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:

PRADEEP G

(1BM24CS414)

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 Pradeep G (1BM24CS414) during the 5th Semester August 2025-December 2025

Signature of the Faculty Incharge:

Sarala D V
Assistant Professor

Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

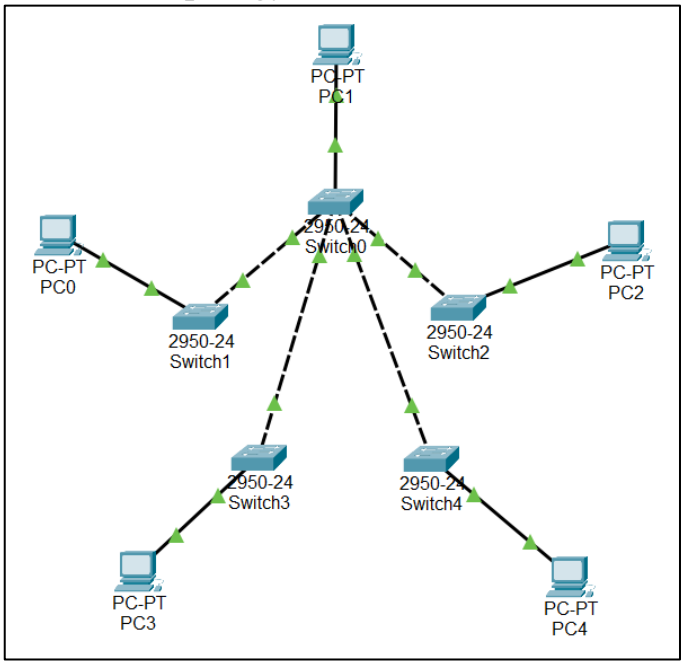
Table of Contents

PART - A

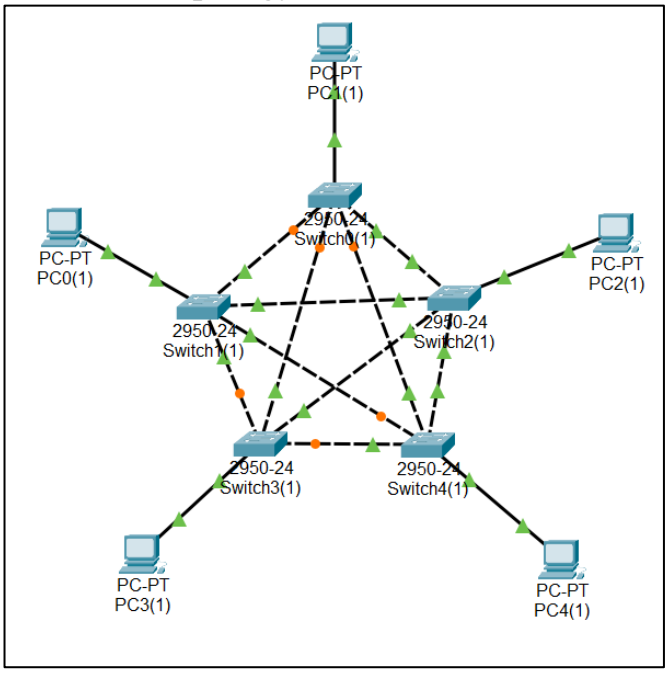
Program 1: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

Network diagram:

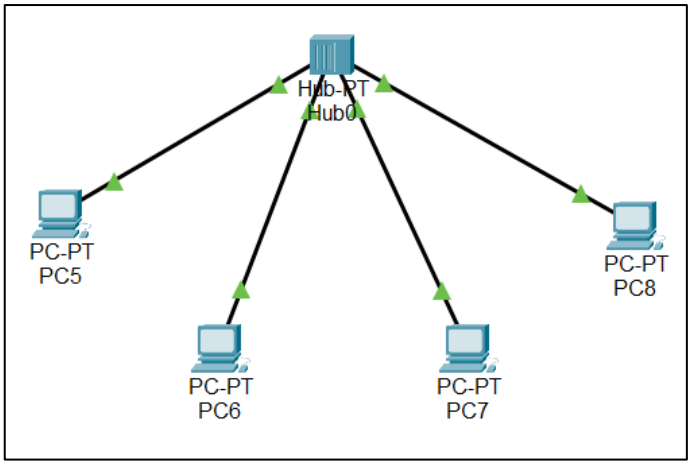
1. STAR Topology with Switch:



2. MESH Topology with Switch:



3. HUB-Based Network Topology:



Configuration:

Program 1

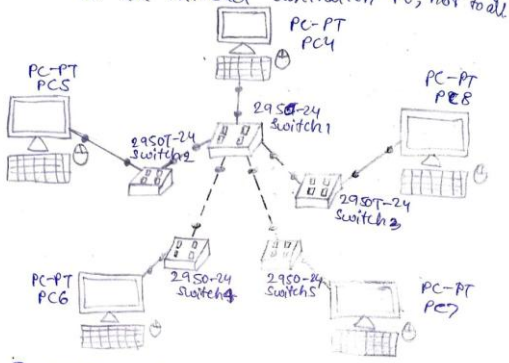
Aim: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message. using CISCO Packet Tracer.

Star Topology using Switch:

① Name: Implementation of Star Topology using cisco Packet Tracer.

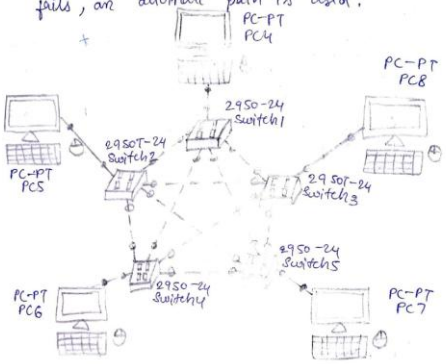
- Procedure:
  1. Open a new project in Packet Tracer.
  2. Drag and drop:
     - 1 central switch (Switch 0)
     - 4 Edge Switches (Switch 2, Switch 3, Switch 4, Switch 5)
     - 5 PCs (PC4, PC5, PC6, PC7, PC8)
  3. Connect each PC to its respective switch using copper Straight-Through cables.
  4. Connect all edge switches to the central switch.
  5. Assign IP addresses: [click on PC icon/device → Desktop → IP configuration → enter IP address]
     - PC4 → 192.168.1.2
     - PC5 → 192.168.1.3
     - PC6 → 192.168.1.4
     - PC7 → 192.168.1.5
     - PC8 → 192.168.1.6

     & Subnet mask: 255.255.255.0 is directly entered
- Simulation:
  1. Switch to Simulation Mode (bottom-right corner).
  2. click the Add simple PDU (envelop icon) tool.
  3. Select source PC → then Destination PC.
  4. In the Event list, a packet creation entry will appear.
  5. Click Auto Capture/Play or Capture/Forward to watch packet flow.
  6. With Switch, the packet travels only to the intended destination PC, not to all.



② MESH Topology using Switches

- Name: Implementation of Mesh Topology using Switches in Cisco Packet Tracer.
- Procedure:
  1. Open a new project in Packet Tracer.
  2. Drag and Drop:
     - 5 switches (Switch1, Switch2, Switch3, Switch4, Switch5)
     - 5 PC's (PC4, PC5, PC6, PC7, PC8)
  3. Connect each PC to a switch.
  4. Interconnect all switches (mesh connection).
  5. Assign IP addresses:
     - PC4 → 192.168.1.2
     - PC5 → 192.168.1.3
     - PC6 → 192.168.1.4
     - PC7 → 192.168.1.5
     - PC8 → 192.168.1.6
- Simulation:
  1. Switch to Simulation Mode.
  2. Click the Add simple PDU tool.
  3. Choose source PC → then destination PC.
  4. In the Event List, packet transmission will be logged.
  5. Use Auto Capture/Play or Capture/Forward to observe the packet flow.
  6. In mesh topology, packets have multiple possible paths between switches, if one link fails, an alternate path is used.



