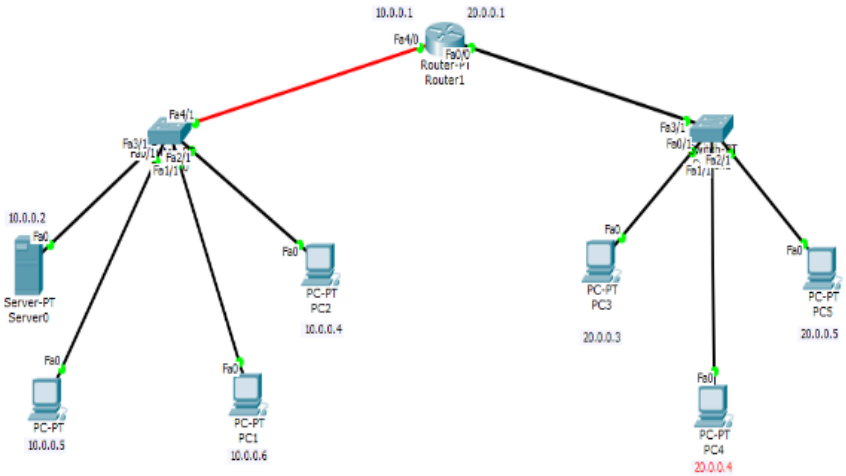
thu 13/11/24

Topology:

(within Lan)

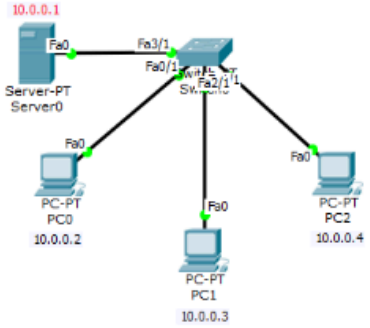


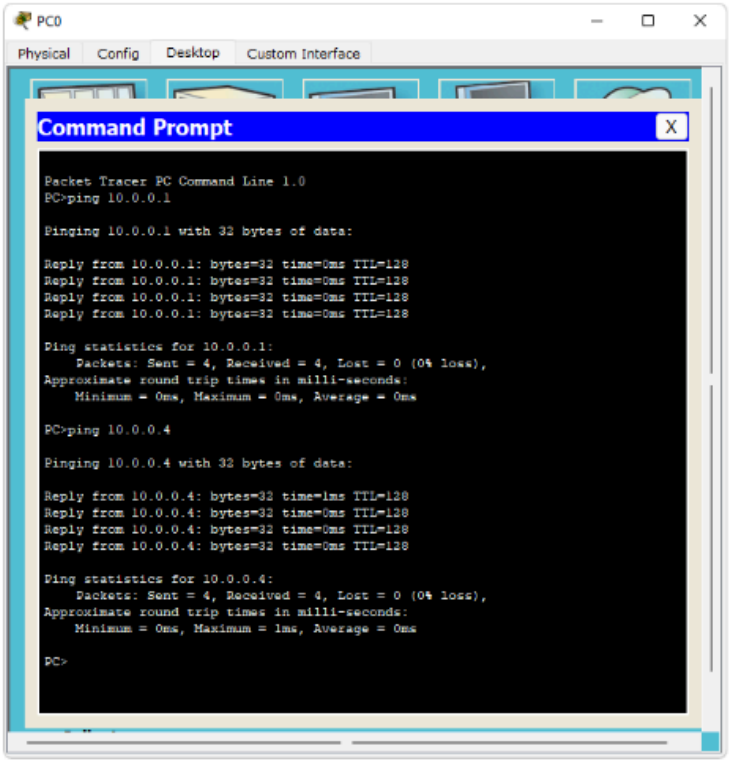
(outside Lan)



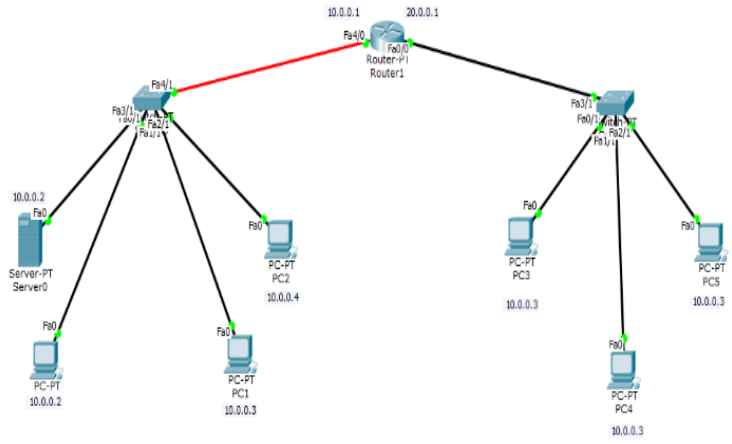
Output:

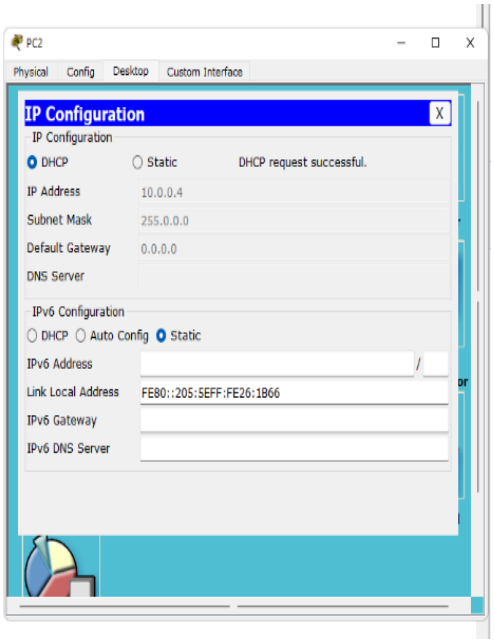
(within Lan)

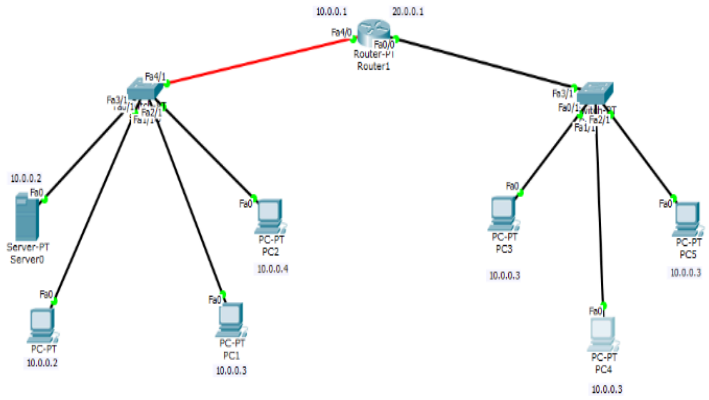


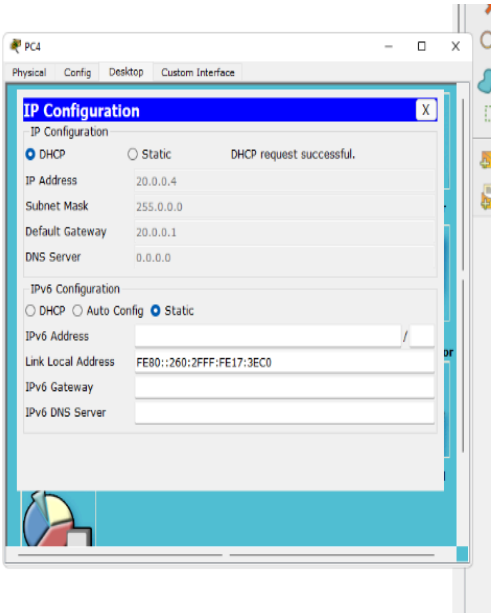


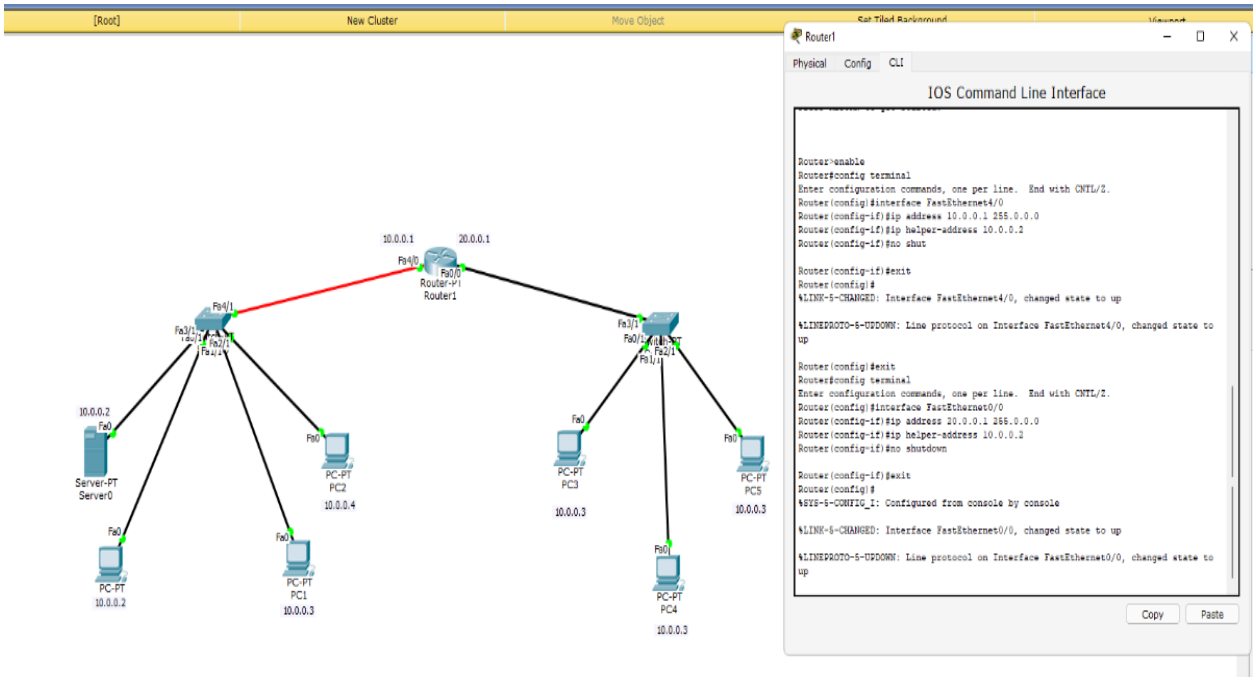
(outside Lan)






10.0.0.1
20.0.0.1
Router-PT
Router1
Server-PT
Server0
10.0.0.2
PC-PT
PC2
10.0.0.4
PC-PT
PC1
10.0.0.3
PC-PT
10.0.0.2
PC-PT
PC3
10.0.0.3
PC-PT
PC4
10.0.0.3
PC-PT
PC5
10.0.0.3

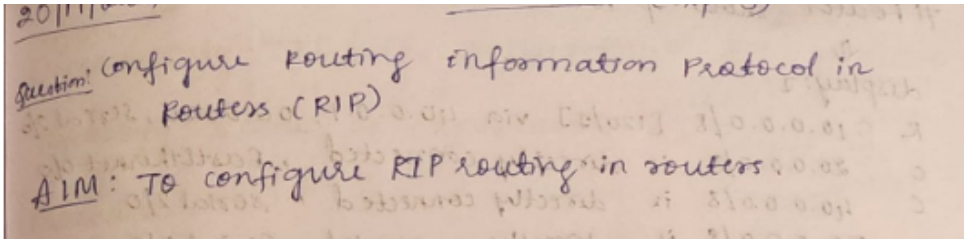

PC4
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
20.0.0.4
Subnet Mask
255.0.0.0
Default Gateway
20.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::260:2FFF:FE17:3EC0
IPv6 Gateway
IPv6 DNS Server


[Root]
New Cluster
Move Object
10.0.0.1
20.0.0.1
Router-PT
Router1
Server-PT
Server0
10.0.0.2
PC-PT
PC2
10.0.0.4
PC-PT
PC1
10.0.0.3
PC-PT
10.0.0.2
PC-PT
PC3
10.0.0.3
PC-PT
PC4
10.0.0.3
PC-PT
PC5
10.0.0.3
Router1
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet4/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
%LINK-5-CHANGED: Interface FastEthernet4/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet4/0, changed state to up
Router(config)#exit
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#ip helper-address 10.0.0.3
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
%SYS-5-CONFIG_I: Configured from console by console
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Copy
Paste

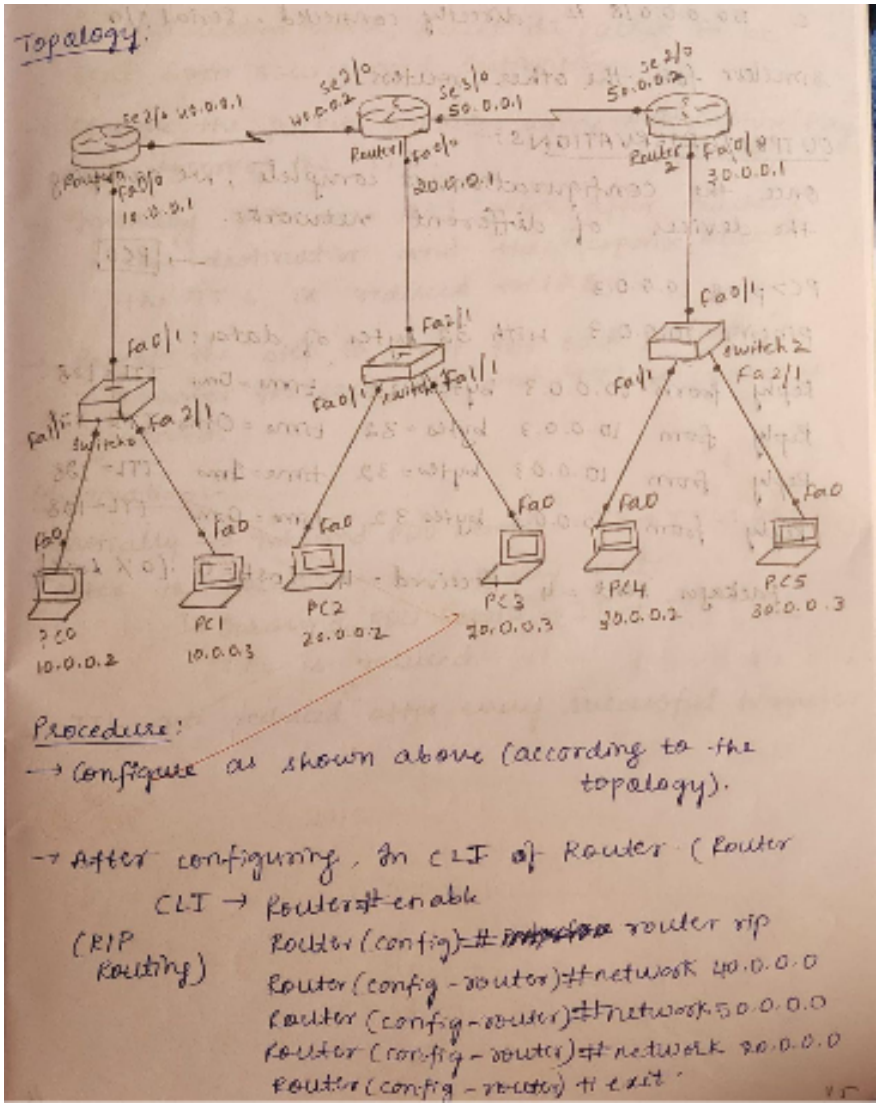
Program 5

Question:

Configure RIP routing Protocol in Routers



Observation writeup:



Router show ip route

display:2

R 10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:24, Serial 2/0
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
C 40.0.0.0/8 is directly connected, Serial 2/0
C 50.0.0.0/8 is directly connected, Serial 3/0

similar for the other routers.

OUTPUT/OBSERVATIONS:-

Once the configuration is complete, we can ping the devices of different networks.

→ PC0

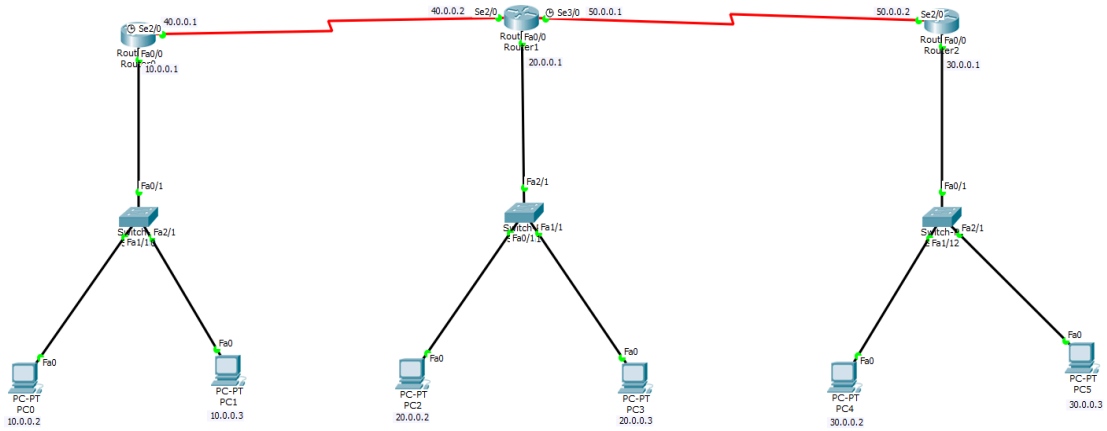
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

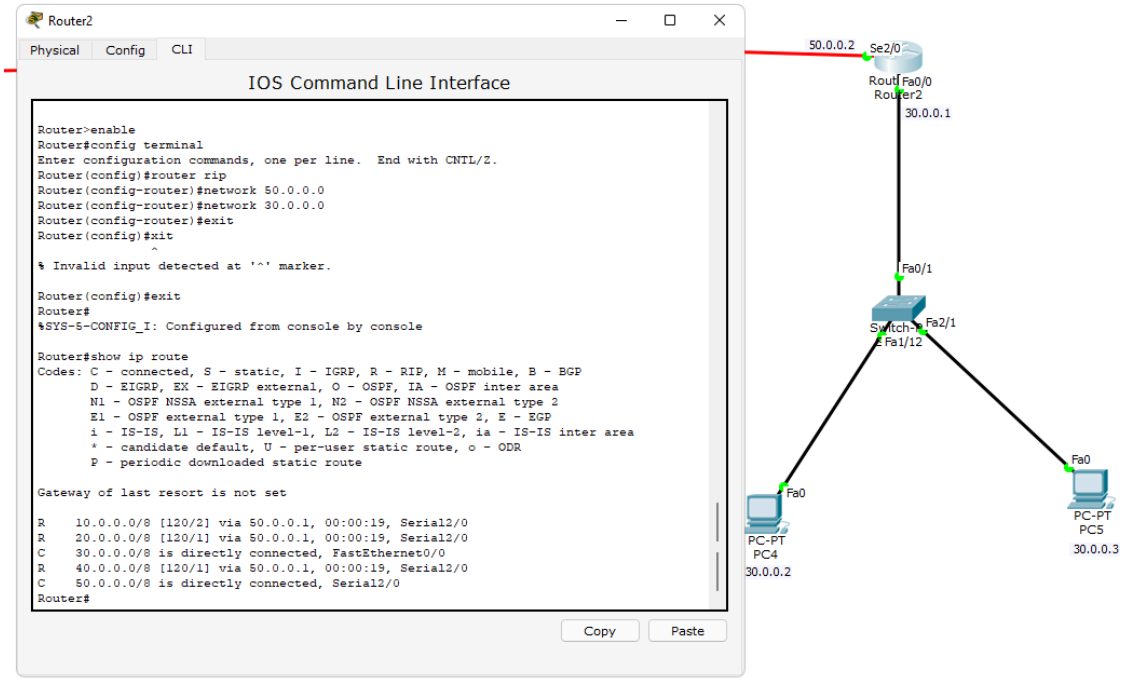
Reply from 10.0.0.3 bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 bytes 32 time=0ms TTL=128

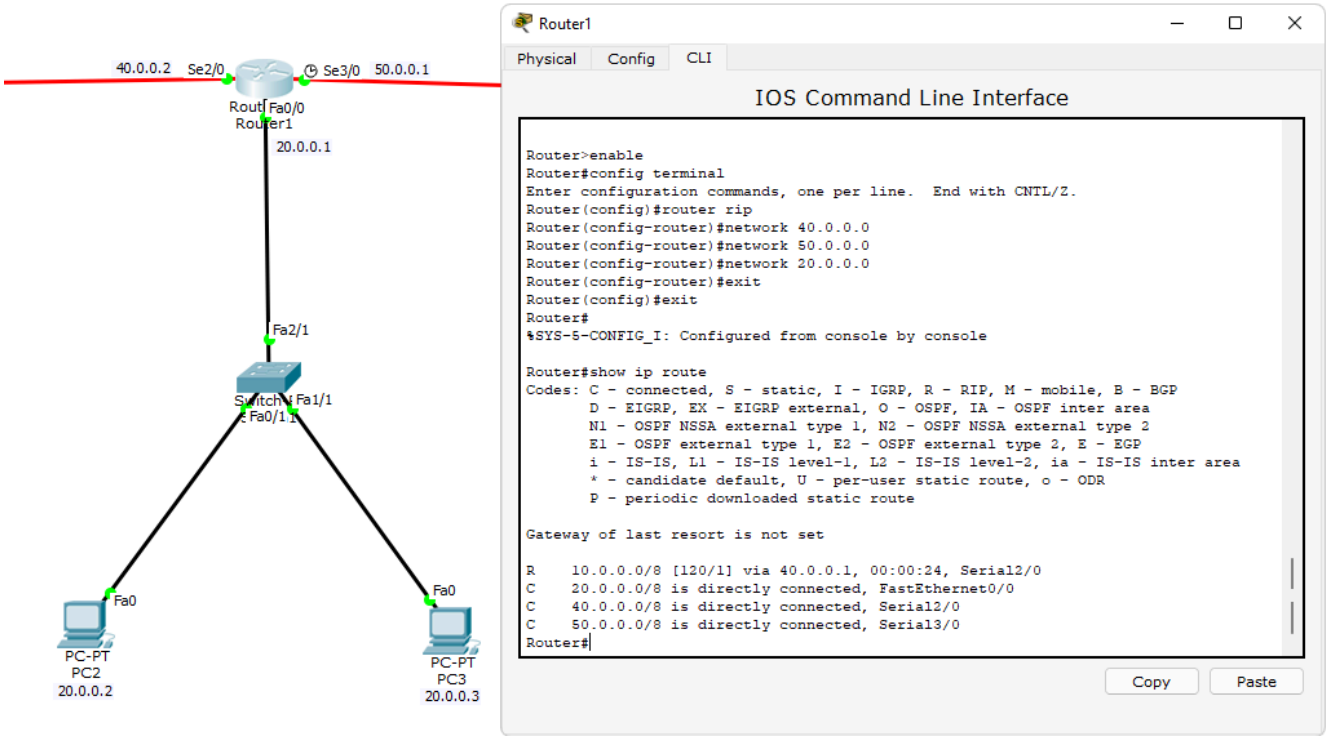
Packages sent = 4, Received = 4, Lost = 0 (0% Loss).

