VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

BHAVYA HANWITHA (1BM22CS095)

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 E Bhavya Hanwitha(1BM22CS095) 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.

DR. LATHA N.R
Assistant Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda
Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

CYCLE 1

Sl. No.	Date	Experiment Title	Page No.
1	25-09-2024	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.	
2	9-10-2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	
3	23-10-2024	Configure default route, static route to the Router	
4	13-11-2024	Configure DHCP within a LAN and outside LAN	
5	20-11-2024	Configure RIP routing Protocol in Routers	
6	27-11-2024	Configure OSPF routing protocol	
7	20-11-2024	Demonstrate the TTL/ Life of a Packet	
8	18-12-2024	Configure Web Server, DNS within a LAN.	
9	18-12-2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	
10	18-12-2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	
11	18-12-2024	To construct a VLAN and make the PC's communicate among a VLAN	
12	18-12-2024	To construct a WLAN and make the nodes communicate wirelessly	

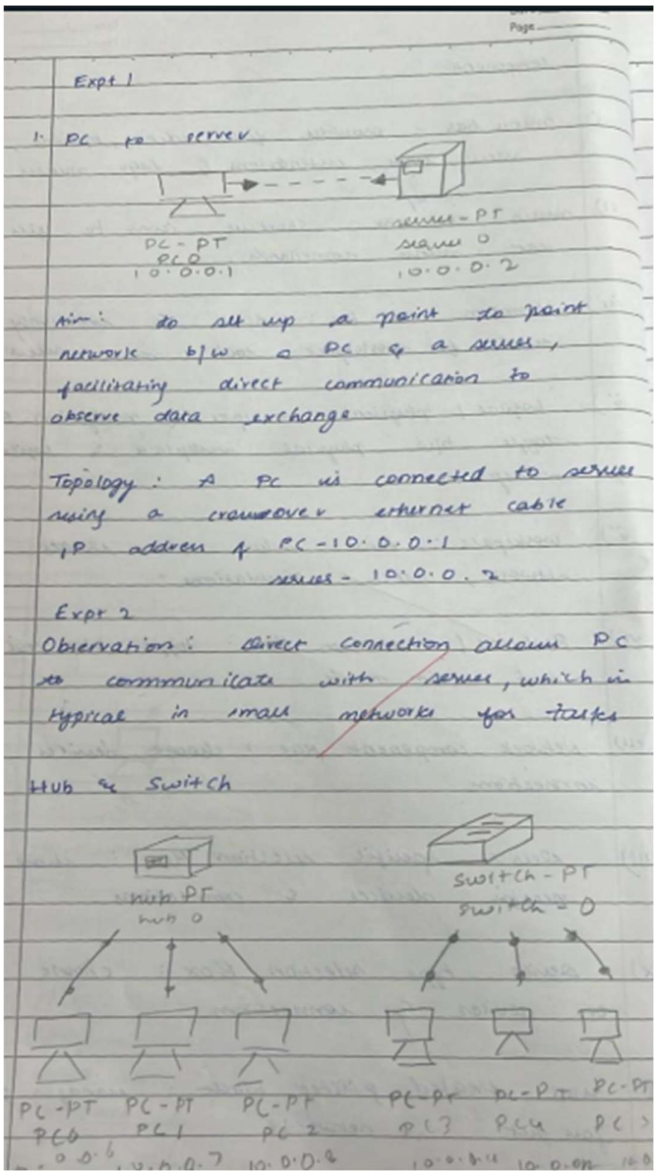
INDEX

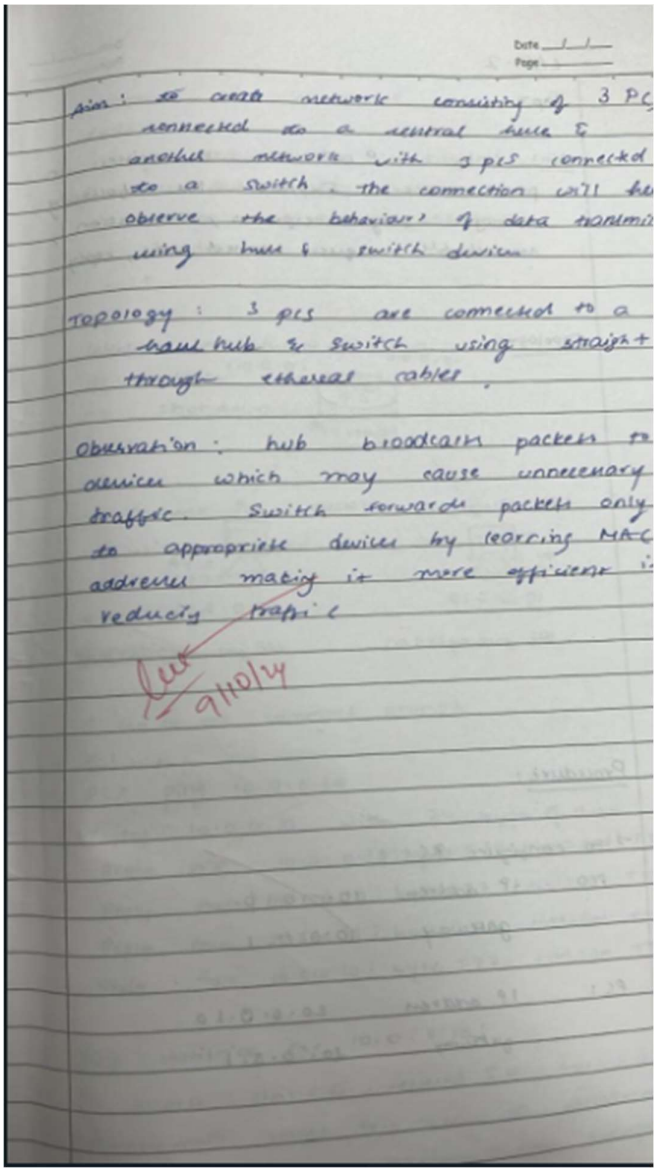
CYCLE 2

Sl. No.	Date	Experiment Title	Page No.
1	1-1-2025	Write a program for error detecting code using CRC-CCITT (16-bits).	
2	1-1-2025	Write a program for congestion control using Leaky bucket algorithm.	
3	2-1-2025	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.	
4	3-1-2025	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.	
5	3-1-2025	Tool Exploration –Wireshark	

Cycle -1
Experiment 1:
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message

Observation:




Date / /
Page

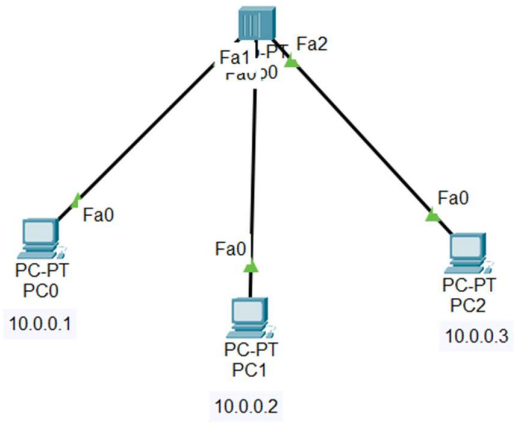
Aim: To create network consisting of 3 PC connected to a central hub & another network with 3 pcs connected to a switch. the connection will be observe the behaviour of data transmi using hub & switch device.

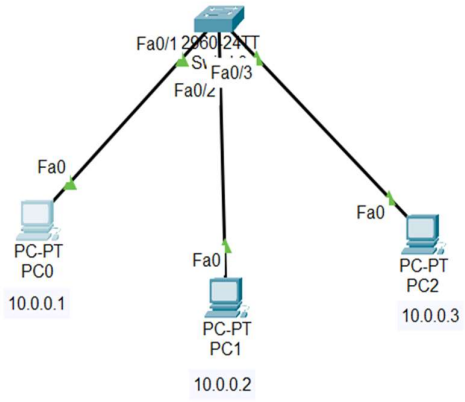
Topology: 3 pcs are connected to a hub hub & switch using straight through ethernet cables.

Observation: hub broadcasts packets to devices which may cause unnecessary traffic. Switch forwards packets only to appropriate devices by learning MAC address making it more efficient i reducing traffic

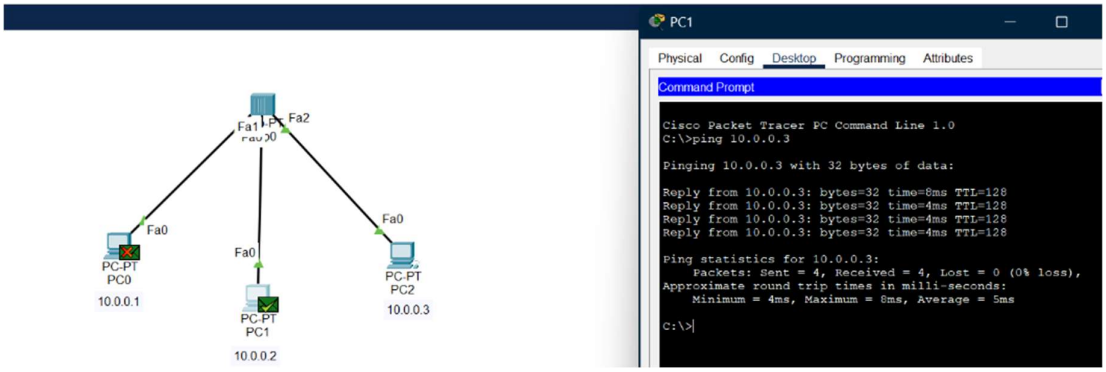
9/10/24

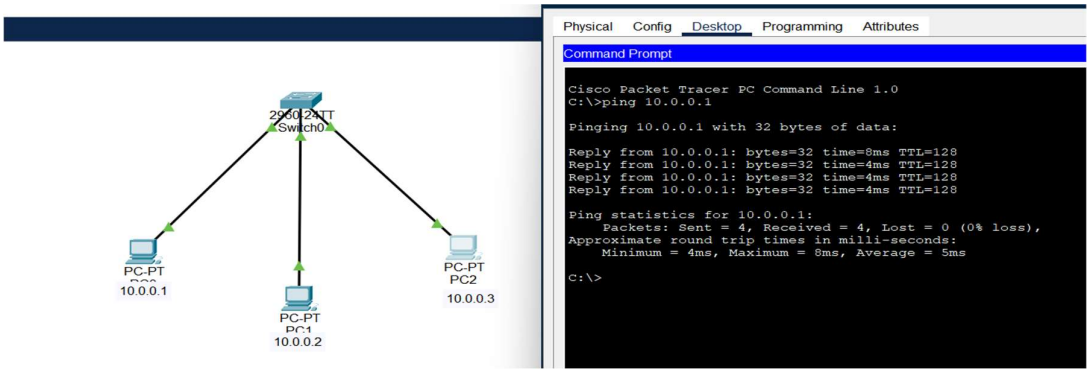
Topology:


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


Fa0/3
Fa0
Fa0
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3

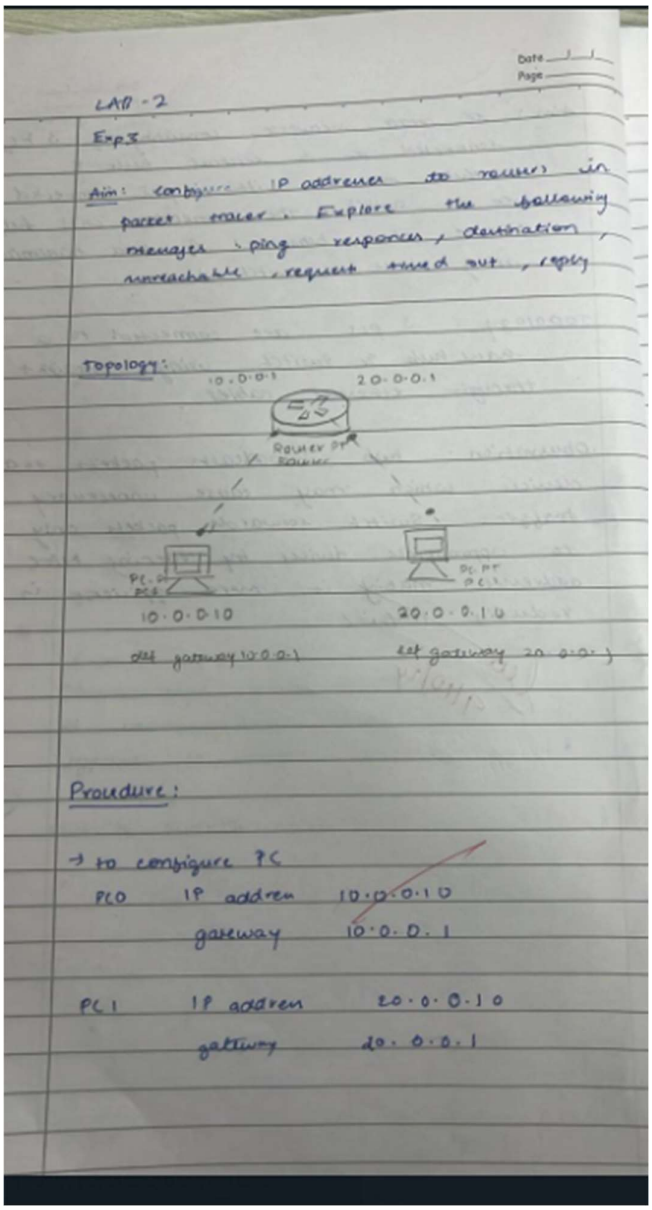
Output:


PC1
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=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:\>
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


Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=8ms TTL=128
Reply from 10.0.0.1: bytes=32 time=4ms TTL=128
Reply from 10.0.0.1: bytes=32 time=4ms TTL=128
Reply from 10.0.0.1: bytes=32 time=4ms TTL=128
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 8ms, Average = 5ms
C:\>
2950-24TT
Switch0
PC-PT
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3

Experiment 2:
Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

Observation Book:



Date / /
Page

command line interface :
enable
config terminal
interface fastethernet 0/0
ip addres 10.0.0.10 255.0.0.0
no shutdown
exit

interface fastethernet 0/1
ip address 20.0.0.1
no shut down
exit

→ connect the router to both the PC using copper cross over wire

observation:

→ go to the command prompt
PC1
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=127
Reply from 10.0.0.10: bytes=32 time=0ms TTL=127
Reply from 10.0.0.10: bytes=32 time=0ms TTL=127
Reply from 10.0.0.10: bytes=32 time=0ms TTL=127

Ping statistics for 10.0.0.10:
packets: sent=4, recieved=4, lost=0, 0% loss
Approximate round trip times in milliseconds
Minimum=0ms, Maximum=0ms, Average=0ms

Date / /
Page

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, FastEthernet 0/0

C 20.0.0.0/8 is directly connected, FastEthernet 1/0

9/10/24

CNB-1

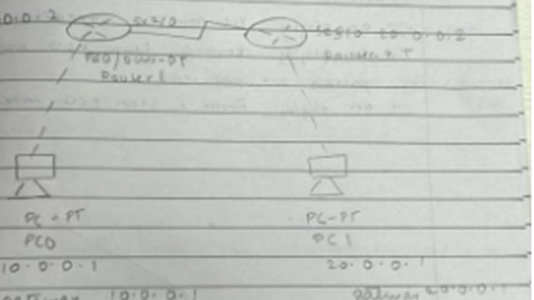
Exp-2

Date / /
Page

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

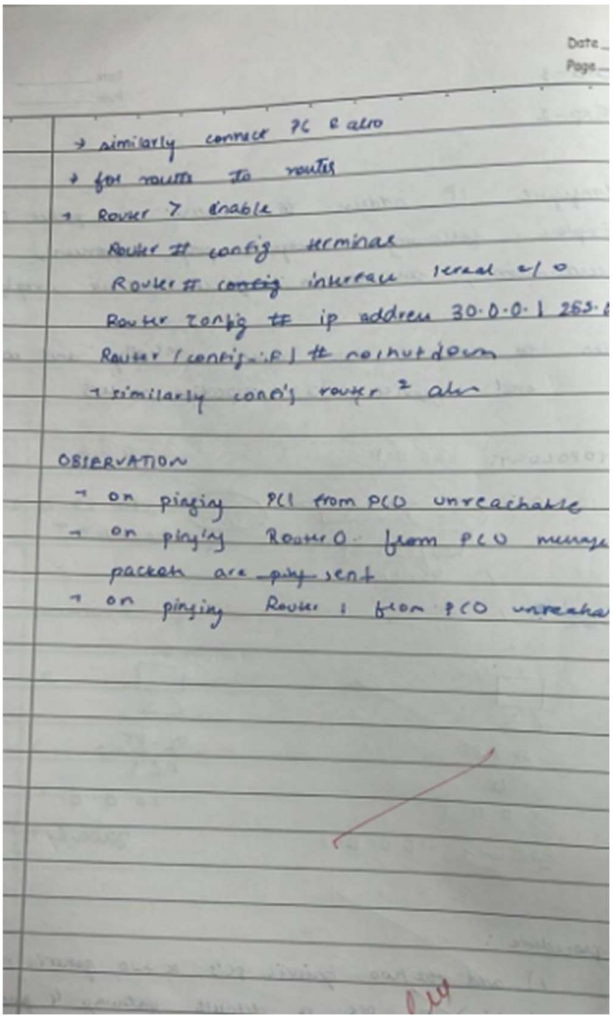
Aim: To connect two routers serially and connect end devices to respective routers

Topology:



Procedure:

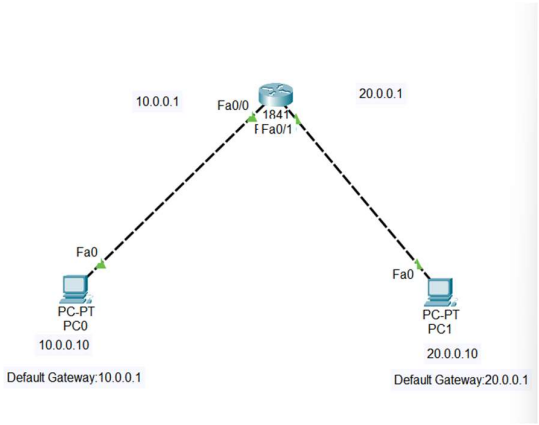
1) add two generic PC's & two generic routers
2) Give the PCs a default gateway & address
3) each PC is connected with copper with to router
4) click on the router and go to CLI command
5) router > enable
Router # config terminal
Router (config)# interface fastethernet 0/0
Router (config-if)# no shutdown
exit