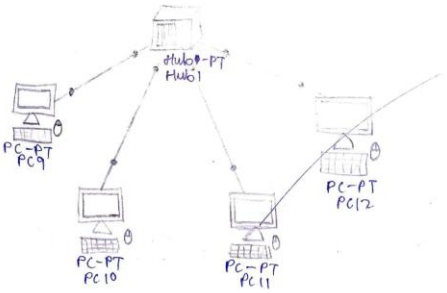
③ Hub-Based Network Topology:

Name: Implementation of Hub-Based Network Topology in Cisco Packet Tracer.

Procedure:

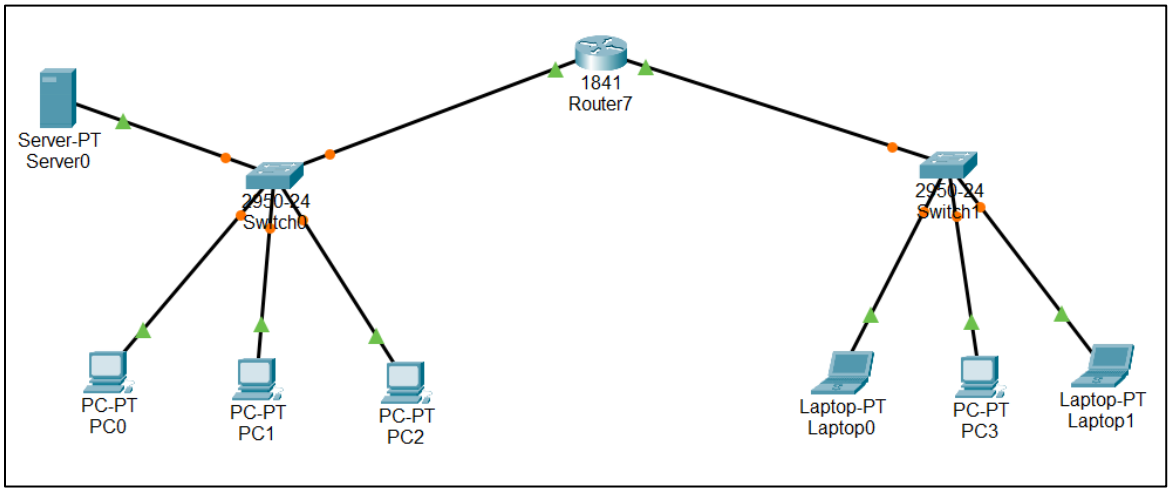
1. Open a new project in Packet Tracer.
2. Drag and drop:
   - 1 Hub (Hub1)
   - 4 PCs (PC9, PC10, PC11, PC12)
3. Connect all PCs to the hub using copper-straight-through cables.
4. Assign IP addresses:
   - PC9 → 192.168.1.2
   - PC10 → 192.168.1.3
   - PC11 → 192.168.1.4
   - PC12 → 192.168.1.5

   (Subnet Mask: 255.255.255.0)
- Simulation:
  1. Switch to Simulation Mode
  2. click the Add Simple PDU tool.
  3. choose Source PC → then Destination PC
  4. Check the Event List for packet creation.
  5. Use Auto Capture/Play or Capture/Forward to watch transmission.
  6. With a Hub, the packet is broadcast to all devices, but only the destination PC accepts it while others discard it.



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

Network diagram:



Configuration:

9/9/25 program 2

* Configure DHCP within a LAN & outside a LAN

Procedure:

① click on Server

② Go to desktop → IP configuration → select static & give (192.168.10.2) → & give Gateway (192.168.10.1)

③ Go to Services → DHCP → Disktop IP config → Static → 192.168.10.2
Gateway → 192.168.10.1

④ Services → DHCP → Pool Name : SwitchOne
Gateway : 192.168.20.1
Start IPadd : 192.168.10.3
Subnet mask : 255.255.255.0