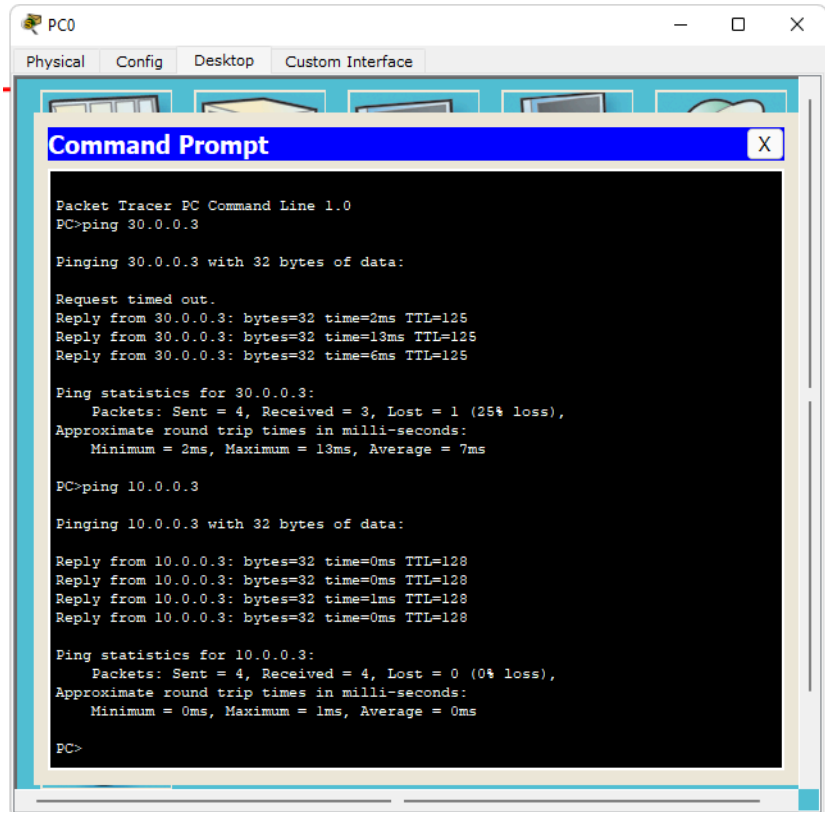
Topology:

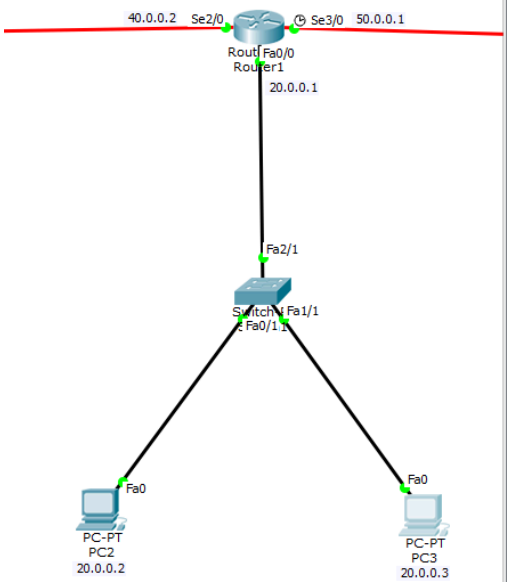


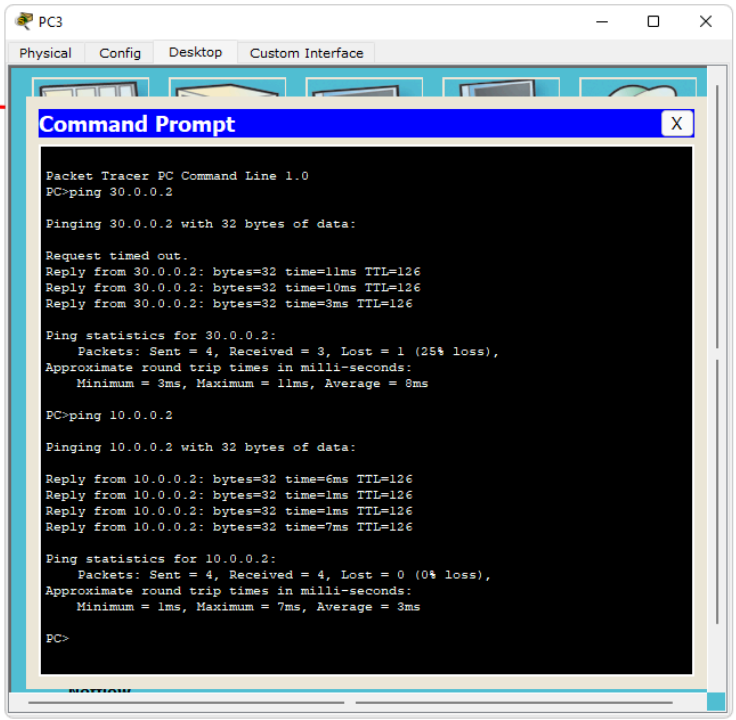
Output:


Router2
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 50.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#xit
^
% Invalid input detected at '^' marker.
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 [120/2] via 50.0.0.1, 00:00:19, Serial2/0
R 20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:19, Serial2/0
C 30.0.0.0/8 is directly connected, FastEthernet0/0
R 40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:19, Serial2/0
C 50.0.0.0/8 is directly connected, Serial2/0
Router#
Copy
Paste
50.0.0.2
Se2/0
Fa0/0
30.0.0.1
Fa0/1
Fa2/1
Fa1/12
Fa0
PC-PT
PC4
30.0.0.2
Fa0
PC-PT
PC5
30.0.0.3


40.0.0.2
Se2/0
Se3/0
50.0.0.1
Fa0/0
20.0.0.1
Fa2/1
Fa1/1
Fa0/1/1
Fa0
PC-PT
PC2
20.0.0.2
Fa0
PC-PT
PC3
20.0.0.3
Router1
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#config terminal
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)#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 40.0.0.1, 00:00:24, Serial2/0
C 20.0.0.0/8 is directly connected, FastEthernet0/0
C 40.0.0.0/8 is directly connected, Serial2/0
C 50.0.0.0/8 is directly connected, Serial3/0
Router#
Copy
Paste


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.3
Pinging 30.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 30.0.0.3: bytes=32 time=2ms TTL=125
Reply from 30.0.0.3: bytes=32 time=13ms TTL=125
Reply from 30.0.0.3: bytes=32 time=6ms TTL=125
Ping statistics for 30.0.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 13ms, Average = 7ms
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

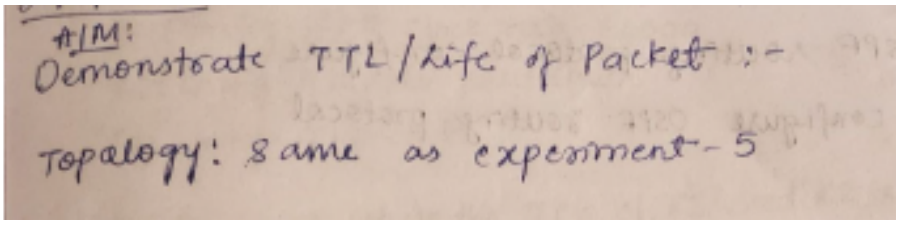

40.0.0.2
Se2/0
Se3/0
50.0.0.1
Router1
Fa0/0
20.0.0.1
Fa2/1
Switch1
Fa1/1
Fa0/1
Fa0
PC-PT
PC2
20.0.0.2
Fa0
PC-PT
PC3
20.0.0.3


PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 30.0.0.2: bytes=32 time=11ms TTL=126
Reply from 30.0.0.2: bytes=32 time=10ms TTL=126
Reply from 30.0.0.2: bytes=32 time=3ms TTL=126
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 3ms, Maximum = 11ms, Average = 8ms
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=6ms TTL=126
Reply from 10.0.0.2: bytes=32 time=1ms TTL=126
Reply from 10.0.0.2: bytes=32 time=1ms TTL=126
Reply from 10.0.0.2: bytes=32 time=7ms TTL=126
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 7ms, Average = 3ms
PC>

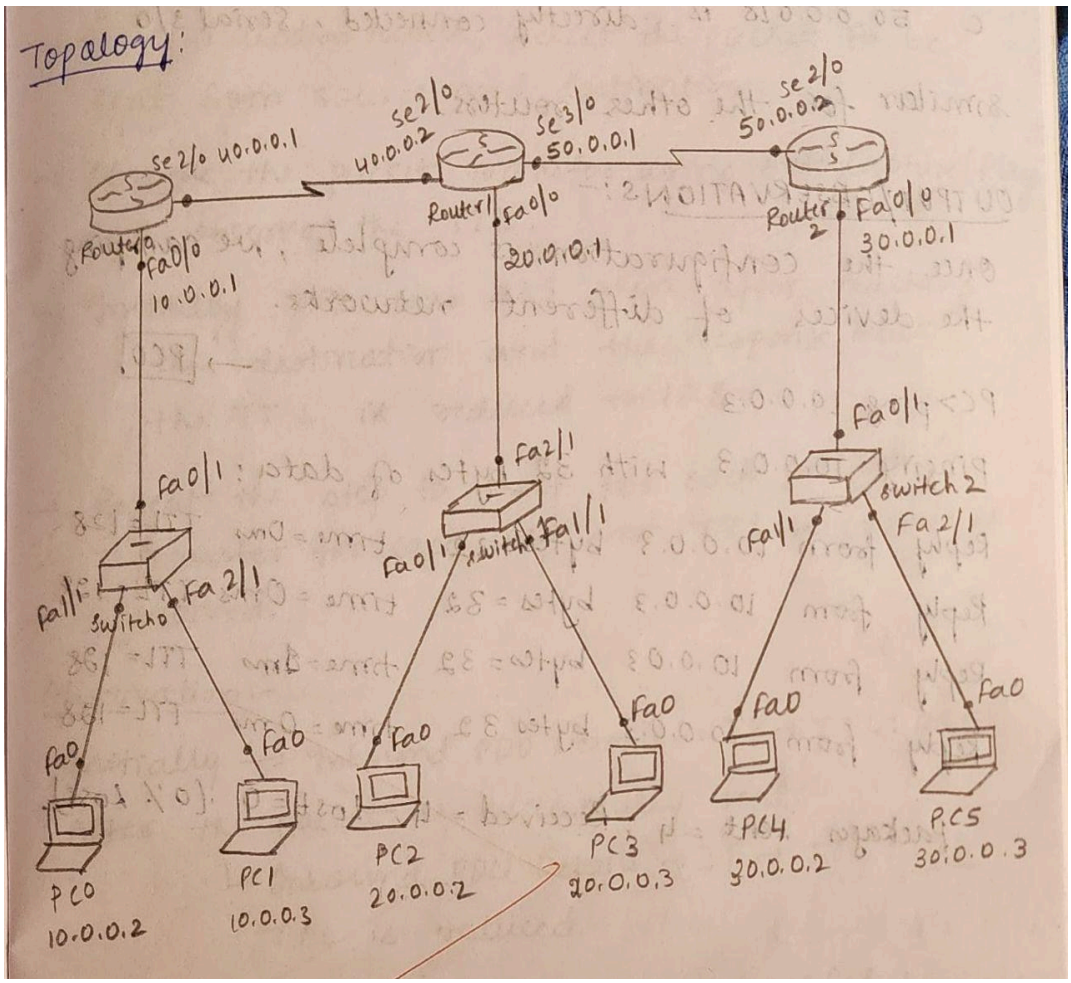
Program 6

Question:

Demonstrate the TTL/ Life of a Packet



Observation writeup:



Procedure:

→ In simulation Mode, select the packet to be sent from source and destination.

→ Observe the packet transfer using Auto Capture/Play and observe the TTL.

→ Initially TTL was 225 then after reaching the destination and the response back the TTL is reduced to 128.

→ Repeat the step to check for each packets transfer for each step and TTL decrease is observed.

Observation:-

Initially → Inbound PDU Details → IP → TTL: 255

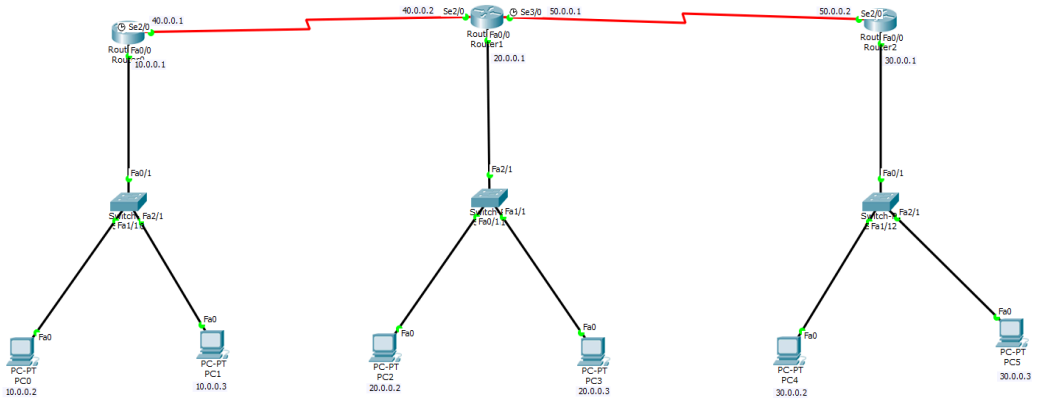
After the packet is successfully sent

↳ Inbound PDU Details → IP → TTL: 126

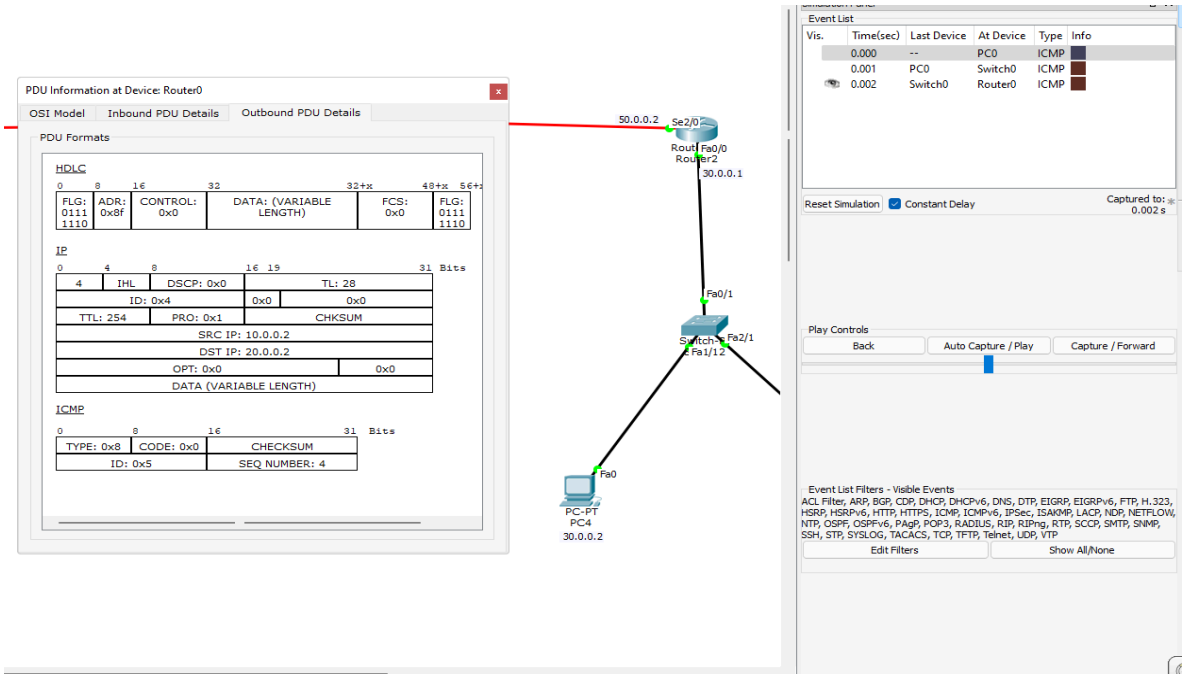
TTL is reduced.

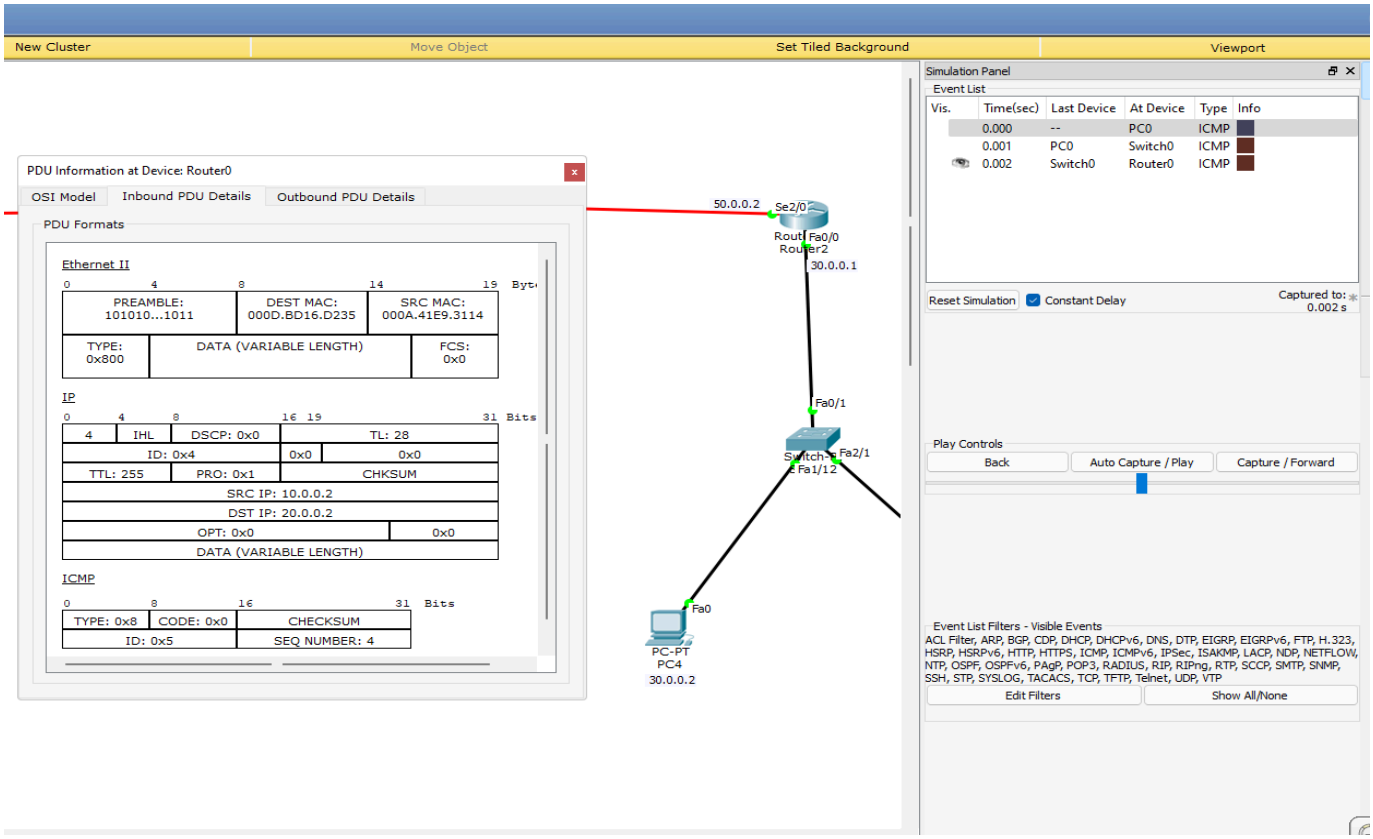
• TTL gets reduced after every successful transfer