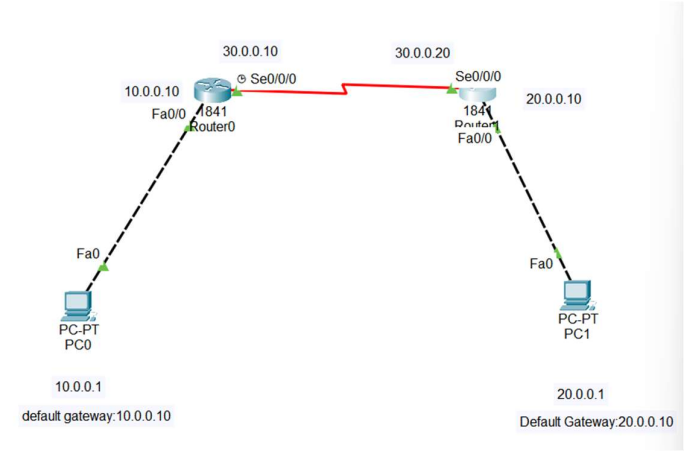


→ similarly connect PC & also
→ for router to router
→ Router > enable
Router # config terminal
Router # config interface serial 2/0
Router config ## ip address 30.0.0.1 255.
Router (config-if) # no shutdown
→ similarly config's router 2 also

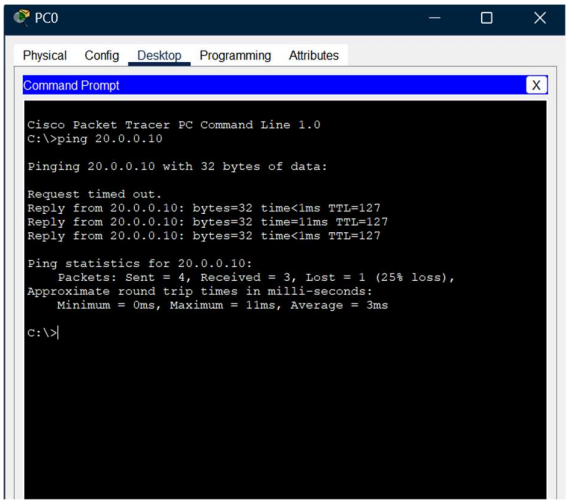
OBSERVATION
→ on pinging PC1 from PC0 unreachable
→ on pinging Router0 from PC0 messa
packets are ping sent
→ on pinging Router1 from PC0 unreacha

Topology:

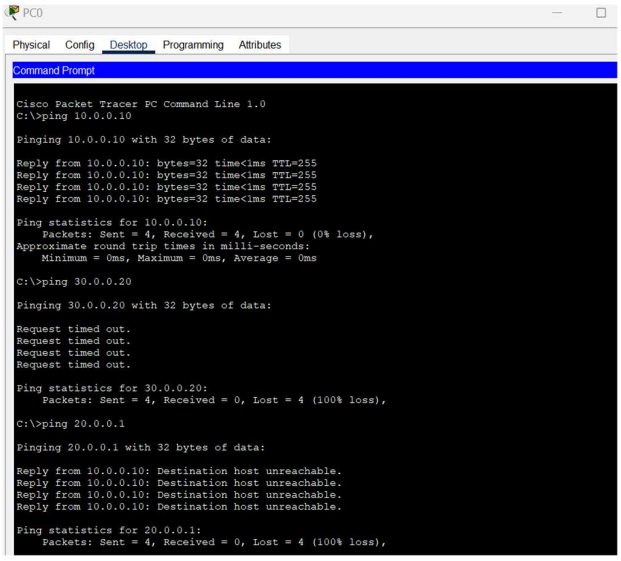

10.0.0.1
Fa0/0
1841
Fa0/1
20.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
Default Gateway:10.0.0.1
Fa0
PC-PT
PC1
20.0.0.10
Default Gateway:20.0.0.1


30.0.0.10
30.0.0.20
Se0/0/0
Se0/0/0
10.0.0.10
Fa0/0
1841
Router0
20.0.0.10
Fa0/0
Fa0
PC-PT
PC0
10.0.0.1
default gateway:10.0.0.10
Fa0
PC-PT
PC1
20.0.0.1
Default Gateway:20.0.0.10

Output:


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.10
Pinging 20.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time=11ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Ping statistics for 20.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 11ms, Average = 3ms
C:\>

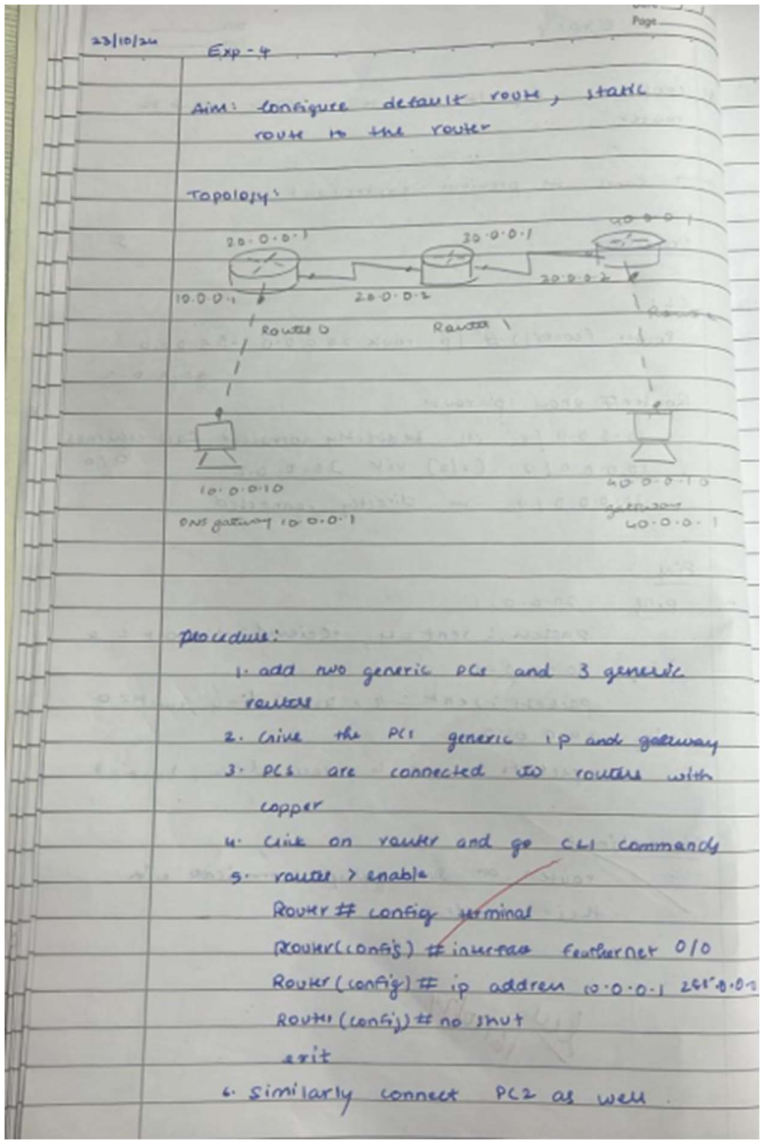

Router0
Physical
Config
CLI
Attributes
IOS Command Line Interface
Press RETURN to get started.
Router>enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - 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 not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 30.0.0.0/8 is directly connected, Serial0/0/0


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.10
Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time<1ms TTL=255
Reply from 10.0.0.10: bytes=32 time<1ms TTL=255
Reply from 10.0.0.10: bytes=32 time<1ms TTL=255
Reply from 10.0.0.10: bytes=32 time<1ms 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 = 0ms, Average = 0ms
C:\>ping 30.0.0.20
Pinging 30.0.0.20 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 30.0.0.20:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

Experiment 3:

Configure default route, static route to the Router

Observation Book:



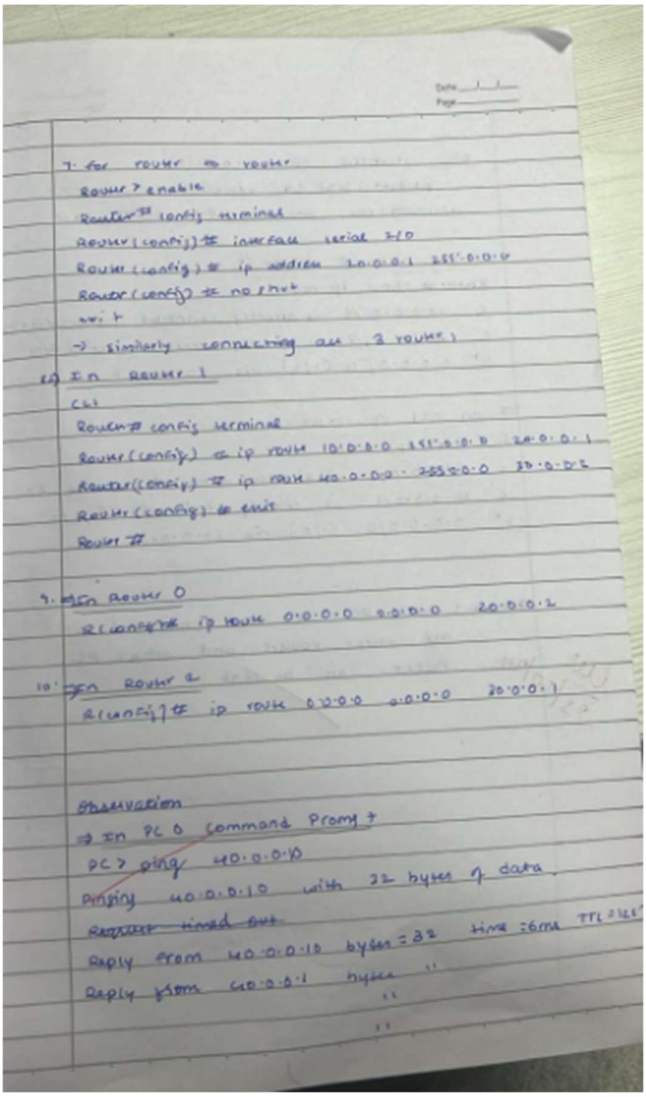
23/10/24 Exp - 4

Aim: Configure default route, static route to the router

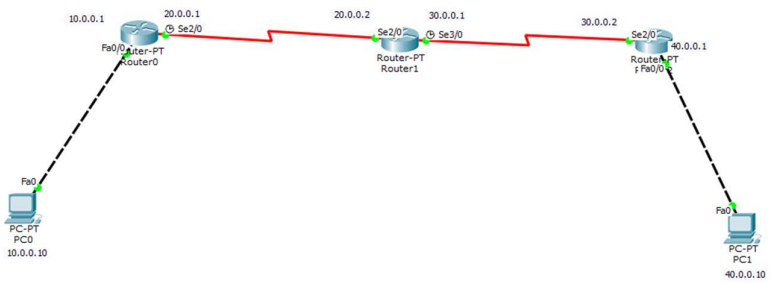
Topology:

Procedure:

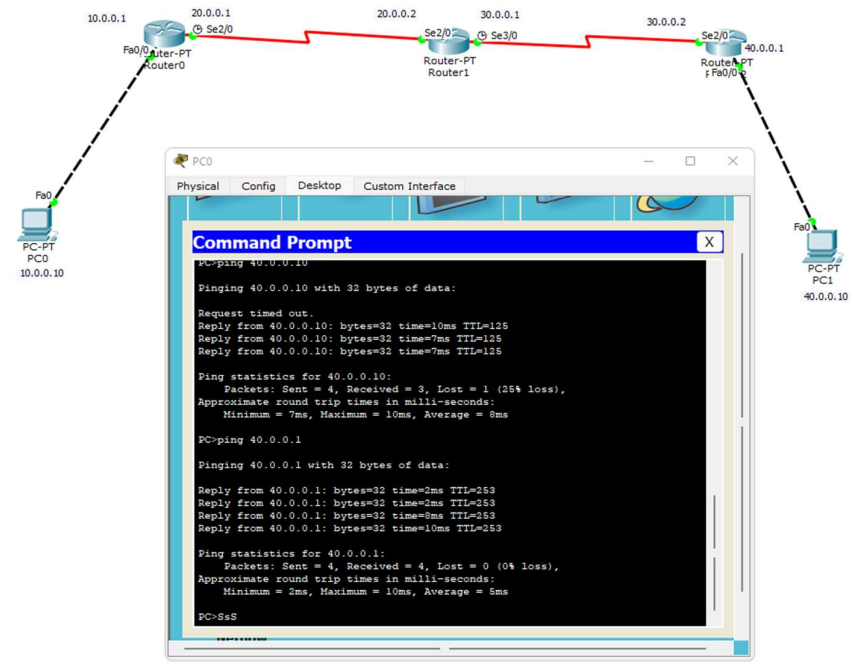
1. add two generic PCs and 3 generic routers
2. Give the PCs generic ip and gateway
3. PCs are connected to routers with copper
4. Click on router and go CLI commands
5. router > enable
   Router # config terminal
   Router(config) # interface FastEthernet 0/0
   Router(config) # ip address 10.0.0.1 255.0.0.0
   Router(config) # no shut
   exit
6. Similarly connect PC2 as well

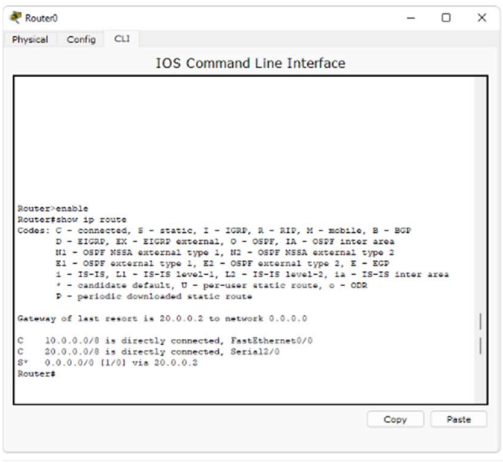

7. for router ⇒ router
Router > enable
Router# config terminal
Router(config)# interface serial 2/0
Router(config)# ip address 20.0.0.1 255.0.0.0
Router(config)# no shut
ex: 1
→ similarly connecting all 3 routers
8) In Router 1
CLI
Router# config terminal
Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)# exit
Router #
9. In Router 0
R(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2
10. In Router 2
R(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1
Observation
⇒ In PC 0 command Prompt
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 40.0.0.1 bytes=32

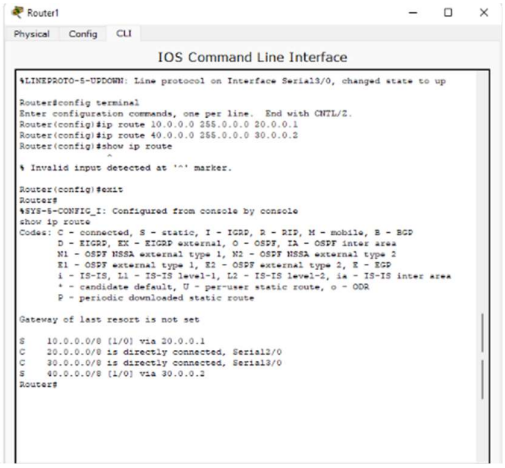
Topology:


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

Output:

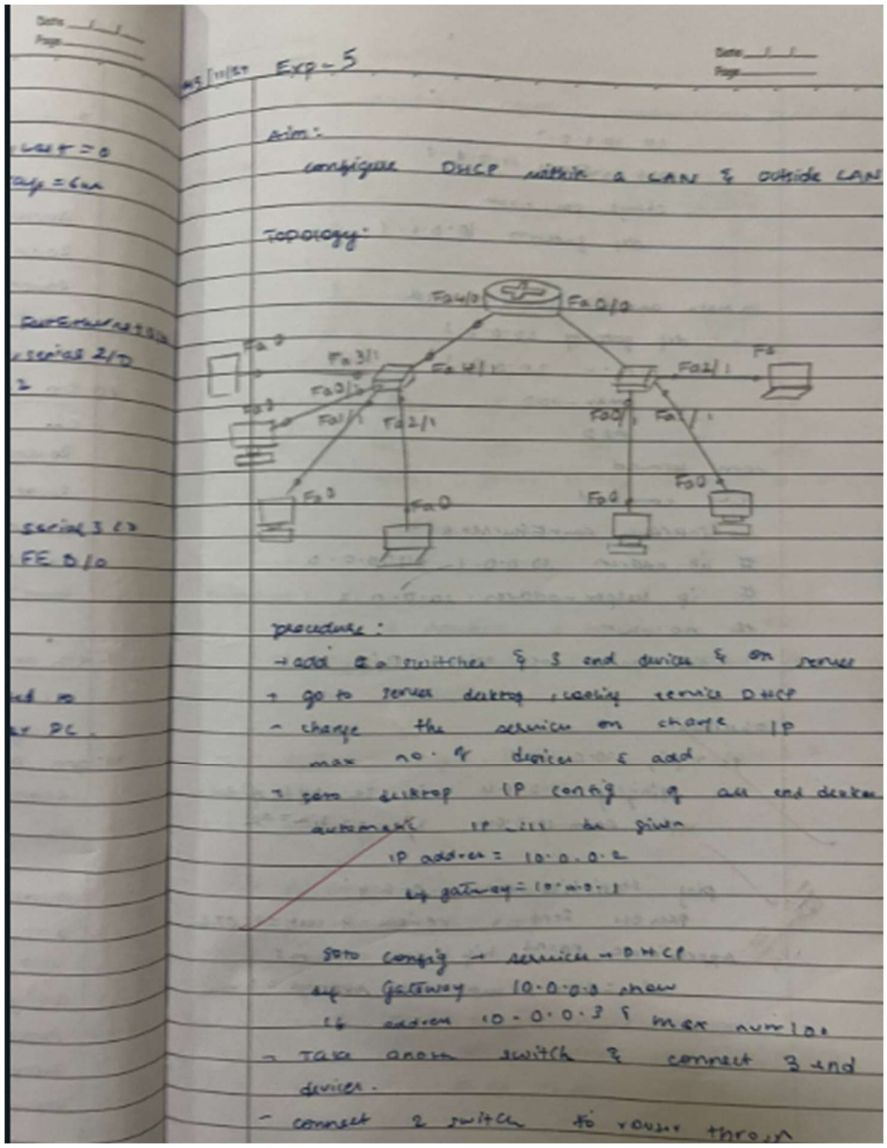

10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
Se2/0
40.0.0.1
Router-PT
Fa0/0
Fa0
PC-PT
PC0
10.0.0.10
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
PC>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply 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
Fa0
PC-PT
PC1
40.0.0.10