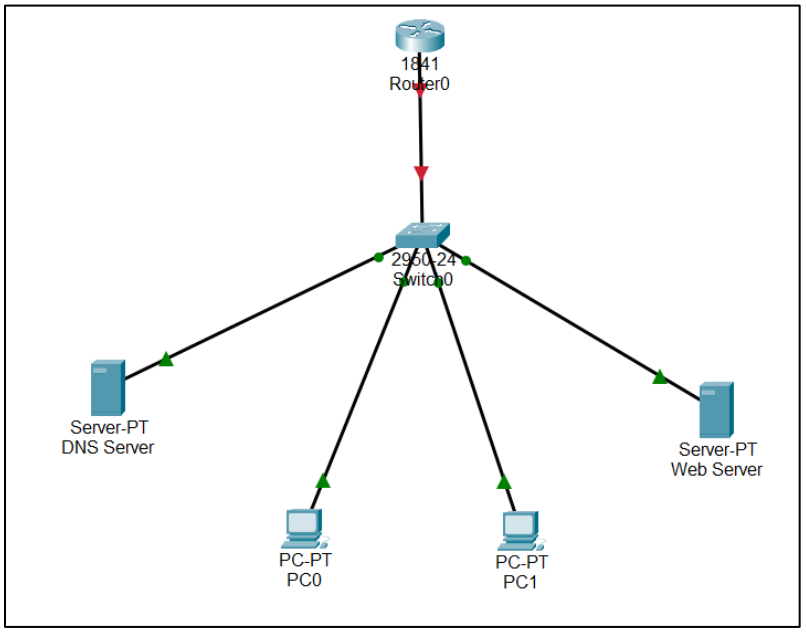
Pool Name : Switch Two
Gate way : 192.168.20.1
Start IP add : 192.168.20.2
Subnet mask : 255.255.255.0

Add Add

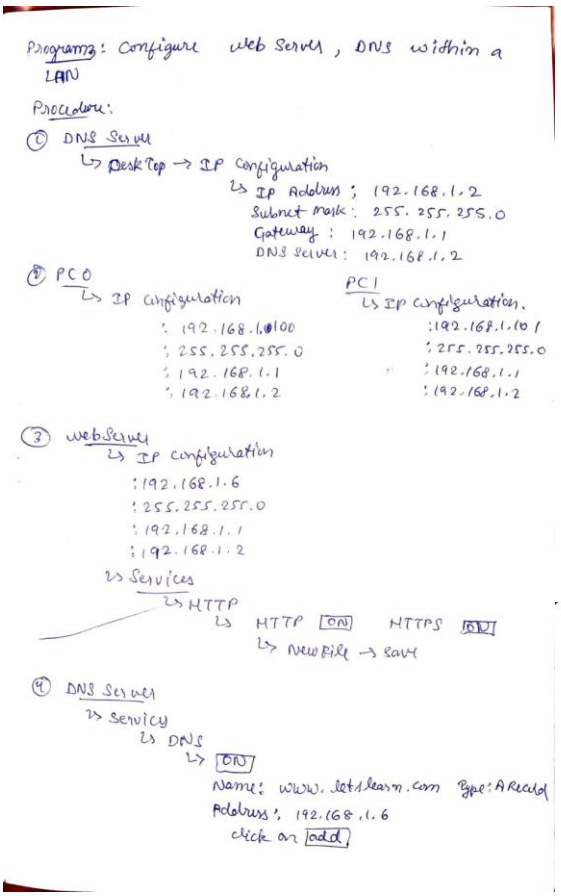
⑤ Router → CLI → no enter.
Router # enable
conf t
① # int Fa0/0
ip address 192.168.10.1 255.255.255.0
ip helper-address 192.168.10.2
no shutdown
do write memory
exit
② # int Fa 0/1
IP address 192.168.20.1 255.255.255.0
ip helper-address 192.168.10.2
no shutdown
exit
do write memory.