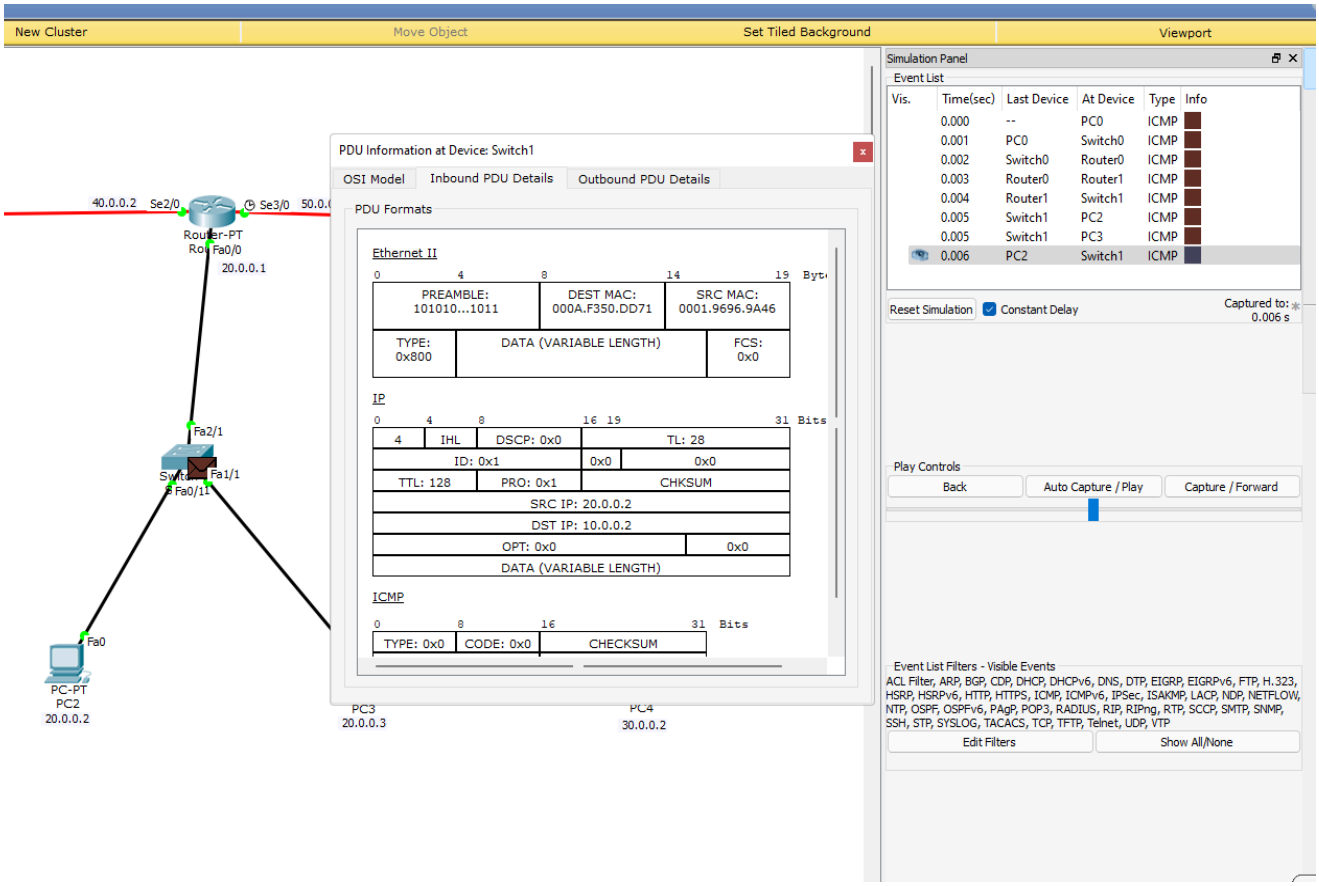
12/4/24



Output:


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: 0x4
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x5
SEQ NUMBER: 4
50.0.0.2
Se2/0
Router2
Fa0/0
30.0.0.1
Fa0/1
Switch
Fa1/12
Fa2/1
Fa0
PC-PT
PC4
30.0.0.2
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.000
--
PC0
ICMP
0.001
PC0
Switch0
ICMP
0.002
Switch0
Router0
ICMP
Reset Simulation
Constant Delay
Captured to:
0.002 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None


New Cluster
Move Object
Set Tiled Background
Viewport
PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 000D.BD16.D235
SRC MAC: 000A.41E9.3114
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x4
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x5
SEQ NUMBER: 4
50.0.0.2
Se2/0
Router2
Fa0/0
30.0.0.1
Fa0/1
Switch
Fa1/12
Fa2/1
Fa0
PC-PT
PC4
30.0.0.2
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.000
--
PC0
ICMP
0.001
PC0
Switch0
ICMP
0.002
Switch0
Router0
ICMP
Reset Simulation
Constant Delay
Captured to:
0.002 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None


New Cluster
Move Object
Set Tiled Background
Viewport
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.000 -- PC0 ICMP
0.001 PC0 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Router1 ICMP
0.004 Router1 Switch1 ICMP
0.005 Switch1 PC2 ICMP
0.005 Switch1 PC3 ICMP
0.006 PC2 Switch1 ICMP
Reset Simulation
Constant Delay
Captured to: 0.006 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
PDU Information at Device: Switch1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 000A.F350.DD71
SRC MAC: 0001.9696.9A46
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4 IHL DSCP: 0x0 TL: 28
ID: 0x1 0x0 0x0
TTL: 128 PRO: 0x1 CHKSUM
SRC IP: 20.0.0.2
DST IP: 10.0.0.2
OPT: 0x0 0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x0 CODE: 0x0 CHECKSUM
40.0.0.2
Se2/0
Se3/0
Router-PT
Fa0/0
20.0.0.1
Fa2/1
Fa1/1
Fa0/1
Fa0
PC-PT
PC2
20.0.0.2
PC3
20.0.0.3
PC4
30.0.0.2

Program 7

Question:

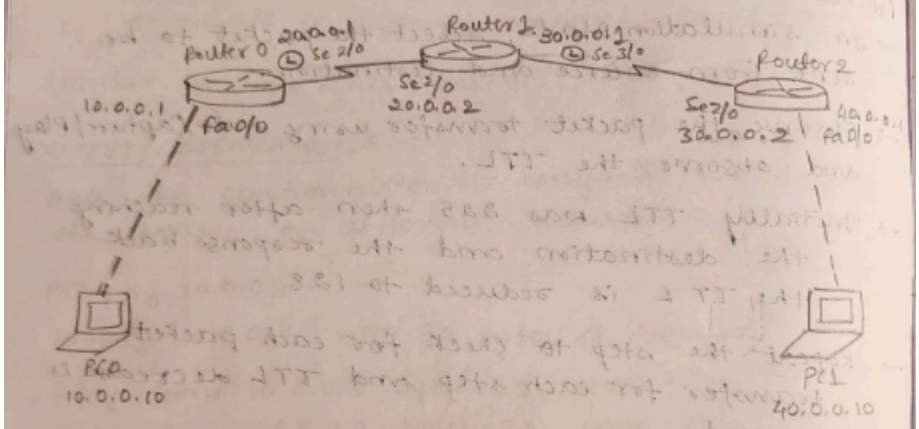
Configure OSPF routing protocol

Question: OSPF routing protocol configure

Aim: To configure OSPF routing protocol

Observation writeup:

Topology:



Procedure:

(1) Connect the device in the same manner as shown above.
Click on end devices → config → settings → set the default gateway (IP address of it's router) → then click on fast ethernet() → set the IP address of the end device and subnet mask.

Click on Router

For Router0 → CLI

(setting up fast Ethernet)

R0 (config)# interface fastethernet 0/0

R0 (config-if)# ip address 10.0.0.1 255.0.0.0

R0 (config-if)# no shutdown

R0 (config-if)# exit

(setting up serial connection)

R0 (config)# interface serial 2/0

R0 (config)# IP address 20.0.0.1 255.0.0.0

R0(config-if)# encapsulation PPP
R0 (config-if)# clock rate 64000
R0(config-if)#no shutdown
R0(config-if)#exit.

similarly we set up the IP's of R1 and R2 while the setup of fast Ethernet remains, the setting up of serial connections has 2 extra lines (encapsulation PPP, clock rate 64000) clock rate 6400 must only be written if the serially connected port show a ⓒ symbal.

Here, we write clock rate command for R0 serial 2/0, R1 serial 3/0

After this step all the connected must have turned green.

(2.) To enable IP routing by configuring OSPF routing protocol in all routers.

Router R0 → CLI
R0 (config)#router OSPF 1
R0 (config -router) # router -id 1.1.1.1
R0 (config - router)-# network 10.0.0.0 0.255.255.255 area3
R0 (config- router)# network 20.0.0.0 0.255.255.255 area1
R0 (config- router)#exit

Router R1 - CLI
R1 (config)#router ospf 1
R1 (config-router)#router-id 2.2.2.2
R1 (config-router)#network 20.0.0.0 0.255.255.255 area1
R1 (config-router)#network 30.0.0.0 0.255.255.255 area0
R1 (config-router)#exit

On router R2 → CLI

```
R2(config)#router ospf 1
R2(config-router)#router-id 3.3.3.3
R2(config-router)#network 30.0.0.0 0.255.255.255 area 0
R2(config-router)#network 40.0.0.0 0.255.255.255 area 2
R2(config-router)#exit
```

(3) Once the setting up of networking area is done use configure loopback address to router.

```
R0(config-if)#interface loopback 0
R0(config-if)#ip add 172.16.1.252 255.255.0.0
R0(config-if)#no shutdown

R1(config-if)#interface loopback 0
R1(config-if)#ip add 172.16.1.253 255.255.0.0
R1(config-if)#no shutdown

R2(config-if)#interface loopback 0
R2(config-if)#ip add 172.16.1.254 255.255.0.0
R2(config-if)#no shutdown
```

(4) On checking routing table of R2 using show ip route we can see that R2 doesn't know about area 3 Gateway of last resort is not set.

```
O IA  20.0.0.0 0/8 [110/125] via 30.0.0.1 serial 1/0
C     40.0.0.0/8 is directly connected fastEthernet 0/0
C     30.0.0.0/8 is directly connected serial 2/0
```

since R2 doesn't know about area3, we have to create a virtual link between R0 and R1.

(5) Creating Virtual link between R1, R0

On Router R0

```
R0(config)#router ospf 1
R0(config-router)#area 1 virtual-link 2.2.2.2
R0(config-router)#exit
```

On Router R1

```
R1(config)#router ospf 1
R1(config-router)#area 1 virtual-link 1.1.1.1
R1(config-router)#exit
```

(6) Now, check routing table of R2

Once all these steps are completed, the message can now be pinged from 1 end device to other.

OBSERVATION:

In R2,

Router# show ip route

O IA 20.0.0.0/8 [110/128] via 30.0.0.1 09:57:23, Serial 2/0

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

O IA 10.0.0.0/8 [110/128] via 30.0.0.1 00:57:05, Serial 2/0

C 30.0.0.0/8 is directly connected, Serial 2/0

C 172.16.0.0/16 is directly connected, Loopback 0

similarly the output is shown for Router 0 and 1

Ping output:

(from PC0 to PC1)

PC0 → command prompt

C:\> ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data

Request timed out

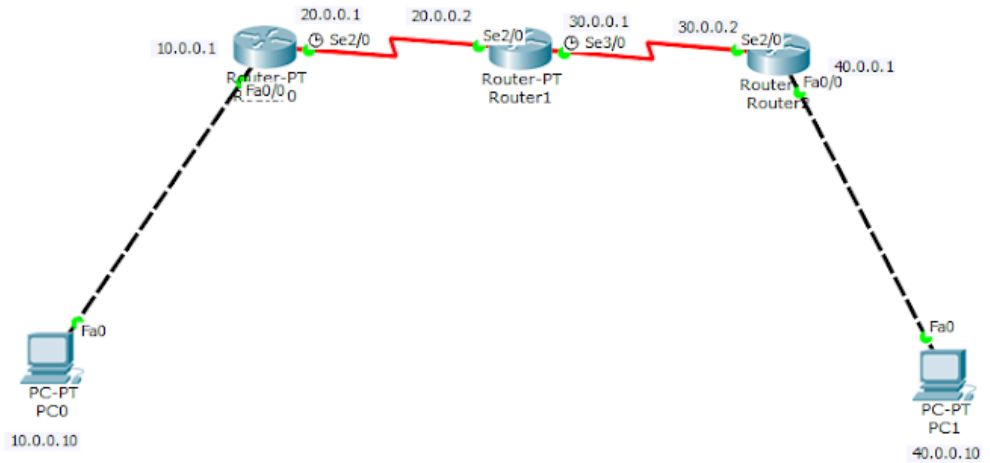
Reply from 40.0.0.10 bytes=32 time=21ms TTL=125

Reply from 40.0.0.10 bytes=32 time=2ms TTL=125

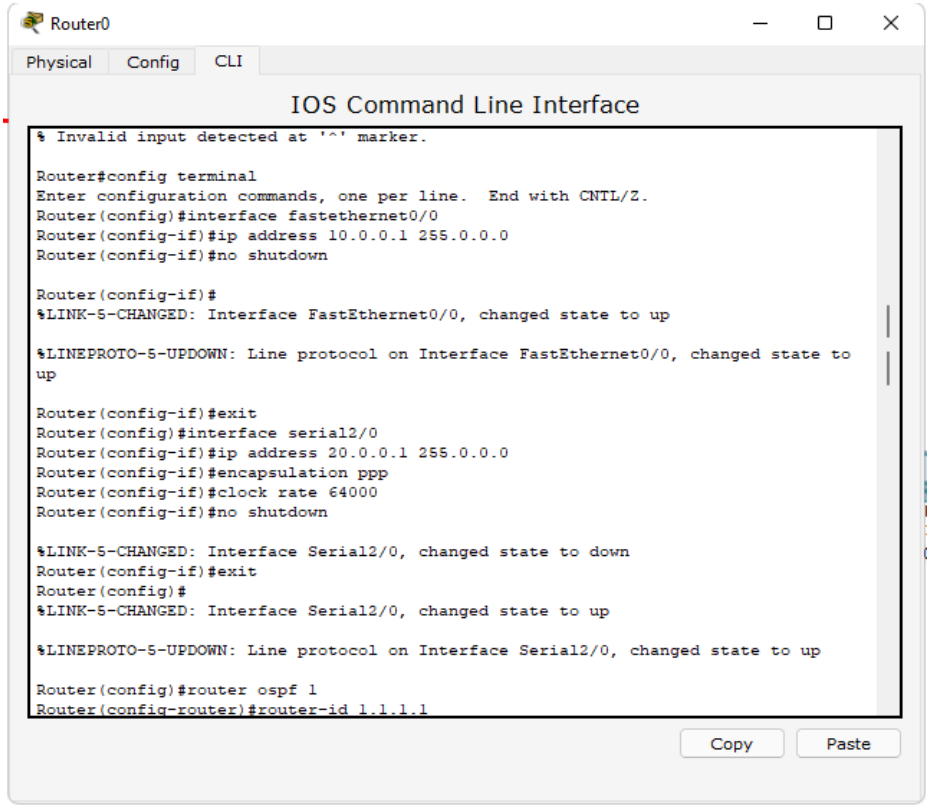
Reply from 40.0.0.10 bytes=32 time=28ms TTL=125

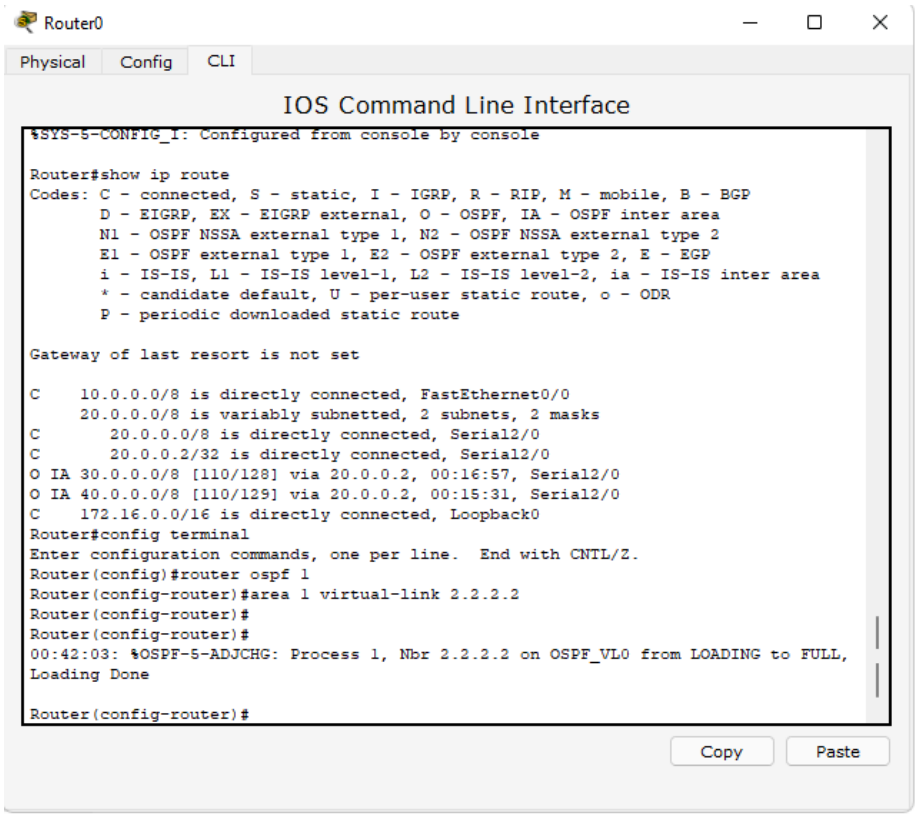
Pinging statistics for 40.0.0.10

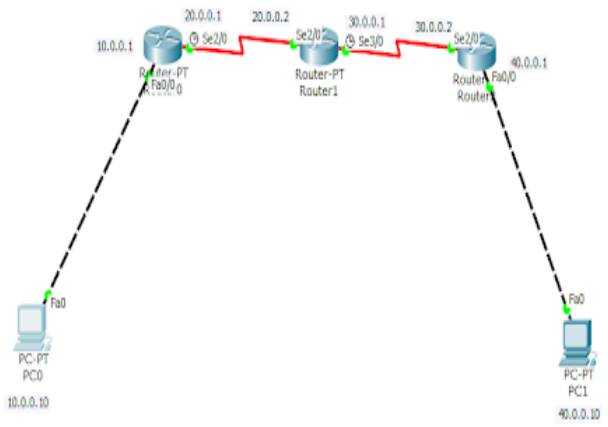
Packets: sent=4, received=3, lost=1 (25% loss).

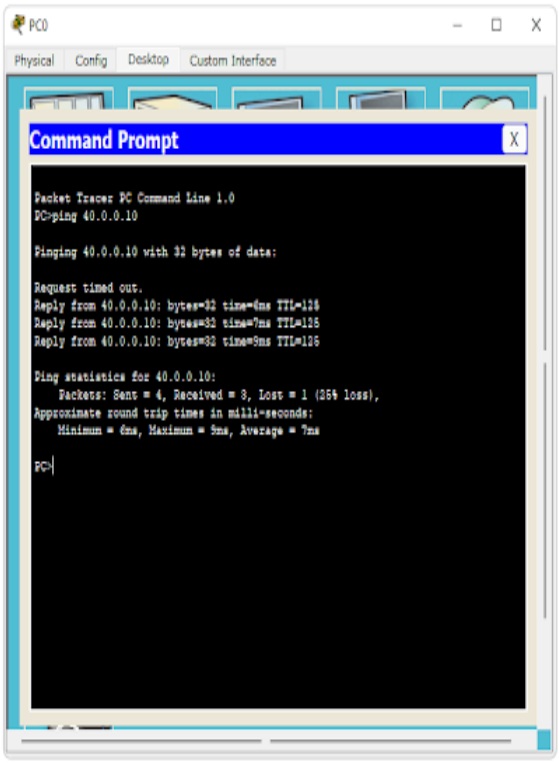


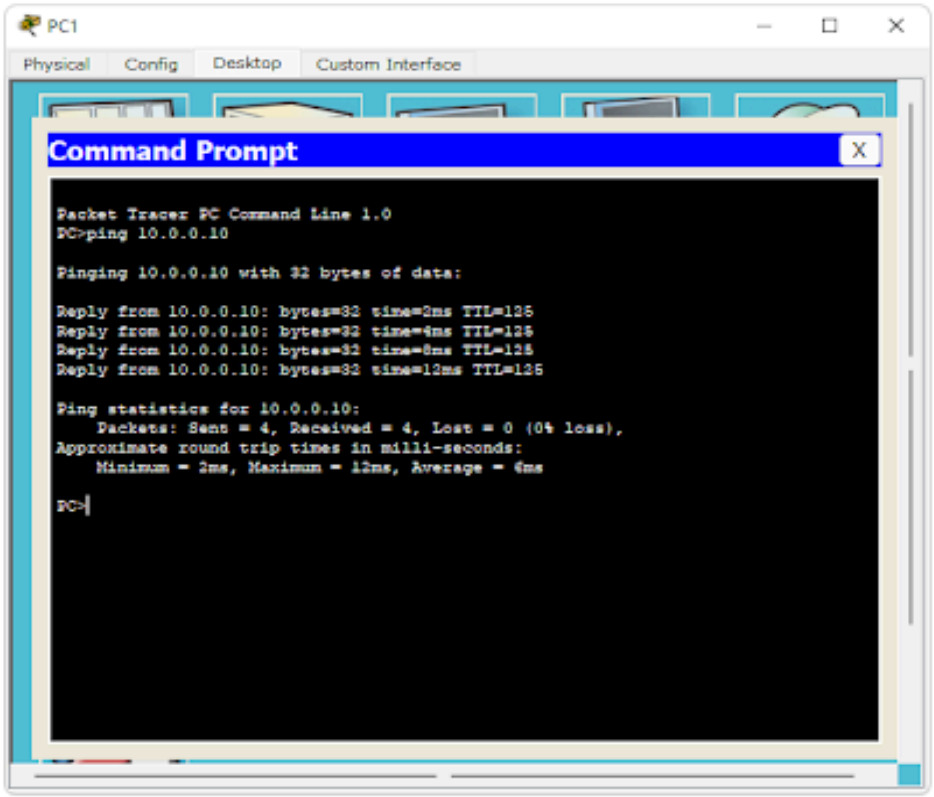
Output:


Router0
Physical
Config
CLI
IOS Command Line Interface
% Invalid input detected at '^' marker.
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#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 serial2/0
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)#
%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)#router ospf 1
Router(config-router)#router-id 1.1.1.1
Copy
Paste


Router0
Physical
Config
CLI
IOS Command Line Interface
%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
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, 00:16:57, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:15:31, Serial2/0
C 172.16.0.0/16 is directly connected, Loopback0
Router#config terminal
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)#
Router(config-router)#
00:42:03: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on OSPF_VL0 from LOADING to FULL, Loading Done
Router(config-router)#
Copy
Paste


20.0.0.1
20.0.0.2
30.0.0.1
30.0.0.2
10.0.0.1
Se2/0
Se3/0
Router-PT
Router1
Fa0/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10

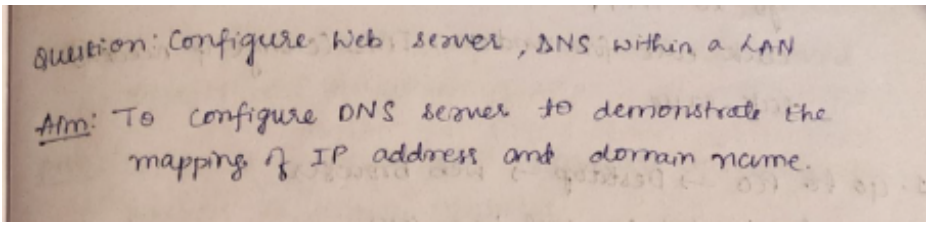

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=9ms 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 = 6ms, Maximum = 9ms, Average = 7ms
PC>


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
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=2ms TTL=125
Reply from 10.0.0.10: bytes=32 time=4ms TTL=125
Reply from 10.0.0.10: bytes=32 time=8ms TTL=125
Reply from 10.0.0.10: bytes=32 time=12ms 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 = 2ms, Maximum = 12ms, Average = 6ms
PC>

Program 8

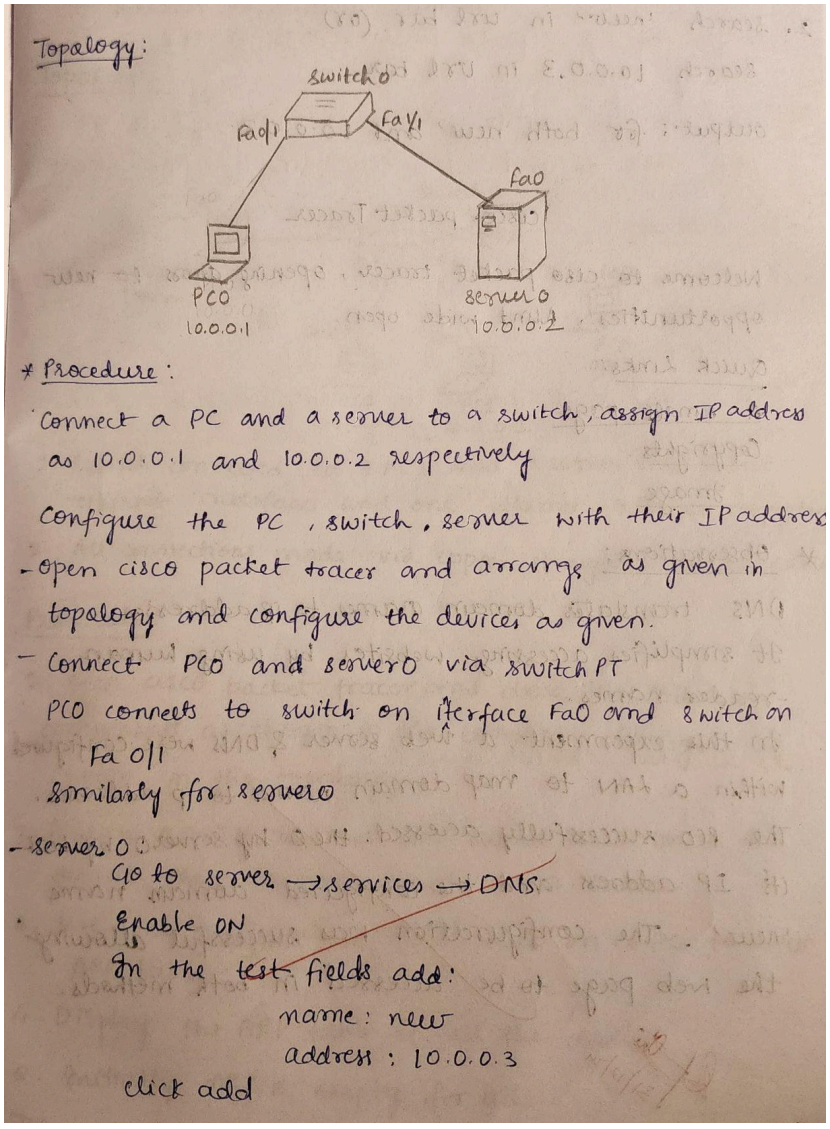
Question:

Configure Web Server, DNS within a LAN


Question: Configure Web server, DNS within a LAN

Aim: To configure DNS server to demonstrate the mapping of IP address and domain name.

