

B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



Lab Record

Computer Networks – 23CS5PCCON

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

AARYAN PRAKASH

(1BM23CS006)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
August 2025-December 2025

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Computer Networks (23CS5PCCON) laboratory has been carried out by Aaryan Prakash (1BM23CS006) during the 5th Semester August 2025-December 2025.

Signature of Faculty In charge:

Praveen N
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

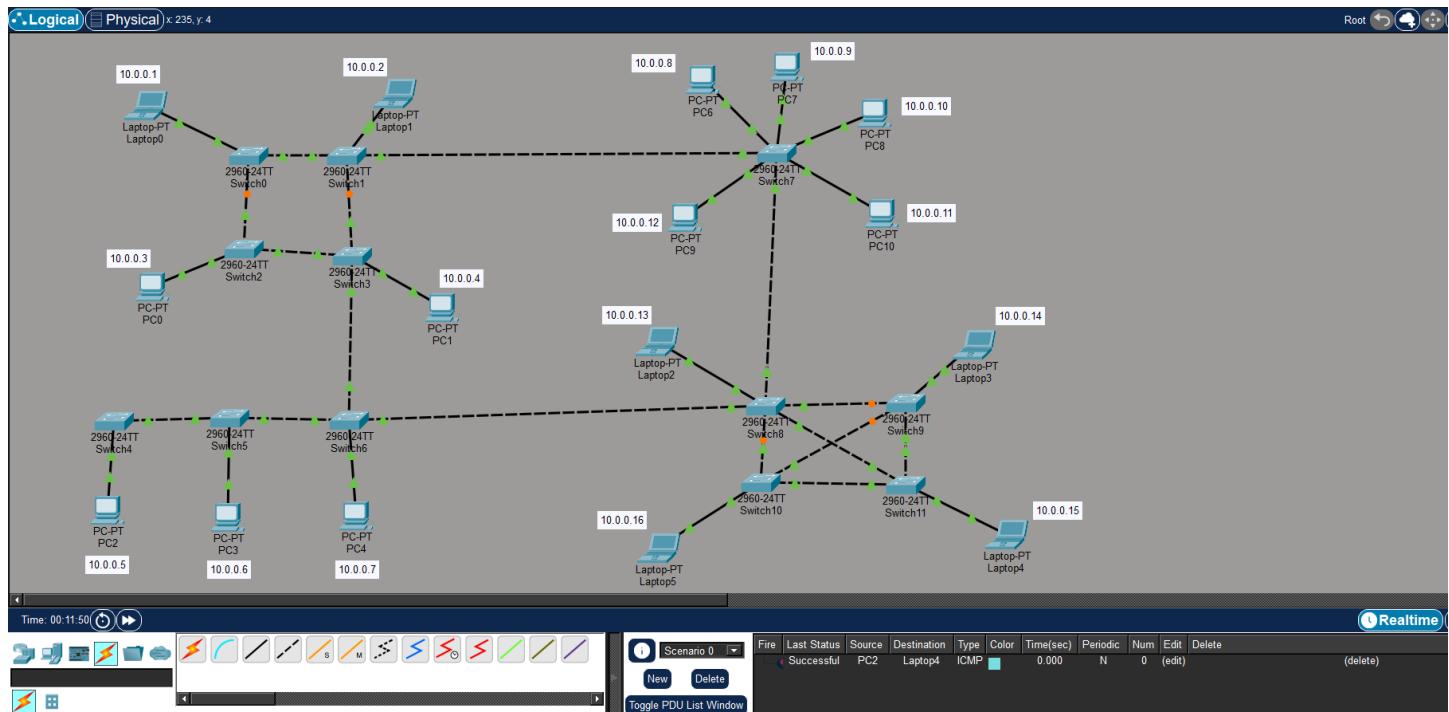
PART - A		
Sl. No.	Name of Experiment	Page No.
1.	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.	4
2.	Configure DHCP within a LAN and outside LAN.	6
3.	Configure Web Server, DNS within a LAN.	11
4.	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.	14
5.	Configure default route, static route to the Router.	16
6.	Configure RIP routing Protocol in Routers.	18
7.	Configure OSPF routing protocol.	20
8.	To construct a VLAN and make the PCs communicate among a VLAN.	23
9.	To construct a WLAN and make the nodes communicate wirelessly.	25
10.	Demonstrate the TTL/ Life of a Packet.	27
11.	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	28
12.	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).	30

PART – B		
Sl. No.	Name of Experiment	Page No.
1.	Write a program for congestion control using Leaky bucket algorithm.	32
2.	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.	35
3.	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.	37
4.	Write a program for error detecting code using CRC-CCITT (16-bits).	39

PART - A

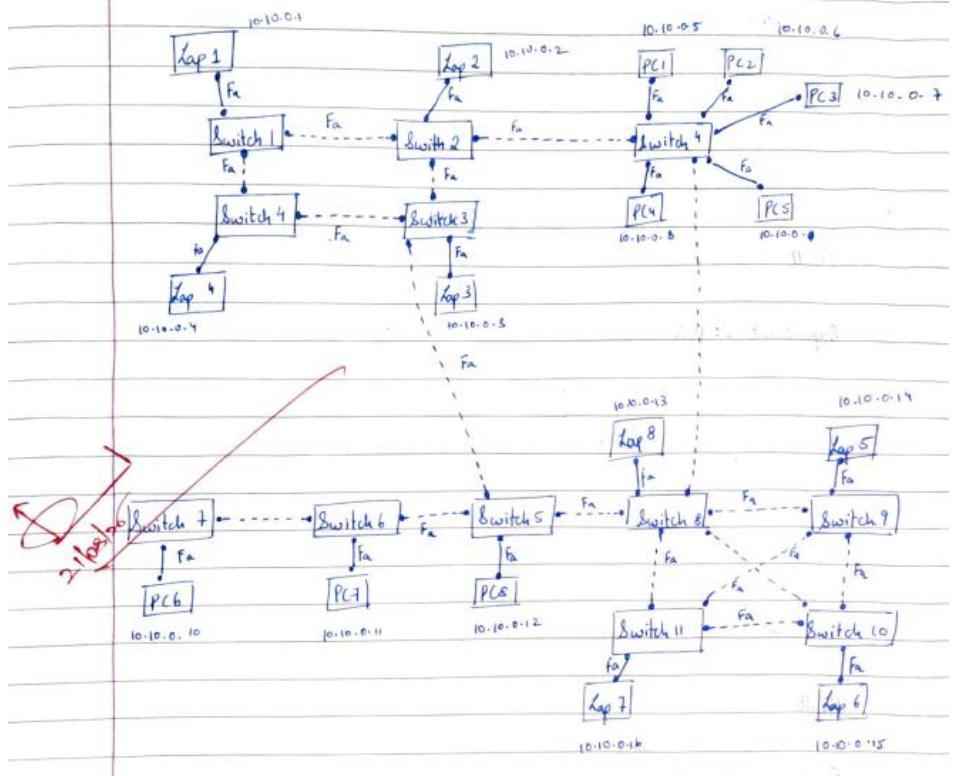
Program 1: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

Network diagram:



Configuration:

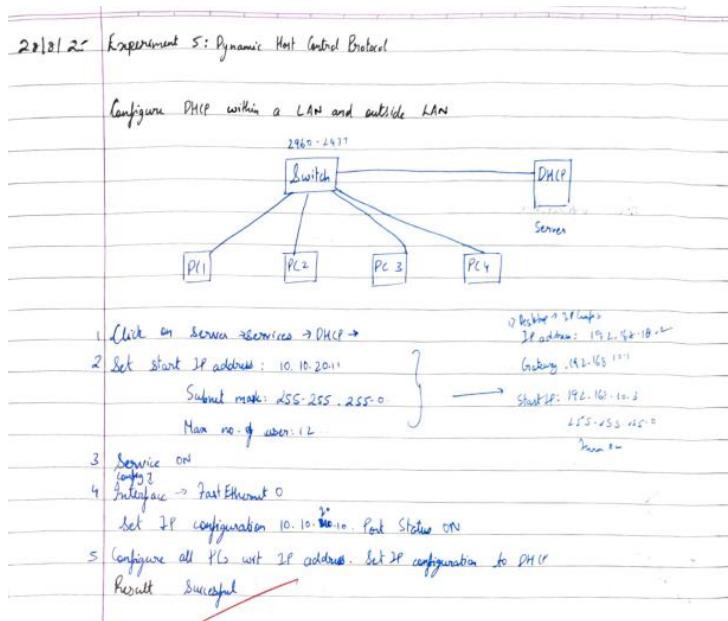
Experiment 4



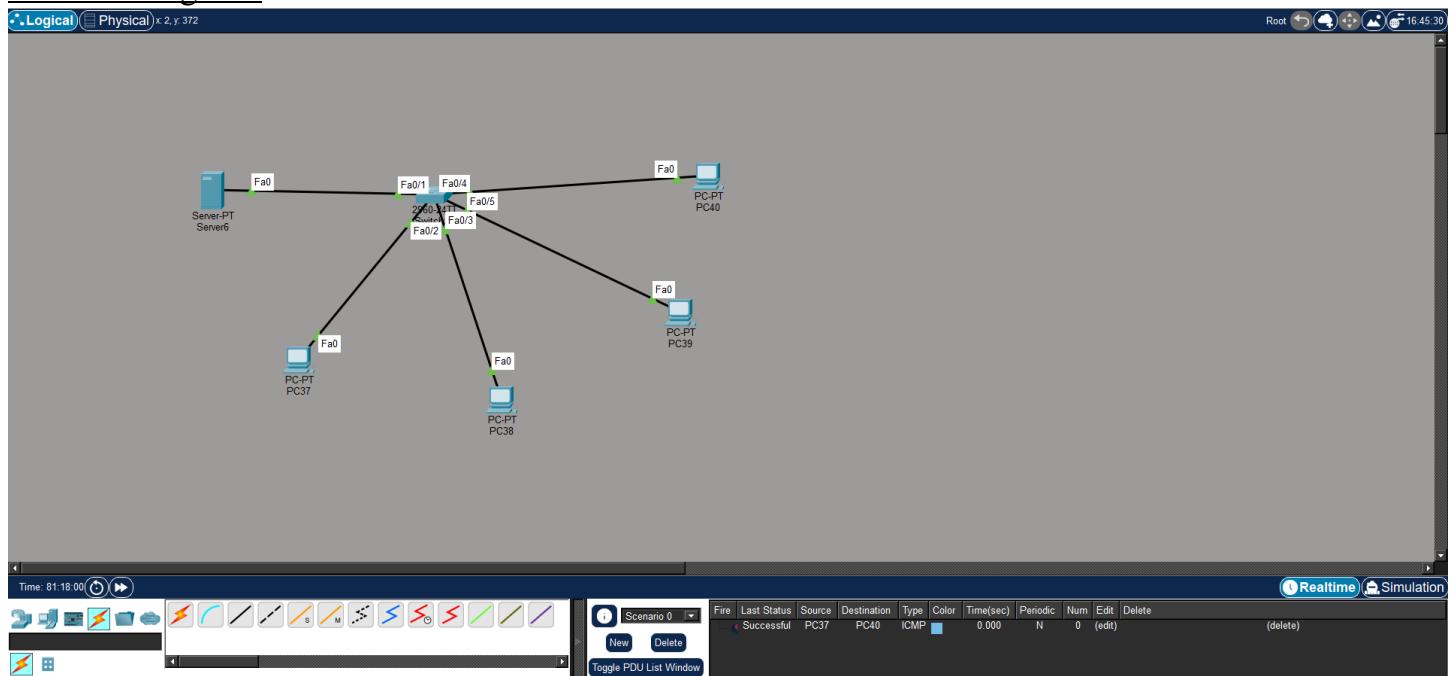
- Place all end devices and switches
- Connect using generic cables
- Set IP addresses
- Send message from Laptop 1 to PC 3
- Result successful

Program 2: Configure DHCP within a LAN and outside LAN.