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#

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

Observation Book:



IP 10.0.0.2
def gateway 10.0.0.1
change pool name
def gateway 60.0.0.1

- make another pool & add
- def gateway 20.0.0.1
  IP - 20.0.0.3
  max - 100
  ADD

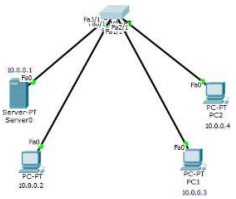
router terminal
CLI command
interface fastEthernet 0/0
ip address 10.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut
do same for other 0/0

observation
PC > ping 10.0.0.4
pinging 10.0.0.4 with 32 bytes of data
reply from 10.0.0.4 bytes=32 time=0ms

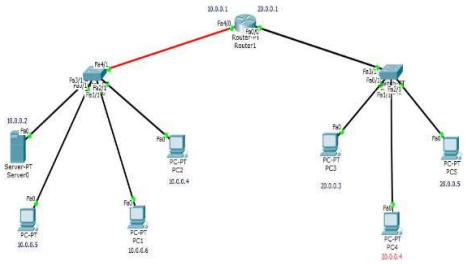
ping statistics for 10.0.0.4:
Packets Sent=4 recieved=4 lost=0 0% loss
Approximate round trip time in ms
min=0ms, max=0ms, Average=0ms

Topology:

(within Lan)

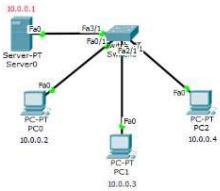


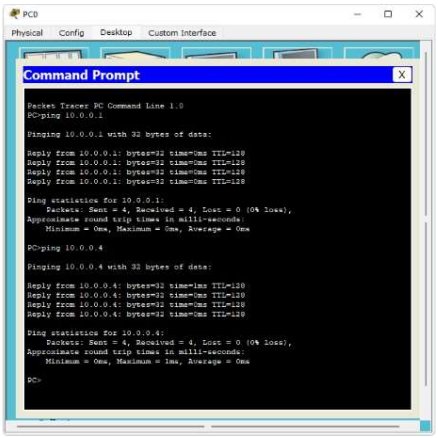
(outside Lan)



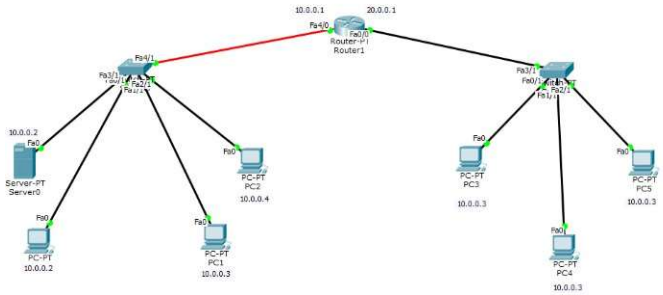
Output:

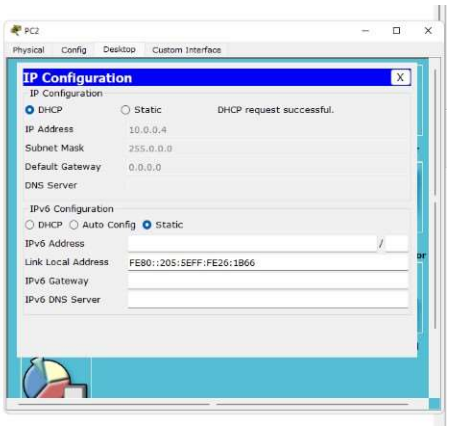
(within Lan)

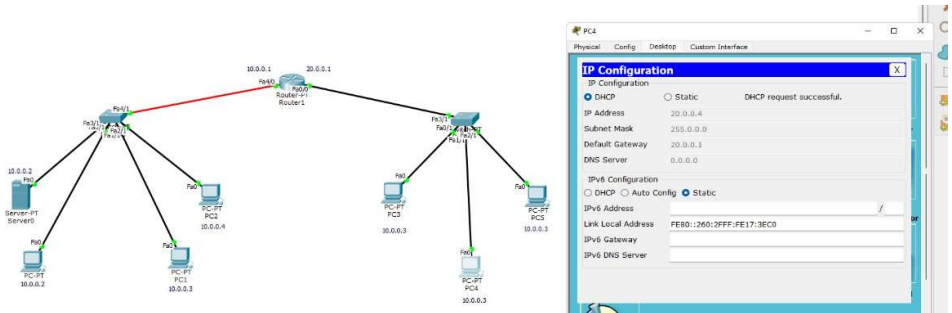




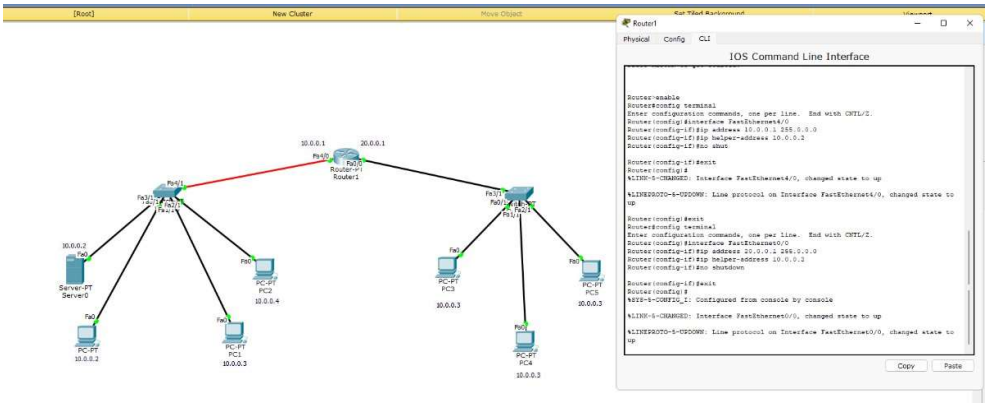
(outside Lan)






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

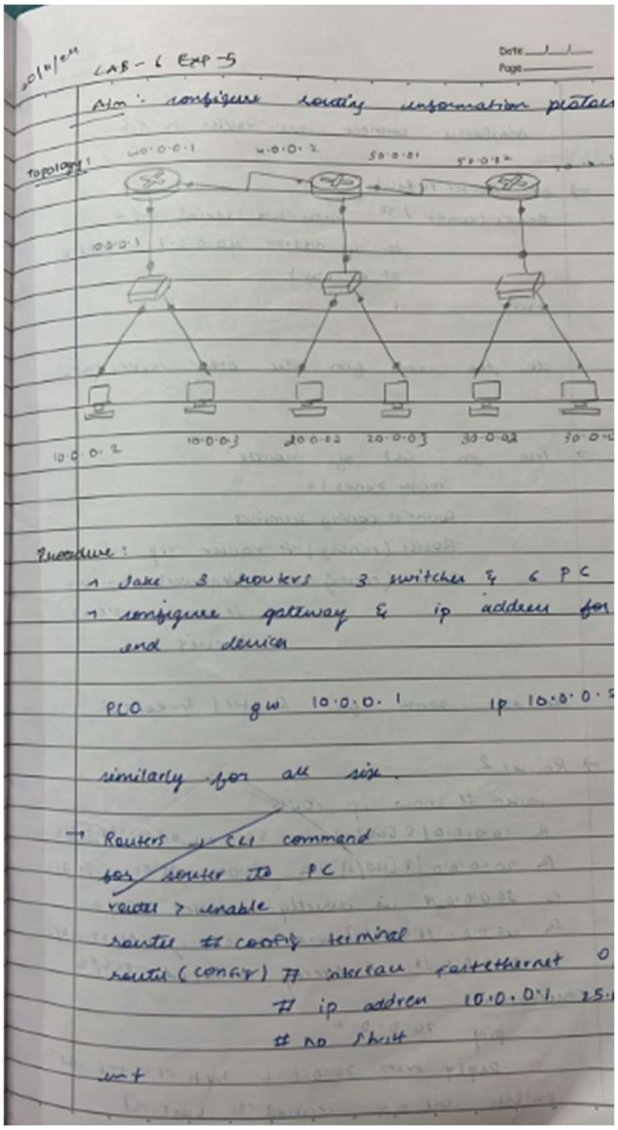
PC4
Physical
Config
Desktop
Custom Interface
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
20.0.0.4
Subnet Mask
255.0.0.0
Default Gateway
20.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
Auto Config
IPv6 Address
Link Local Address
FE80::260:2FFF:FE17:3EC0
IPv6 Gateway
IPv6 DNS Server

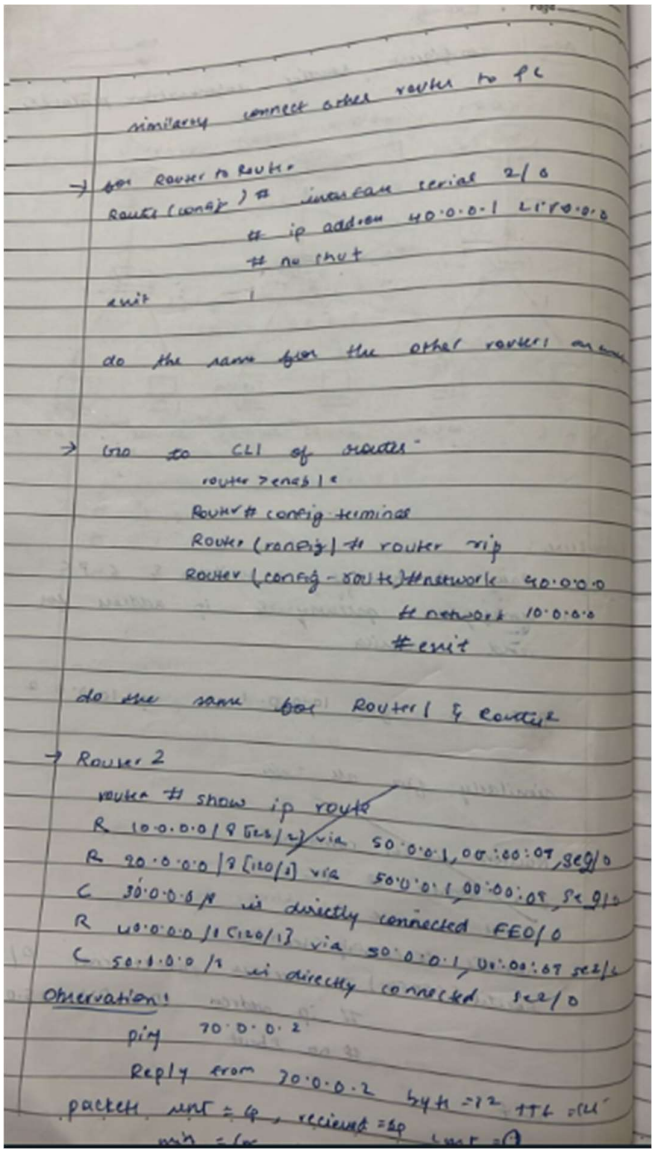

[Root]
New Cluster
10.0.0.1
20.0.0.1
Router-PT
Router1
Server-PT
Server0
10.0.0.2
PC-PT
PC1
10.0.0.3
PC2
10.0.0.4
PC3
PC4
PC5

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 10.0.0.2
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

Experiment 5:
Configure RIP routing Protocol in Routers

Observation Book:




similarly connect other routers to PC

→ for Router to Router

Router (config)# interface serial 2/0
ip address 40.0.0.1 255.0.0.0
no shut
exit

do the same for the other routers as well

→ Go to CLI of router

router >enable
Router# config terminal
Router (config)# router rip
Router (config-router)#network 40.0.0.0
network 10.0.0.0
#exit

do the same for Router1 & router2

→ Router 2

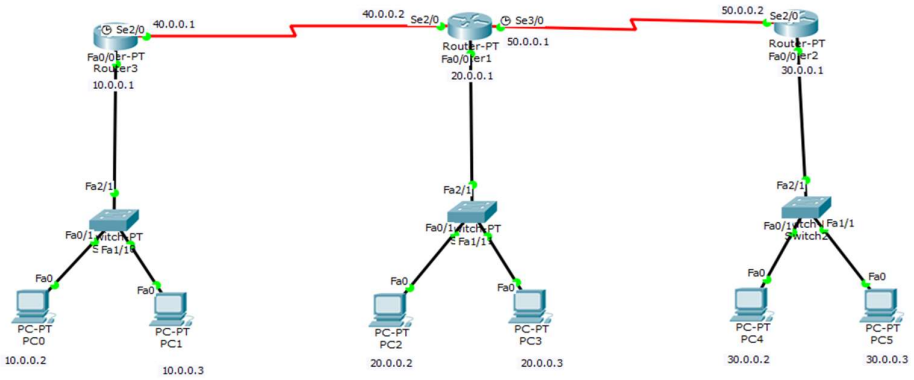
router # show ip route
R 10.0.0.0/8 [120/1] via 50.0.0.1, 00:00:07, se2/0
R 20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:07, se2/0
C 30.0.0.0/8 is directly connected FE0/0
R 40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:07 se2/0
C 50.0.0.0/8 is directly connected se2/0

Observation:

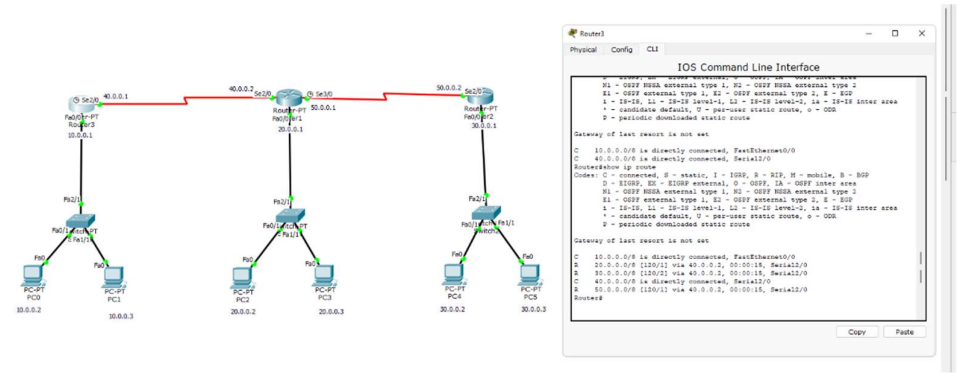
ping 70.0.0.2

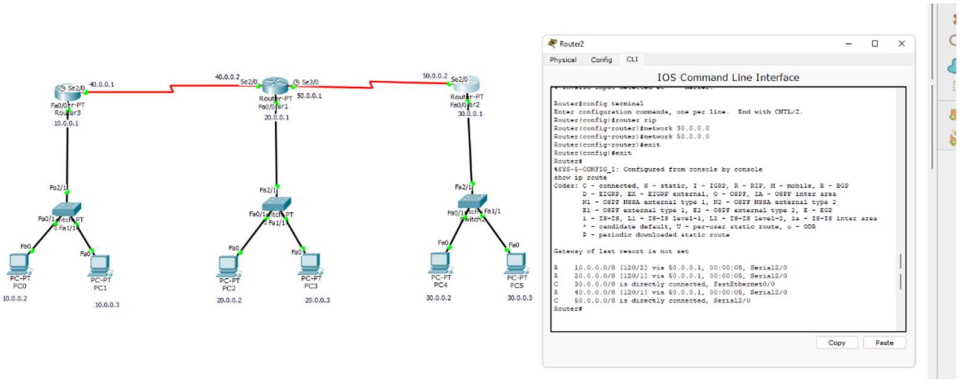
Reply from 70.0.0.2 byte=32 TTL=125
packet sent = 4, received = 4, lost = 0

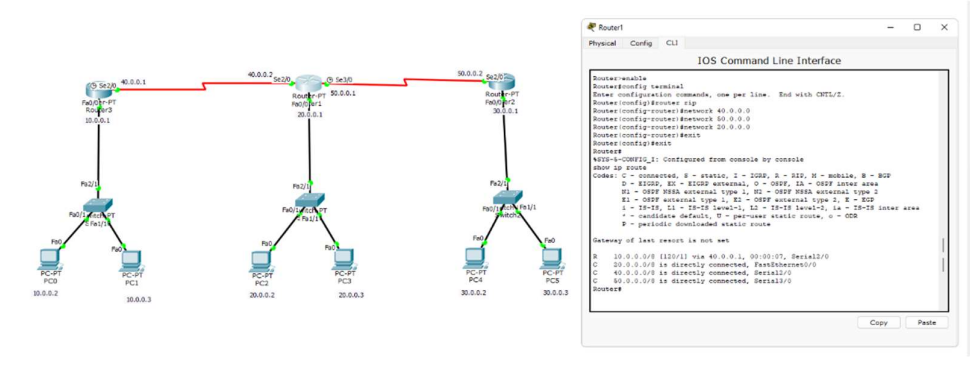
Topology:

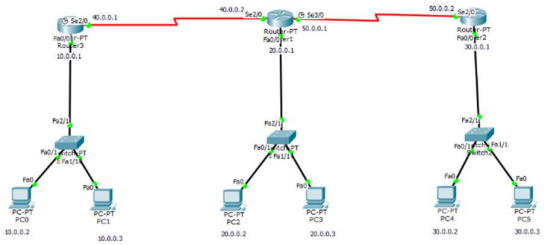

40.0.0.1
40.0.0.2
Se2/0
Se3/0
50.0.0.1
50.0.0.2
Router3
Router1
Router2
10.0.0.1
20.0.0.1
30.0.0.1
Fa2/1
Fa0/1
Fa1/1
Switch
PC0
PC1
PC2
PC3
PC4
PC5
10.0.0.2
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3

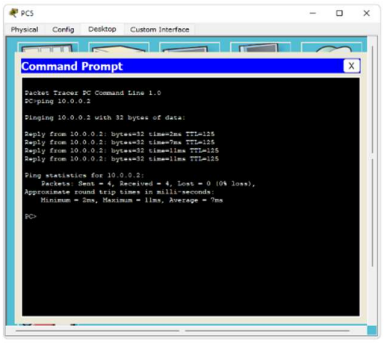
Output:


IOS Command Line Interface
Router#show ip route
Gateway of last resort is not set
Copy
Paste


IOS Command Line Interface
Router#config terminal
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Gateway of last resort is not set
Copy
Paste

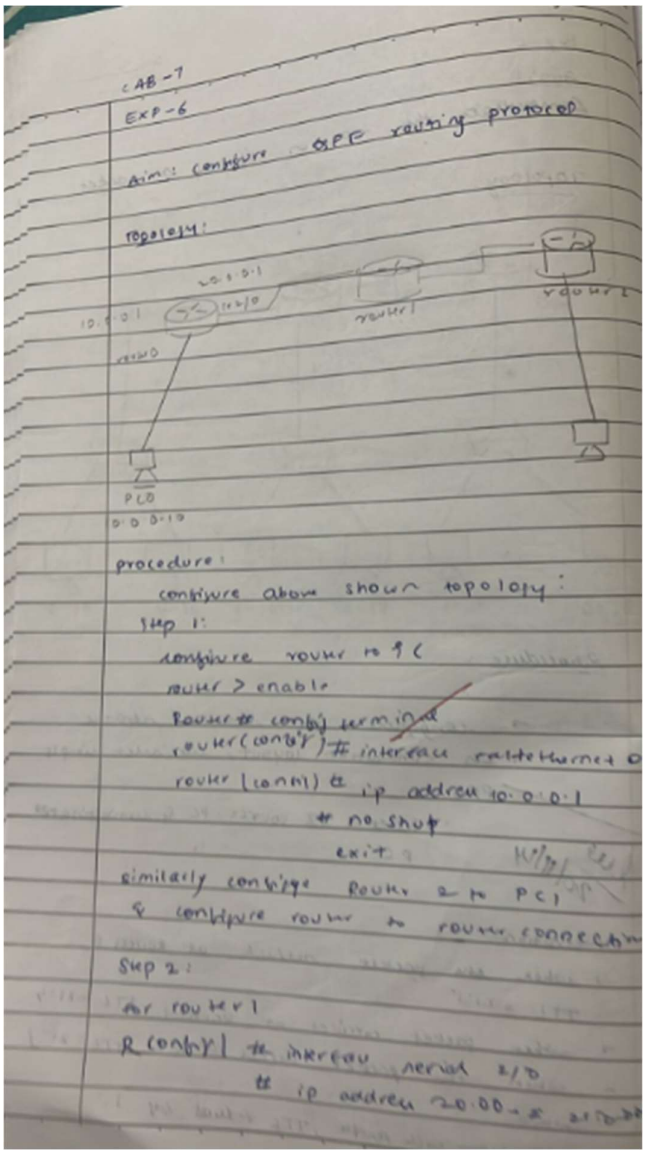

IOS Command Line Interface
Router>enable
Router#config terminal
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Gateway of last resort is not set
Copy
Paste

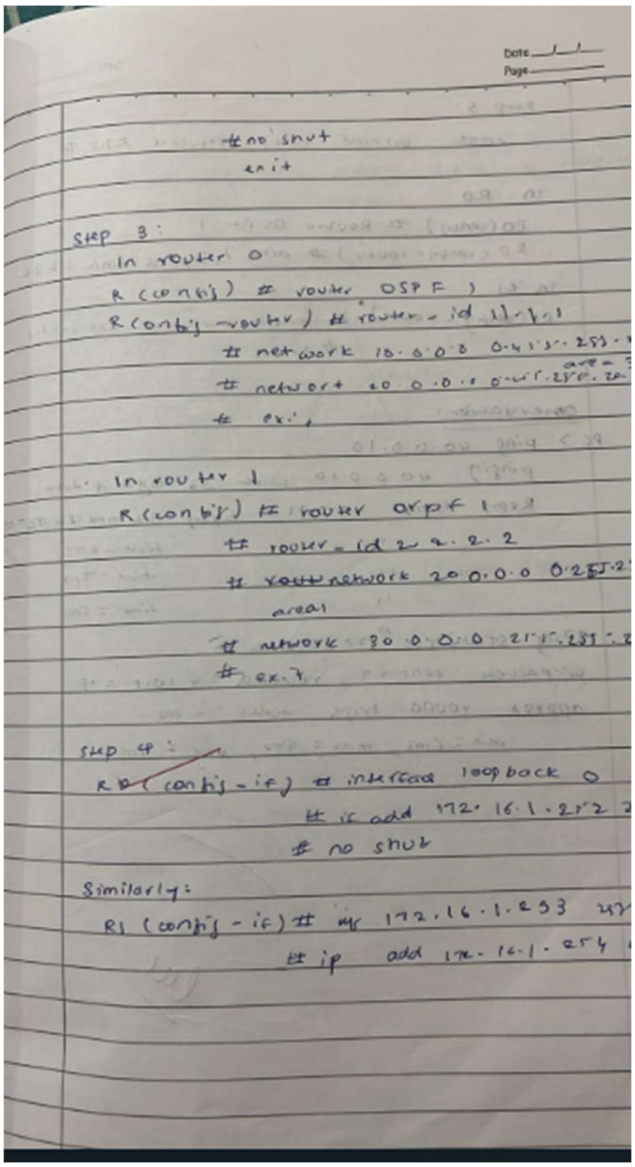

40.0.0.1
40.0.0.2
50.0.0.1
50.0.0.2
10.0.0.1
20.0.0.1
30.0.0.1
PC-PC PC0
PC-PC PC1
PC-PC PC2
PC-PC PC3
PC-PC PC4
PC-PC PC5
10.0.0.2
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3


PC5
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=2ms TTL=125
Reply from 10.0.0.2: bytes=32 time=7ms TTL=125
Reply from 10.0.0.2: bytes=32 time=11ms TTL=125
Reply from 10.0.0.2: bytes=32 time=11ms TTL=125
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 11ms, Average = 7ms
PC>

Experiment 6:
Configure OSPF routing protocol

Observation Book:





Date / /
Page

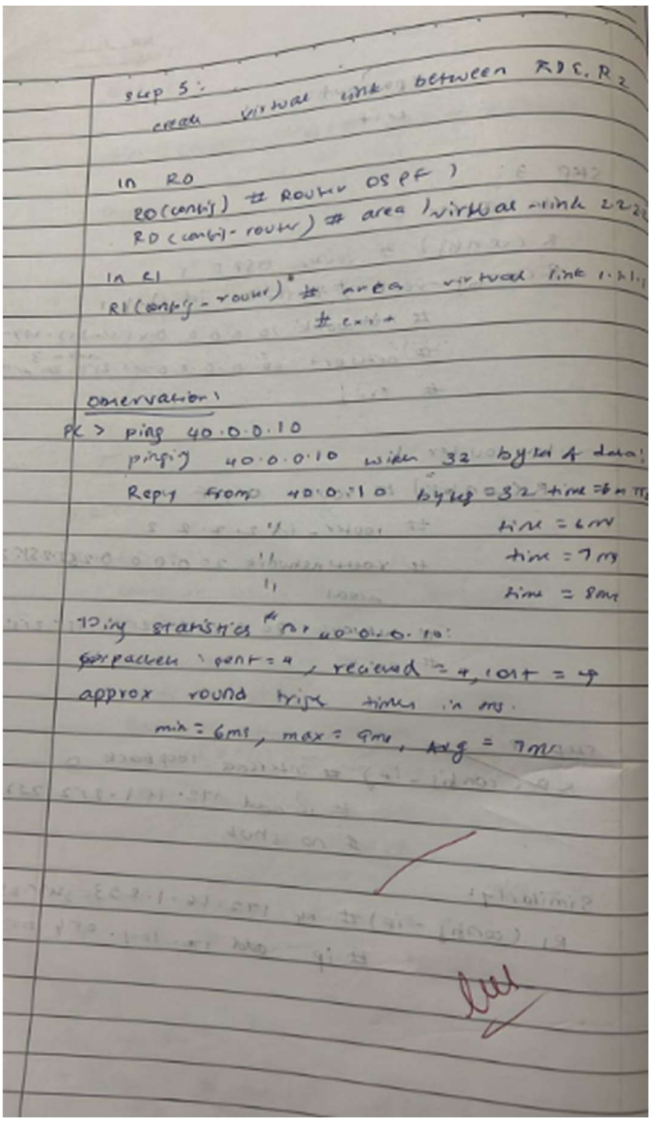
no shut
exit

Step 3:
In router 0
R(config) # router OSPF 1
R(config-router) # router-id 1.1.1.1
network 10.0.0.0 0.255.255.255 area 1
network 20.0.0.0 0.255.255.255 area 1
ex.

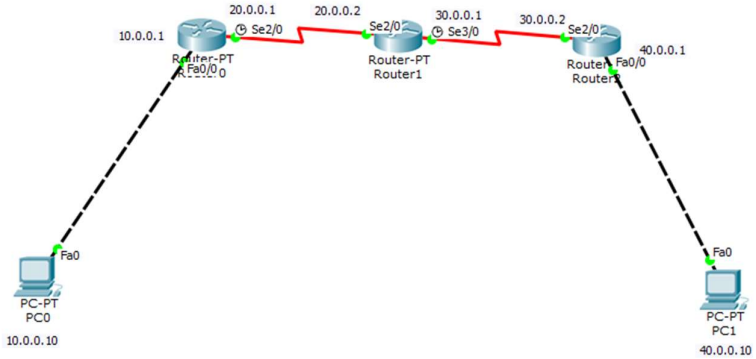
In router 1
R(config) # router ospf 1
router-id 2.2.2.2
router network 20.0.0.0 0.255.255.255 area 1
network 30.0.0.0 255.255.255.255 area 1
ex.

Step 4:
R1(config-if) # interface loopback 0
ip add 172.16.1.252 255.255.255.255
no shut

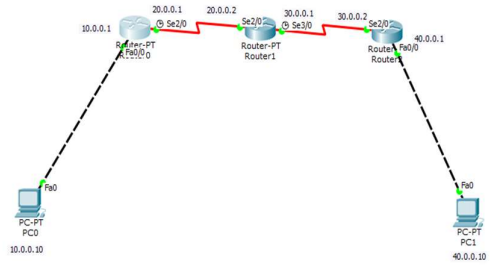
Similarly:
R1(config-if) # ip 172.16.1.253 255.255.255.255
ip add 172.16.1.254 255.255.255.255

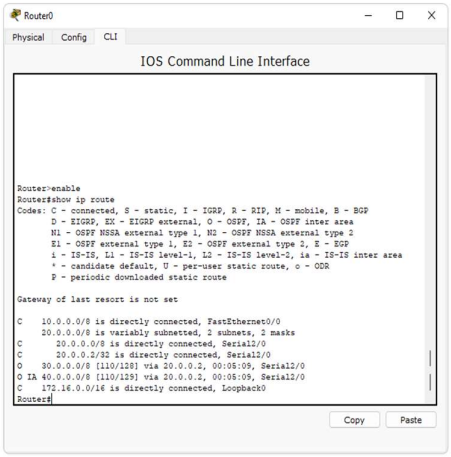

step 5:
create virtual link between R0 & R2
in R0
R0(config) # Router OSPF 1
R0(config-router) # area 1 virtual-link 2.2.2.2
in R1
R1(config-router) # area virtual link 1.1.1.1
exit
observation:
PC > ping 40.0.0.10
pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10 bytes=32 time=6ms TTL=125
time = 6ms
time = 7ms
time = 8ms
Ping statistics for 40.0.0.10:
packets: sent = 4, received = 4, lost = 0
approx round trip times in ms:
min = 6ms, max = 8ms, Avg = 7ms

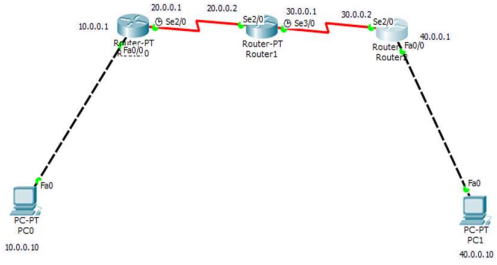
Topology:

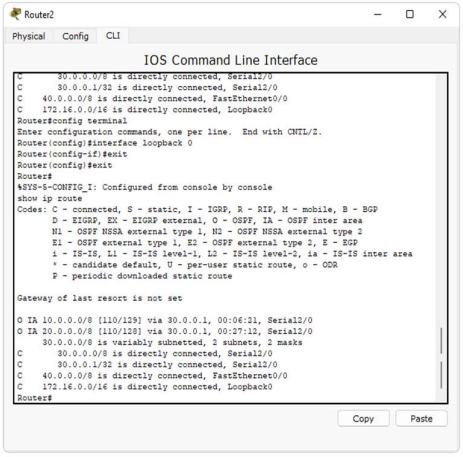

10.0.0.1
20.0.0.1
Se2/0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Router-PT
Router1
Fa0
PC-PT
PC0
10.0.0.10
Fa0
PC-PT
PC1
40.0.0.10

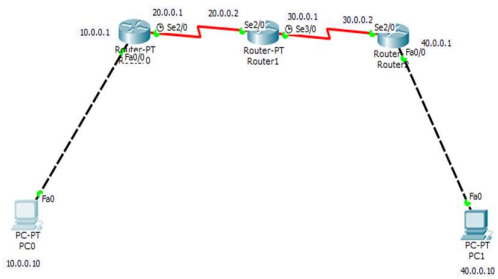
Output:

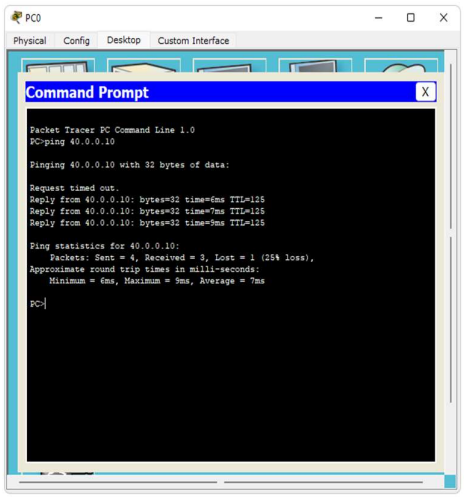

20.0.0.1
20.0.0.2
30.0.0.1
30.0.0.2
10.0.0.1
Se2/0
Se3/0
Router-PT
Router1
Fa0/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC1
40.0.0.10

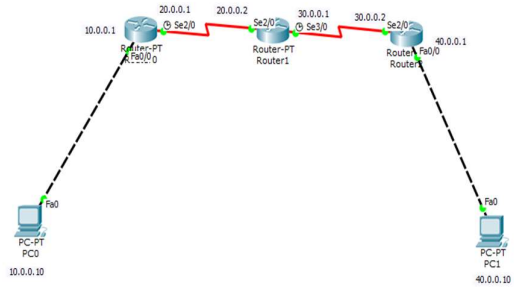

Router0
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external 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 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 30.0.0.0/8 [110/128] via 20.0.0.2, 00:05:09, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:05:09, Serial2/0
C 172.16.0.0/16 is directly connected, Loopback0
Router#
Copy
Paste

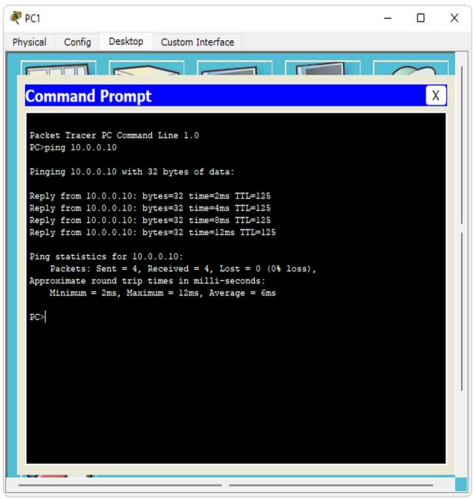

20.0.0.1
20.0.0.2
30.0.0.1
30.0.0.2
10.0.0.1
Se2/0
Se3/0
Router-PT
Router1
Fa0/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC1
40.0.0.10


Router2
Physical
Config
CLI
IOS Command Line Interface
C 30.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.1/32 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
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)#interface loopback 0
Router(config-if)#exit
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
O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:06:21, Serial2/0
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:27:12, Serial2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.1/32 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
C 172.16.0.0/16 is directly connected, Loopback0
Router#
Copy
Paste


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


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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>

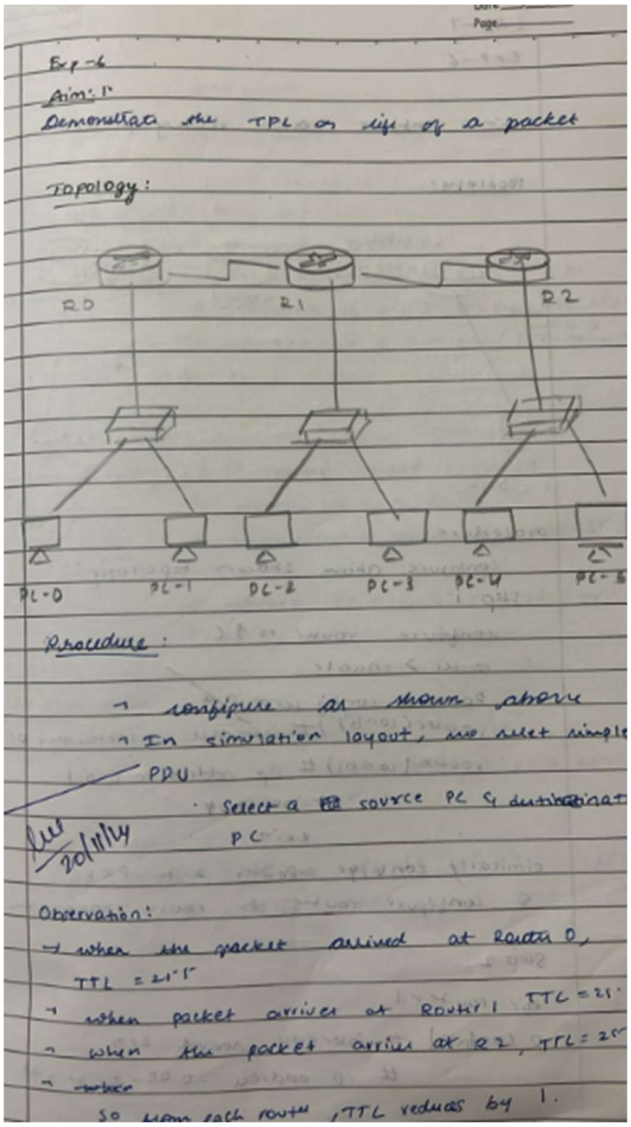

10.0.0.1
20.0.0.1
Se2/0
20.0.0.2
Se2/0
Router-PT
Router1
30.0.0.1
Se3/0
30.0.0.2
Se2/0
Router
Fa0/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
Fa0
PC-PT
PC1
40.0.0.10