Observation writeup:


Topology:

* Procedure:

Connect a PC and a server to a switch, assign IP address as 10.0.0.1 and 10.0.0.2 respectively

Configure the PC, switch, server with their IP address

- Open cisco packet tracer and arrange as given in topology and configure the devices as given.
- Connect PC0 and server0 via switch PT

PC0 connects to switch on interface Fa0 and switch on Fa 0/1

Similarly for server0

- server 0:

Go to server → services → DNS

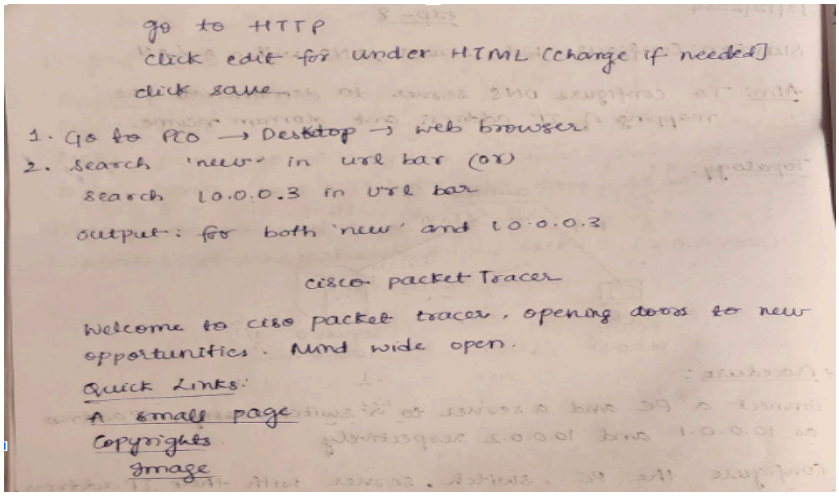
Enable ON

In the ~~test~~ fields add:

name: new

address: 10.0.0.3

click add



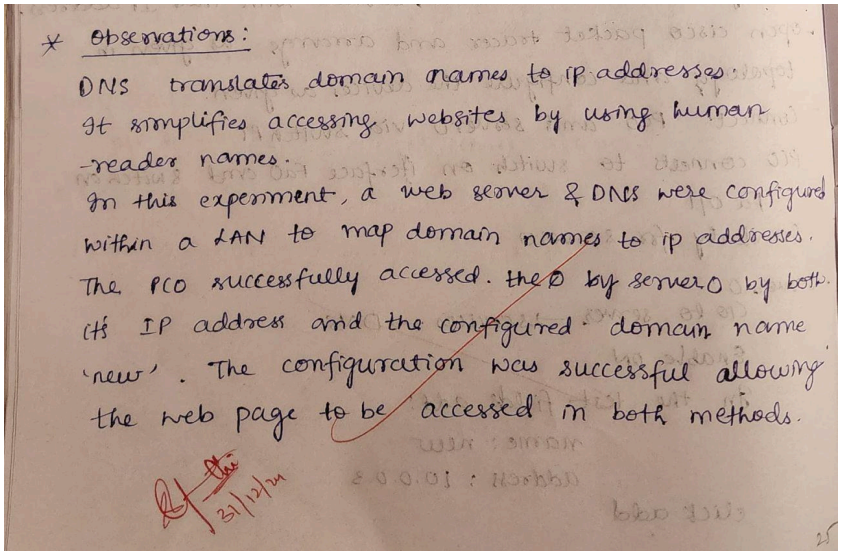
Go to HTTP
click edit for under HTML (change if needed)
click save

1. Go to PC0 → Desktop → web browser
2. Search 'new' in url bar (or)
   Search 10.0.0.3 in URL bar
   output: for both 'new' and 10.0.0.3

cisco packet Tracer

Welcome to ciso packet tracer, opening doors to new opportunities. Mind wide open.

Quick Links:
A small page
Copyrights
Image



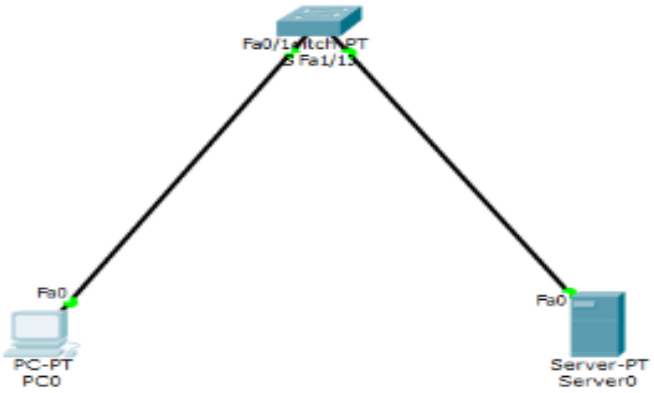
* Observations:

DNS translates domain names to ip addresses. It simplifies accessing websites by using human-reader names.
In this experiment, a web server & DNS were configured within a LAN to map domain names to ip addresses. The PC0 successfully accessed the DNS by server0 by both its IP address and the configured domain name 'new'. The configuration was successful allowing the web page to be accessed in both methods.

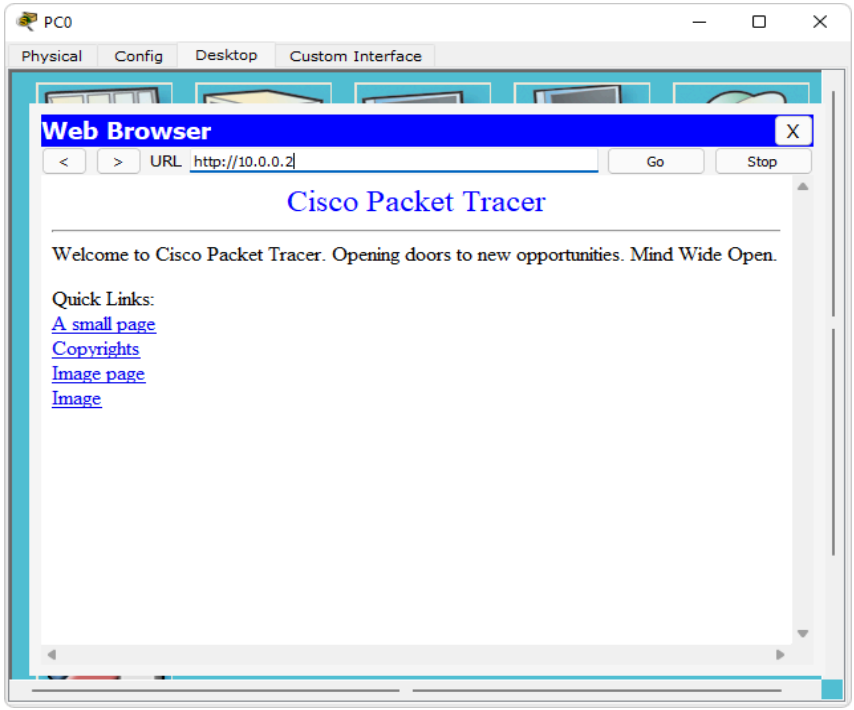
31/12/24

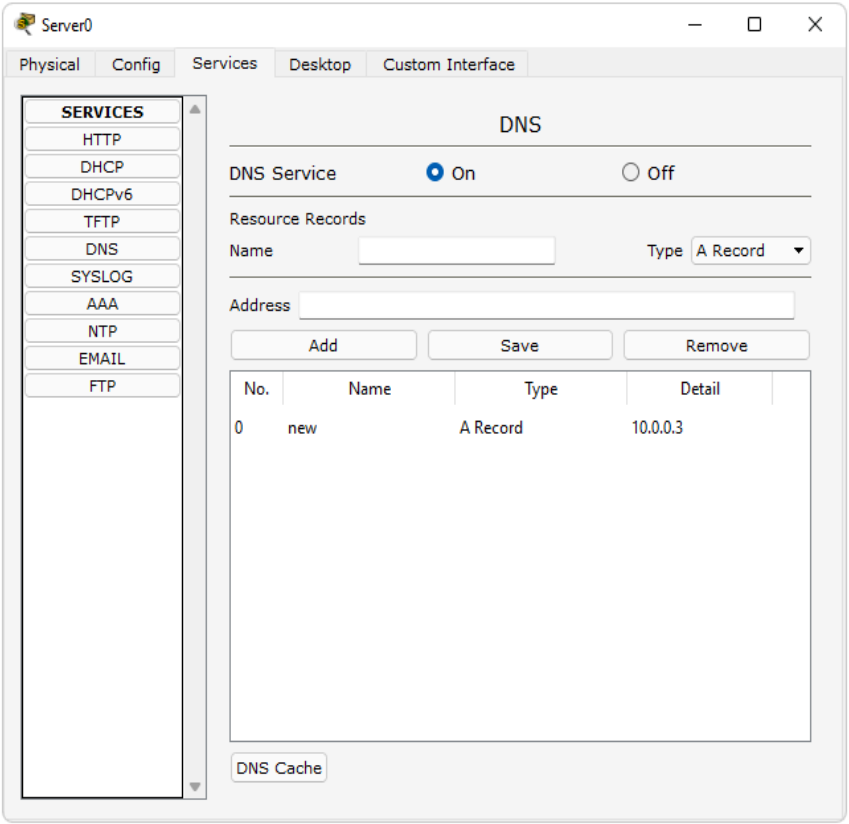
25

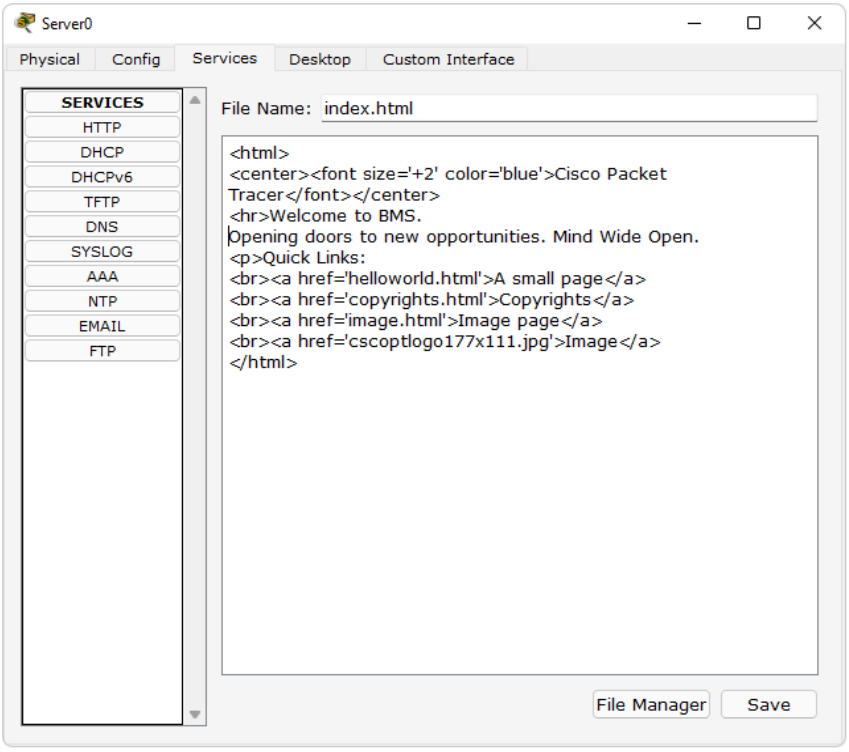
Topology:



Output:


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
URL
http://10.0.0.2
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image


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

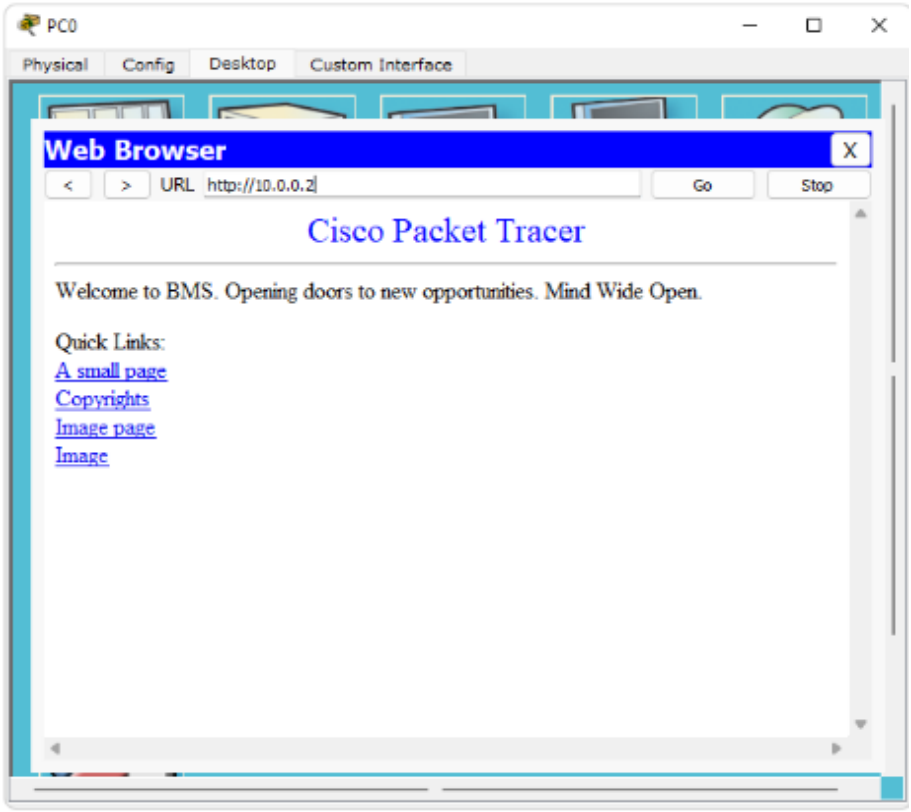

Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
File Name: index.html
<html>
Cisco Packet Tracer
<hr>Welcome to BMS.
Opening doors to new opportunities. Mind Wide Open.
<p>Quick Links:

A small page

Copyrights

Image page

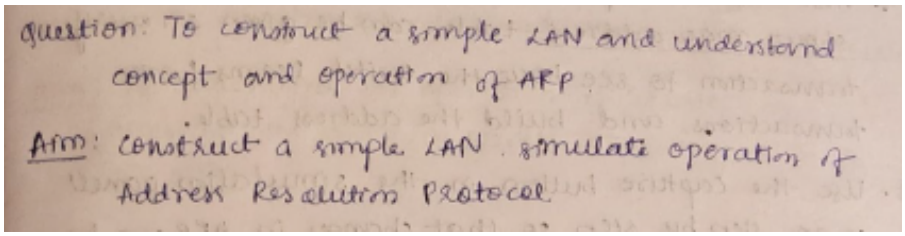
Image
</html>
File Manager
Save


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL http://10.0.0.2
Go
Stop
Cisco Packet Tracer
Welcome to BMS. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image

Program 9

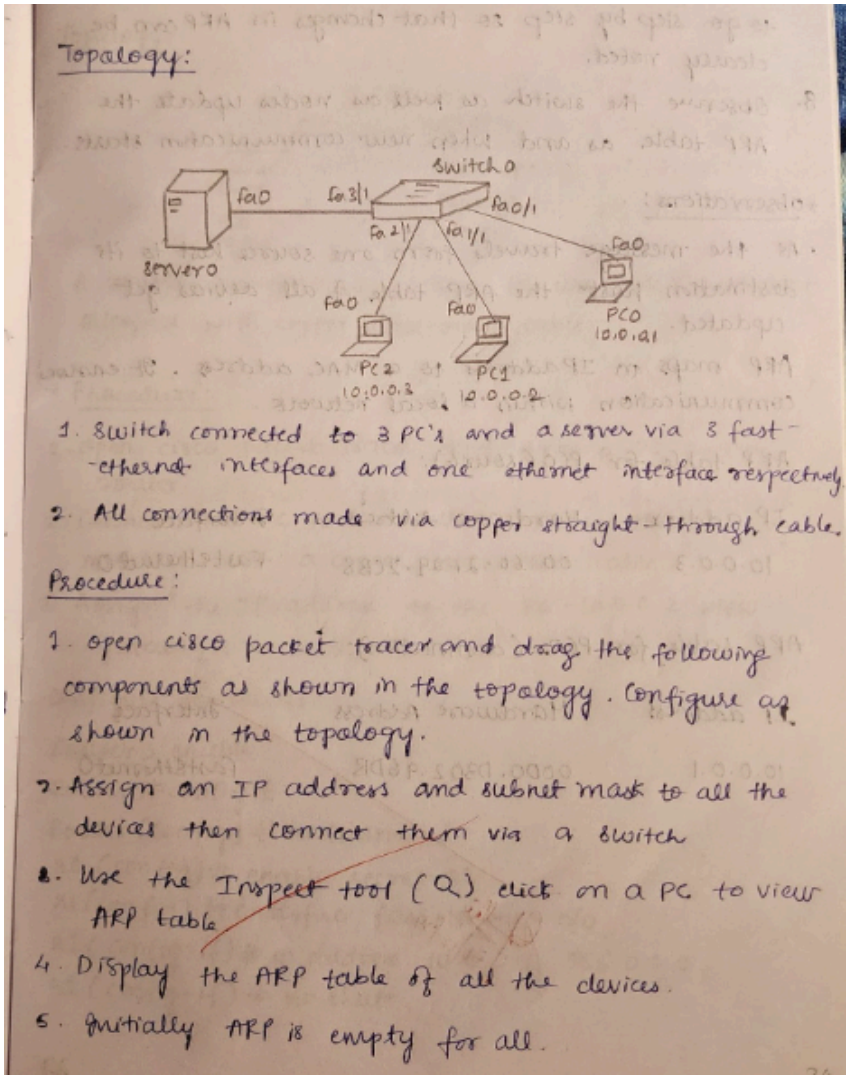
Question:

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


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

Aim: Construct a simple LAN, simulate operation of Address Resolution Protocol

Observation writeup:


Topology:

1. Switch connected to 3 PC's and a server via 3 fast-ethernet interfaces and one ethernet interface respectively
2. All connections made via copper straight-through cable.

Procedure:

1. open cisco packet tracer and drag the following components as shown in the topology. Configure as shown in the topology.
2. Assign an IP address and subnet mask to all the devices then connect them via a switch
3. Use the Inspect tool (Q) click on a PC to view ARP table
4. Display the ARP table of all the devices.
5. Initially ARP is empty for all.

6. Also in CLI of switch the command - show mac address-table can be given on every transaction to see how the switch learns from transactions and build the address table.
7. Use the capture button in the simulation panel to go step by step so that changes in ARP can be clearly noted.
8. Observe the switch as well as nodes update the ARP table as and when new communication starts.

*Observations:

- As the message travels from one source host to its destination host the ARP table of all devices get updated.

ARP maps an IP address to a MAC address. It ensures communication within a local network.