(i) Inside LAN



Network diagram:



Configuration:

Server6

Physical Config Services **Desktop** Programming Attributes

IP Configuration

IP Configuration

DHCP Static

IPv4 Address	192.168.10.2
Subnet Mask	255.255.255.0
Default Gateway	192.168.10.1
DNS Server	0.0.0.0

Server6

Physical Config **Services** Desktop Programming Attributes

SERVICES

HTTP	FastEthernet0	serverPool		
DHCP		0.0.0		
DHCIPv6		0.0.0		
TFTP				
DNS	192	168	10	0
SYSLOG	255	255	255	0
AAA		512		
NTP		0.0.0		
EMAIL		0.0.0		
FTP				
IoT				
VM Management				
Radius EAP				
PRP	Add	Save	Remove	

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
switch	192.168.10.1	0.0.0	192.168.10.3	255.255.255.0	12	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0	192.168.10.0	255.255.255.0	512	0.0.0.0	0.0.0.0

PC37

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

DHCP Static

IPv4 Address	192.168.10.1
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
DNS Server	0.0.0.0

PC38

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

<input checked="" type="radio"/> DHCP	<input checked="" type="radio"/> Static
IPv4 Address	192.168.10.3
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
DNS Server	0.0.0.0

PC39

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

<input checked="" type="radio"/> DHCP	<input checked="" type="radio"/> Static
IPv4 Address	192.168.10.4
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
DNS Server	0.0.0.0

PC40

Physical Config Desktop Programming Attributes

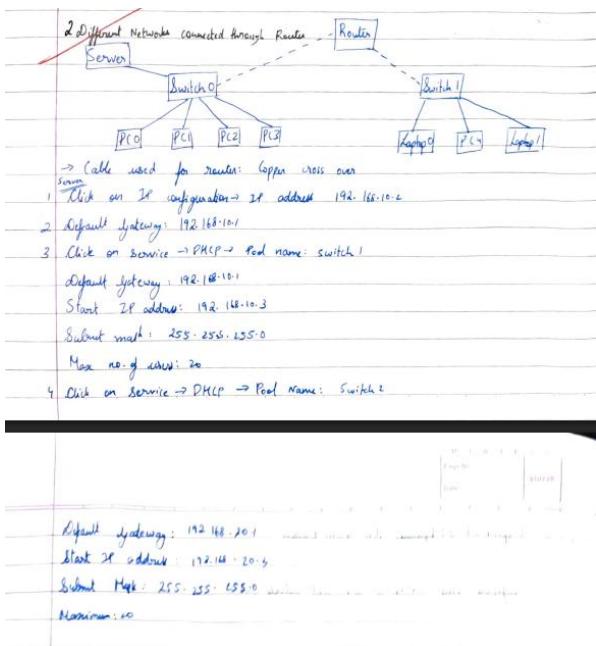
IP Configuration

Interface FastEthernet0

IP Configuration

<input checked="" type="radio"/> DHCP	<input checked="" type="radio"/> Static
IPv4 Address	192.168.10.5
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
DNS Server	0.0.0.0

(ii) Outside LAN



Router Configuration:

- Click L2I → Continue with configuration? no
- Enter return
- Router > Enable
- Router # conf t
- Router (config)# int fa 0/0
- Router (config-if) # ip address 192.168.10.1 255.255.255.0 (Gateway Address)
- Router (config-if) # ip default-router 192.168.10.1 (Default Address Selection)
- Router (config-if) # no shutdown

do write memory

Router (config) # exit

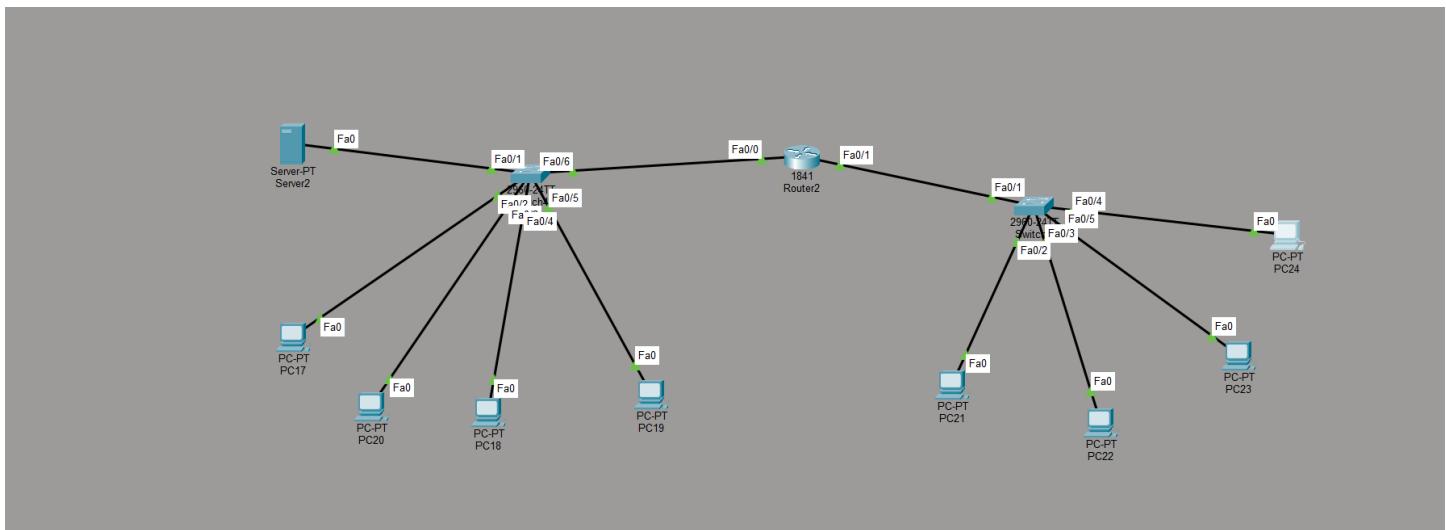
Router (config) # exit

H. R. shutdown

do write memory

4 Configure IP Laptop 1 → DHCP successful
 To Router → IP config = 1.1.1.1 for all

Network Diagram:



Configuration:

The top window shows the 'IP Configuration' tab with the following settings:

DHCP	Static
IPv4 Address	192.168.10.2
Subnet Mask	255.255.255.0
Default Gateway	192.168.10.1
DNS Server	0.0.0.0

The bottom window shows the 'Services' tab for 'FastEthernet0' with a 'serverPool' entry. The pool table lists the following entries:

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
s2	192.168.20.1	0.0.0.0	192.168.20.3	255.255.255.0	20	0.0.0.0	0.0.0.0
s1	192.168.10.1	0.0.0.0	192.168.10.3	255.255.255.0	20	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	192.168.10.0	255.255.255.0	512	0.0.0.0	0.0.0.0

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#ip add 192.168.10.1 255.255.255.0
Router(config-if)#ip helper-address 192.168.10.2
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

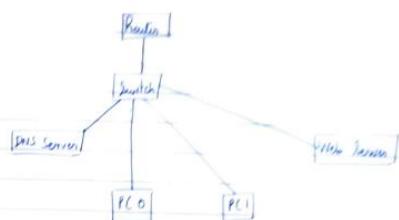
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
exit
Router(config)#int fa0/1
Router(config-if)#ip add 192.168.20.1 255.255.255.0
Router(config-if)#ip helper-address 192.168.10.2
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
exit
Router(config)#do wr
Building configuration...
[OK]
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Program 3: Configure Web Server, DNS within a LAN.

Configure web server and DNS within a LAN



- 1 PLO \rightarrow IP Address: 192.168.1.100 (demon IP address)
Subnet mask: 255.255.255.0
Default gateway: 192.168.1.1

Sabot 149th. 255 255. 255. 0

Default Gateway: 192.168.1.1 (Laptop 2P address)

PME Survey: 172.18-15 (Survey 18 - 11-11)

- 2 PCI → IP Address: 192.168.1.103

Plant material: 192-168-1-1

Page Server: 192.168.1.5

- ~~3 Web Server → Service → HTTP → Down or HTTPS, H2S~~

- Click on `hello-world.html` → edit text → add your own text

- Desktop → 78 ports → 78 address: 192.168.1.6

Oct. 11, 1955 - 182 160-1

repeat survey: 112.10

- 4 DNS Server → Service & DNS → Name: "www.letslearn.com" → Type: A Record → Address: 192.168.1.6 (web server IP)

→ IP Configuration : IP address : 192.168.1.3

Default settings 1192-168-1

Page Sixty-Three

100.000 - 1.000.000

100 m³/h

- 5 P(D \Rightarrow forward) = 1 - p(D \Rightarrow forward) = 1 - 0.2145 = 0.7855

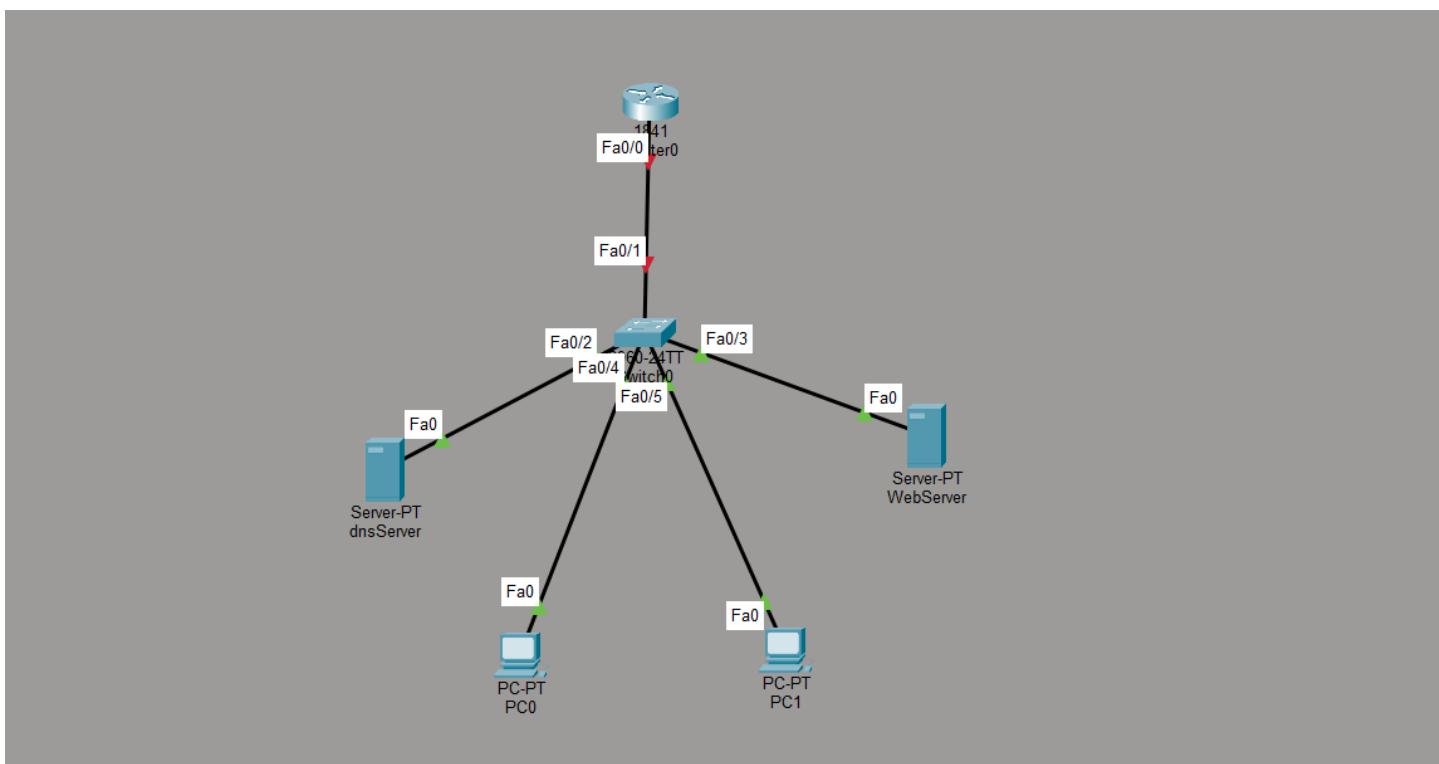
? Check if it is valid

⇒ Web Browser ⇒ http://www.abc.com

→ a link a href="http://www.1

→ next page should open
But it didn't

Network diagram:



Configuration:

WebServer

Physical Config Services Desktop **Programming Attributes**

IP Configuration

Interface FastEthernet0

IP Configuration

DHCP Static

IPv4 Address	192.168.1.6
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.1
DNS Server	192.168.1.5

dnsServer

Physical Config Services Desktop **Programming Attributes**

IP Configuration

IP Configuration

DHCP Static

IPv4 Address	192.168.1.5
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.1
DNS Server	192.168.1.5