PC1
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=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>

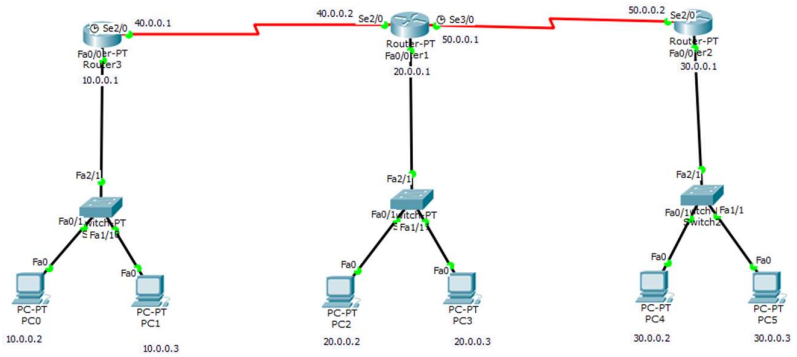
Experiment 7:

Demonstrate the TTL/ Life of a Packet

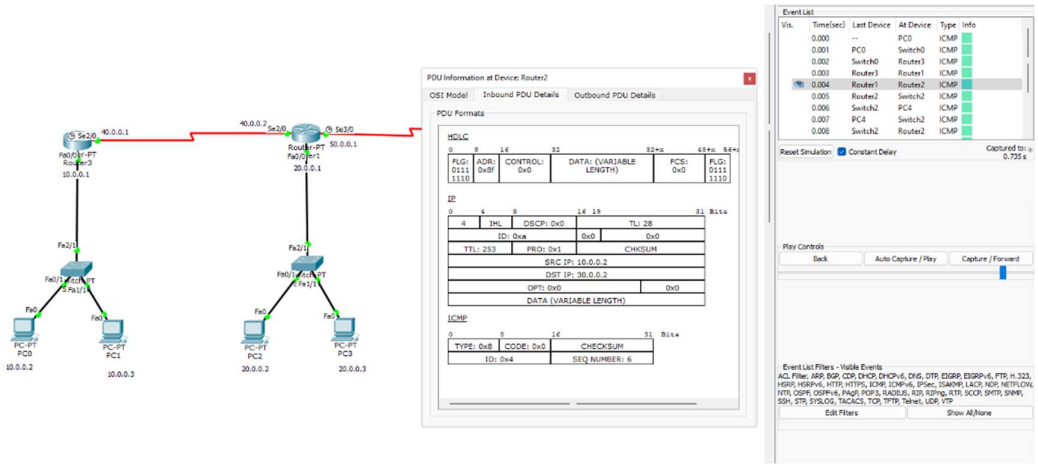
Observation Book:

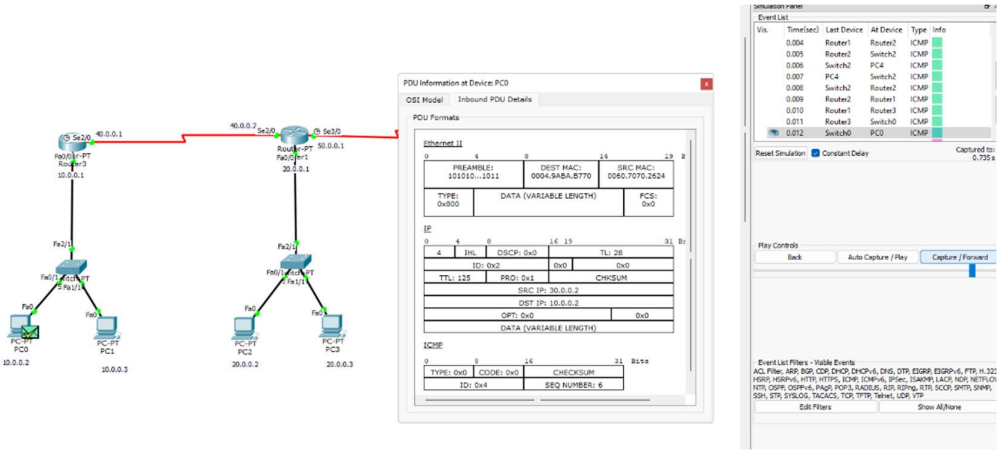


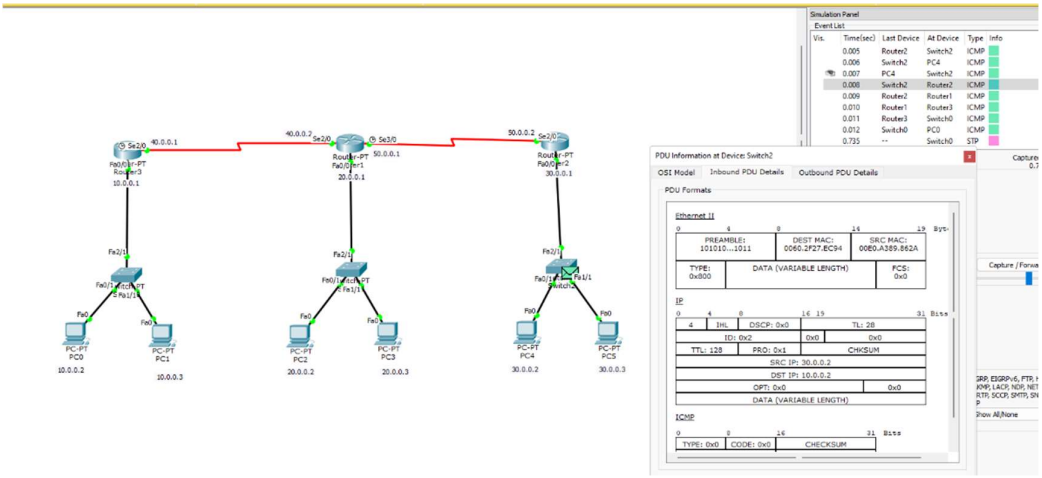
Topology:


Router3
Router1
Router2
Switch0
Switch1
Switch2
PC0
PC1
PC2
PC3
PC4
PC5
10.0.0.1
20.0.0.1
30.0.0.1
40.0.0.1
40.0.0.2
50.0.0.1
50.0.0.2
10.0.0.2
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3

Output:


PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
IP
ICMP
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None

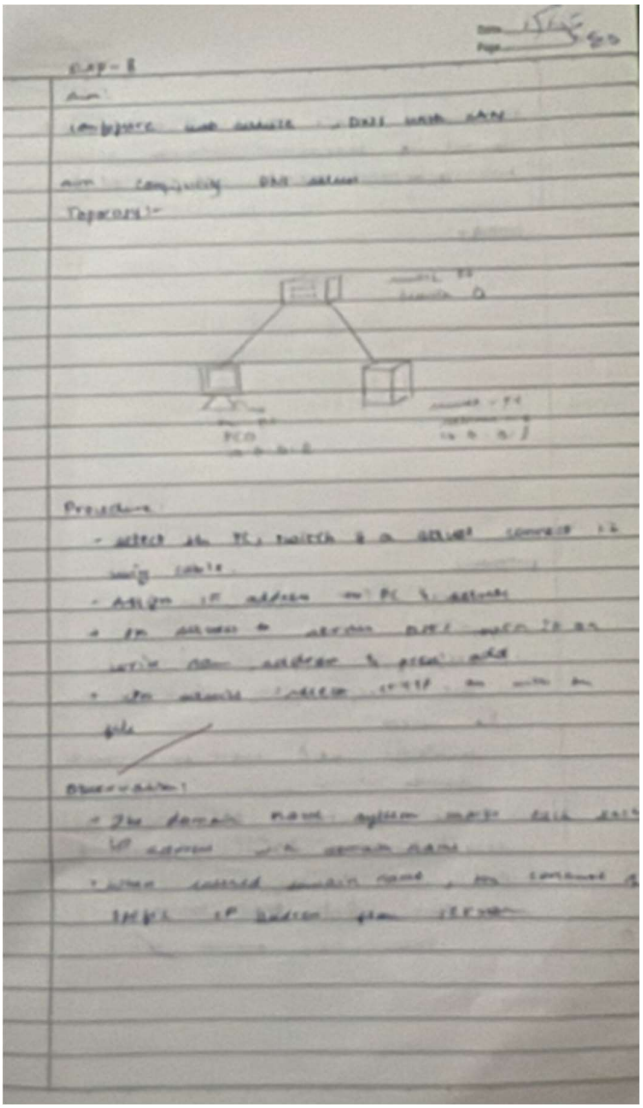

PDU Information at Device: PC0
OSI Model
Inbound PDU Details
PDU Formats
Ethernet II
IP
ICMP
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None


Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.005 Router2 Switch2 ICMP
0.006 Switch2 PC4 ICMP
0.007 PC4 Switch2 ICMP
0.008 Switch2 Router2 ICMP
0.009 Router2 Router1 ICMP
0.010 Router1 Router3 ICMP
0.011 Router3 Switch0 ICMP
0.012 Switch0 PC0 ICMP
0.735 -- Switch0 STP
PDU Information at Device: Switch2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 10101010...1011
DEST MAC: 0060.2F27.EC94
SRC MAC: 00E0.A389.862A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
IHL
DSCP: 0x0
TL: 28
ID: 0x2
TTL: 128
PRO: 0x1
CHKSUM
SRC IP: 30.0.0.2
DST IP: 10.0.0.2
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
Router-PT
PC-PT
Switch-PT
10.0.0.1
20.0.0.1
30.0.0.1
40.0.0.1
40.0.0.2
50.0.0.1
50.0.0.2
10.0.0.2
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3

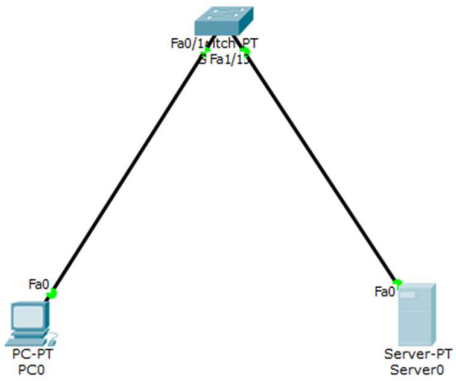
Experiment 8:

Configure Web Server, DNS within a LAN

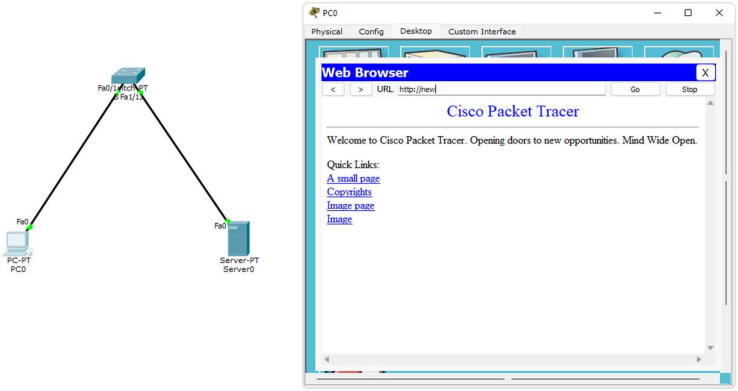
Observation Book:

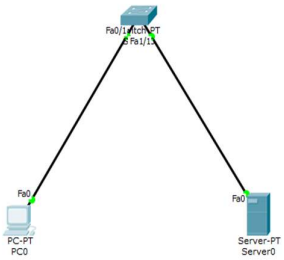


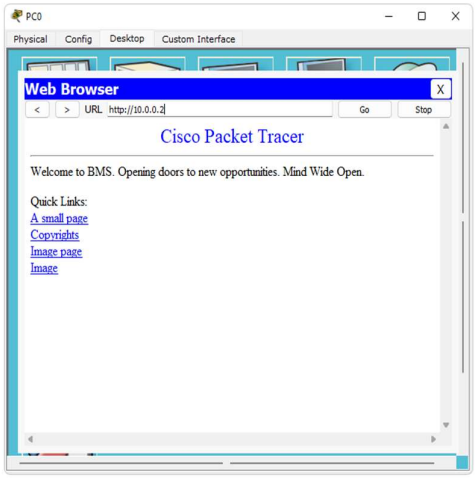
Topology:

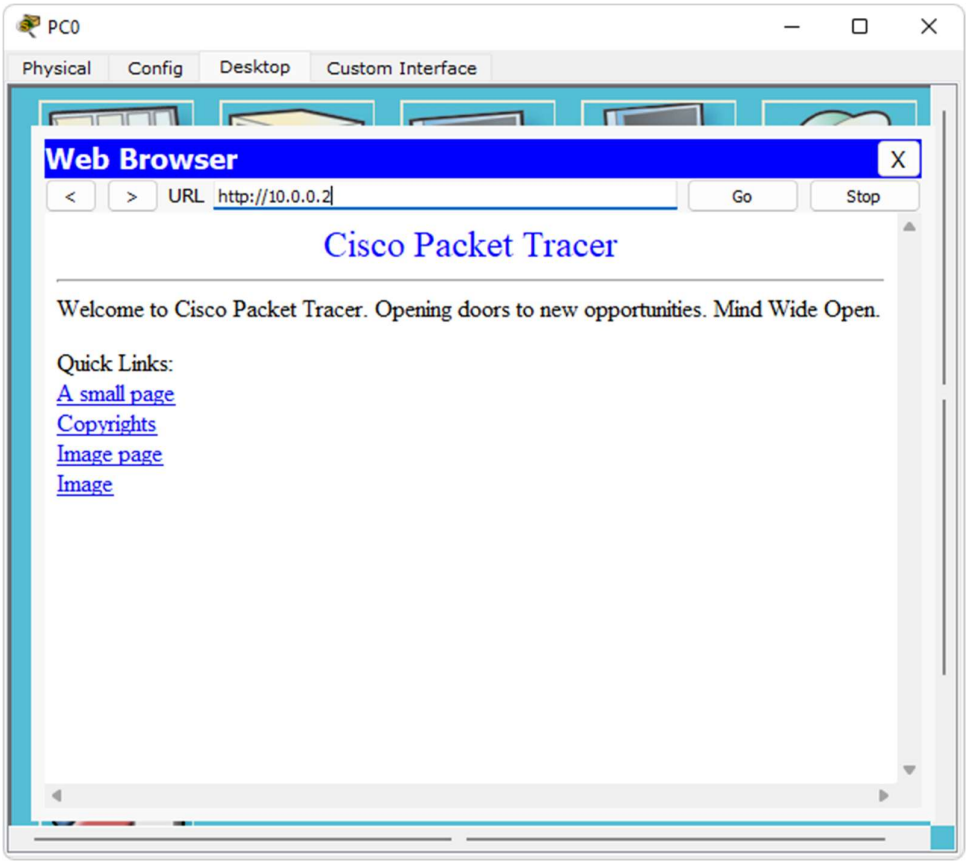


Output:




Fa0/1
Fa1/1
Fa0
PC-PT
PC0
Fa0
Server-PT
Server0

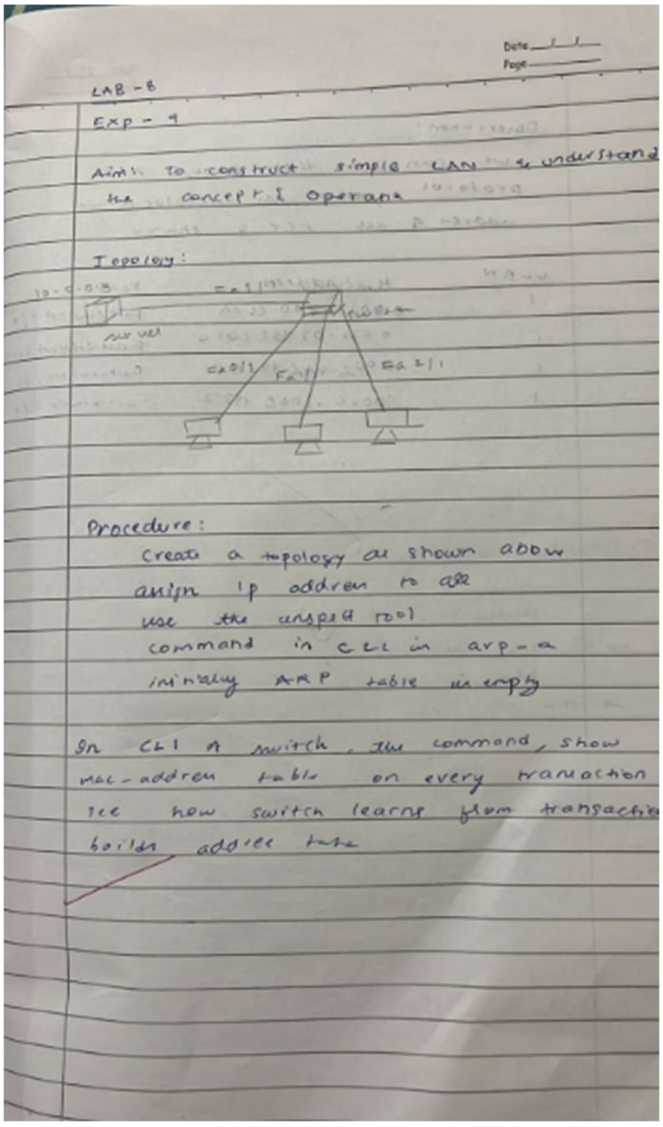

PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL http://10.0.0.2
Go
Stop
Cisco Packet Tracer
Welcome to BMS. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
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

Experiment 9:

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

Observation Book:



LAB - 8

EXP - 9

Aim: To construct simple LAN & understand the concept & Operation

Topology:

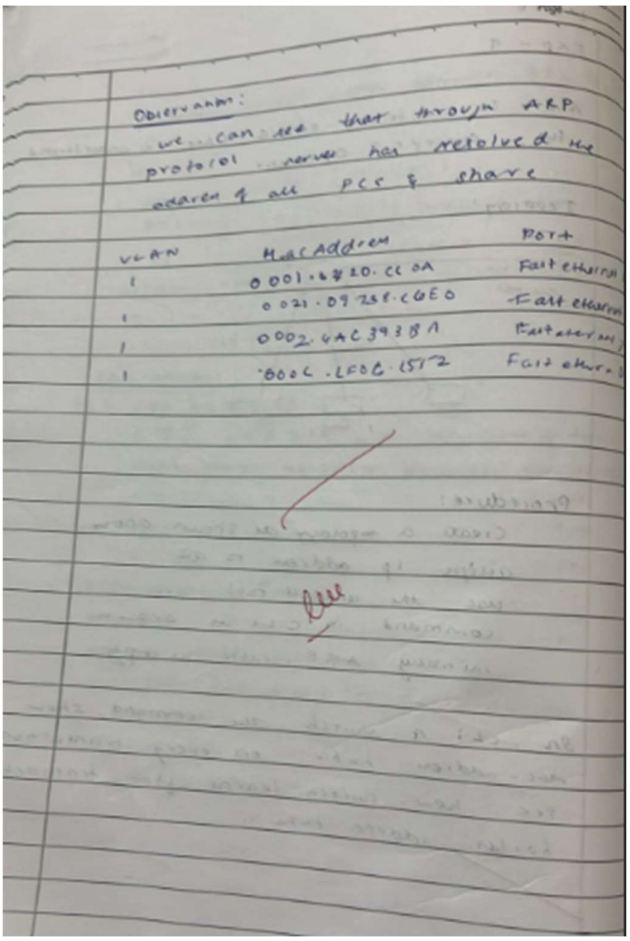
Procedure:

Create a topology as shown above

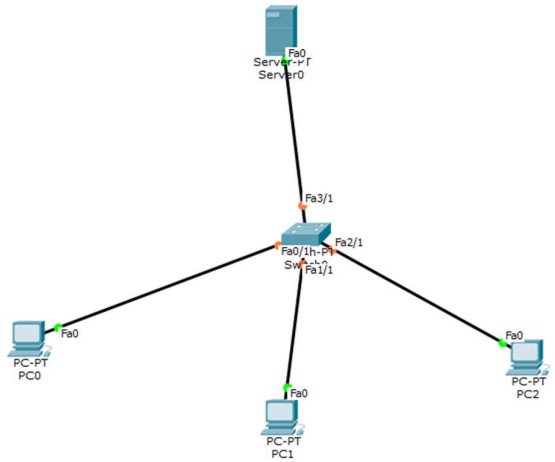
assign ip address to PCs

use the unspecified command in CLI is arp -a initially ARP table is empty

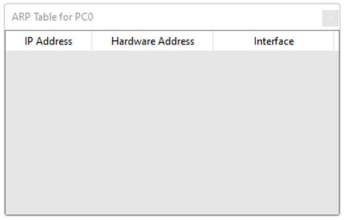
In CLI of switch, use command, show mac-address table on every transaction see how switch learns from transaction builds address table

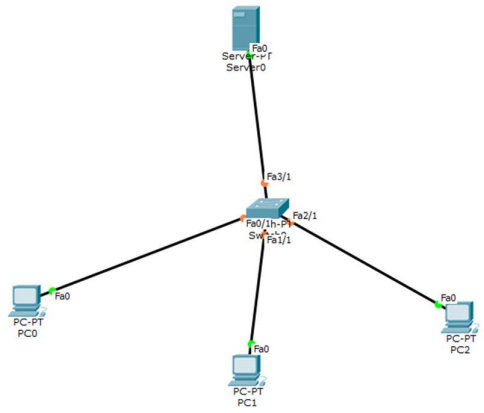


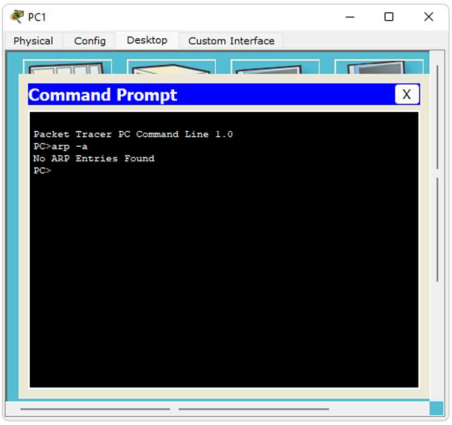
Topology:

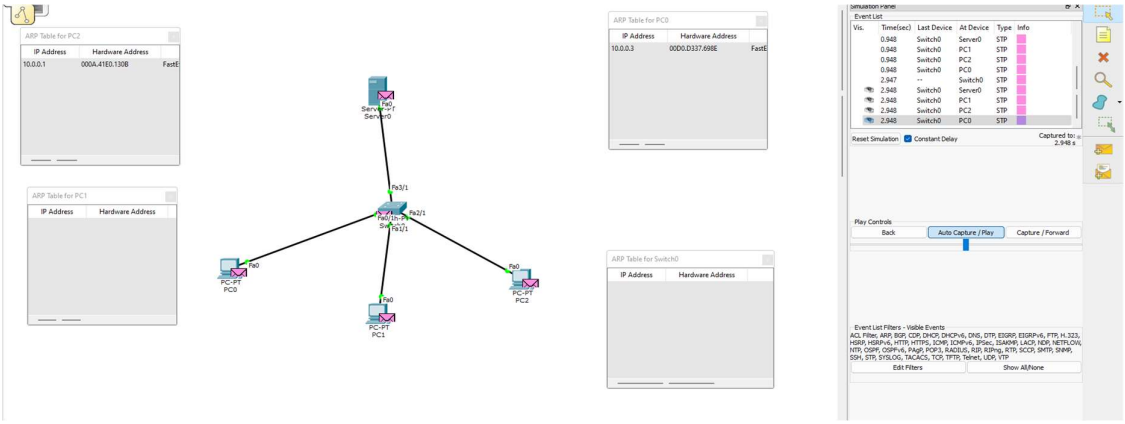


Output:


ARP Table for PC0
IP Address
Hardware Address
Interface


Server-PT
Server0
Fa0
Fa3/1
Fa0/1
Fa2/1
Fa1/1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2

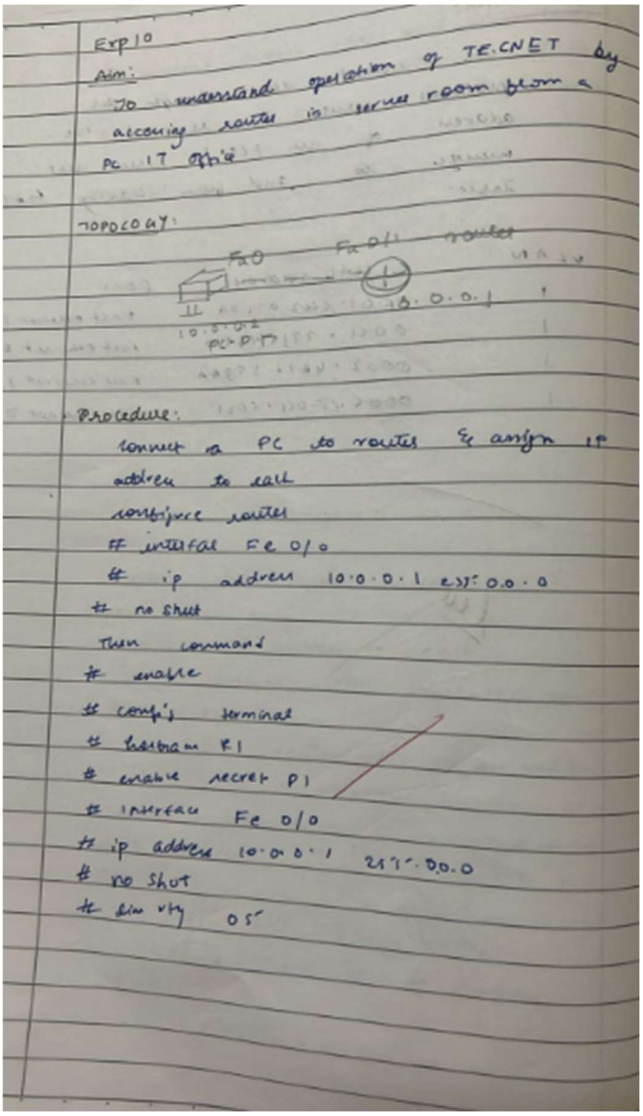

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>


ARP Table for PC2
ARP Table for PC1
ARP Table for PC0
ARP Table for Switch0
Server-PT
Server0
Fa3/1
Fa2/1
Fa1/1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None

Experiment 10:

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

Observation Book:



login
password p0
exit
exit
R1 # wr

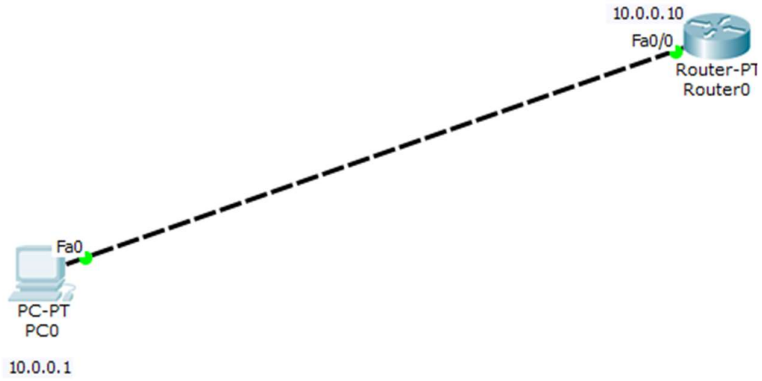
Observations:

It is observed that through telnet hostname & password is given to CLI router in any other device.
In PC type ping to know its conn
telnet 10.0.0.1 command access to
host in PC0.

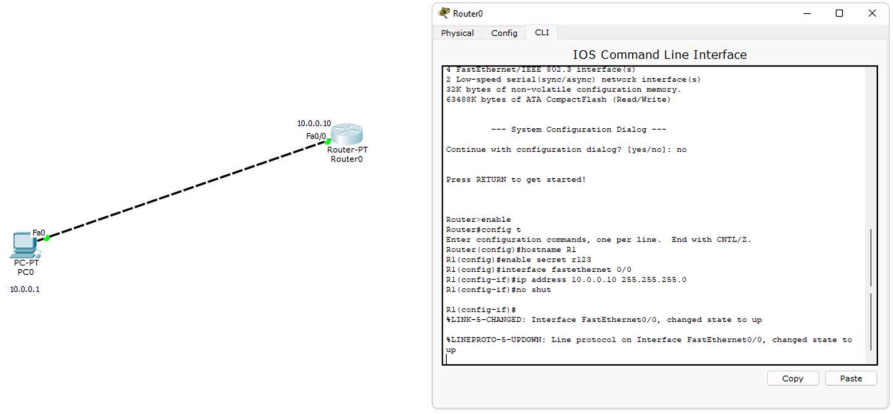
User access verification
password: P0
R1 > enable
password: P1
R1#

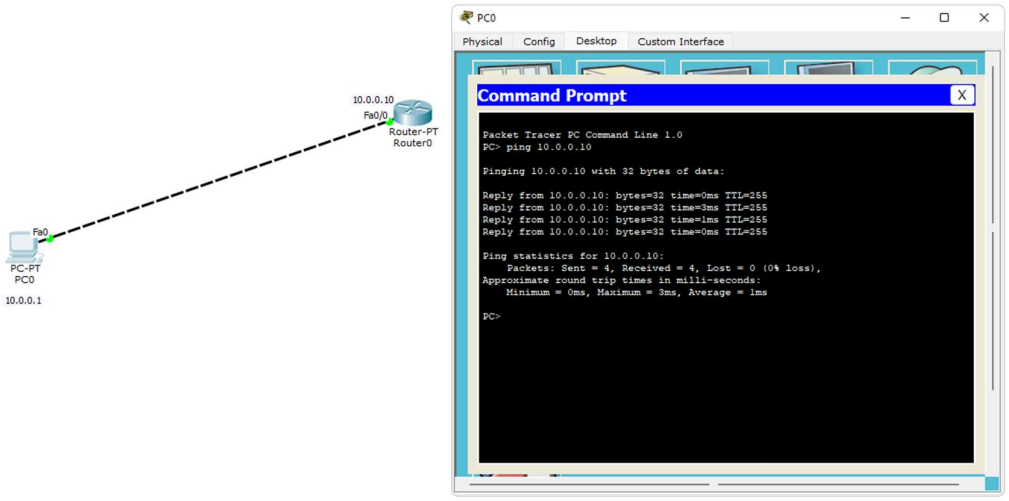
Lab

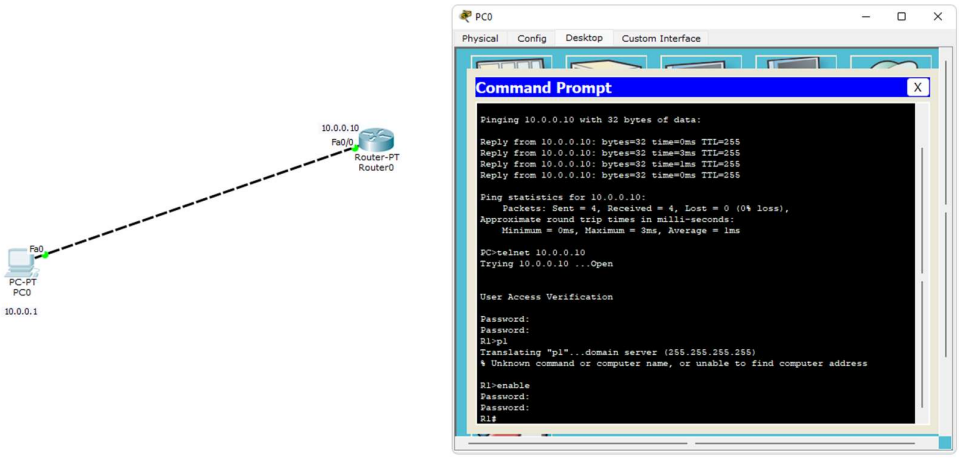
Topology:



Output:

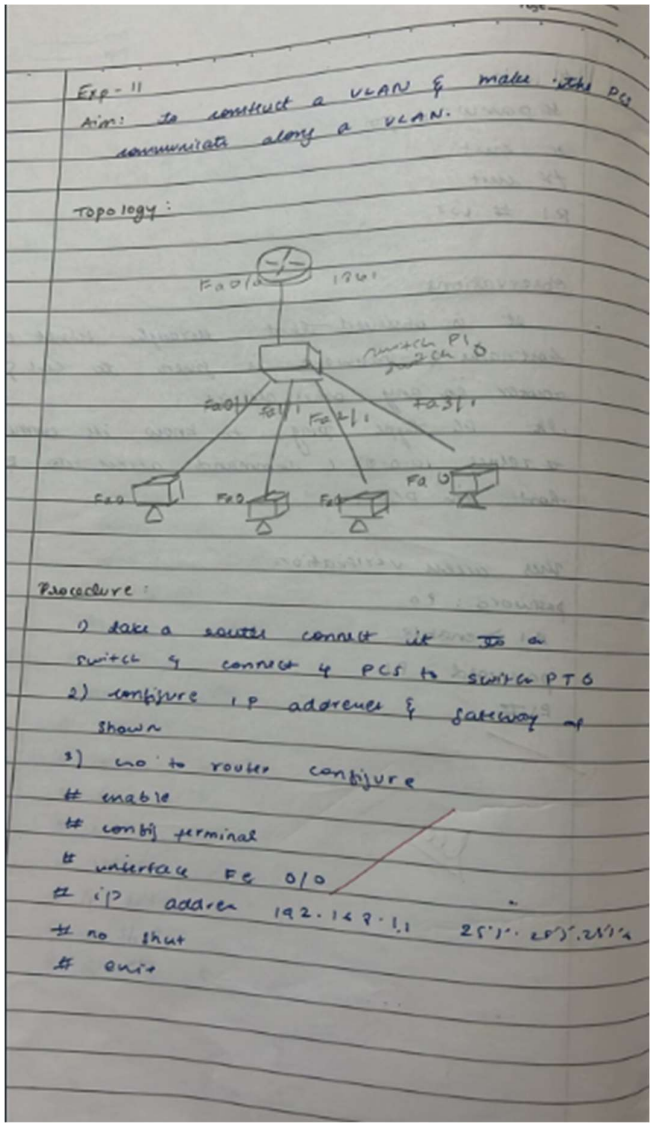


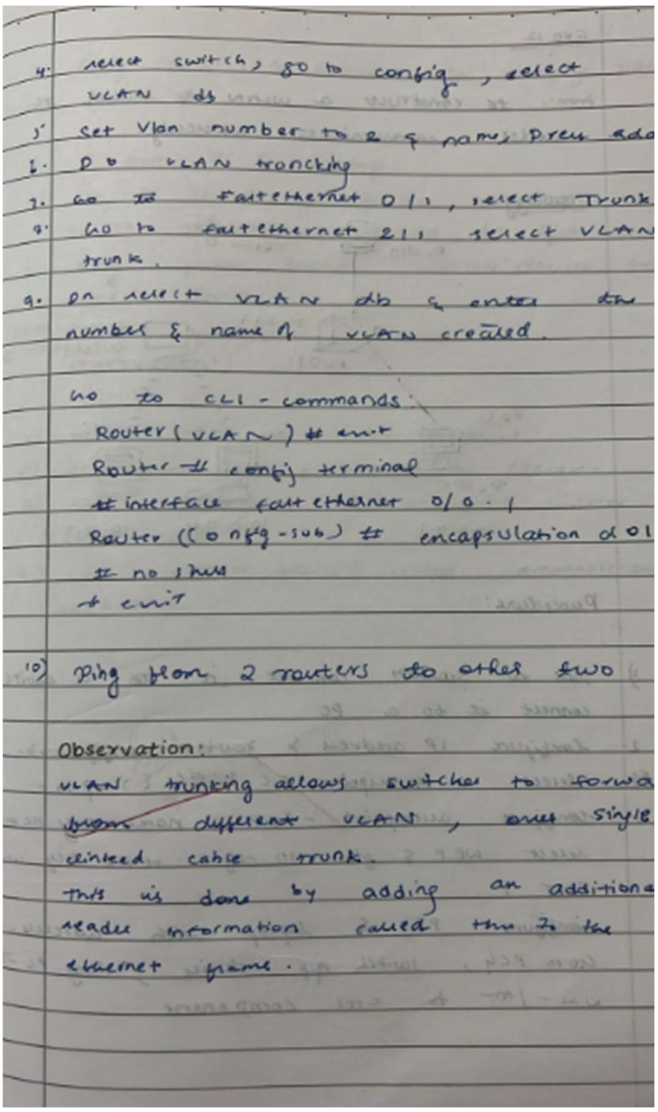

10.0.0.10
Fa0/0
Router-PT
Router0
Fa0
PC-PT
PC0
10.0.0.1
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC> ping 10.0.0.10
Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Reply from 10.0.0.10: bytes=32 time=3ms 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>


10.0.0.10
Fa0/0
Router-PT
Router0
Fa0
PC-PT
PC0
10.0.0.1
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#

Experiment 11:
To construct a VLAN and make the PC's communicate among a VLAN

Observation:





4. select switch, go to config, select VLAN db
5. Set Vlan number to 2 & name, Press add
6. Do VLAN troncking
7. Go to fastethernet 0/1, select Trunk
8. Go to fastethernet 2/1, select VLAN trunk.
9. On select VLAN db & enter the number & name of VLAN created.