ARP table for PC0 (source):

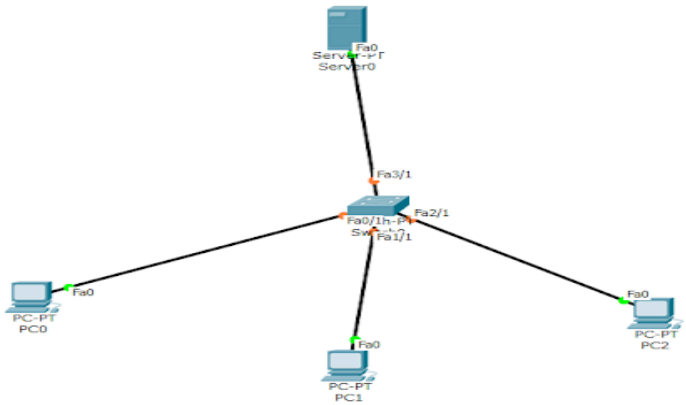
IP address	Hardware Address	Interface
10.0.0.3	00:60-2F89.2CB8	FastEthernet0

ARP table for PC2 (destination):

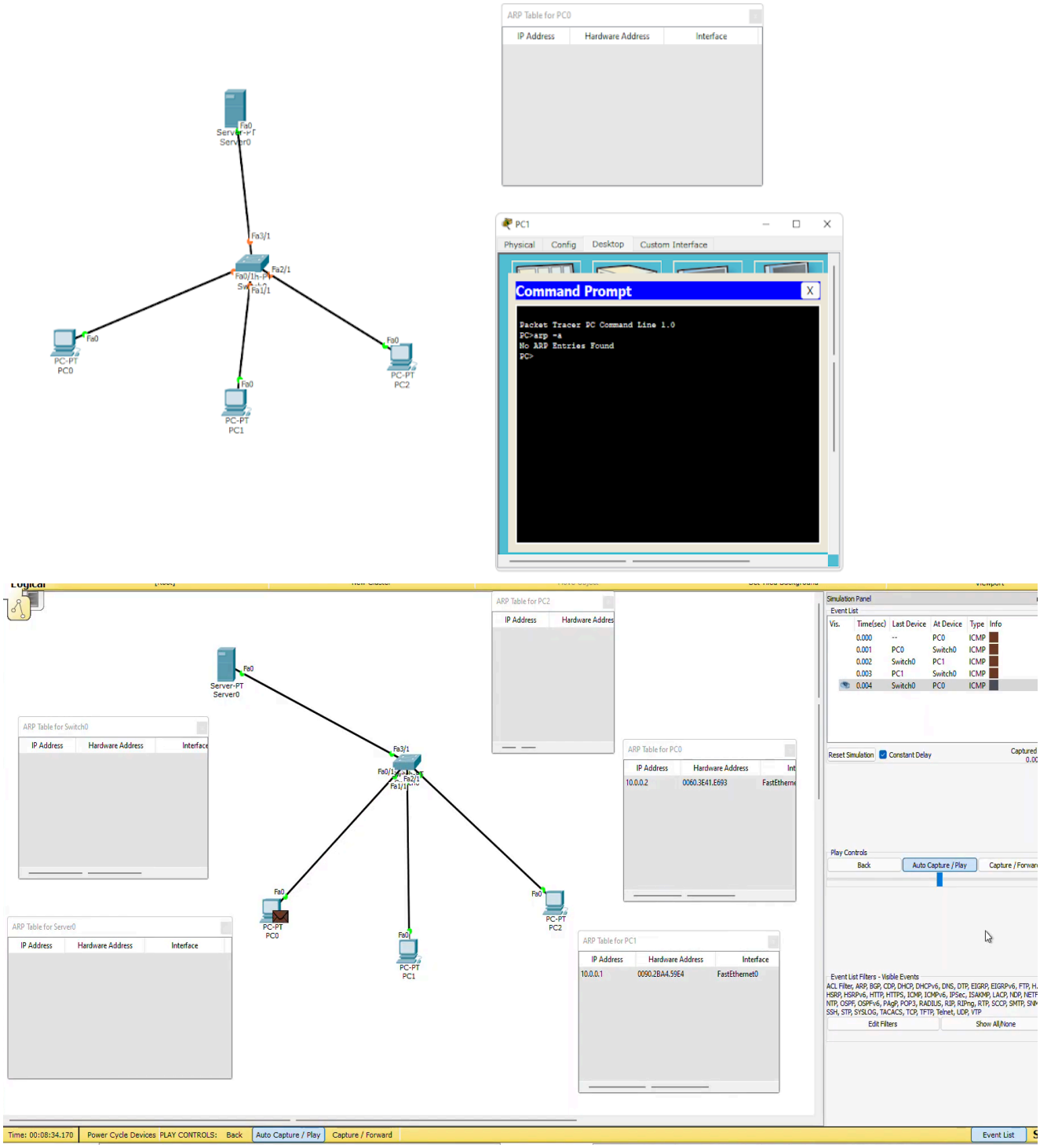
IP address	Hardware Address	Interface
10.0.0.1	0000.D302.96DB	FastEthernet0

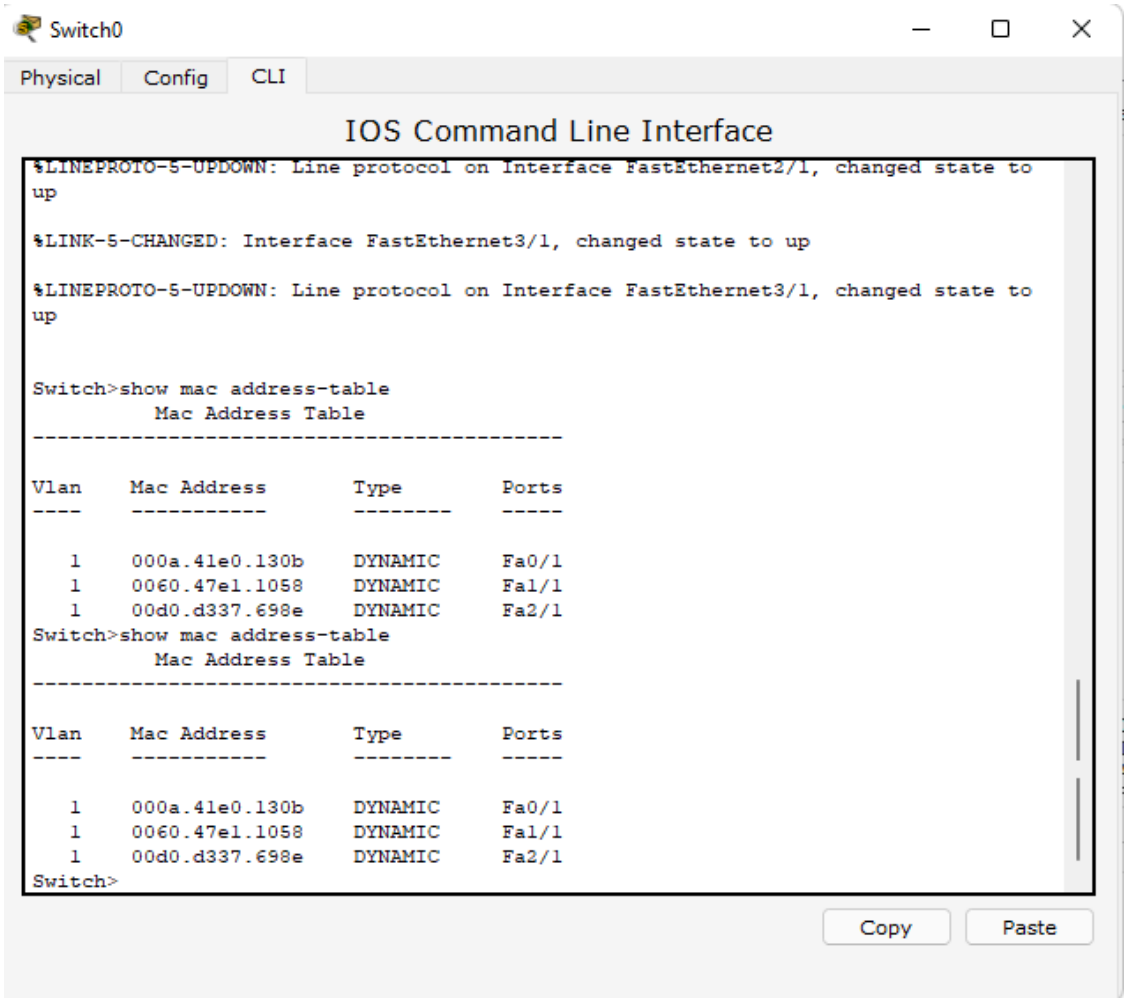
Tue 3/1/24

Topology:



Output:

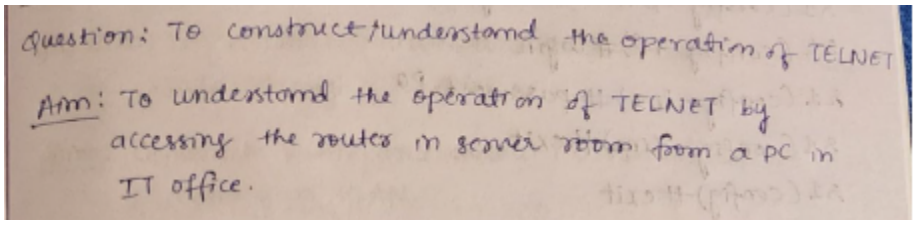

ARP Table for PC0
IP Address
Hardware Address
Interface
Server-PT
Server0
Fa0
Fa3/1
Fa2/1
Fa1/1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>
ARP Table for PC2
ARP Table for Switch0
ARP Table for PC0
10.0.0.2
0060.3E41.E693
ARP Table for Server0
ARP Table for PC1
10.0.0.1
0090.2BA4.59E4
FastEthernet0
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.000
PC0
ICMP
0.001
PC0
Switch0
ICMP
0.002
Switch0
PC1
ICMP
0.003
PC1
Switch0
ICMP
0.004
Switch0
PC0
ICMP
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:08:34.170
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List


Switch0
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state to up
Switch>show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 000a.41e0.130b DYNAMIC Fa0/1
1 0060.47e1.1058 DYNAMIC Fa1/1
1 00d0.d337.698e DYNAMIC Fa2/1
Switch>show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 000a.41e0.130b DYNAMIC Fa0/1
1 0060.47e1.1058 DYNAMIC Fa1/1
1 00d0.d337.698e DYNAMIC Fa2/1
Switch>
Copy
Paste

Program 10

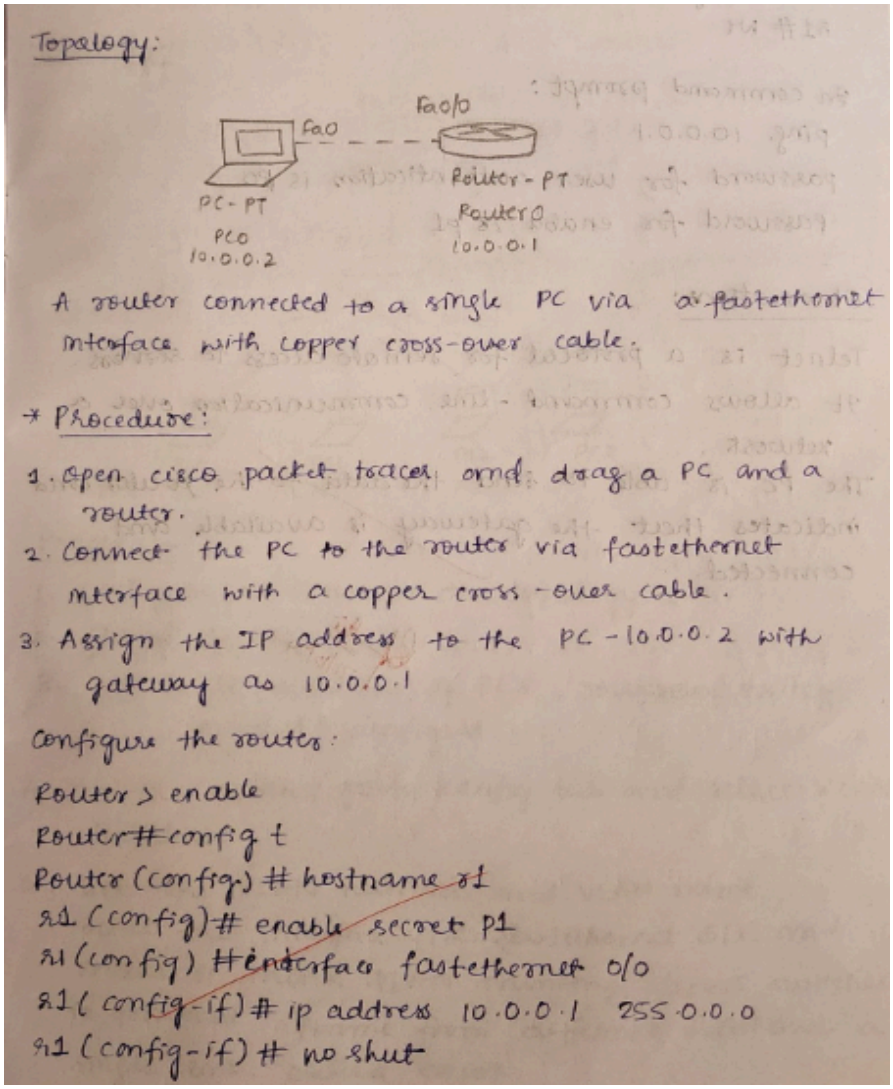
Question:

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


Question: To construct/understand the operation of TELNET

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

Observation writeup:


Topology:

A router connected to a single PC via a fastethernet interface with copper cross-over cable.

* Procedure:

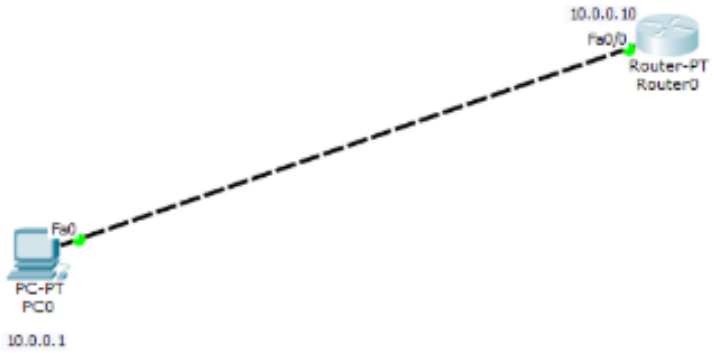
1. Open cisco packet tracer and drag a PC and a router.
2. Connect the PC to the router via fastethernet interface with a copper cross-over cable.
3. Assign the IP address to the PC - 10.0.0.2 with gateway as 10.0.0.1

Configure the router:

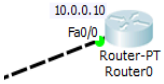
```
Router > enable
Router# config t
Router (config) # hostname r1
r1 (config)# enable secret p1
r1 (config) #interface fastethernet 0/0
r1 (config-if)# ip address 10.0.0.1 255.0.0.0
r1 (config-if)# no shut
```

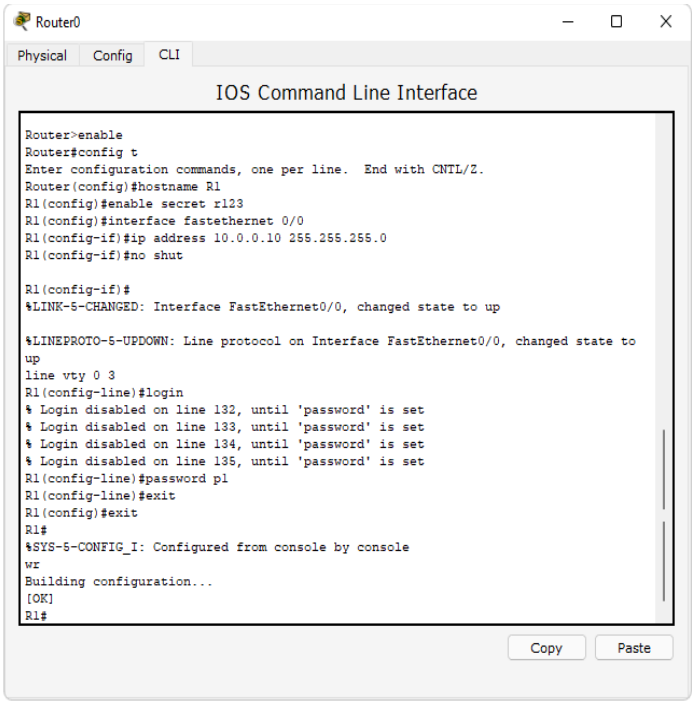
```
R1(config-if)# line vty 0 5
R1(config-if)#login
R1(config-line)#password p0
R1(config-line)#exit
R1(config)#exit
R1# wr

In command prompt:
ping 10.0.0.1
password for user authentication is p0
password for enable is p1
```

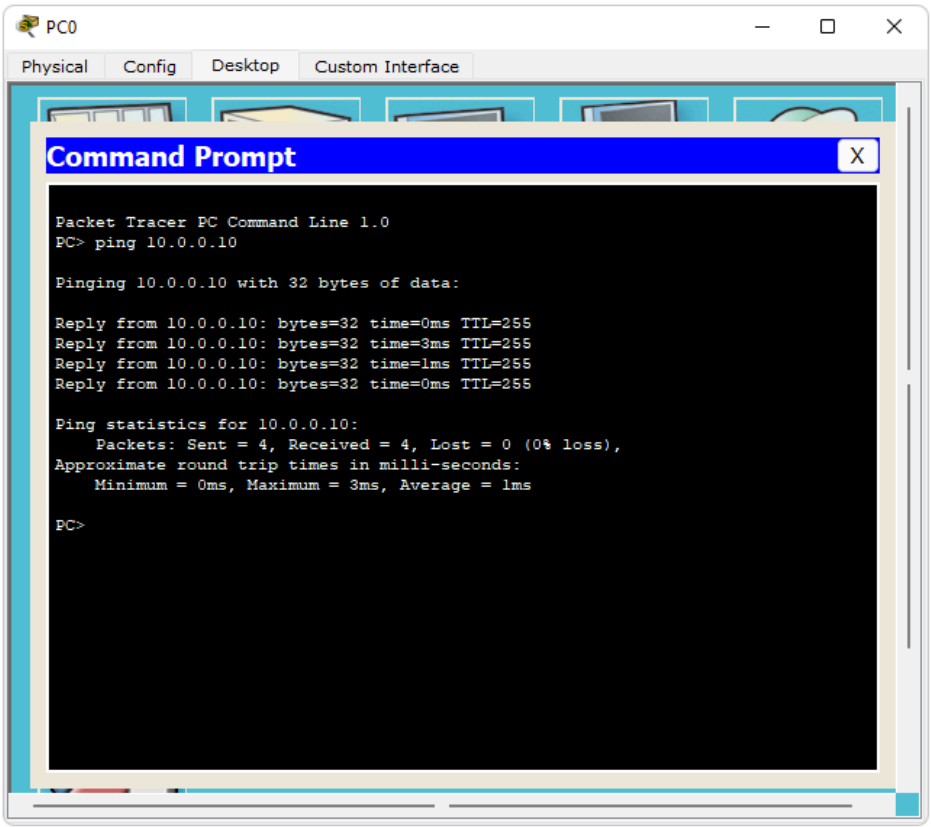
Topology:

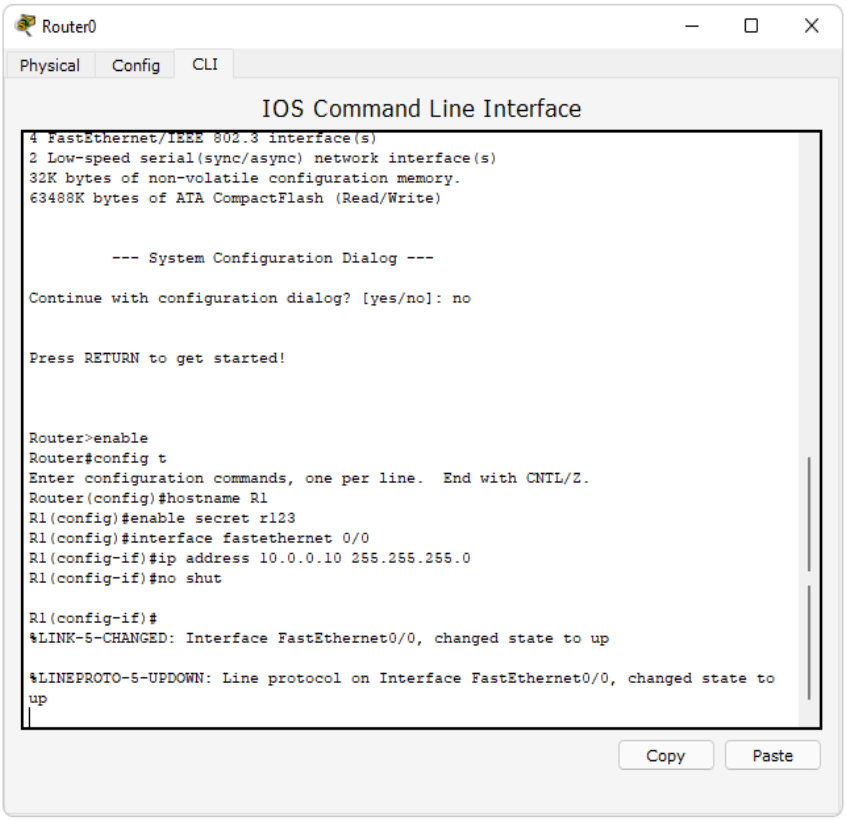


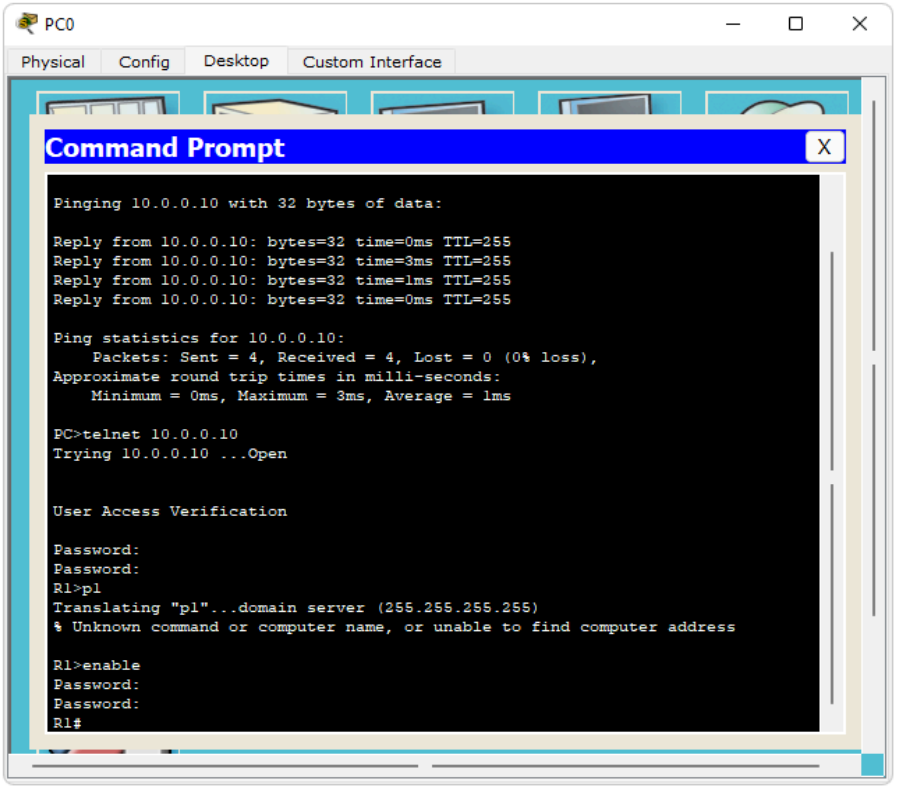
Output:


10.0.0.10
Fa0/0
Router-PT
Router0


Router0
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#enable secret r123
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.10 255.255.255.0
R1(config-if)#no shut
R1(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
line vty 0 3
R1(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
R1(config-line)#password p1
R1(config-line)#exit
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
wr
Building configuration...
[OK]
R1#
Copy
Paste


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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=0ms TTL=255
Reply from 10.0.0.10: bytes=32 time=3ms TTL=255
Reply from 10.0.0.10: bytes=32 time=1ms TTL=255
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 1ms
PC>

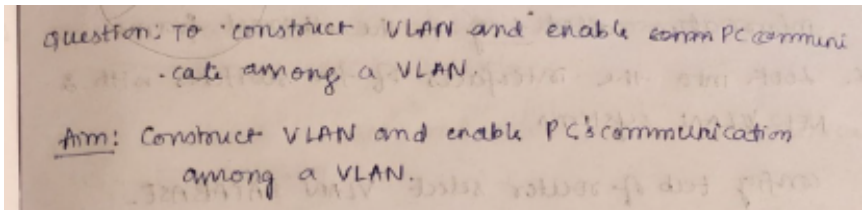

Router0
Physical
Config
CLI
IOS Command Line Interface
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]: no
Press RETURN to get started!
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#enable secret r123
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.10 255.255.255.0
R1(config-if)#no shut
R1(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
Copy
Paste


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Reply from 10.0.0.10: bytes=32 time=3ms TTL=255
Reply from 10.0.0.10: bytes=32 time=1ms TTL=255
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 1ms
PC>telnet 10.0.0.10
Trying 10.0.0.10 ...Open
User Access Verification
Password:
Password:
R1>p1
Translating "p1"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
R1>enable
Password:
Password:
R1#

Program 11

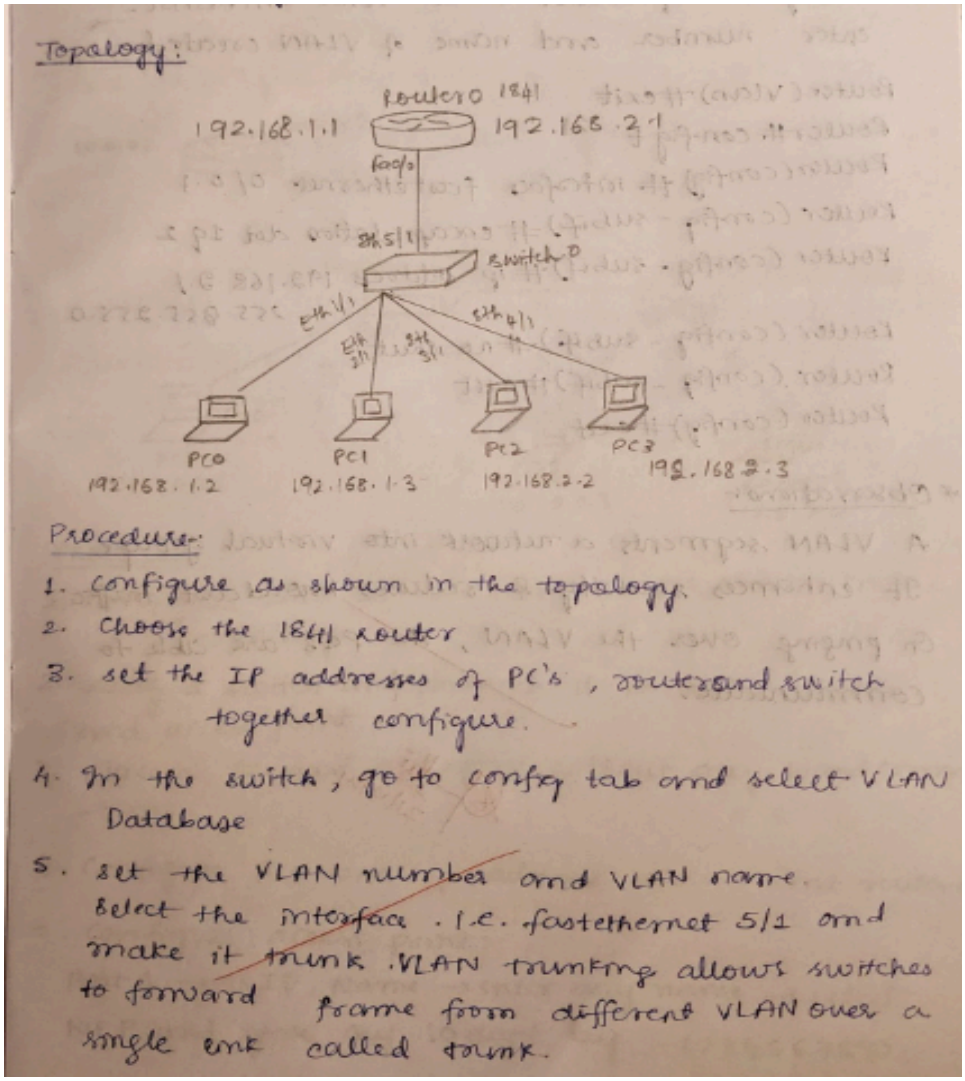
Question:

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


Question: To construct VLAN and enable comm PC communi -cate among a VLAN.

Aim: Construct VLAN and enable PC's communication among a VLAN.

Observation writeup:



Procedure:

1. Configure as shown in the topology.
2. Choose the 1841 router
3. Set the IP addresses of PC's, router and switch together configure.
4. In the switch, go to config tab and select VLAN Database
5. Set the VLAN number and VLAN name. Select the interface. i.e. fastethernet 5/1 and make it trunk. VLAN trunking allows switches to forward frame from different VLAN over a single link called trunk.

6. This is done by adding an additional header information called 'tag' to the ethernet frame.
7. Look into the interfaces of the switches with a NEW VLAN systems

config tab of router select VLAN DATABASE
enter number and name of VLAN created

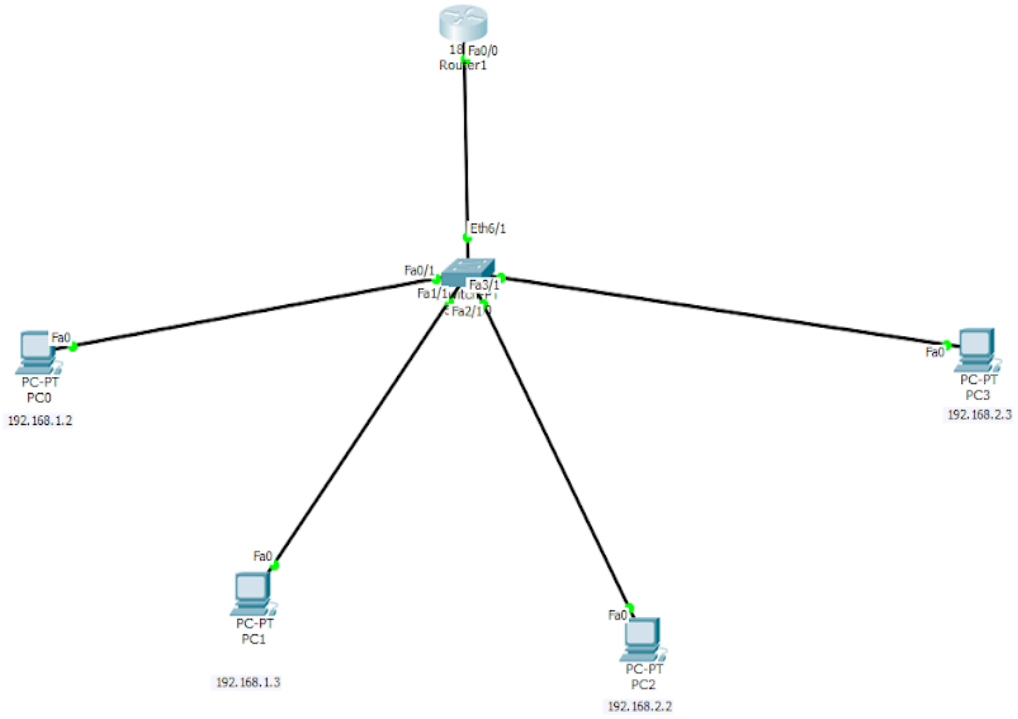
```
Router(Vlan)#exit
Router#config t
Router(config)#interface fastethernet 0/0.1
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 192.168.2.1
                                255.255.255.0
Router(config-subif)#no shut
Router(config-subif)#exit
Router(config)#exit
```

* Observations:-

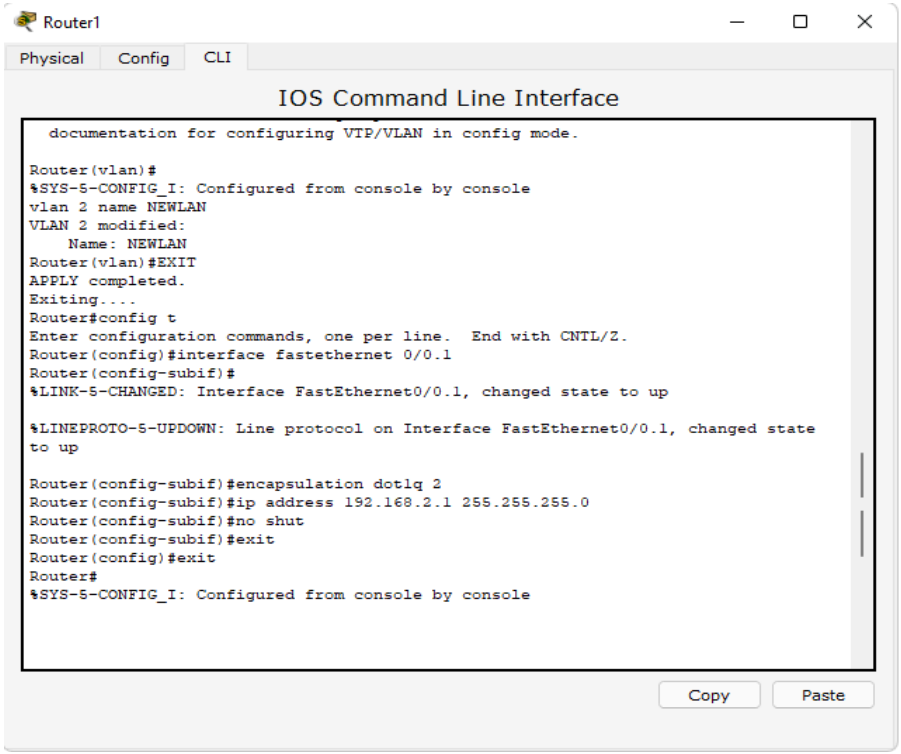
A VLAN segments a network into virtual groups
It enhances security & reduces broadcast traffic.
On pinging over the VLAN, the PCs are able to communicate.

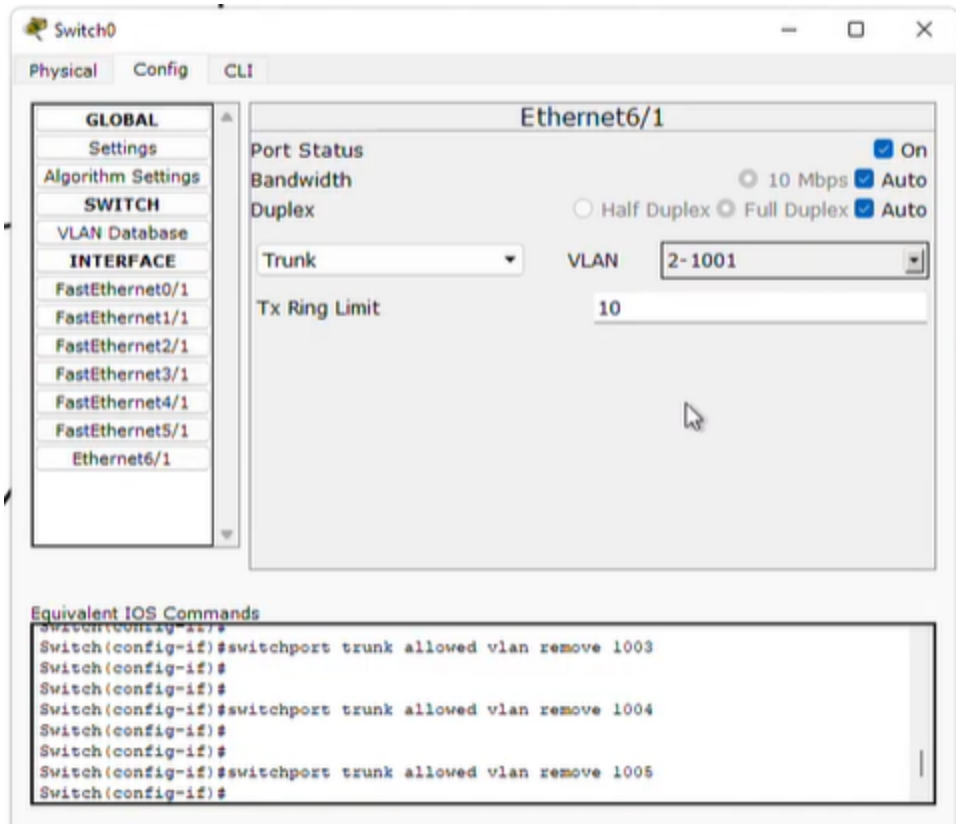
31/12/24

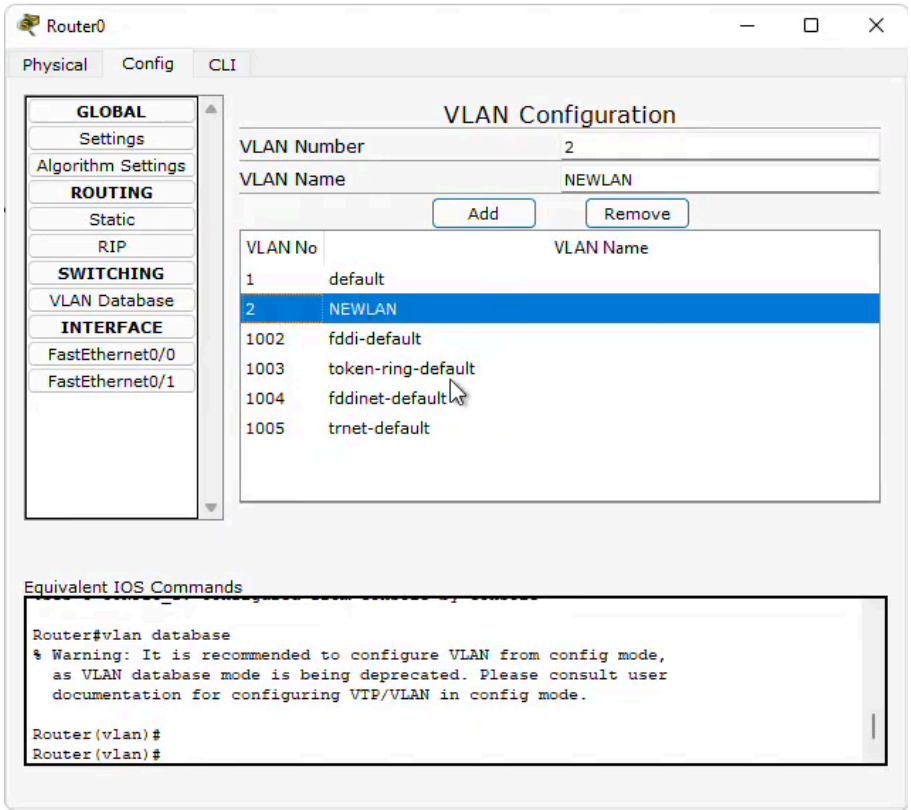
Topology:

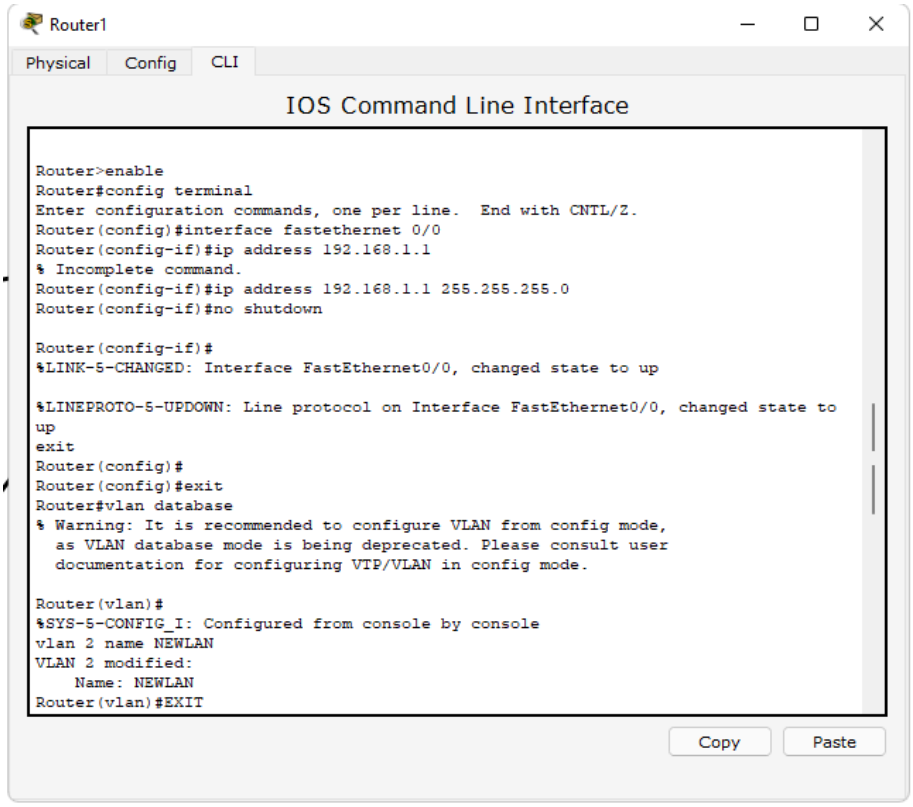


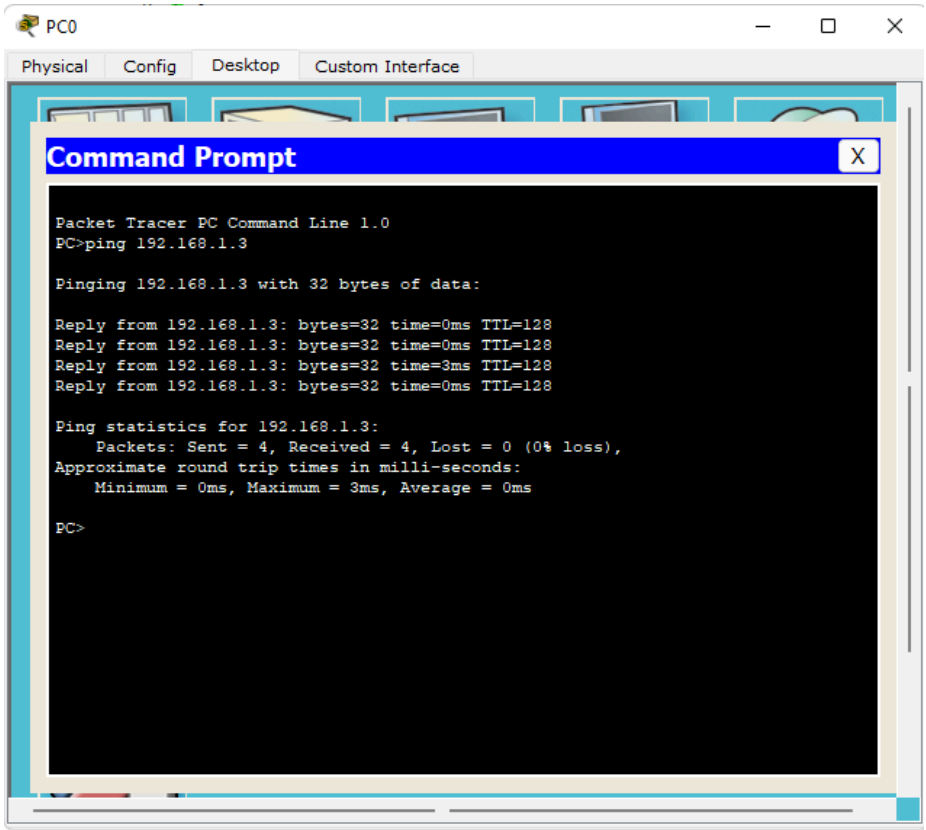
Output:




Switch0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet1/1
FastEthernet2/1
FastEthernet3/1
FastEthernet4/1
FastEthernet5/1
Ethernet6/1
Ethernet6/1
Port Status
On
Bandwidth
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
Trunk
VLAN
2-1001
Tx Ring Limit
10
Equivalent IOS Commands
Switch(config-if)#switchport trunk allowed vlan remove 1003
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#switchport trunk allowed vlan remove 1004
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#switchport trunk allowed vlan remove 1005
Switch(config-if)#


Router0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
VLAN Configuration
VLAN Number
2
VLAN Name
NEWLAN
Add
Remove
VLAN No
VLAN Name
1
default
2
NEWLAN
1002
fddi-default
1003
token-ring-default
1004
fddinet-default
1005
trnet-default
Equivalent IOS Commands
Router#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.
Router(vlan)#
Router(vlan)#