WebServer

Physical Config **Services Desktop Programming Attributes**

SERVICES

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP
- PRP

File Name Edit Delete

1 copyrights.html	(edit)	(delete)
2 cscoptlogo177x111.jpg		(delete)
3 hellohi.html	(edit)	(delete)
4 helloworld.html	(edit)	(delete)
5 hihi.html	(edit)	(delete)
6 image.html	(edit)	(delete)
7 index.html	(edit)	(delete)

dnsServer

Physical Config Services Desktop Programming Attributes

SERVICES

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

www.letslearn.com A Record

192.168.1.6

Add Save Remove

No.	Name	Type	Detail
0	www.letslearn.com	A Record	192.168.1.6

PC0

Physical Config Desktop Programming Attributes

Web Browser

< > URL http://www.letslearn.com

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](#)

Program 4: Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

i. Configure IP address to router in packet tracer. Explore the following message:

- Ping Message
- Destination Unreachable
- Request Timeout
- Reply

i. ping 192.168.1.5 // another PC IP
pinging 192.168.1.5 with 32 bytes of data:
Reply from 192.168.1.5: bytes=32 time=1ms

ii. ping 192.168.1.1 // router IP
pinging 192.168.1.1 with 32 bytes
Request time out

iv. ping 192.168.1.5 // PC IP
pinging 192.168.1.5 with 32 bytes of data:
Reply from 192.168.1.5: bytes=22 time=6ms TTL=128
Reply from 192.168.1.5: bytes=32 time=0ms TTL=128

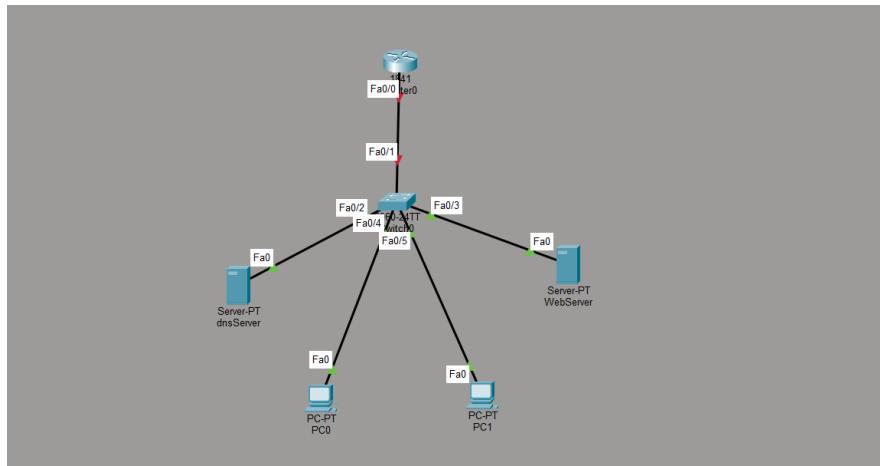
(calculated)

Network Diagram:

```

graph TD
    Router[Router (192.168.1.1)] --- Fa0_0[Fa0/0]
    Router --- Fa0_1[Fa0/1]
    Switch[Switch] --- Fa0_2[Fa0/2]
    Switch --- Fa0_4[Fa0/4]
    Switch --- Fa0_5[Fa0/5]
    DNS_Server[DNS Server] --- Fa0_4
    Web_Server[Web Server] --- Fa0_5
    PC0[PC-PT PC0] --- Fa0_2
    PC1[PC-PT PC1] --- Fa0_5
  
```

Network diagram:



Configuration:

```
C:\>ping 192.168.1.5          C:\>ping 192.168.1.5
Pinging 192.168.1.5 with 32 bytes of data:      Pinging 192.168.1.5 with 32 bytes of data:
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128  Request timed out.
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128  Request timed out.
Reply from 192.168.1.5: bytes=32 time=1ms TTL=128  Request timed out.
Reply from 192.168.1.5: bytes=32 time=1ms TTL=128  Request timed out.

Ping statistics for 192.168.1.5:                  Ping statistics for 192.168.1.5:
  Packets: Sent = 4, Received = 4 (0% loss),
  Approximate round trip times in milli-seconds:   Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Program 5: Configure default route, static route to the Router.

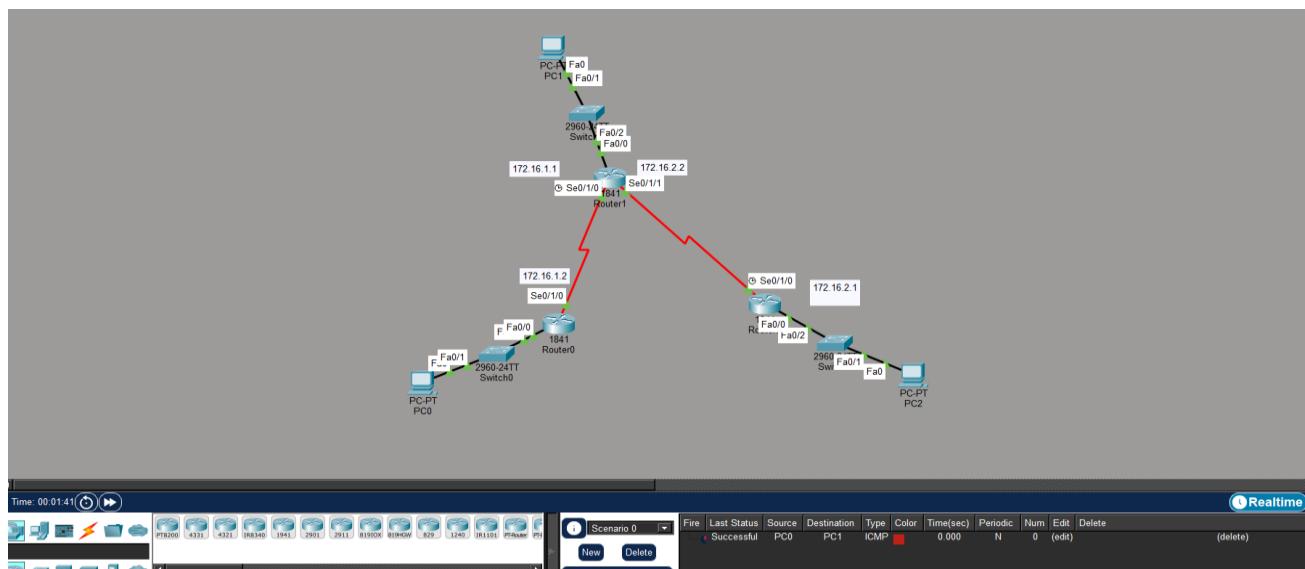
Lab 5: Static and Dynamic Routing

B. Configure default route, static routes to the routers

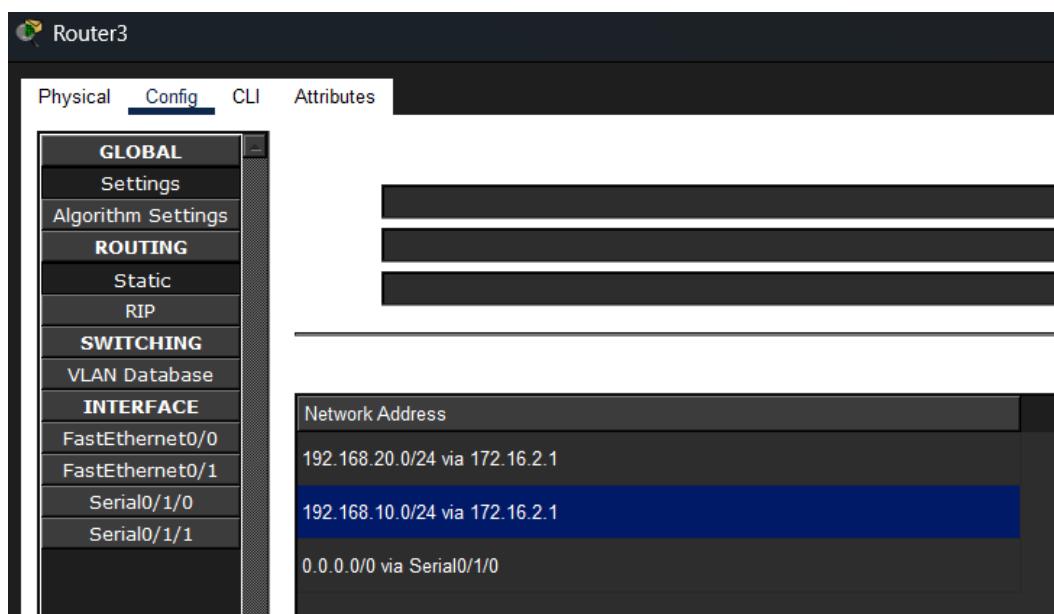
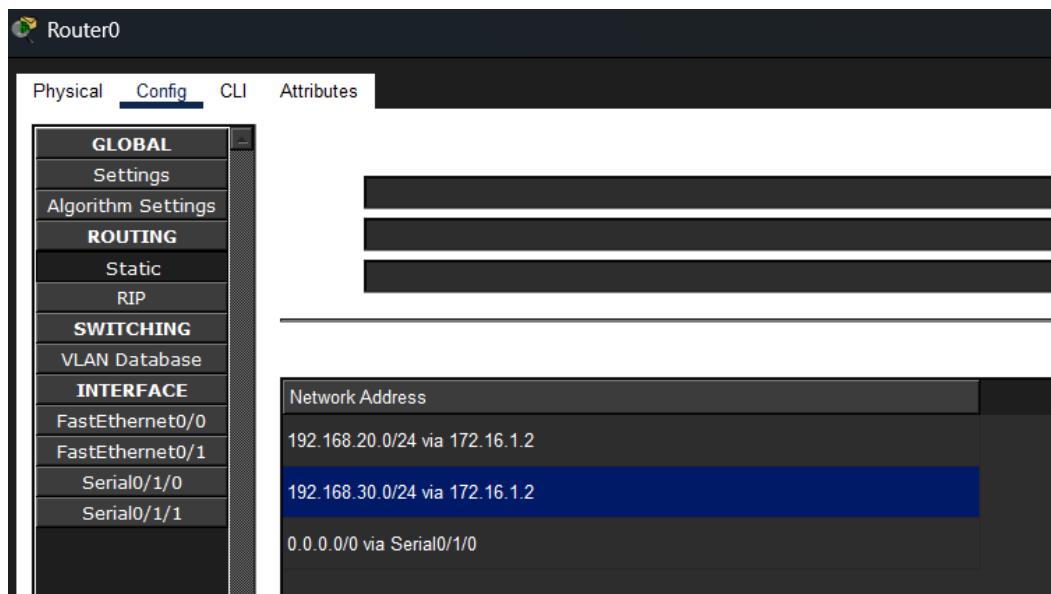
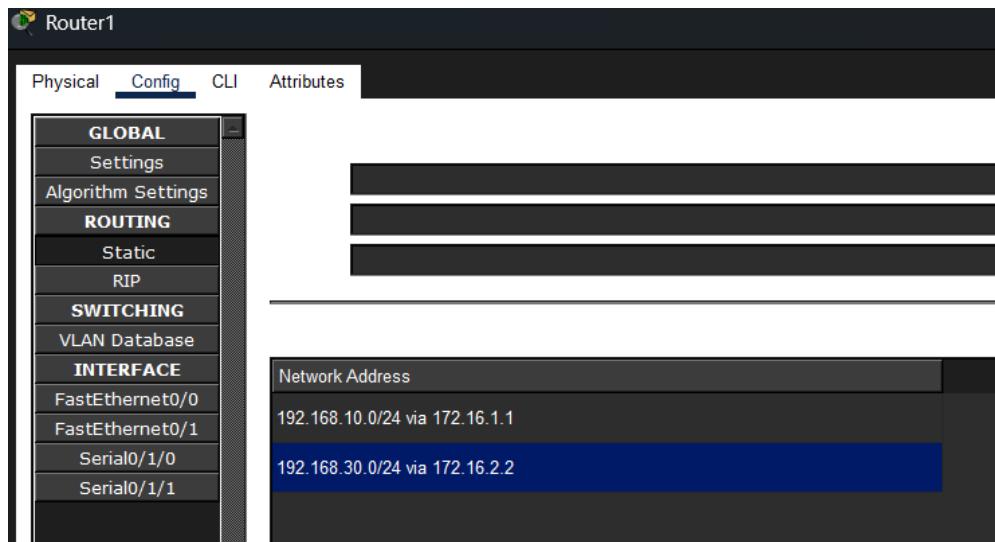
- Add 3 routes (R1 to R3) and label of R1, R2, R3
- Add 3 switches
- Add 3 PCs
- Connect them all w. wires
- Router 1 → Config
- Physical → Assign HWIC 2/1 into 2 blank slots and then power
 - ↳ Send config
- PCs reachable + config = IP: 192.168.10.10, subnet mask: 255.255.255.0, gateway: 192.168.10.1
- Connect R1 to R2 with serial PCle wire (See photo page)
- Router 1 → LCD
- no
- enable
- # config t
- # int se0/0/0
- # ip address 192.168.10.1 255.255.255.0
 - (config of R1 to R2)
- # no shutdown
- # exit
- # interface fa0/0
- # ip address 192.168.10.1 255.255.255.0
 - # no shutdown
 - no write memory
- Router 2 → LCD
- no
- enable
- # config t
- # ip route 192.168.10.0 255.255.255.0 192.168.10.1
 - (config of R2 to R1)
- # ip route 192.168.30.0 255.255.255.0 192.168.10.1
 - (config of R2 to R3)
- # ip route 0.0.0.0 0.0.0.0 192.168.10.1
 - (config of R2 to R1)
- # exit
- Set up IP address for R3

IP1	192.168.10.10	192.168.20.10	192.168.30.10
IP2	192.168.10.11	192.168.20.11	192.168.30.11
- Click on Router 3
- enable
- # config t
- # hostname R3
 - (config of R3)
- # ip route 192.168.20.0 255.255.255.0 192.168.10.1
 - (config of R3 to R1)
- # ip route 192.168.30.0 255.255.255.0 192.168.10.1
 - (config of R3 to R2)
- # ip route 0.0.0.0 0.0.0.0 192.168.10.1
 - (config of R3 to R1)
- # exit
- Write
- Send PC from PLC to PLC
 - Result: Successful

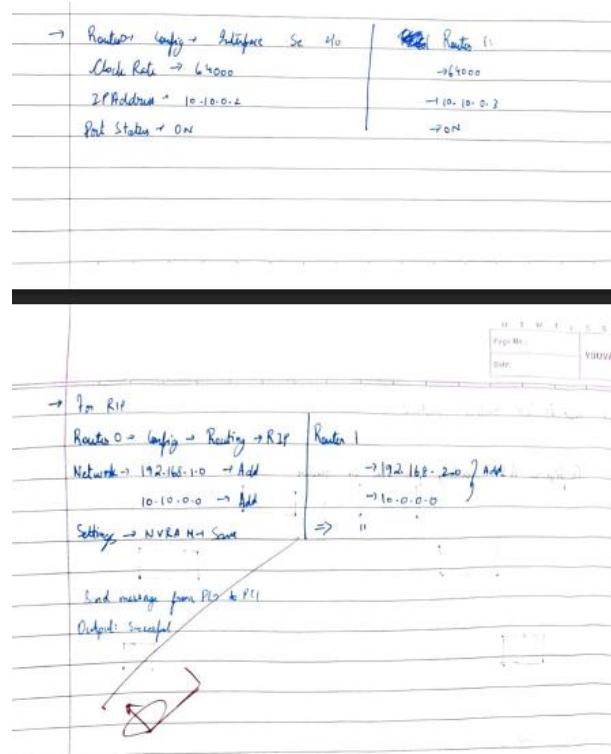
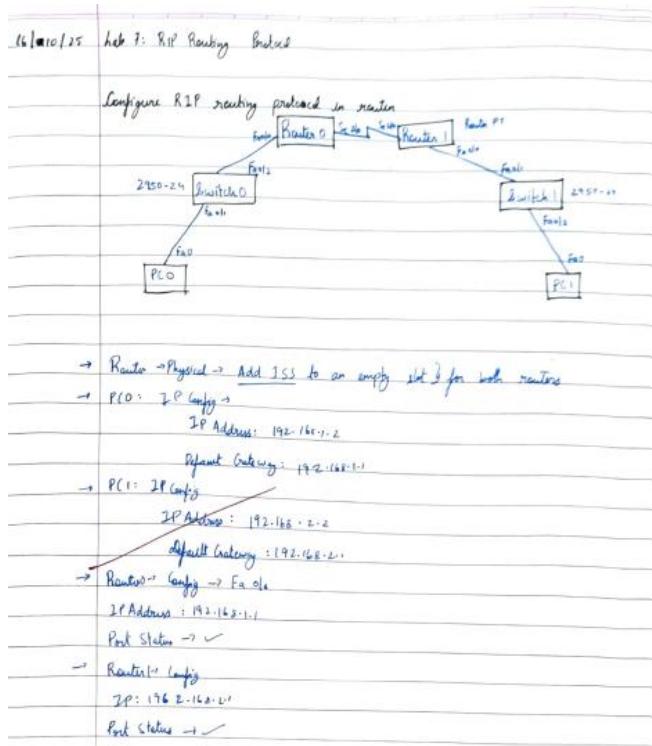
Network diagram:



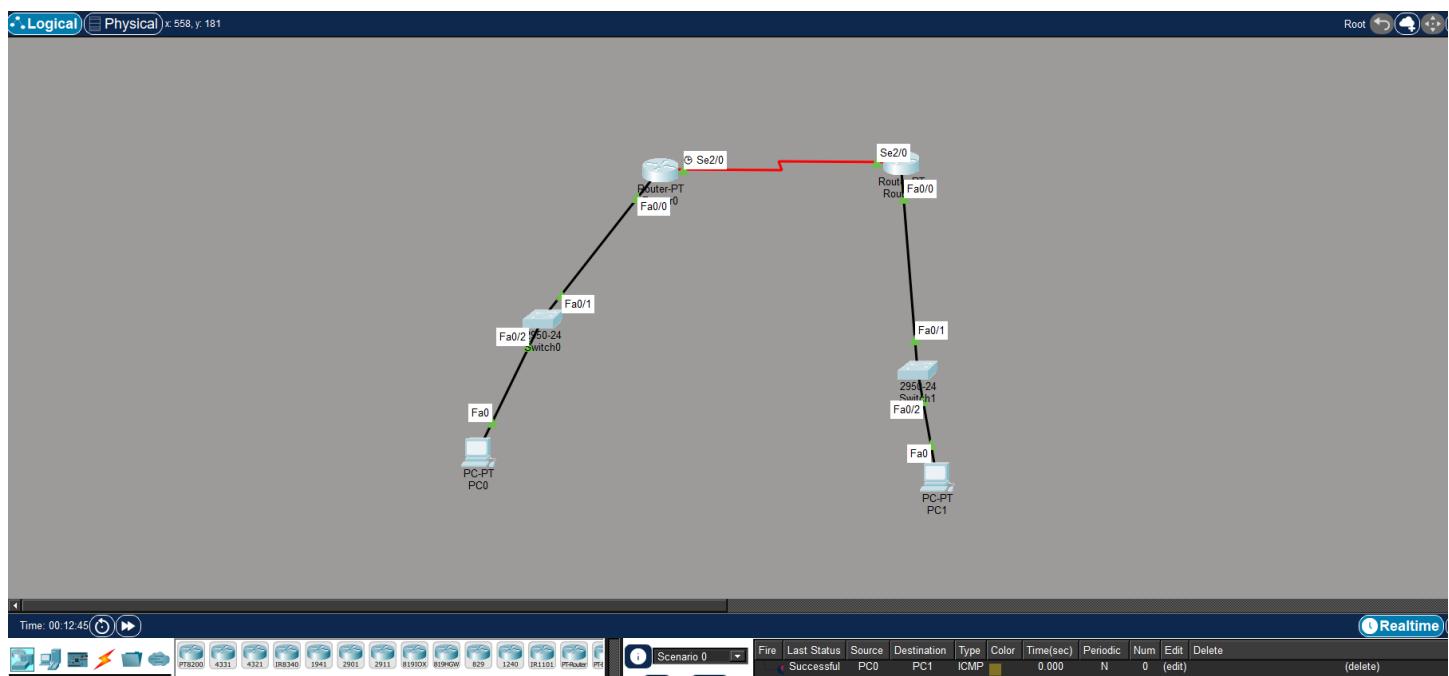
Configuration:



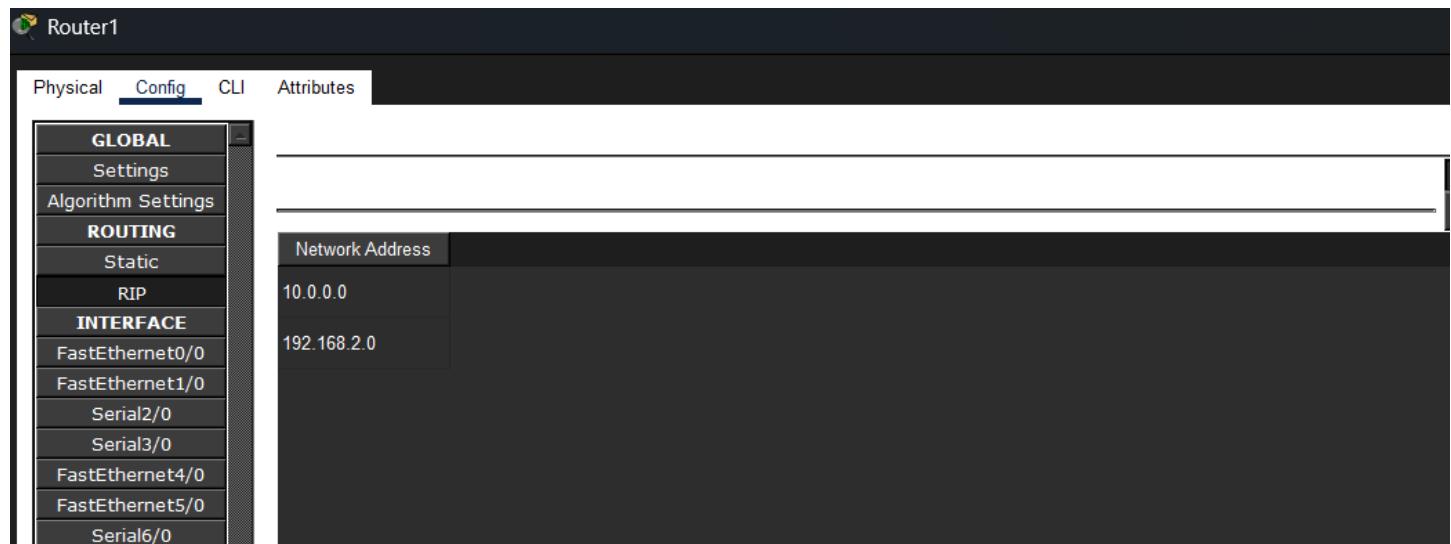
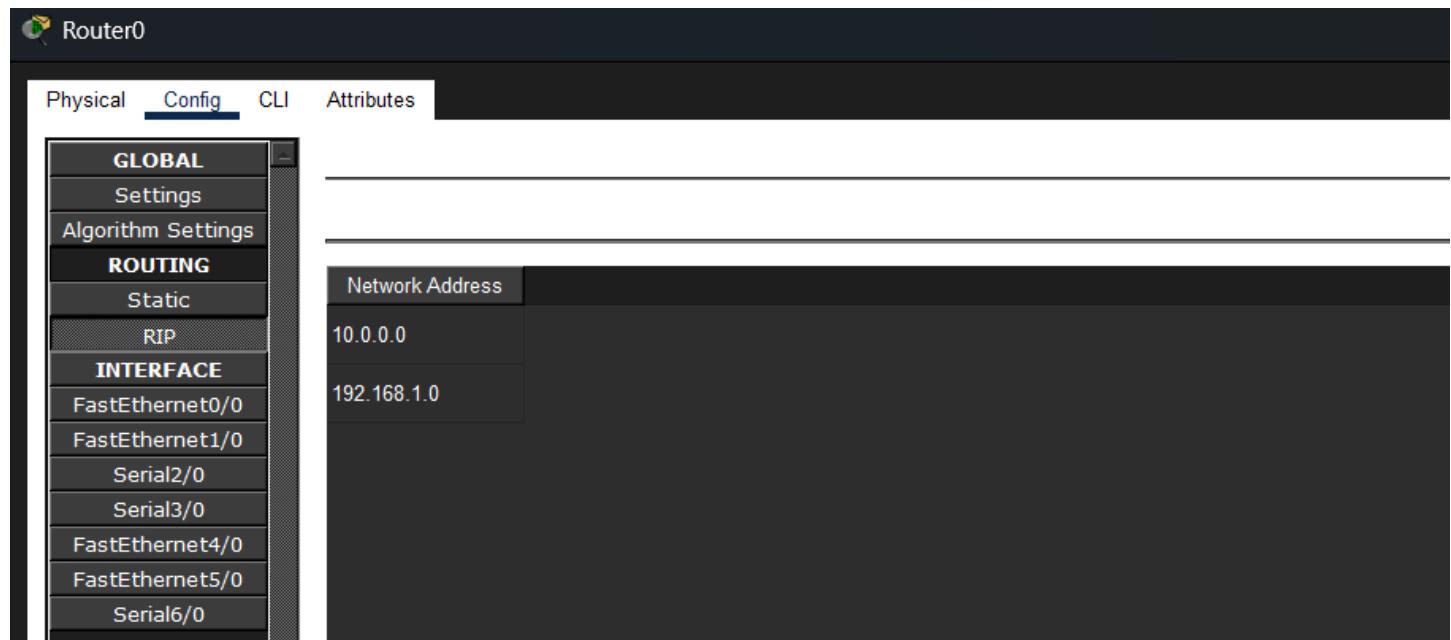
Program 6: Configure RIP routing Protocol in Routers.



Network diagram:



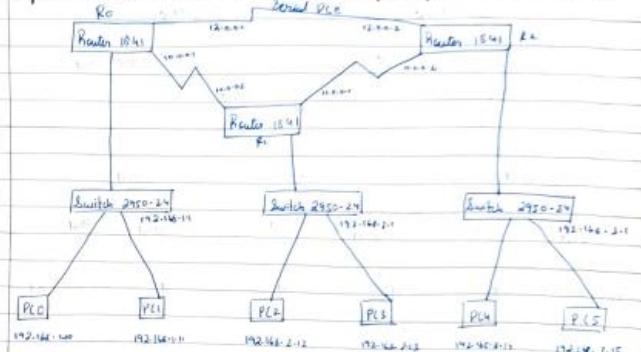
Configuration:



Program 7: Configure OSPF routing protocol.

06/01/25 Lab 9: OSPF

Configure OSPF and demonstrate how packets are transferred from node A to node E.



→ IP Config

R1: 192.168.1.11 } Default gateway: 192.168.1.1
PC1: 192.168.1.11

PC2: 192.168.2.12 } Default gateway: 192.168.2.1
PC3: 192.168.2.13

PC4: 192.168.3.15 } Default gateway: 192.168.3.1
PC5: 192.168.3.15

→ For routers R1 and R2, do the following by changing ports and IP address:

no shutdown
enable
config
hostname R1

int Fa0/0

ip add 192.168.1.1 255.255.255.0

no shutdown

exit

int Se0/0/0

ip add 10.0.0.1 255.0.0.0

clock rate 64000

no shutdown

exit

int Se0/0/1

ip add 12.0.0.1 255.0.0.0

clock rate 64000

no shutdown

exit

router ospf 1

network 192.168.1.0 0.0.0.255 area 0

network 10.0.0.0 0.0.0.255 area 0

network 12.0.0.0 0.0.0.255 area 0

exit

exit

→ Router 2 ospf

router ospf 1

network 192.168.2.0 0.0.0.255 area 0

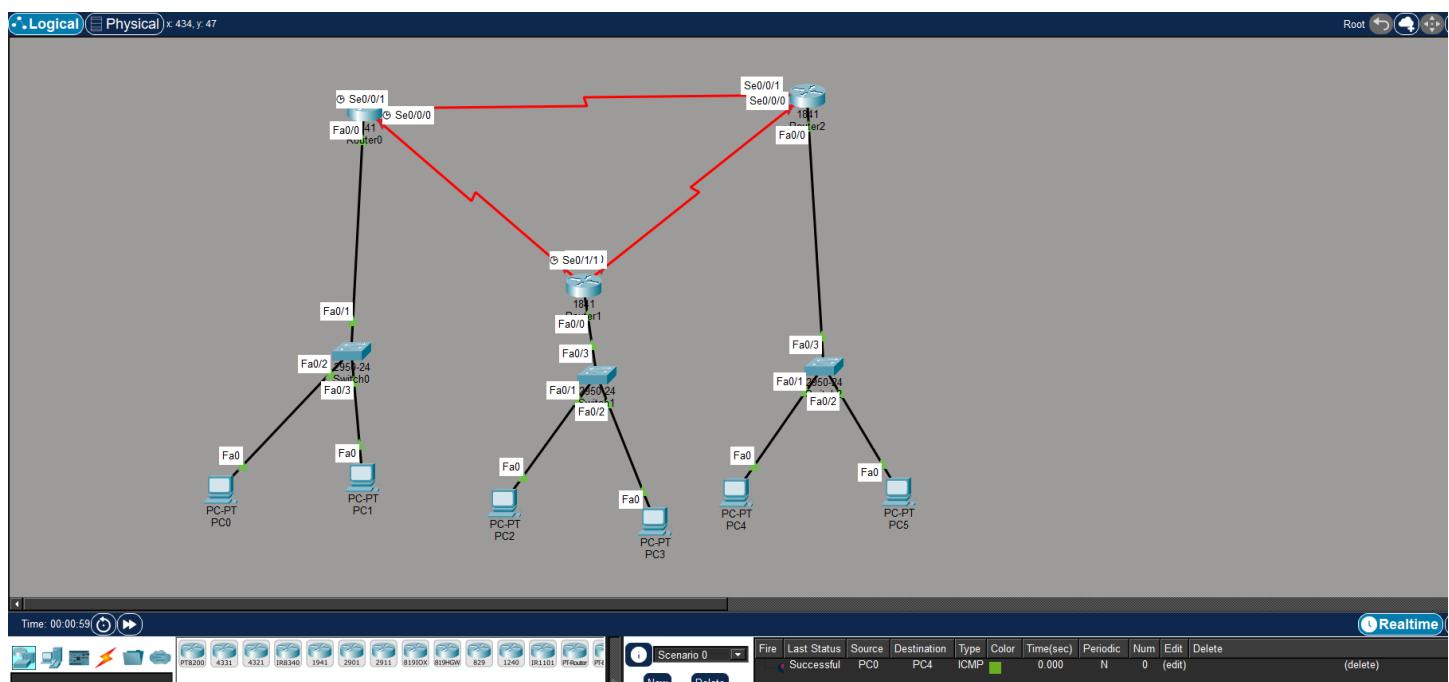
network 10.0.0.0 0.0.0.255 area 0

network 12.0.0.0 0.0.0.255 area 0

exit

exit

Network diagram:



Configuration:

```
Router0
```

Physical Config **CLI** Attributes

```
* FastEthernet/IEEE 802.3 Interface(s)
4 Low-speed serial(sync/async) network interface(s)
191K bytes of NVRAM.
63488K bytes of ATA CompactFlash (Read/Write)
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE SOFTWARE 
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team

Press RETURN to get started!

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial0/0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/1, changed state to up
00:00:10: *OSPF-5-ADJCHG: Process 1, Nbr 192.168.3.1 on Serial0/0/1 from LOADING to FULL, Loading I

r0>enable
r0#
r0#configure terminal
Entered configuration commands, one per line. End with CNTL/Z.
r0(config)#interface FastEthernet0/0
r0(config-if)#
r0(config-if)#exit
r0(config)#router rip
r0(config-router)#
r0(config-router)#end
r0#configure terminal
Entered configuration commands, one per line. End with CNTL/Z.
r0(config)#
r0(config)#
%SYS-5-CONFIG_I: Configured from console by console

r0(config)#interface FastEthernet0/1
r0(config-if)#
r0(config-if)#exit
r0(config)#interface Serial0/0/0
r0(config-if)#
r0(config-if)#exit
r0(config)#interface Serial0/1/1
r0(config-if)#
r0(config)#exit
r0(config)#router ospf 1
r0(config-router)#network 192.168.1.0 0.0.0.255 area 0
r0(config-router)#network 10.0.0.0 0.255.255.255 area 0
r0(config-router)#network 12.0.0.0 0.255.255.255 area 0
r0(config-router)#exit
r0(config)#do wr
Building configuration...
[OK]
^Z
```

Router1

Physical Config **CLI** Attributes

```
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team
Image text-base: 0x60080608, data-base: 0x6270CD50
```

```
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
```

```
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
```

```
If you require further assistance please contact us by sending email to
export@cisco.com.
```

```
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
```

```
Processor board ID FTX0947Z18E
M860 processor; part number 0, mask 49
2 FastEthernet/IEEE 802.3 interface(s)
4 Low-speed serial(sync/async) network interface(s)
191K bytes of NVRAM.
63488K bytes of ATA CompactFlash (Read/Write)
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team
```

```
Press RETURN to get started!
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.2.0 0.0.0.255 area 0
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router(config-router)#network 11.0.0.0 0.255.255.255 area 0
Router(config-router)#exit
Router(config)#do wr
Building configuration...
[OK]
Router(config) #
```

Router2

Physical Config **CLI** Attributes

```
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
```

```
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
```

```
If you require further assistance please contact us by sending email to
export@cisco.com.
```

```
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
```

```
Processor board ID FTX0947Z18E
M860 processor; part number 0, mask 49
2 FastEthernet/IEEE 802.3 interface(s)
4 Low-speed serial(sync/async) network interface(s)
191K bytes of NVRAM.
63488K bytes of ATA CompactFlash (Read/Write)
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team
```

```
Press RETURN to get started!
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

```
%LINK-5-CHANGED: Interface Serial0/0/1, changed state to up
```

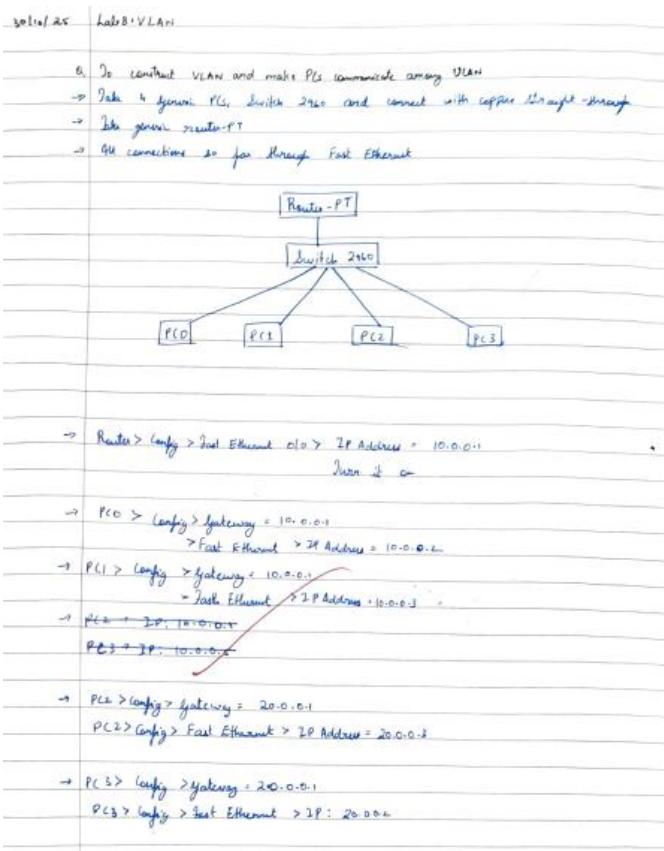
```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/1, changed state to up
```

```
00:00:10: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on Serial0/0/1 from LOADING to FULL, Loading Done
```

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#network 192.168.3.0 0.0.0.255 area 0
^
% Invalid input detected at '^' marker.
```