Go to CLI - commands:

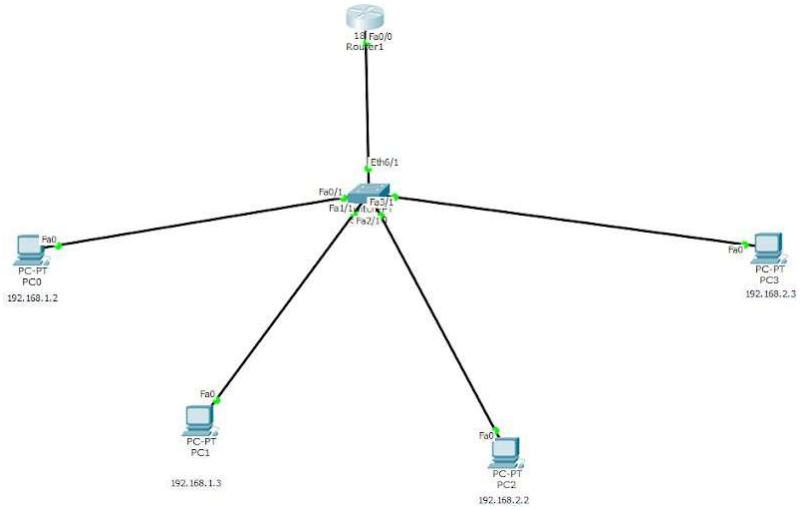
Router(VLAN)# exit
Router# config terminal
interface fastethernet 0/0.1
Router((config-sub)# encapsulation dot1
no shut
exit

10) Ping from 2 routers to other two

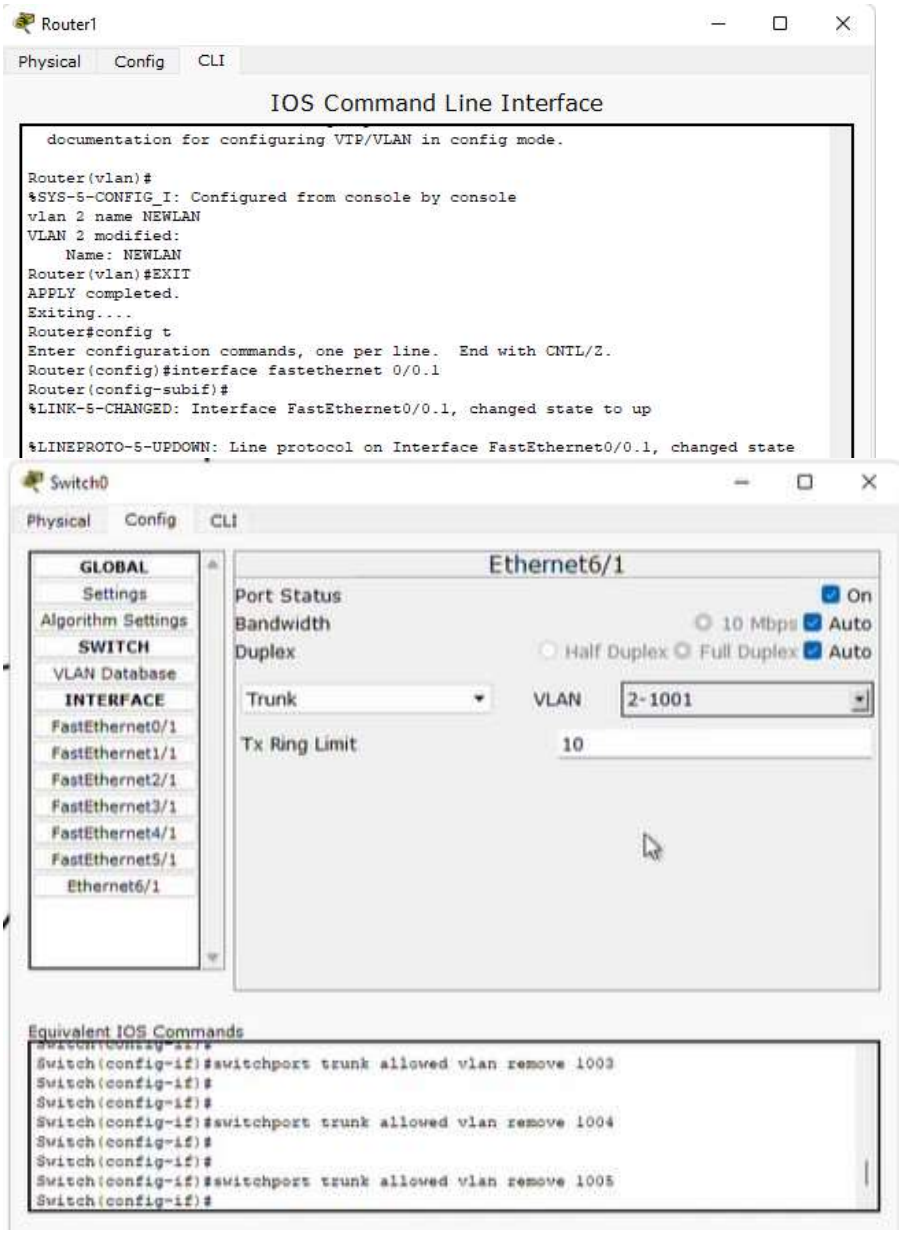
Observation:

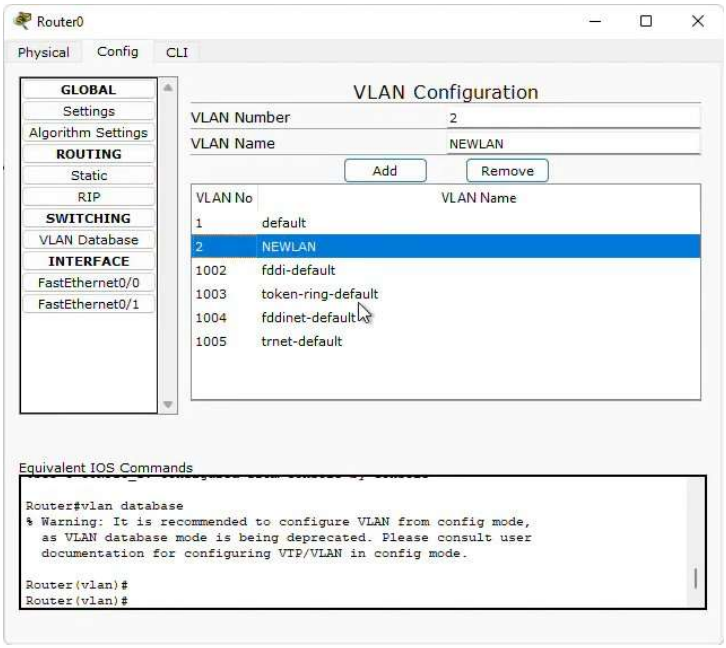
VLAN trunking allows switches to forward from different VLAN, over single united cable trunk.
This is done by adding an additional header information called the 2 to the ethernet frame.

Topology:

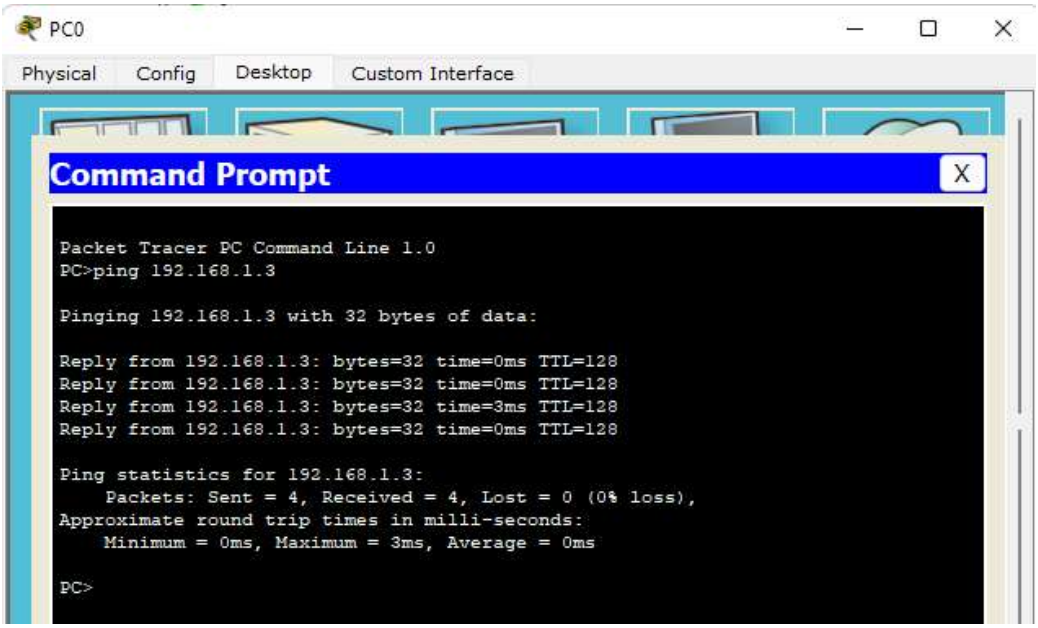

Router1
Fa0/0
Eth6/1
Fa0/1
Fa1/1
Fa2/1
Fa3/1
PC-PT
PC0
192.168.1.2
PC-PT
PC3
192.168.2.3
PC-PT
PC1
192.168.1.3
PC-PT
PC2
192.168.2.2

Output:


Router1
Physical
Config
CLI
IOS Command Line Interface
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
APPLY completed.
Exiting....
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet 0/0.1
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.1, changed state
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)#