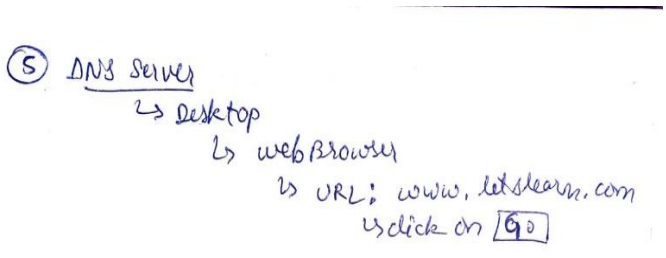
Program 3: Configure Web Server, DNS within a LAN.

Network diagram:



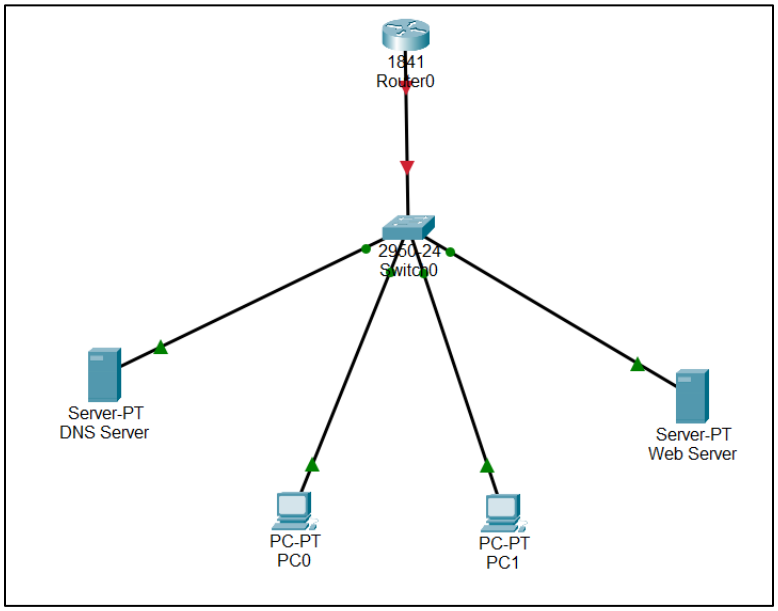
Configuration:





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

Network diagram:



Configuration:

Program 4: Configure IP address to the router in packet tracer explore the following messages
1. Ping response
2. Destination unreachable
3. Request timeout
4. Reply

Aim:
Network Connection (Diagram:
Same as prev experiment (DNS - Web Server)

Procedure
1. Assign IP address as follow

Device	Interface	IP Address	Subnet mask	Gateway
Router0 (Fa0/0)	—	192.168.1.1	255.255.255.0	—
PC0	Fa0	192.168.1.10	255.255.255.0	192.168.1.1
PC1	Fa0	192.168.1.20	255.255.255.0	192.168.1.1
DNS Server	Fa0	192.168.1.100	255.255.255.0	192.168.1.1
Web Server	Fa0	192.168.1.200	255.255.255.0	192.168.1.1

2. Configure Router interface
=> Router > enable
Router # configure terminal
Router(config)# #interface fa0/0
Router(config-if) # ip address 192.168.1.1 255.255.255.0
Router(config-if) # no shutdown
3. Save Configuration:
Router# copy
4. Configure IP & Default Gateway in PCs/Servers
(Desktop → IP configuration)
5. Test connectivity using ping command from PCs.
6. Change conditions (wrong IP, wrong gateway, shut interface, power off device) to observe diff ping messages

Observations:
Case 1 : Ping Response
Ping Command : ping 192.168.1.20
(PC0 → PC1 in same network)
Message observed : Ping Response
Reason: ICMP Echo Request and Echo Reply exchanged successfully b/w two active devices.