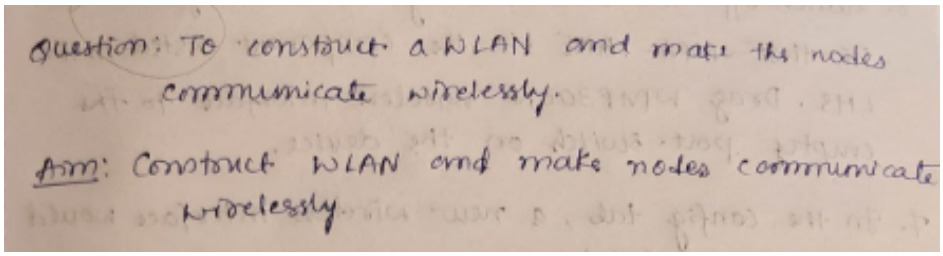

Router1
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 192.168.1.1
% Incomplete command.
Router(config-if)#ip address 192.168.1.1 255.255.255.0
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
exit
Router(config)#
Router(config)#exit
Router#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.
Router(vlan)#
%SYS-5-CONFIG_I: Configured from console by console
vlan 2 name NEWLAN
VLAN 2 modified:
Name: NEWLAN
Router(vlan)#EXIT
Copy
Paste


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.3
Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=3ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 0ms
PC>

Program 12

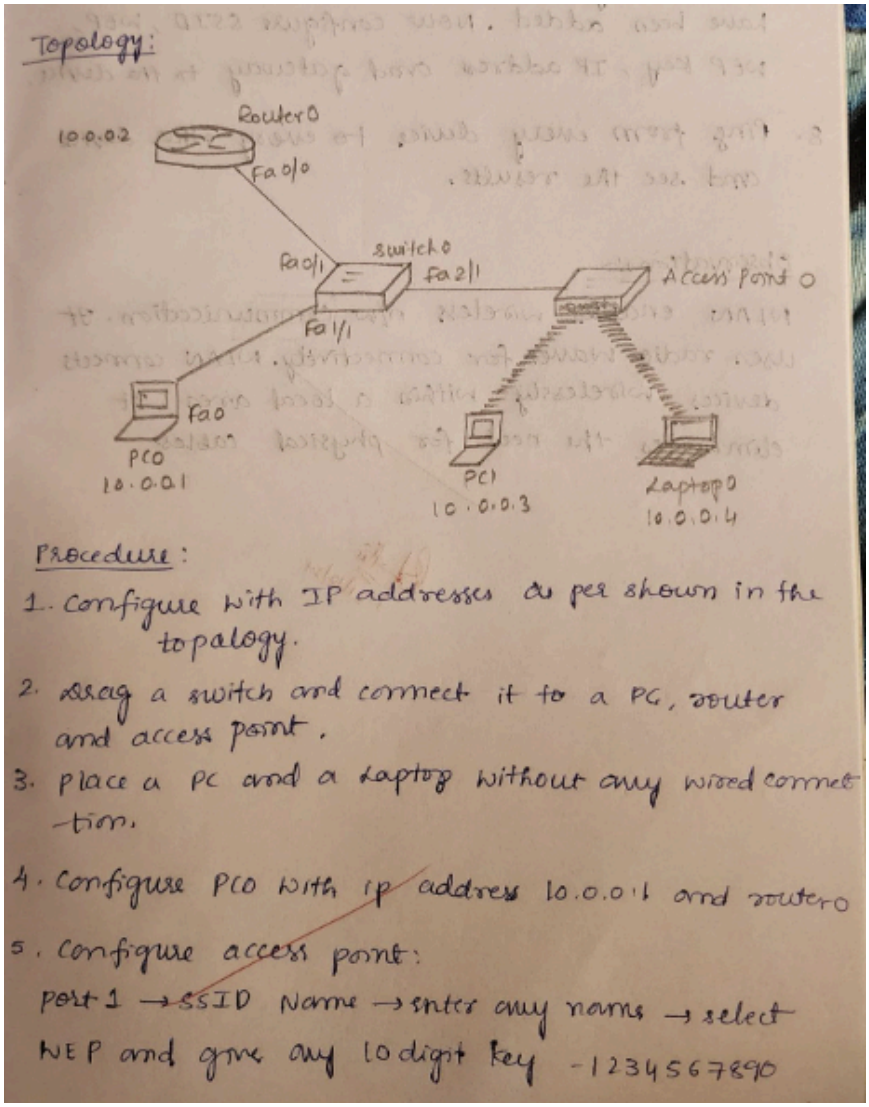
Question:

To construct a WLAN and make the nodes communicate wirelessly


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

Aim: Construct WLAN and make nodes communicate wirelessly.

Observation writeup:



Procedure:

1. Configure with IP addresses as per shown in the topology.
2. Drag a switch and connect it to a PC, router and access point.
3. Place a PC and a laptop without any wired connection.
4. Configure PC0 with ip address 10.0.0.1 and router0
5. Configure access point:
   port 1 → SSID Name → enter any name → select WEP and give any 10 digit key - 1234567890

6. Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag WMP300N wireless interface to the empty port. Switch on the device.

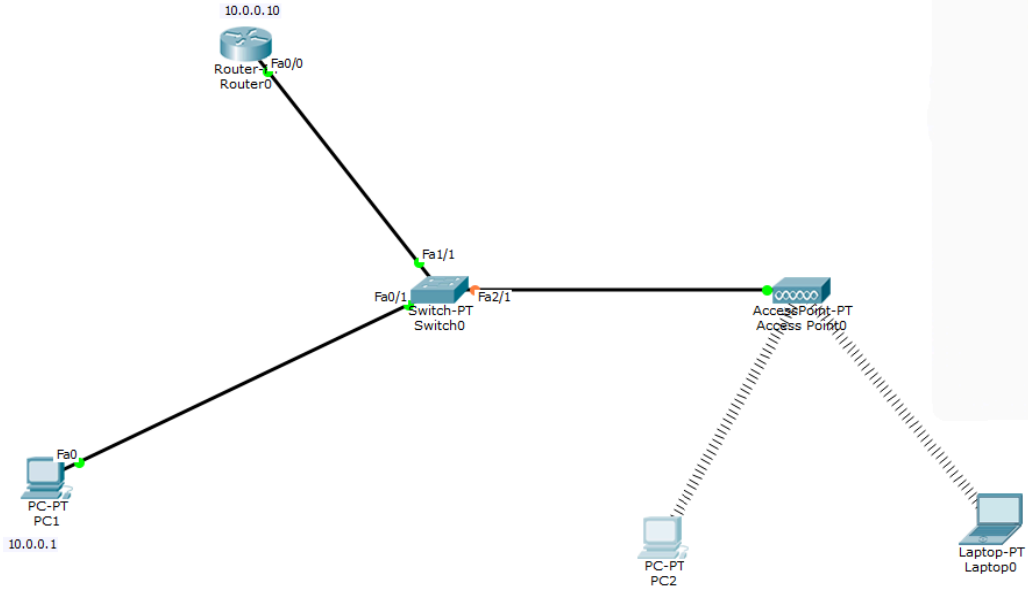
7. In the config tab, a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and gateway to the device.

8. Ping from every device to every other device and see the results.

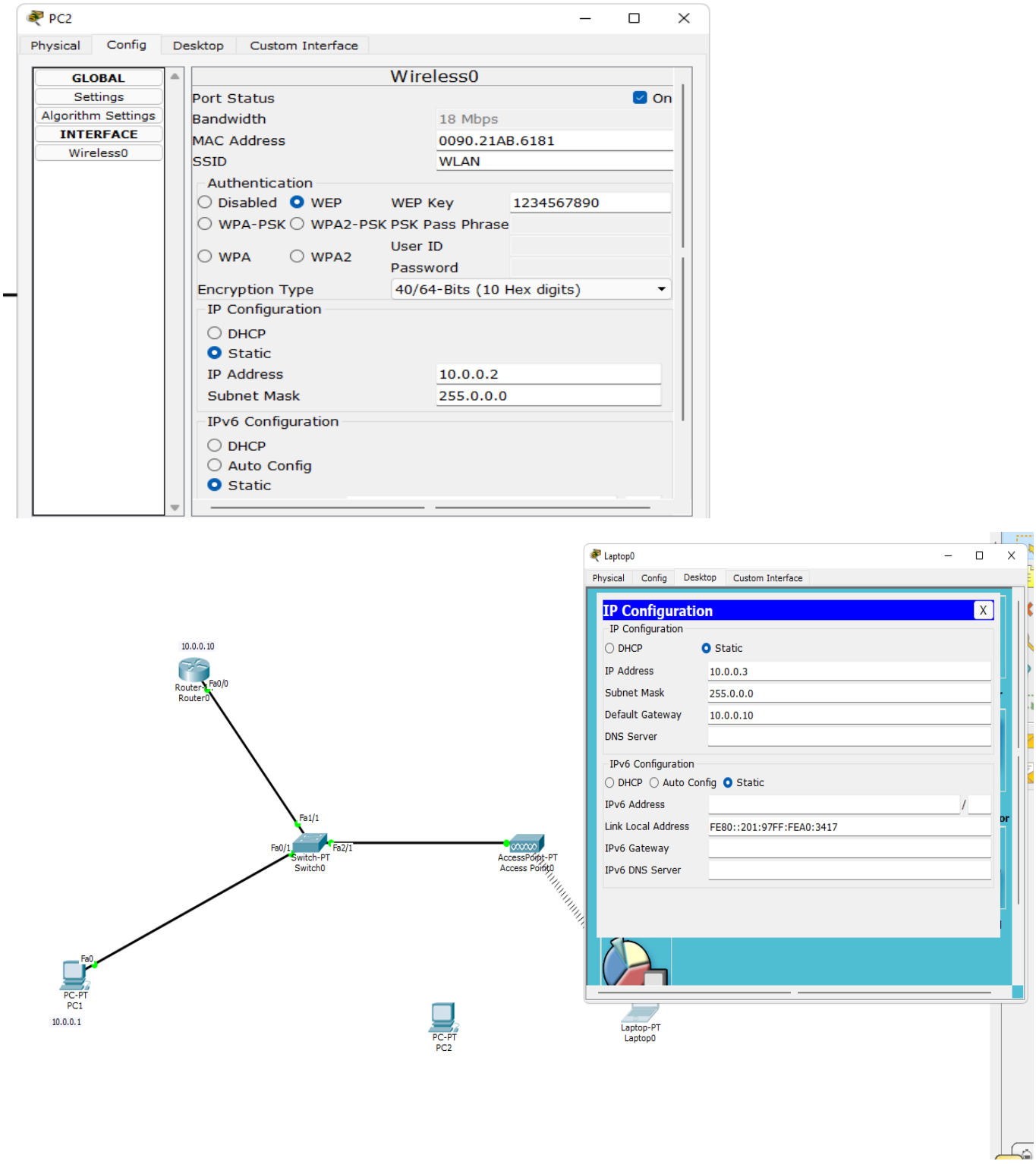
Observations:-

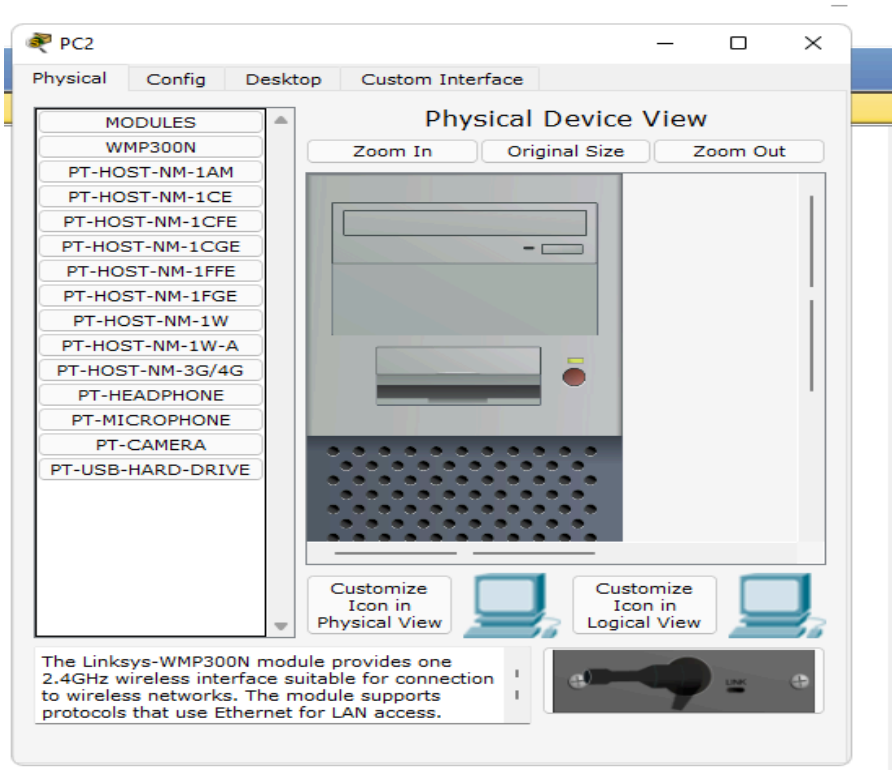
WLAN enables wireless n/w communication. It uses radio waves for connectivity. WLAN connects devices wirelessly within a local area. It eliminates the need for physical cables.

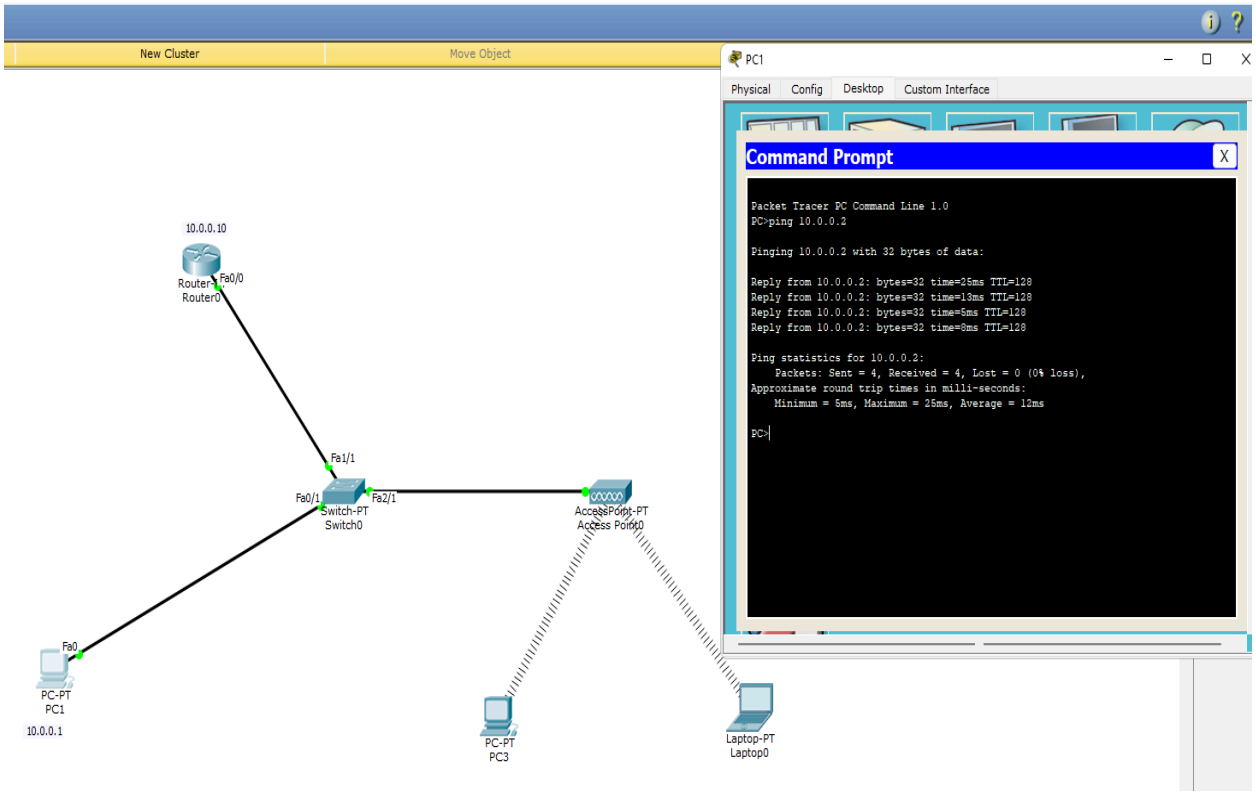
Topology:



Output:


PC2
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
Wireless0
Port Status
On
Bandwidth
18 Mbps
MAC Address
0090.21AB.6181
SSID
WLAN
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
WPA
WPA2
User ID
Password
Encryption Type
40/64-Bits (10 Hex digits)
IP Configuration
DHCP
Static
IP Address
10.0.0.2
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
Laptop0
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.10
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::201:97FF:FEA0:3417
IPv6 Gateway
IPv6 DNS Server
10.0.0.10
Router-PT
Router0
Fa0/0
Fa1/1
Fa0/1
Switch-PT
Switch0
Fa2/1
AccessPoint-PT
Access Point0
Fa0
PC-PT
PC1
10.0.0.1
PC-PT
PC2
Laptop-PT
Laptop0


PC2
Physical
Config
Desktop
Custom Interface
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

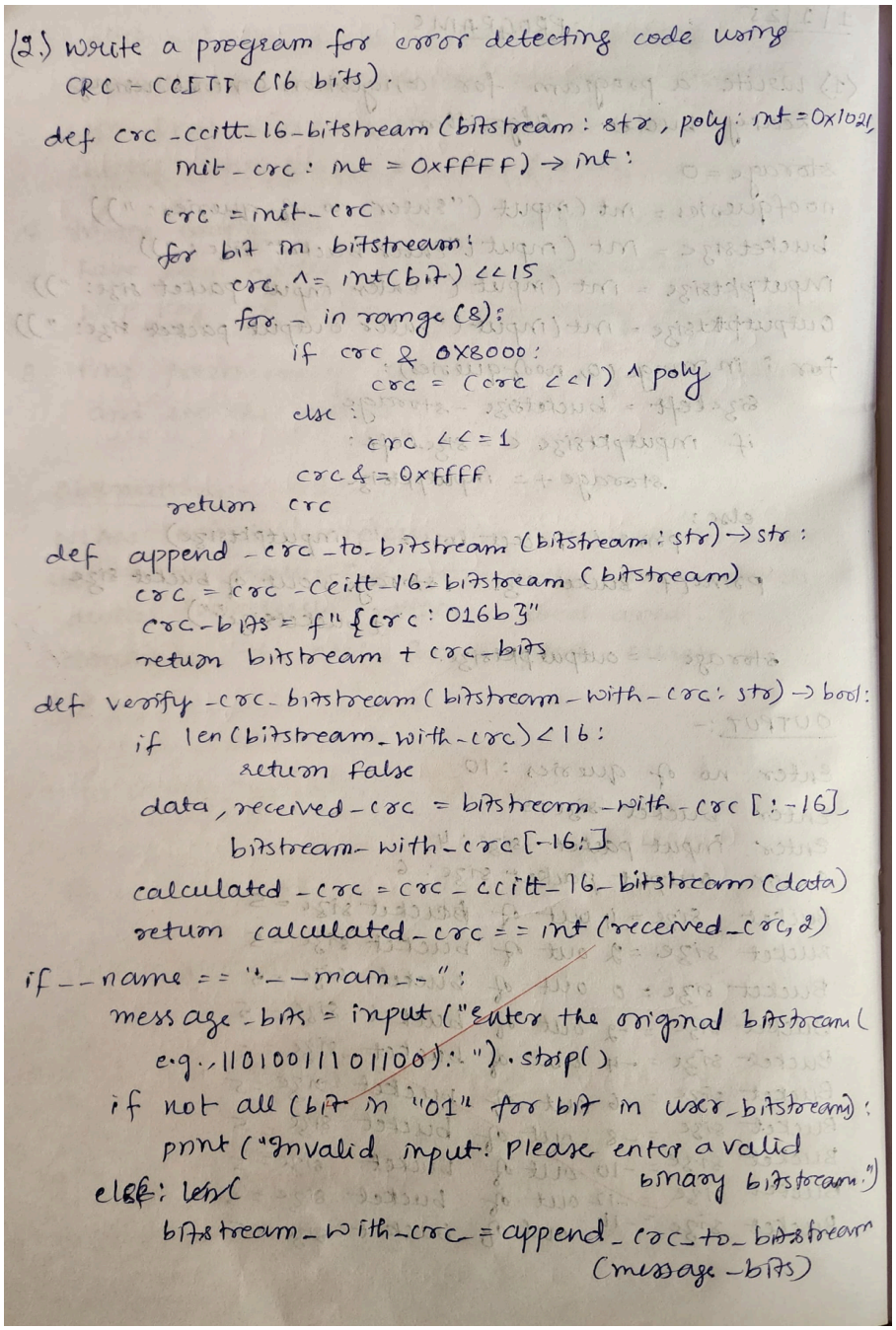

New Cluster
Move Object
PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=25ms TTL=128
Reply from 10.0.0.2: bytes=32 time=13ms TTL=128
Reply from 10.0.0.2: bytes=32 time=5ms TTL=128
Reply from 10.0.0.2: bytes=32 time=8ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 25ms, Average = 12ms
PC>
10.0.0.10
Router-PT
Router0
Fa0/0
Fa1/1
Fa0/1
Fa2/1
Switch-PT
Switch0
Access Point-PT
Access Point0
Fa0
PC-PT
PC1
10.0.0.1
PC-PT
PC3
Laptop-PT
Laptop0

CYCLE 2

Program 1:

Question:

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



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

def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021,
        init_crc: int = 0xFFFF) -> int:
    crc = init_crc
    for bit in bitstream:
        crc ^= int(bit) << 15
        for _ in range(8):
            if crc & 0x8000:
                crc = (crc << 1) ^ poly
            else:
                crc <<= 1
            crc &= 0xFFFF
    return crc

def append_crc_to_bitstream(bitstream: str) -> str:
    crc = crc_ccitt_16_bitstream(bitstream)
    crc_bits = f"{crc:016b}"
    return bitstream + crc_bits

def verify_crc_bitstream(bitstream_with_crc: str) -> bool:
    if len(bitstream_with_crc) < 16:
        return False
    data, received_crc = bitstream_with_crc[:-16],
        bitstream_with_crc[-16:]
    calculated_crc = crc_ccitt_16_bitstream(data)
    return calculated_crc == int(received_crc, 2)

if __name__ == "__main__":
    message_bits = input("Enter the original bitstream (
        e.g., 1101001110100): ").strip()
    if not all(bit in "01" for bit in user_bitstream):
        print("Invalid input. Please enter a valid
                                        binary bitstream.")
    else:
        bitstream_with_crc = append_crc_to_bitstream
                                        (message_bits)
```