```
Router(config)#router ospf 1
Router(config-router)#network 192.168.3.0 0.0.0.255 area 0
Router(config-router)#network 11.0.0.0 0.255.255.255 area 0
Router(config-router)#network 12.0.0.0 0.255.255.255 area 0
Router(config-router)#exit
Router(config)#do wr
Building configuration...
[OK]
Router(config) #
```

Program 8: To construct a VLAN and make the PCs communicate among a VLAN.



→ Switch > VLAN Database > VLAN Number = 2 # Add
 VLAN Name: VLAN

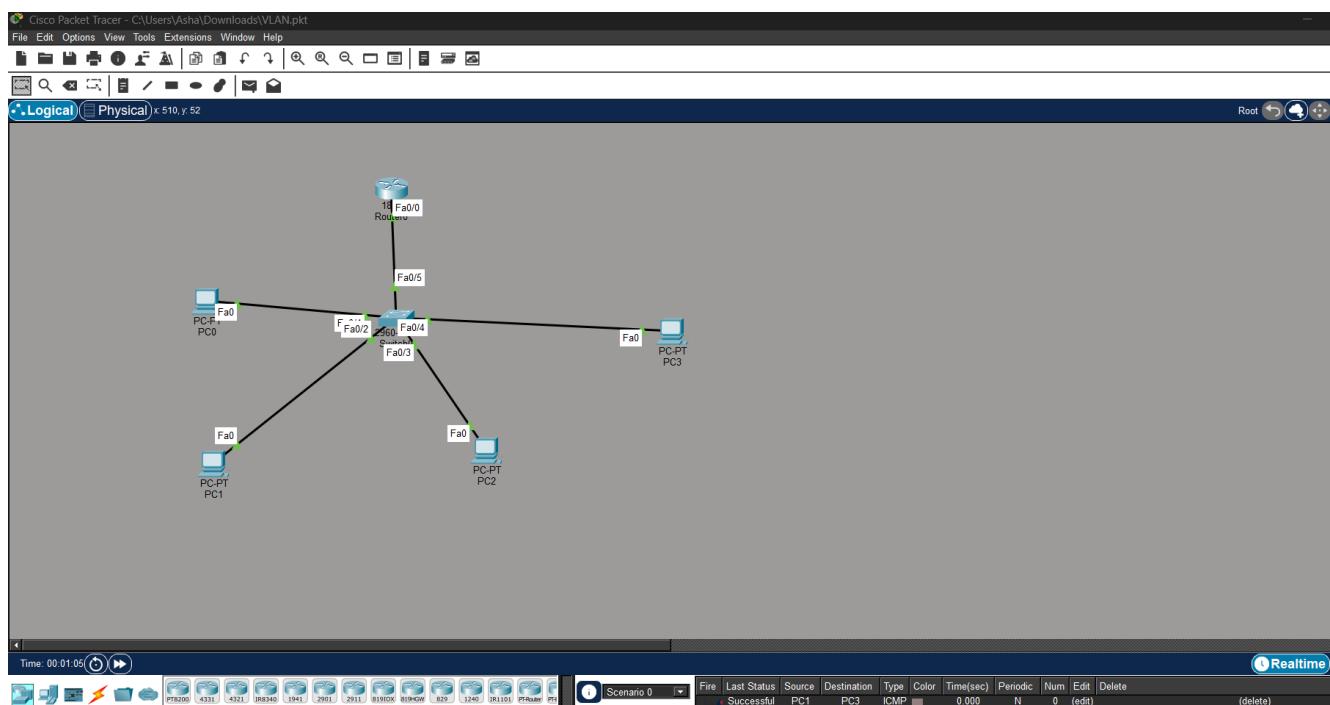
→ Check more port field going from router to switch
 Switch > Router Fast Ethernet 0/5 > Access > Trunk
 (Under VLAN dropdown)

→ Switch > config > Fast Ethernet 0/5 > 2: VLAN
 Fast Ethernet 0/5 > 2: VLAN

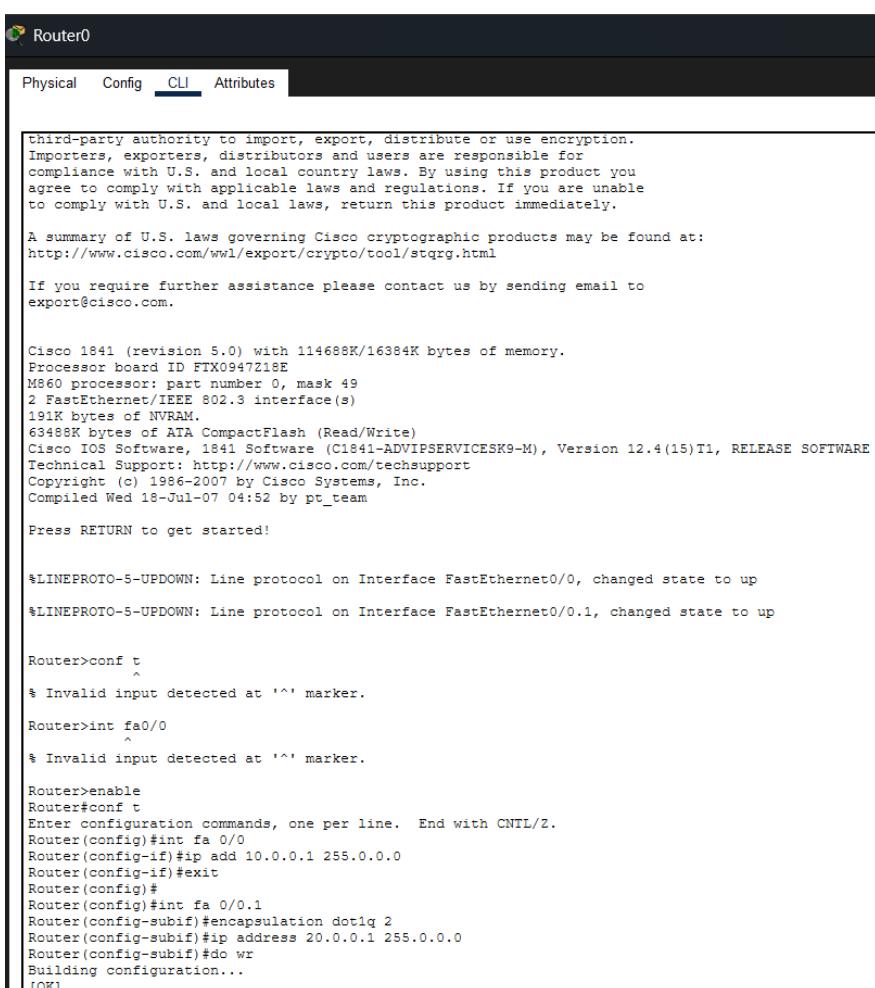
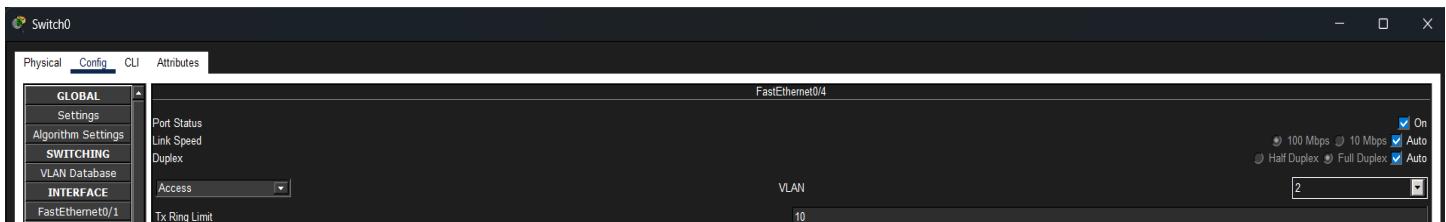
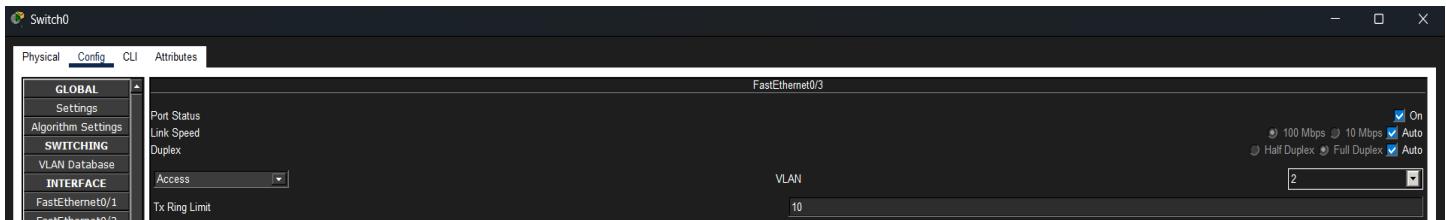
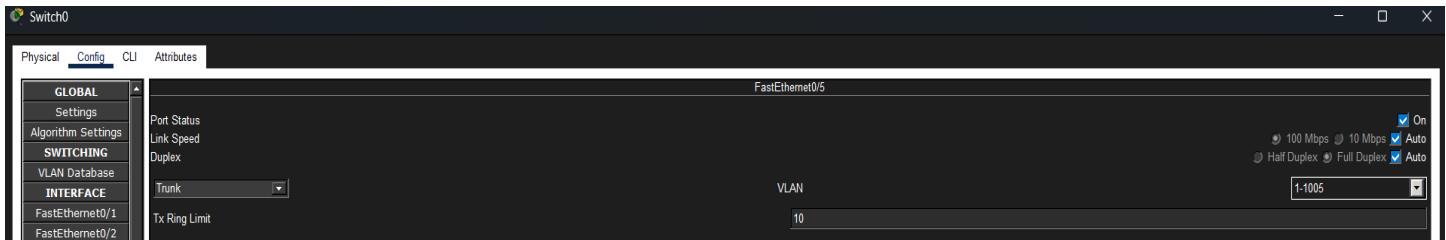
→ Router > CLI > Enable
 Router(Config)# config
 # interface fastEthernet 0/0
 # ip address 10.0.0.1 255.0.0.0
 # no shutdown
 # exit
 # interface Fa0/0/1
 #
 # encapsulation dot1q 2
 # ip address 20.0.0.1 255.0.0.0
 # no shutdown
 # exit

✓ 30/10/25

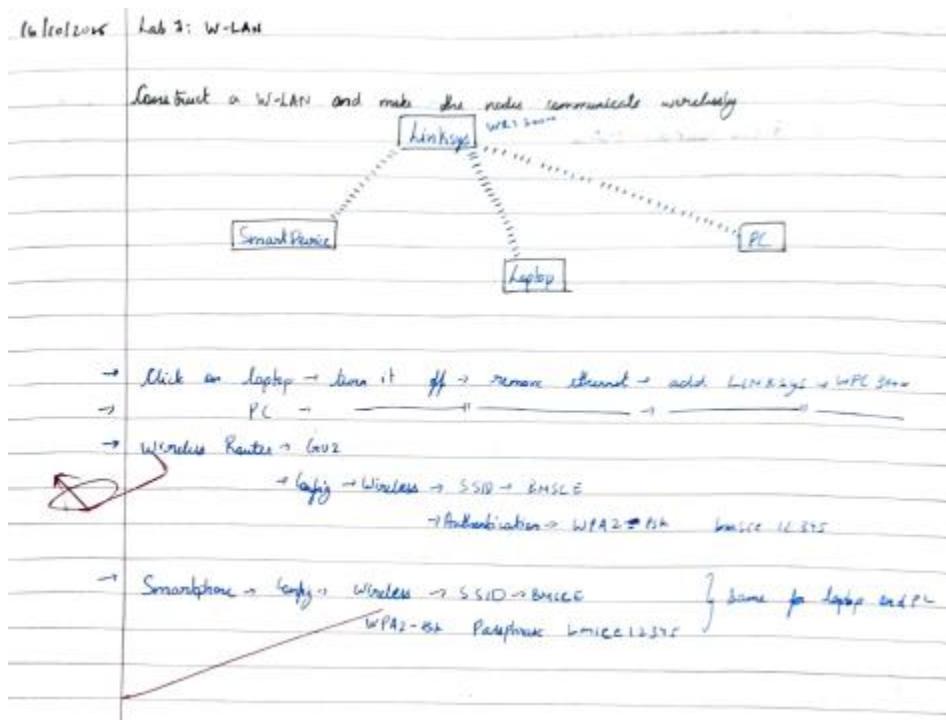
Network diagram:



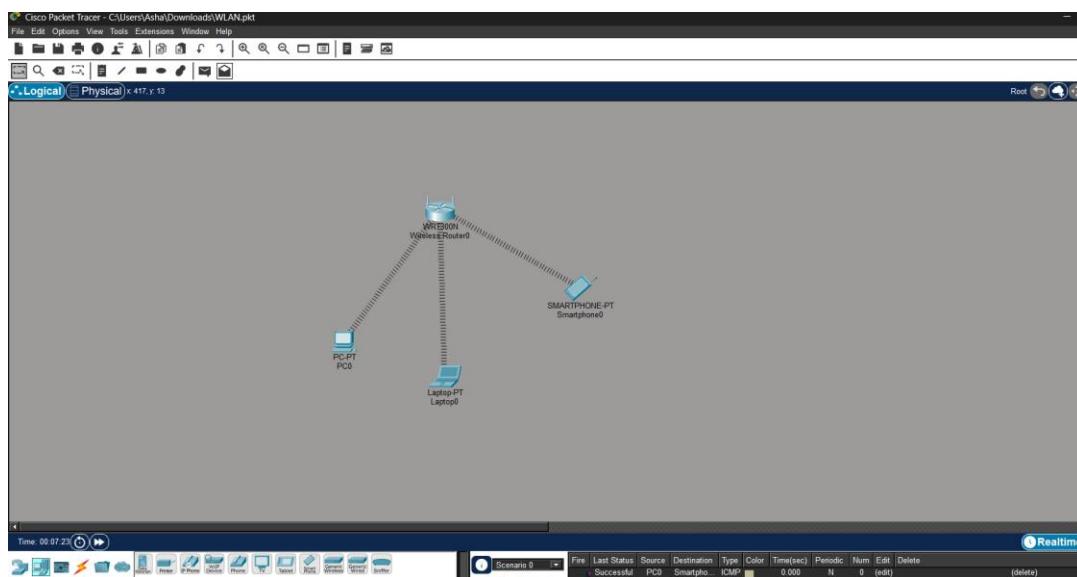
Configuration:



Program 9: To construct a WLAN and make the nodes communicate wirelessly.



Network diagram:



Configuration:

The image displays four separate windows of a network configuration interface, each representing a different device: Wireless Router0, PC0, Laptop0, and Smartphone0. Each window shows the 'Config' tab selected, with the 'Wireless' interface highlighted.

Wireless Router0 Configuration:

- SSID:** bmsce
- 2.4 GHz Channel:** 1 - 2.412GHz
- Coverage Range (meters):** 260.00
- Authentication:** WPA2-PSK (selected)
- WEP Key:** bmsce123
- PSK Pass Phrase:** bmsce123
- RADIUS Server Settings:** IP Address: 192.168.1.1, Shared Secret: bmsce123, Encryption Type: AES

PC0 Configuration:

- SSID:** bmsce
- Bandwidth:** 300 Mbps
- MAC Address:** 0005:5E2A:923A
- Authentication:** WPA2-PSK (selected)
- WEP Key:** bmsce123
- User ID:** MDS
- Password:** MDS
- Method:** WPA2
- Encryption Type:** AES

Laptop0 Configuration:

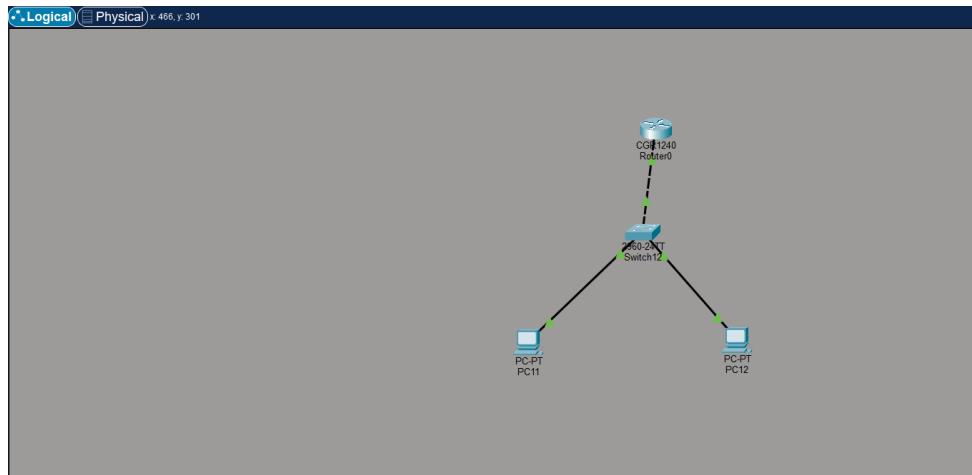
- SSID:** bmsce
- Bandwidth:** 300 Mbps
- MAC Address:** 0006:80EC:DBC4
- Authentication:** WPA2-PSK (selected)
- WEP Key:** bmsce123
- User ID:** MDS
- Password:** MDS
- Method:** WPA2
- Encryption Type:** AES

Smartphone0 Configuration:

- SSID:** bmsce
- Bandwidth:** 300 Mbps
- MAC Address:** 0009:5CC1:4C6A
- Authentication:** WPA2-PSK (selected)
- WEP Key:** bmsce123
- User ID:** MDS
- Password:** MDS
- Method:** WPA2
- Encryption Type:** AES

Program 10: Demonstrate the TTL/ Life of a Packet.

Network diagram:



Configuration:

```
C:\>ping 192.168.10.3

Pinging 192.168.10.3 with 32 bytes of data:

Reply from 192.168.10.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Program 11: To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Answers for 6: TELNET

B. Explain TELNET to access remote router remotely

→ TELNET is used to access remote server/router and it is a simple command line tool that runs on your computer and it allows you to send commands remotely through a device's administration.

→ It is also used to manage other devices like router, switch and also to check if the ports are open or closed.

1. Set up PC

→ IPconfig: IP of router = 192.168.1.1
IP Addr: 192.168.1.2

→ Router > C:\> netstat -an

> enable
con1
shutdown -r1

* # enable default ip
int Fa0/0
ip add address 192.168.1.1 255.255.255.0
no shutdown Fa0/0

* # done vrf 0
login password tp
password tp
con1
exit
wqy

* # show ip interface brief

PL2 Command prompt

ping 192.168.1.1

* > telnet 192.168.1.1

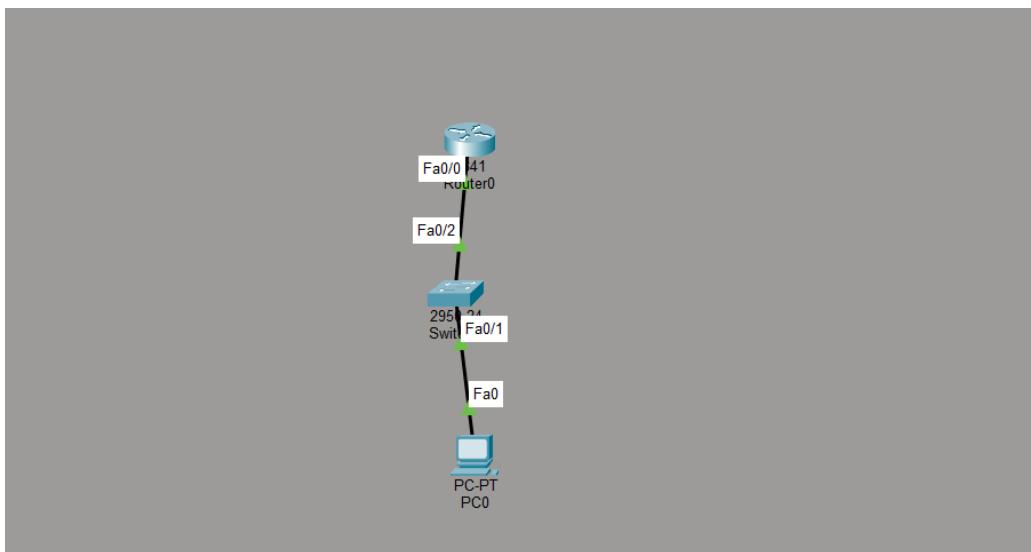
Router>

> enable
#enable ip
int con1
int Fa0/0
ip add 192.168.1.1 255.255.255.0
#down ip interface con1
do their ip interface config

Observation Router can be managed using telnet through the PC

8/2

Network diagram:



Configuration:

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname r1
r1(config)#enable secret rp
r1(config)#int fa0/1
r1(config-if)#exit
r1(config)#int fa0/0
r1(config-if)#ip add 192.168.10.1
* Incomplete command.
r1(config-if)#ip add 192.168.10.1 255.255.255.0
r1(config-if)#no shutdown

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

r1(config-if)#line vty 0 5
r1(config-line)#login
* Login disabled on line 194, until 'password' is set
* Login disabled on line 195, until 'password' is set
* Login disabled on line 196, until 'password' is set
* Login disabled on line 197, until 'password' is set
* Login disabled on line 198, until 'password' is set
* Login disabled on line 199, until 'password' is set
r1(config-line)#password tp
r1(config-line)#exit
r1(config)#exit
r1#
%SYS-5-CONFIG_I: Configured from console by console
wr
Building configuration...
[OK]
r1#show int brief
^
* Invalid input detected at '^' marker.

r1#show ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    192.168.10.1   YES manual up           up
FastEthernet0/1    unassigned     YES unset administratively down down
Vlan1             unassigned     YES unset administratively down down
```

Pinging 192.168.10.1 with 32 bytes of data:

```
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time=12ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
```

Ping statistics for 192.168.10.1:

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

Approximate round trip times in milli-seconds:

 Minimum = 0ms, Maximum = 12ms, Average = 3ms

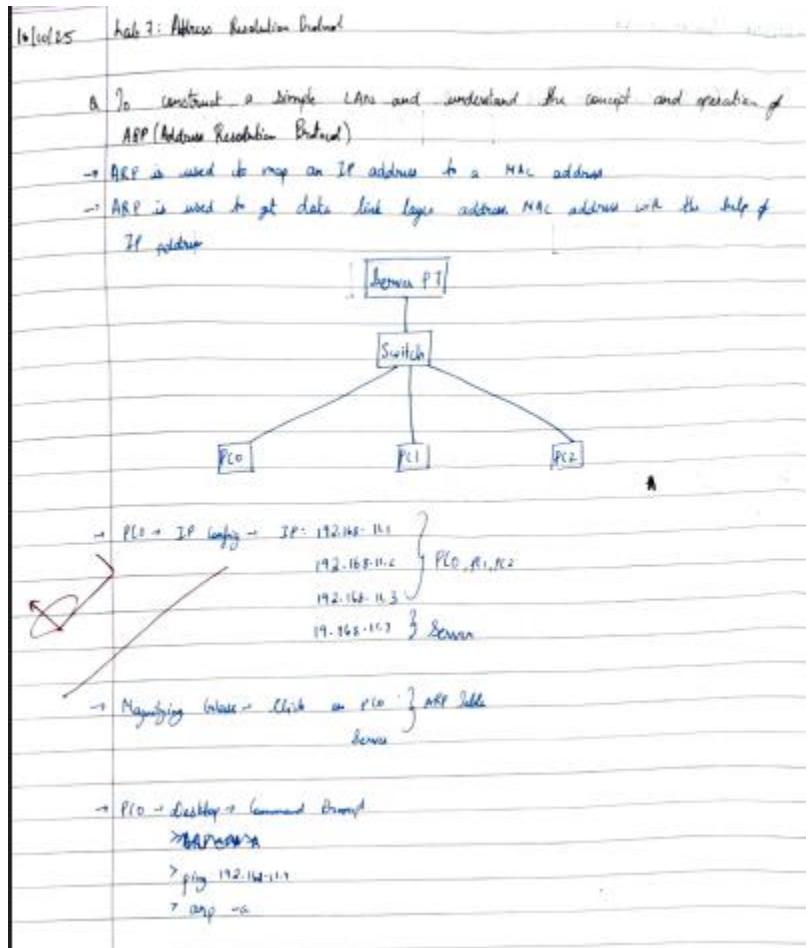
```
C:\>telnet 192.168.10.1
Trying 192.168.10.1 ...Open
```

User Access Verification

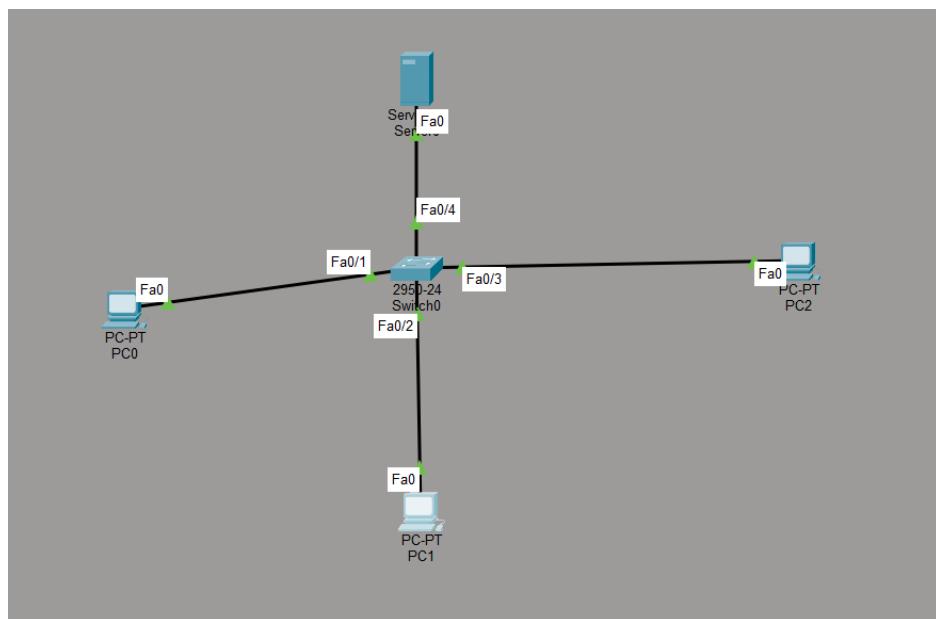
```
Password:
r1>enable
Password:
r1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
r1(config)#int fa0/1
r1(config-if)#ip add 192.168.20.1 255.255.255.0
r1(config-if)#do show ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    192.168.10.1   YES manual up           up
FastEthernet0/1    192.168.20.1   YES manual administratively down down
Vlan1             unassigned     YES unset administratively down down
r1(config-if)#

```

Program 12: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).



Network diagram:



Configuration:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
C:\>ping 192.168.11.4

Pinging 192.168.11.4 with 32 bytes of data:

Reply from 192.168.11.4: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.11.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

ARP Table for PC1

IP Address	Hardware Address	Interface
192.168.11.4	0040.0B12.2299	FastEthernet0

PART - B

Program 1: Write a program for congestion control using Leaky bucket algorithm.

12/11/2016 Part 1: Lab 10: Leaky Bucket Algorithm

Write a program for congestion control using leaky bucket algorithm