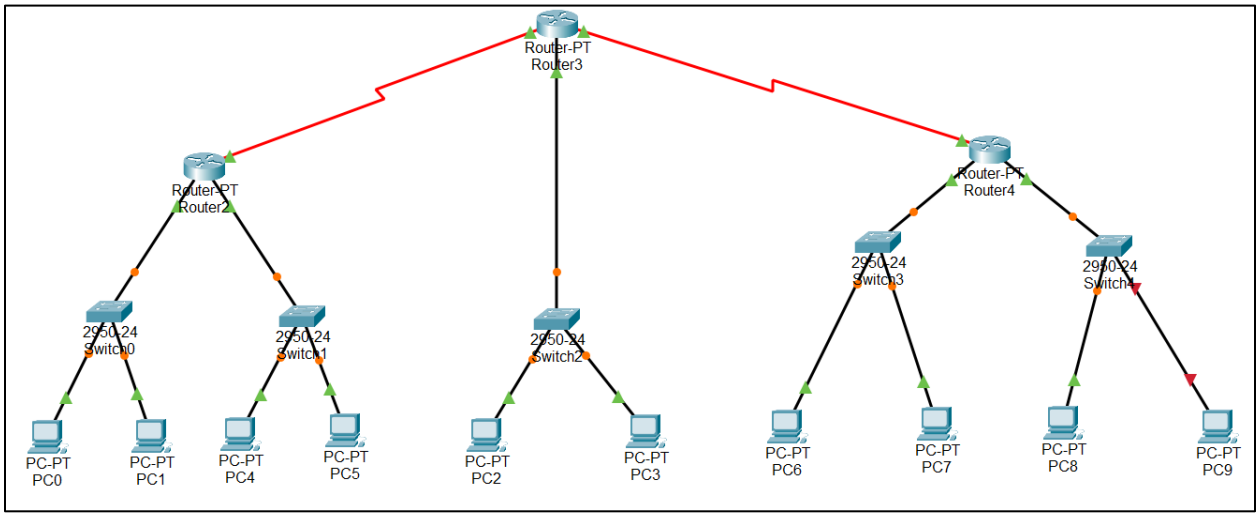
Case 2 : Reply
Ping Command : ping 192.168.1.100
(PC0 → DNS Server)
Message observed: Reply from 192.168.1.100
Reason: Destination device is active, reachable & properly configured.

Case 3: Destination Unreachable
Ping Command : ping 192.168.1.200
(towards web server)
First Remove Gateway on PC0
Message observed: Destination Host Unreachable
Reason: Router cannot be reached due to missing/incorrect gateway, so no route exists.

Case 4: Request timed out
Ping Command : ping 192.168.1.1500
(Non-existent device)
Message observed: Request timed out
Reason: No reply received since IP does not exist/device is off.

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

Network diagram:



Configuration:

23/9/25

Program 5

Configure Default Route, Static Route to the Router:

① Static Route:

It is a manually configured path that tells a router how to reach a specific network.
It is entered by Network administrator.

Connections:

* Router 1: Select [Router-PT]

config → Routing → Static

known data: 1,2,4
unknown data: 3,5,6,7

①	②	③	④
Network: 192.168.3.0	Network: 192.168.5.0	Network: 192.168.6.0	192.168.7.0
Mask:	Mask:	Mask:	
Next Hop: 192.168.4.2	Next Hop: 192.168.4.2	Next Hop: 192.168.4.2	192.168.4.2

Then click on add (for each)

→ Interface

→ FastEthernet 0/0 (FE 0/0)
- on ☑
- IP address: 192.168.1.1
- Subnet mask: 255.255.255.0

→ FastEthernet 1/0 (FE 1/0)
- on ☑
- IP address: 192.168.2.1
- Subnet mask: 255.255.255.0

→ Serial 2/0 (Se 2/0)
- on ☑
- IP address: 192.168.4.1
- Subnet Mask: 255.255.255.0

* Router 2: Select [Router-PT]

config → Routing → Static

known data: 3,4,7
unknown data: 1,2,5,6

①	②	③	④
Network: 192.168.1.0	192.168.2.0	192.168.5.0	192.168.6.0
Mask:			
Next Hop: 192.168.4.1	192.168.4.1	192.168.7.2	192.168.7.2

Then click on add for each

→ Interface

→ FastEthernet (Fa 0/0)
- on ☑
- IP Address: 192.168.3.1
- Subnet mask: 255.255.255.0