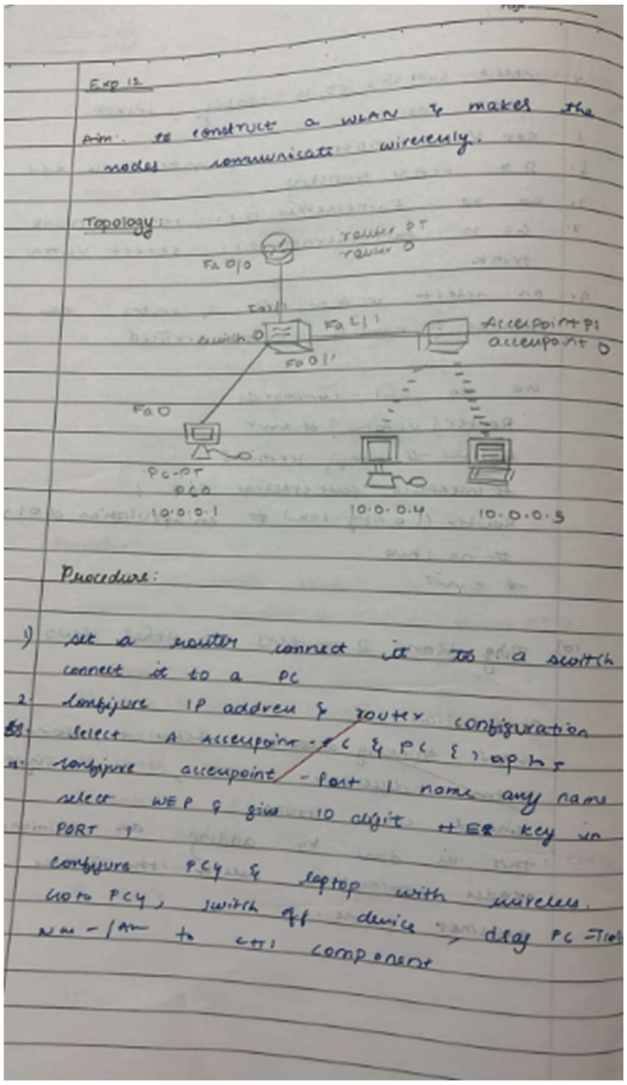


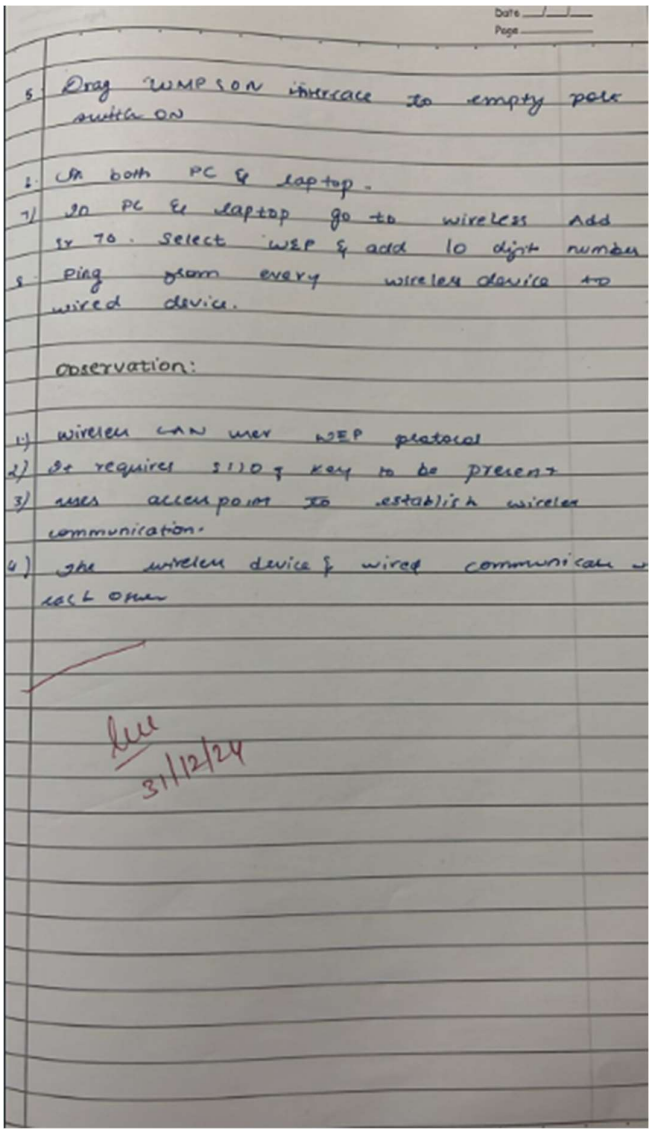

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to
up
exit
Router(config)#
Router(config)#exit
Router#vlan database
% Warning: It is recommended to configure VLAN from config mode,
  as VLAN database mode is being deprecated. Please consult user
  documentation for 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
```



Experiment 12:
To construct a WLAN and make the nodes communicate wirelessly

Observation Book:




Date __/__/__
Page _____

5. Drag JUMPSON interface to empty port switch ON

6. On both PC & laptop.

7) In PC & laptop go to wireless Add it to. Select WEP & add 10 digit number

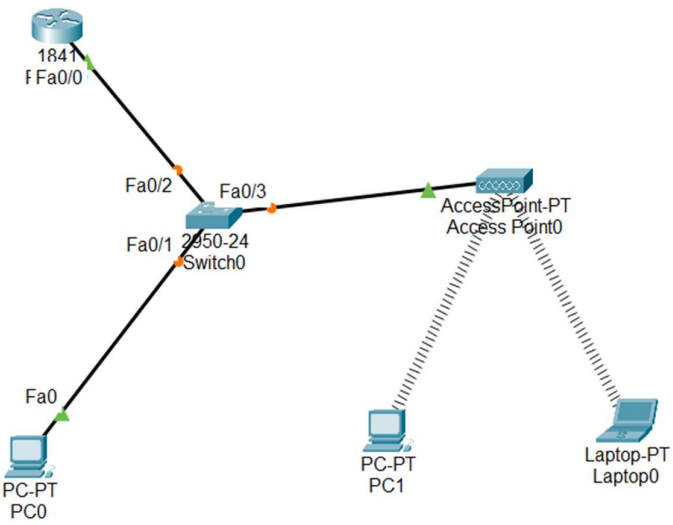
8. Ping from every wireless device to wired device.

Observation:

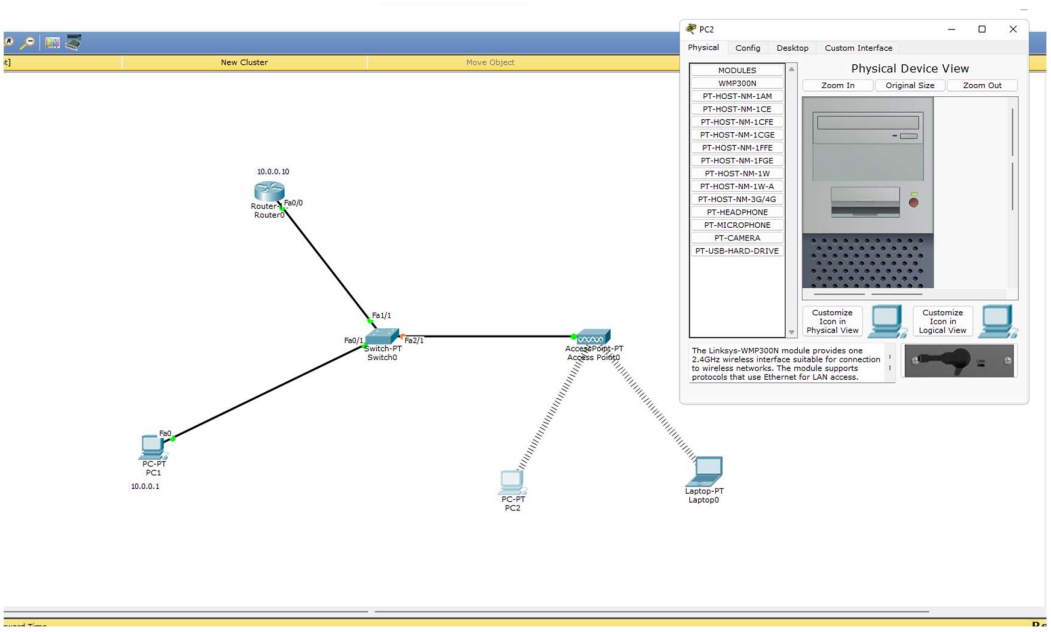
1) Wireless LAN uses WEP protocol
2) It requires SSID & key to be present
3) uses access point to establish wireless communication.
4) the wireless device & wired communicate with each other

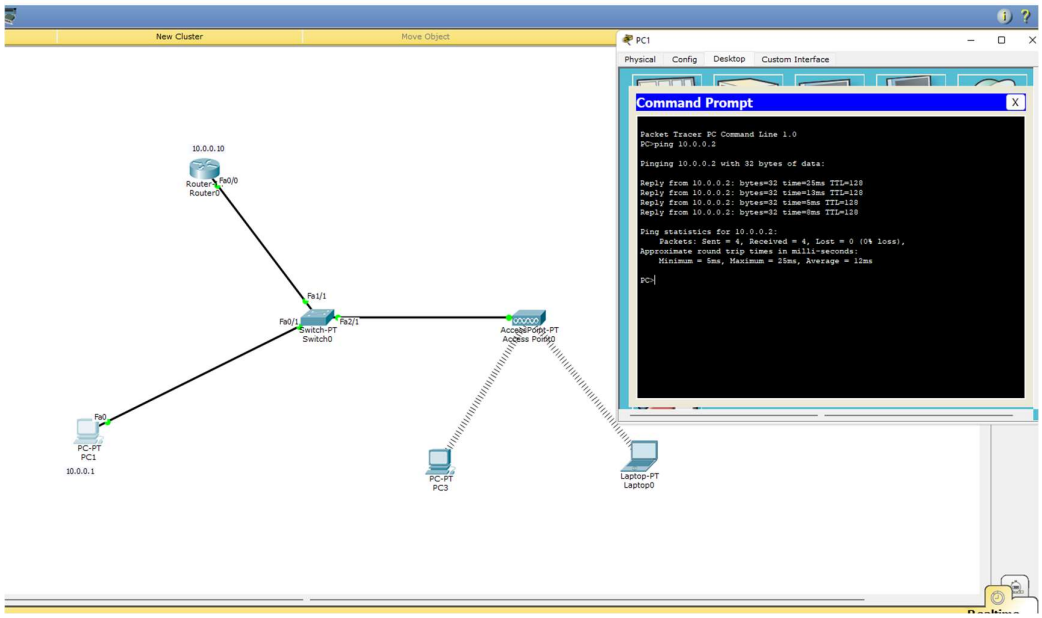
31/12/24

Topology:


1841
Fa0/0
Fa0/2
Fa0/3
Fa0/1
2950-24
Switch0
AccessPoint-PT
Access Point0
Fa0
PC-PT
PC0
PC-PT
PC1
Laptop-PT
Laptop0

Output:


New Cluster
Move Object
PC2
Physical
Config
Desktop
Custom Interface
Physical Device View
Zoom In
Original Size
Zoom Out
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
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.
10.0.0.10
Router0
Fa0/0
Fa1/1
Fa0/1
Switch-PT
Switch0
Fa2/1
Fa0
PC-PT
PC1
10.0.0.1
PC-PT
PC2
Laptop-PT
Laptop0

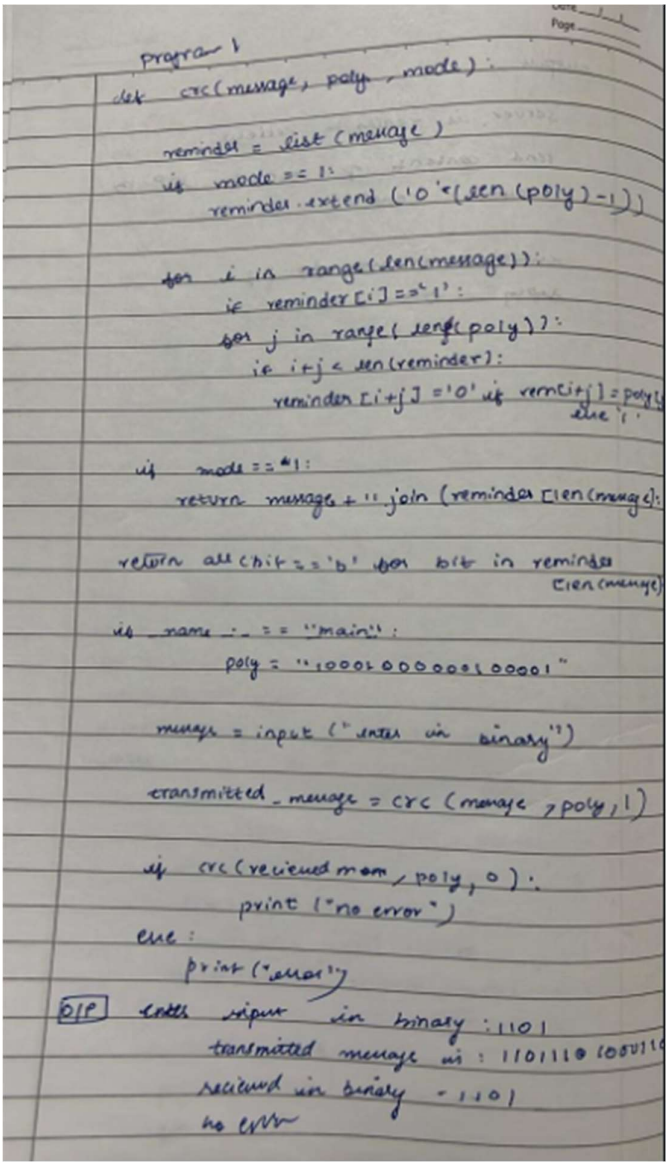

New Cluster
Move Object
10.0.0.10
Router-PT
Router0
Fa0/0
Fa1/1
Fa0/1
Switch-PT
Switch0
Fa2/1
Access Point-PT
Access Point0
Fa0
PC-PT
PC1
10.0.0.1
PC-PT
PC3
Laptop-PT
Laptop0
PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 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>

Cycle-2

Program 1:

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

Observation:



Code:

```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    """
    Calculate the 16-bit CRC-CCITT checksum for a given binary string.
    """
    crc = init_crc
```

```
    for bit in bitstream:
        crc ^= int(bit) << 15  # Align the bit with CRC's uppermost bit
        for _ in range(8):  # Process each bit
            if crc & 0x8000:  # Check if the leftmost bit is set
                crc = (crc << 1) ^ poly
            else:
                crc <<= 1
            crc &= 0xFFFF  # Ensure CRC remains 16-bit
    return crc


def append_crc_to_bitstream(bitstream: str) -> str:
    """
    Append the calculated 16-bit CRC to the given bitstream.
    """
    crc = crc_ccitt_16_bitstream(bitstream)
    crc_bits = f"{crc:016b}"  # Convert CRC to a 16-bit binary string
    return bitstream + crc_bits


def verify_crc_bitstream(bitstream_with_crc: str) -> bool:
    """
    Verify the CRC of the given bitstream with CRC appended.
    """
    if len(bitstream_with_crc) < 16:
        return False  # Not enough bits to contain CRC
    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)


# Main Program
if __name__ == "__main__":
    # User input for original bitstream
    message_bits = input("Enter the original bitstream (e.g., 11010011101100):
").strip()

    # Validate input
    if not all(bit in "01" for bit in message_bits):
        print("Invalid input. Please enter a binary bitstream (e.g.,
11010011101100).")
    else:
        # Calculate and append CRC
        bitstream_with_crc = append_crc_to_bitstream(message_bits)
        print(f"Transmitted bitstream with CRC: {bitstream_with_crc}")

        # User input for received bitstream
        user_bitstream = input("Enter the received bitstream for verification:
").strip()

        # Validate received input
        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:
            # Verify CRC
            is_valid = verify_crc_bitstream(user_bitstream)
            if is_valid:
                print("No errors detected. CRC valid.")
            else:
                print("Error detected! CRC invalid.")
```

Output:

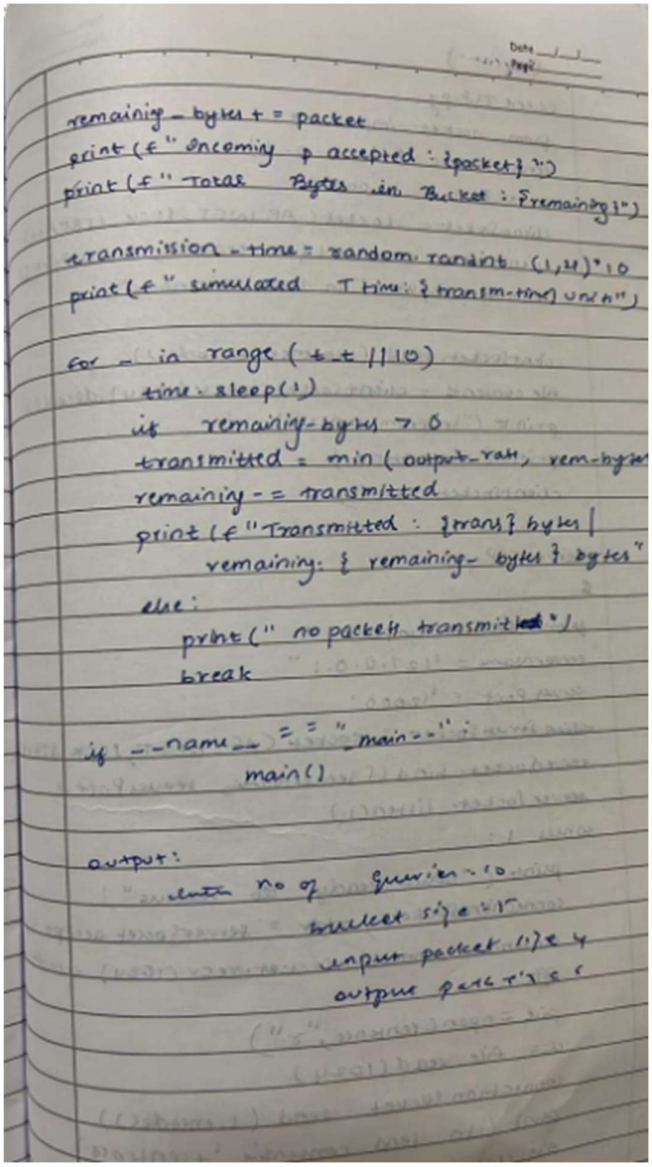
Program 2:

Write a program for congestion control using Leaky bucket algorithm.

Observation:

Program 2

```
def import time
    import random

    NOF_packets = 10

def generate_random_packet_size( max_size ):
    return random.randint (1, max_size //10 ) * 10

def main( ):
    packet_sizes =[gen_random_packet_size (50) for
                _ in range (NOF_PACKETS)]
    print("generated packet sizes:")
    for i, size in enumerate (packet_sizes):
        print (f "Packet[{i}]: {size} bytes")

output_rate = int (input ("\n Enter Output rate:"))
bucket_size = int (input ("Enter size"))
remaining_bytes = 0

for i, packet in enumerate (packet_sizes):
    print (f "\n Processing Packet [{i}] of size
        {packet} bytes")
    if packet > bucket_size:
    print (f" packet size ({packet} bytes)
exceeds capacity {b_size} - Packet rejected")
    continue

    if ( remaining_bytes + packet ) > b_size:
        print("bucket_size exceeded - PACKET-REJ")
        continue
```



Code:

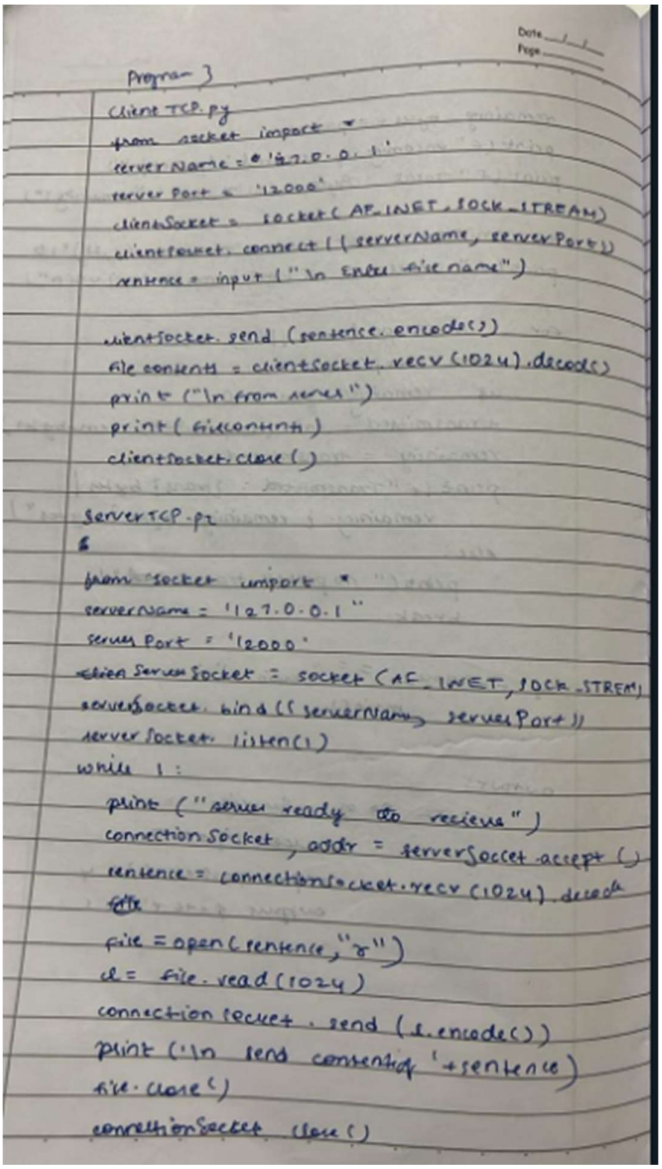
```
storage=0
noofqueries=int(input("Enter no of queries:"))
bucketsize=int(input("Enter bucket size:"))
inputpktsize=int(input("Enter input packet size:"))
outputpktsize=int(input("Enter output packet size:"))
for i in range(0,noofqueries):
    sizeleft=bucketsize-storage
    if inputpktsize<=sizeleft: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

Output:

Program 3:

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

Observation:



Servertcp.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)
    file.close()
    connectionSocket.close()
```

Clienttcp.py

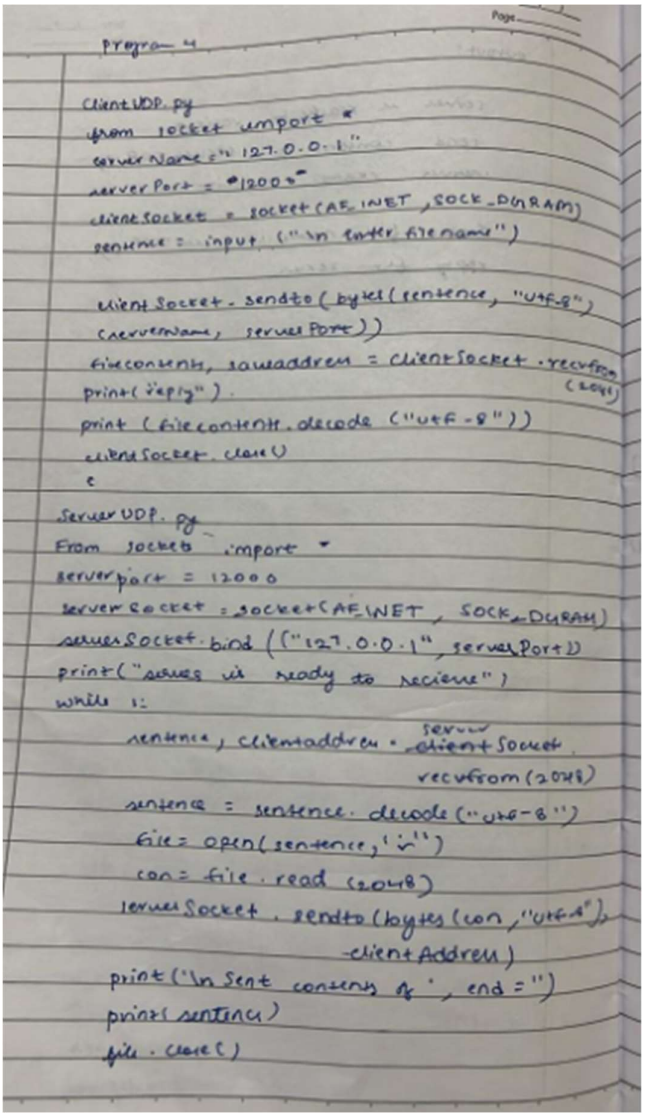
```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

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

Output:

Program 4:

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

Observation:



Prgm - 4

ClientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter filename")

clientSocket.sendto(bytes(sentence, "utf-8"),
(serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom
                                        (2048)
print("reply")
print (fileContents.decode("utf-8"))
clientSocket.close()
```

ServerUDP.py

```
From socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("server is ready to recieve")
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)
    file.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 = ' ')
    print (sentence)
   #  for i in sentence:
       #  print (str(i), end = '')
    file.close()
```

Clienttudp.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()
```

Output:

Program 5:

Tool Exploration –Wireshark

Exp-12

Date __/__/__
Page _____

Wireshark

key features:

1. packet capture
2. protocol analysis
3. following steps
4. visualization

use cases

1. Network trouble shooting
   diagnosing slow network issues
   & other misconfiguration
2. Security analysis:
   detecting malicious traffic & intentions
3. Protocol study
   understanding packet structures & commercial flows

O/P:

1. http : show only http traffic
2. tcp : port : show traffic on TCP port 8
3. ip : 192.168.11.1 : show traffic to & fr
4. udp : show only UDP traffic