```

func leaky-bucket()
    input: bucket-size, output-rate, incoming-packets [n]
    SC T: bucket-content <= 0
    FOR time <= 1 to n DO
        PRINT Time Incoming-Packets
        bucket-content <- bucket-content + incoming-packets [time]
        IF bucket-content > bucket-size THEN
            dropped <- bucket-content - bucket-size
            bucket-content <- bucket-size
        ELSE
            dropped <= 0
        ENDIF
        sent <- MIN(bucket-content, output-rate)
        bucket-content <- bucket-content - sent
        PRINT sent, dropped, bucket-content
    ENDFOR
    while bucket-content > 0
        sent: leaching Remaining-Packets
        sent <- MIN(bucket-content, output-rate)
        bucket-content <- bucket-content - sent
        PRINT sent, bucket-content
    endwhile

```

D

Bucket size = 0, Output rate = 1, No. of Packets = 5

Incoming Packet Size	Dropped	Data pre leak	Data after leak
6	0	6	5
4	0	9	8
8	6	10	9
1	0	10	10
0	0	9	8

Bucket remaining: all packets processed.

Code:

```

#include <stdio.h>
int min(int x, int y)
{
    return (x < y) ? x : y;
}
int main()
{
    int drop = 0, mini, nsec, cap, count = 0, i, inp[25], process;
    printf("Enter the bucket size:\n");
    scanf("%d", &cap);
    printf("Enter the processing rate:\n");
    scanf("%d", &process);

```

```

printf("Enter the number of seconds you want to simulate:\n");
scanf("%d", &nsec);
for (i = 0; i < nsec; i++)
{
    printf("Enter the size of the packet entering at %d sec:\n", i + 1);
    scanf("%d", &inp[i]);
}
printf("\n Second | Packet received | Packet sent | Packet left | Dropped\n");
printf("-----\n");
for (i = 0; i < nsec; i++)
{
    printf("Enter the size of the packet entering at %d sec:\n", i + 1);
    scanf("%d", &inp[i]);
}

printf("\n Second | Packet received | Packet sent | Packet left | Dropped\n");
printf("-----\n");

for (i = 0; i < nsec; i++)
{
    count += inp[i];

    if (count > cap)
    {
        drop = count - cap;
        count = cap;
    }

    printf("%6d | %15d |", i + 1, inp[i]);

    mini = min(count, process);
    printf(" %11d |", mini);

    count -= mini;
    printf(" %12d | %7d\n", count, drop);

    drop = 0;
}
// Process remaining packets after all seconds

```

```

for (; count != 0; i++)
{
if (count > cap)
{
    drop = count - cap;
    count = cap;
}

printf("%6d | %15d |", i + 1, 0);

mini = min(count, process);
printf(" %11d |", mini);

count -= mini;
printf(" %12d | %7d\n", count, drop);
}

return 0;
}

```

Output:

```

Enter initial packets in the bucket: 0
Enter total no. of times bucket content is checked: 4
Enter total no. of packets that can be accommodated in the bucket: 10
Enter no. of packets that enters the bucket at a time: 4
Enter no. of packets that exits the bucket at a time: 1
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10

```

Program 2: 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.

<pre> Julia (222) Lab 11: TCP Server.java import java.net.*; import java.io.*; public class Server { public static void main(String[] args) throws IOException { ServerSocket server = new ServerSocket(1234); System.out.println("Server Ready for connection"); Socket soc = server.accept(); System.out.println("Connection Successful waiting for Chatting"); BufferedReader br = soc.getInputStream(); String from = br.readLine(); BufferedReader fr = new BufferedReader(new InputStreamReader(br)); String content = fr.readLine(); OutputStream os = soc.getOutputStream(); BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(os)); bw.write(content); bw.flush(); while(true & content.equals("quit")) { pr.println("ok"); } } public class { public class { public class { content.read(); } } } } hi hello </pre>	<pre> Client.java import java.net.*; import java.io.*; public class Client { public static void main(String[] args) throws IOException { Socket soc = new Socket("127.0.0.1", 1234); System.out.println("File name ?"); BufferedReader br = new BufferedReader(new InputStreamReader(soc.getInputStream())); String name = br.readLine(); System.out.println("File name : " + name); String str; while((str = br.readLine()) != null) { System.out.println(str); } } } </pre> <p style="text-align: right;">→ java Client.java java Client enter file name: # helloworld hi hello.</p> <p style="text-align: right;"><i>↓</i></p>
--	---

Code:

ClientTCP.py

```

from socket import *

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

```

ServerTCP.py

```

from socket import *

serverName = "127.0.0.1" serverPort = 12000 serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort)) serverSocket.listen(1)

```

while 1:

```

    print("The server is ready to receive") connectionSocket, addr = serverSocket.accept() sentence =
    connectionSocket.recv(1024).decode() file = open(sentence, "r")

```

```

    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('\nSent contents of ' + sentence) file.close() connectionSocket.close()

```

Output:

The screenshot shows two separate instances of Visual Studio Code running side-by-side. Both instances have the same project structure in the Explorer sidebar, which includes files for ClientTCP.py, ServerTCP.py, VLAN.pkt, and WLAN.pkt.

Top Terminal (Python: ServerTCP):

```
PS C:\Users\dell\Desktop\CN_LAB & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ServerTCP.py
Desktop/CN_LAB/ServerTCP.py
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive
```

Bottom Terminal (Python: ClientTCP):

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ClientTCP.py
Enter file name: ServerTCP.py
From Server:
from socket import *
serverName='127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print ('Sent contents of' + sentence)
    file.close()
    connectionSocket.close()
```

Program 3: 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.

20/1/2022 UDP

DServer.java

```

import java.net.*;
import java.util.Scanner;
public class DServer {
    public static void main (String [] args) throws Exception {
        System.out.println ("DServer");
        DatagramSocket ds = new DatagramSocket ();
        Scanner scanner = new Scanner (System.in);
        System.out.println ("\n\nEnter the Message:");
        while (true) {
            String msg = scanner.nextLine ();
            InetAddress ip = InetAddress.getByName ("127.0.0.1");
            DatagramPacket dp = new DatagramPacket (msg.getBytes (), msg.length (), ip, 3000);
            ds.send (dp);
        }
    }
}

```

DSERVER.java

```

import java.net.*;
public class DSERVER {
    public static void main (String [] args) throws Exception {
        System.out.println ("DSERVER");
        DatagramSocket ds = new DatagramSocket (3000);
        ds.receive (dp);
        String msg = new String (dp.getData (), 0, dp.getLength ());
        System.out.println (msg);
    }
}

```

Output

→ java DServer.java
java DServer
Send
Enter Message:
hi my name is jay

→ java DSERVER.java
java DSERVER
Receive
hi my name is jay

Code:

ClientUDP.py

```

from socket import *

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

sentence = input("\nEnter file name: ") clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)

print("\nReply from Server:\n") print(filecontents.decode("utf-8")) clientSocket.close()

```

ServerUDP.py

```

from socket import *

serverPort = 12000 serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort)) print("The server is ready to receive")

while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048) sentence = sentence.decode("utf-8")

```

```
file = open(sentence, "r") con = file.read(2048)
serverSocket.sendto(bytes(con, "utf-8"), clientAddress)

print("\nSent contents of ", end=' ')
print(sentence)
file.close()
```

Output:

Program 4: Write a program for error detecting code using CRC-CCITT (16-bits).

18/11/2023 CRC

B. WAP for error detecting code using CRC-CCITT (16-bits) from (RCC)

input message-bit
Set generator = 0x1021
Set CRC = 0xFFFF

Append 16 0's to message-bit
CRC & CRC XOR (bit 15)
for i=1 to 1 do
if (CRC & 0x8000) ≠ 0
 CRC = (CRC & 0x7FFF) + gen
else
 CRC = CRC<<1
 CRC & CRC and 0xFFFF

Output CRC

Append CRC to original message-bit
if rec-crc(little transmitted-frame) ≠ 0
 print "No error detected"
else
 print "Error detected"

→ Output

Enter data bits: 101101
Enter generator bits: 1101

CRC bits: 010
Transmitted Data (Data + CRC) = 101101010
No error detected. Data received correctly.

Code:

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

int main()
{
    char rem[50], a[50], s[50], c, msj[50], gen[30];
    int i, genlen, t, j, flag = 0, k, n;
```

```

printf("Enter the generator polynomial:\n");
fgets(gen, sizeof(gen), stdin);
gen[strcspn(gen, "\n")] = '\0'; // remove newline if present
printf("Generator polynomial (CRC-CCITT): %s\n", gen);

genlen = strlen(gen);
k = genlen - 1;

printf("Enter the message:\n");
fgets(msj, sizeof(msj), stdin);
msj[strcspn(msj, "\n")] = '\0'; // remove newline

n = strlen(msj);

// Append k zeros to the message
for (i = 0; i < n; i++)
    a[i] = msj[i];
for (i = 0; i < k; i++)
    a[n + i] = '0';
a[n + k] = '\0';

printf("\nMessage polynomial appended with zeros:\n");
puts(a);

// Division (XOR)
for (i = 0; i < n; i++)
{
    if (a[i] == '1')
    {
        t = i;
        for (j = 0; j <= k; j++)
        {
            if (a[t] == gen[j])
                a[t] = '0';
            else
                a[t] = '1';
            t++;
        }
    }
}
// Get remainder
for (i = 0; i < k; i++)
    rem[i] = a[n + i];

```

```

rem[k] = '\0';
printf("\nChecksum (Remainder):\n");
puts(rem);
// Append checksum to message
printf("\nTransmitted message (with checksum):\n");
for (i = 0; i < n; i++)
    a[i] = msj[i];
for (i = 0; i < k; i++)
    a[n + i] = rem[i];
a[n + k] = '\0';
puts(a);
// Receiver side
printf("\nEnter the received message:\n");
fgets(s, sizeof(s), stdin);
s[strcspn(s, "\n")] = '\0'; // remove newline
n = strlen(s);
// Division on received message
for (i = 0; i < n - k; i++)
{
    if (s[i] == '1')
    {
        t = i;
        for (j = 0; j <= k; j++, t++)
        {
            if (s[t] == gen[j])
                s[t] = '0';
            else
                s[t] = '1';
        }
    }
}
for (i = 0; i < k; i++)
    rem[i] = s[n - k + i];
rem[k] = '\0';
// Check for error
flag = 0;
for (i = 0; i < k; i++)
{
    if (rem[i] == '1')
        flag = 1;
}
if (flag == 0)

```

```
    printf("\nReceived message is error-free  \n");
else
    printf("\nReceived message contains errors  \n");
return 0;
}
```

Output:

```
Output
Enter the generator polynomial:
101
Generator polynomial (CRC-CCITT): 101
Enter the message:
110101

Message polynomial appended with zeros:
11010100

Checksum (Remainder):
11

Transmitted message (with checksum):
11010111

Enter the received message:
11010111

Received message is error-free 

==== Code Execution Successful ===
```