→ Serial 2/0 (Se 2/0)
- on ☑
- IP Address: 192.168.4.2
- Subnet mask: 255.255.255.0

→ Serial 3/0 (Se 3/0)
- on ☑
- IP Address: 192.168.7.1
- Subnet mask: 255.255.255.0

* Router 3: Select Router-PT

known data: 5,6,7
unknown data: 1,2,3,4

config → Routing → Static

①	②	③	④
Network: 192.168.1.0	192.168.2.0	192.168.3.0	192.168.4.0
Mask:			
Next Hop: 192.168.7.1	192.168.7.1	192.168.7.1	192.168.7.1

Then click on add (for each)

→ Interface

→ FastEthernet 0/0 (Fa 0/0)
- on ☑
- IP Address: 192.168.5.1
- Subnet Mask: 255.255.255.0

→ FastEthernet 1/0 (Fa 1/0)
- on ☑
- IP address: 192.168.6.1
- Subnet Mask: 255.255.255.0

→ Serial 2/0 (Se 2/0)
- on ☑
- IP address: 192.168.7.2
- Subnet Mask: 255.255.255.0

* PC 0 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.1.2
        Subnet mask : 255.255.255.0
        Default Gateway : 192.168.1.1

* PC 1 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.1.3
        Subnet mask : 255.255.255.0
        Default Gateway : 192.168.1.1

* PC2 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.2.2
        Subnet mask : 255.255.255.0
        Default Gateway : 192.168.2.1

* PC3 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.2.3
        Subnet Mask : 255.255.255.0
        Default Gateway : 192.168.2.1

* PC4 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.3.2
        Subnet Mask : 255.255.255.0
        Default Gateway : 192.168.3.1

* PC5 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.3.3
        Subnet mask : 255.255.255.0
        Default Gateway : 192.168.3.1

* PC6 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.5.2
        Subnet mask : 255.255.255.0
        Default Gateway : 192.168.5.1

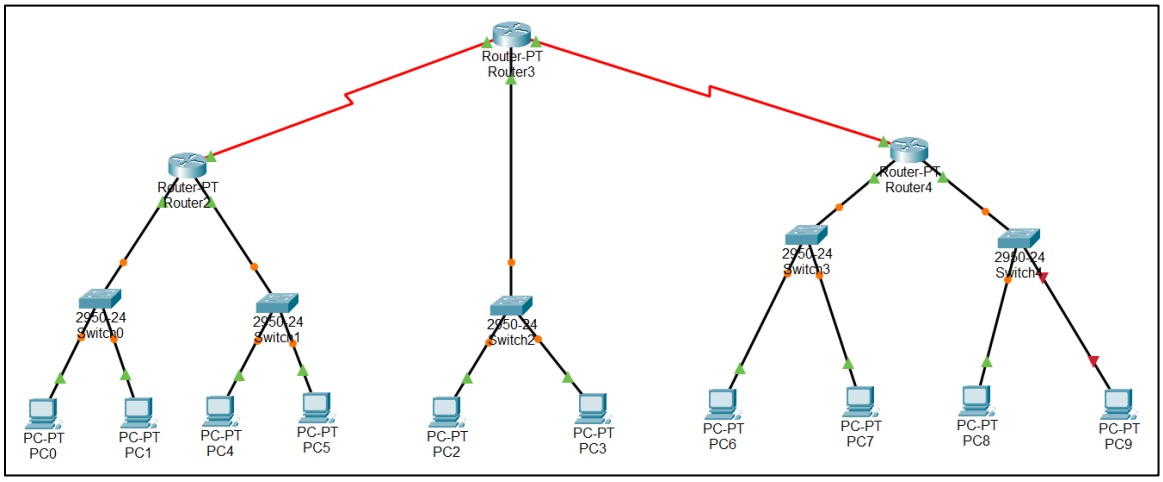
* PC7 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.5.3
        Subnet Mask : 255.255.255.0
        Default Gateway : 192.168.5.1

* PC8 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.6.2
        Subnet Mask : 255.255.255.0
        Default Gateway : 192.168.6.1

* PC9 (Select PC-PT)
  - Desktop
    - IP configuration
      - IP address : 192.168.6.3
        Subnet Mask : 255.255.255.0
        Default Gateway : 192.168.6.1

Program 6: Configure RIP routing Protocol in Routers.