```
print(f"Transmitted bitstream with CRC:
        {bitstream_with_crc}")
user_bitstream = input("Enter the received
    bitstream for verification: ").strip()
if not all(bit in "01" for bit in user_bitstream):
    print("Invalid input. Please enter a valid
        binary bitstream").
elif len(user_bitstream) < 16:
    print("Invalid input. Received bitstream
        must include at least 16 bits for CRC.")
else:
    is_valid = verify_crc_bitstream(user_
                                    bitstream)
    if is_valid:
        print("No errors detected. CRC valid")
    else:
        print("Error detected! CRC invalid")
```

OUTPUT:

```
Enter the original bitstream : 10100011111
Transmitted bitstream with CRC: 10100011111 00101000
                                    11111110

Enter the received bitstream for verification:
                    10100111110610100 1000
Error detected! CRC invalid.
```

Code:

```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    crc = init_crc
    for bit in bitstream:
        crc ^= int(bit) << 15
        for _ in range(8):
            if crc & 0x8000:
                crc = (crc << 1) ^ poly
            else:
                crc <<= 1
            crc &= 0xFFFF
    return crc

def append_crc_to_bitstream(bitstream: str) -> str:
    crc = crc_ccitt_16_bitstream(bitstream)
    crc_bits = f"{crc:016b}"
    return bitstream + crc_bits

def verify_crc_bitstream(bitstream_with_crc: str) -> bool:
    if len(bitstream_with_crc) < 16:
        return False
    data, received_crc = bitstream_with_crc[:-16], bitstream_with_crc[-16:]
    calculated_crc = crc_ccitt_16_bitstream(data)
    return calculated_crc == int(received_crc, 2)

if __name__ == "__main__":
    message_bits = input("Enter the original bitstream (e.g., 11010011101100): ").strip()
    if not all(bit in "01" for bit in message_bits):
        print("Invalid input. Please enter a binary bitstream (e.g., 11010011101100).")
    else:
        bitstream_with_crc = append_crc_to_bitstream(message_bits)
        print(f"Transmitted bitstream with CRC: {bitstream_with_crc}")
        user_bitstream = input("Enter the received bitstream for verification: ").strip()
        if not all(bit in "01" for bit in user_bitstream):
```

```
        print("Invalid input. Please enter a valid binary bitstream.")
    elif len(user_bitstream) < 16:
        print("Invalid input. Received bitstream must include at least 16 bits for CRC.")
    else:
        is_valid = verify_crc_bitstream(user_bitstream)
        if is_valid:
            print("No errors detected. CRC valid.")
        else:
            print("Error detected! CRC invalid.")
```

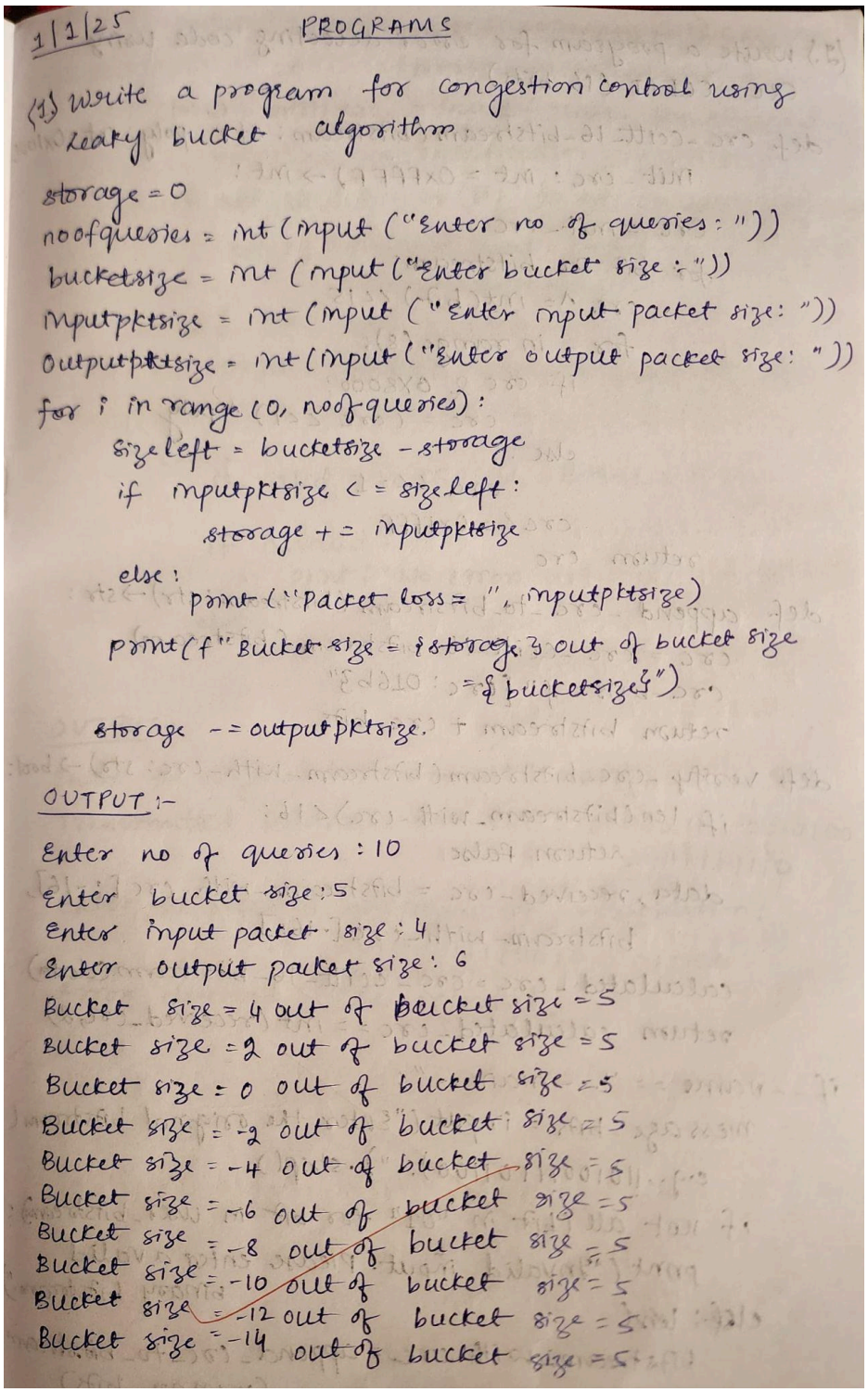
Output:

```
PS C:\Users\Dell\OneDrive\Desktop\code> python crc-ccitt.py
Enter the original bitstream (e.g., 11010011101100): 1010001111
Transmitted bitstream with CRC: 101000111110010100111111110
Enter the received bitstream for verification: 101000111110010100100 0
Error detected! CRC invalid.
```

Program 2:

Question:

Write a program for congestion control using Leaky bucket algorithm.



1/1/25 PROGRAMS

(1) Write a program for congestion control using Leaky bucket algorithm.

```
storage = 0
noofqueries = int(input("Enter no of queries: "))
bucketsize = int(input("Enter bucket size: "))
inputpktsize = int(input("Enter input packet size: "))
Outputpktsize = int(input("Enter output packet size: "))
for i in range(0, noofqueries):
    sizeleft = bucketsize - storage
    if inputpktsize <= sizeleft:
        storage += inputpktsize
    else:
        print("Packet loss = ", inputpktsize)
    print(f"Bucket size = {storage} out of bucket size
                = {bucketsize}")
    storage -= outputpktsize.
```

OUTPUT :-

```
Enter no of queries : 10
Enter bucket size : 5
Enter input packet size : 4
Enter output packet size : 6
Bucket size = 4 out of bucket size = 5
Bucket size = 2 out of bucket size = 5
Bucket size = 0 out of bucket size = 5
Bucket size = -2 out of bucket size = 5
Bucket size = -4 out of bucket size = 5
Bucket size = -6 out of bucket size = 5
Bucket size = -8 out of bucket size = 5
Bucket size = -10 out of bucket size = 5
Bucket size = -12 out of bucket size = 5
Bucket size = -14 out of bucket size = 5
```

Code:

```
storage=0
noofqueries=int(input("Enter no of queries:"))
bucketsize=int(input("Enter bucket size:"))
inputpktsize=int(input("Enter input packet size:"))
outputpktsize=int(input("Enter output packet size:"))
for i in range(0,noofqueries):
   sizeleft=bucketsize-storage
   if inputpktsize<=sizeleft:
      storage+=inputpktsize
   else:
      print("Packet loss=", inputpktsize)
   print(f"Bucket size={storage}out of bucket size={bucketsize}")
   storage-=outputpktsize
```

Output:

```
PS C:\Users\Dell\OneDrive\Desktop\code> python leakybucketalgorithm.py
Enter no of queries:10
Enter bucket size:5
Enter input packet size:4
Enter output packet size:6
Bucket size=4out of bucket size=5
Bucket size=2out of bucket size=5
Bucket size=0out of bucket size=5
Bucket size=-2out of bucket size=5
Bucket size=-4out of bucket size=5
Bucket size=-6out of bucket size=5
Bucket size=-14out of bucket size=5
```

Program 3:

Question:

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.

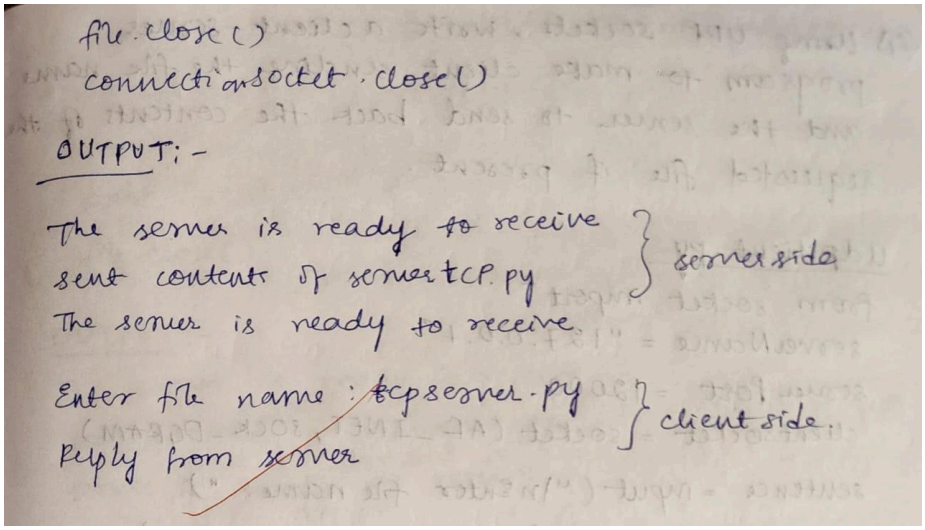
(3) Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

tcpclient.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
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()
```

tcpserver.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
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)
```



Code:

tcpserver.py file:

```
from socket import *
serverName="127.0.0.1"
serverPort = 14000
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()
```

tcpclient.py file:

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
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()
```

Output:

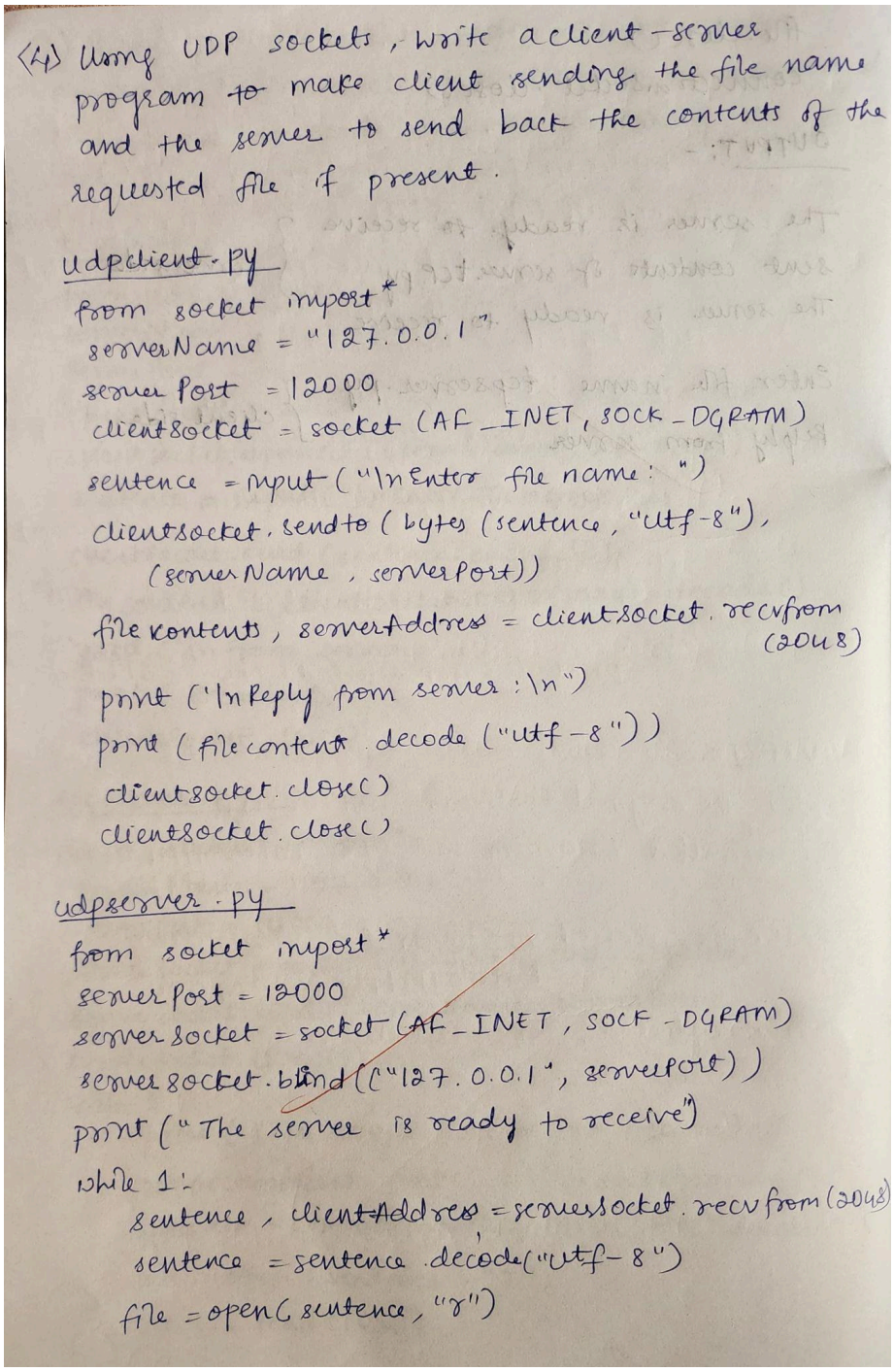
```
PS C:\Users\Dell\OneDrive\Desktop\code> python tcpserver.py
The server is ready to receive

Sent contents of example.txt
The server is ready to receive
```

```
PS C:\Users\Dell\OneDrive\Desktop\code> python tcpclient.py

Enter file name: example.txt

From Server:

Hello, this is a sample file.
It is used for testing the TCP server.
```

Program 4:

Question:

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.



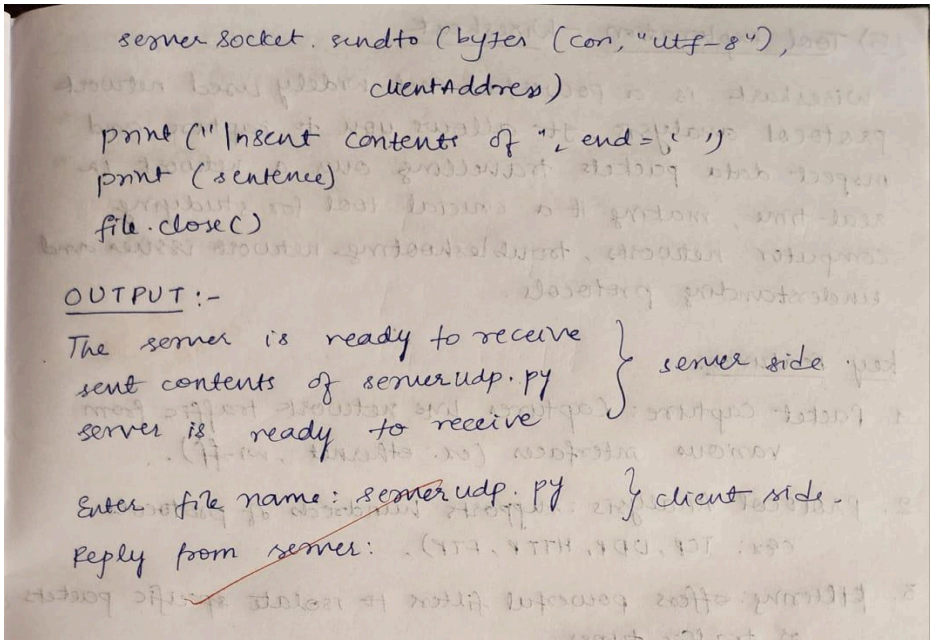
4) Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

udpclient.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(fileContent.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

udpserver.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")
```



Code:

udpserver.py file:

```
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()
```

udpclient.py file:

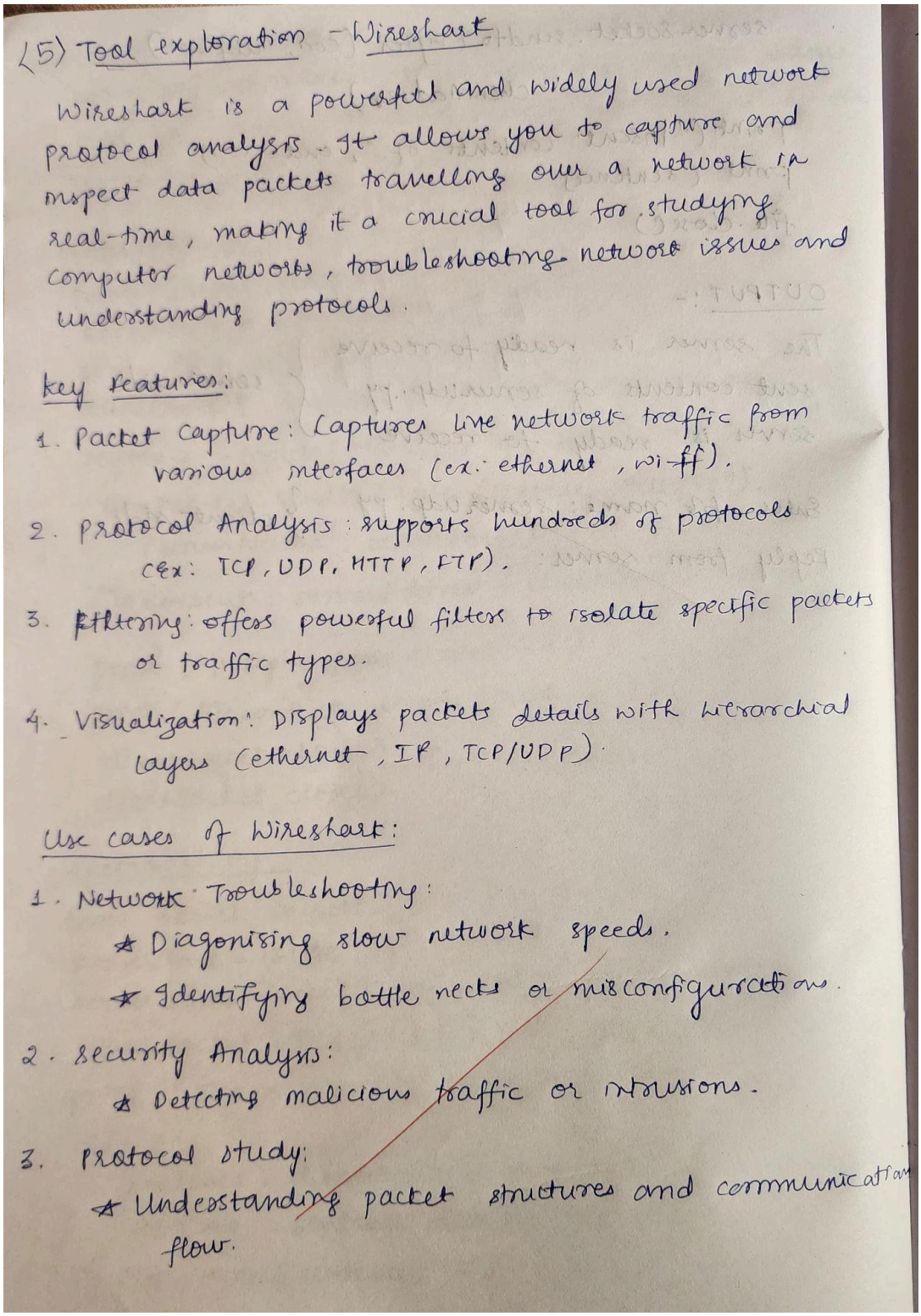
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

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()
```

Output:

```
PS C:\Users\Dell\OneDrive\Desktop\code> python udpserver.py
The server is ready to receive

Sent contents of  example.txt
```

```
PS C:\Users\Dell\OneDrive\Desktop\code> python udpclient.py

Enter file name:  example.txt

Reply from Server:

Hello, this is a sample file.
It is used for testing the TCP server.
```

5. Tool exploration - Wireshark:



(5) Tool exploration - Wireshark

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

Key features:

1. Packet capture: Captures live network traffic from various interfaces (ex: ethernet, wi-ff).
2. Protocol Analysis: supports hundreds of protocols (ex: TCP, UDP, HTTP, FTP).
3. Filtering: offers powerful filters to isolate specific packets or traffic types.
4. Visualization: Displays packets details with hierarchial layers (ethernet, IP, TCP/UDP).

Use cases of Wireshark:

1. Network Troubleshooting:
   - ★ Diagonising slow network speeds.
   - ★ Identifying bottle necks or misconfigurations.
2. Security Analysis:
   - ★ Detecting malicious traffic or intrusions.
3. Protocol study:
   - ★ Understanding packet structures and communication flow.

Common Fitness:

* http : show only HTTP traffic.
* tcp.port == 80 : Show traffic on TCP port 80
* ip.addr == 192.168.1.1 : show packets to or from a specific IP address.
* udp : show only UDP traffic.