Network diagram:



Configuration:

(2) Dynamic Route :-

Dynamic Routing is a networking technique where routers automatically and adaptively share routing information using protocols to find the best path for data to travel across a network.

Connections:

Same as Static Routing, but we have to remove all static Routes [under Routing] from all routers & assign the Dynamic Routing, i.e.,

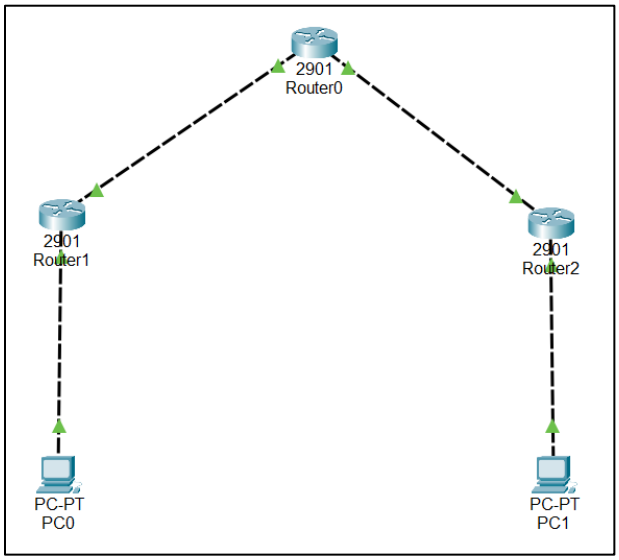
* Router 1 : (select Router-PT)
  - Config
    - Routing
      - RIP Routing
        - Network: 192.168.1.0
          192.168.2.0
          192.168.4.0
          then click on add [for each]

* Router 2 :
  - Config
    - Routing
      - RIP Routing
        - Network: 192.168.3.0
          192.168.4.0
          192.168.7.0
          then click on add [for each]

* Router 3 :
  - Config
    - Routing
      - RIP Routing
        - Network: 192.168.5.0
          192.168.6.0
          192.168.7.0
          then click on add [for each]

Program 7: Configure OSPF routing protocol.

Network diagram:



Configuration:

* Configure OSPF (Open Shortest Path First) Routing Protocol.

Note: If ≤ 15 Router devices then use RIP Routing Protocol.

Connections:

* PC0
  - Desktop
    - IP Configuration
      - IP address: 192.168.44.2
      - Subnet Mask: 255.255.255.0
      - Default Gateway: 192.168.44.1

* PC1
  - Desktop
    - IP Configuration
      - IP address: 10.10.22.2
      - Subnet Mask: 255.0.0.0
      - Default Gateway: 10.10.22.1

* Router1 (Select 2901 Router)
  - Config
    - Interface
      - GigabitEthernet 0/0 (Ge 0/0)
        - on
        - IP Address: 192.168.44.1
        - Subnet Mask: 255.255.255.0
      - GigabitEthernet 0/1 (Ge 0/1)
        - on
        - IP Address: 192.168.55.2
        - Subnet Mask: 255.255.255.0
  - CLI
    - [Enter the following commands one by one]

```
enable
config t
router ospf 1
#network 192.168.55.0 0.0.0.255 area 0
#network 192.168.44.0 0.0.0.255 area 0
#exit
```

* Router 2 (Select 2901 Router)
  - Config
    - Interface
      - GigabitEthernet 0/0 (Ge 0/0)
        - on
        - IP address: 10.10.22.1
        - Subnet Mask: 255.0.0.0
      - GigabitEthernet 0/1 (Ge 0/1)
        - on
        - IP address: 172.16.10.2
        - Subnet Mask: 255.255.0.0
  - CLI
    - [Enter following commands one-by-one]

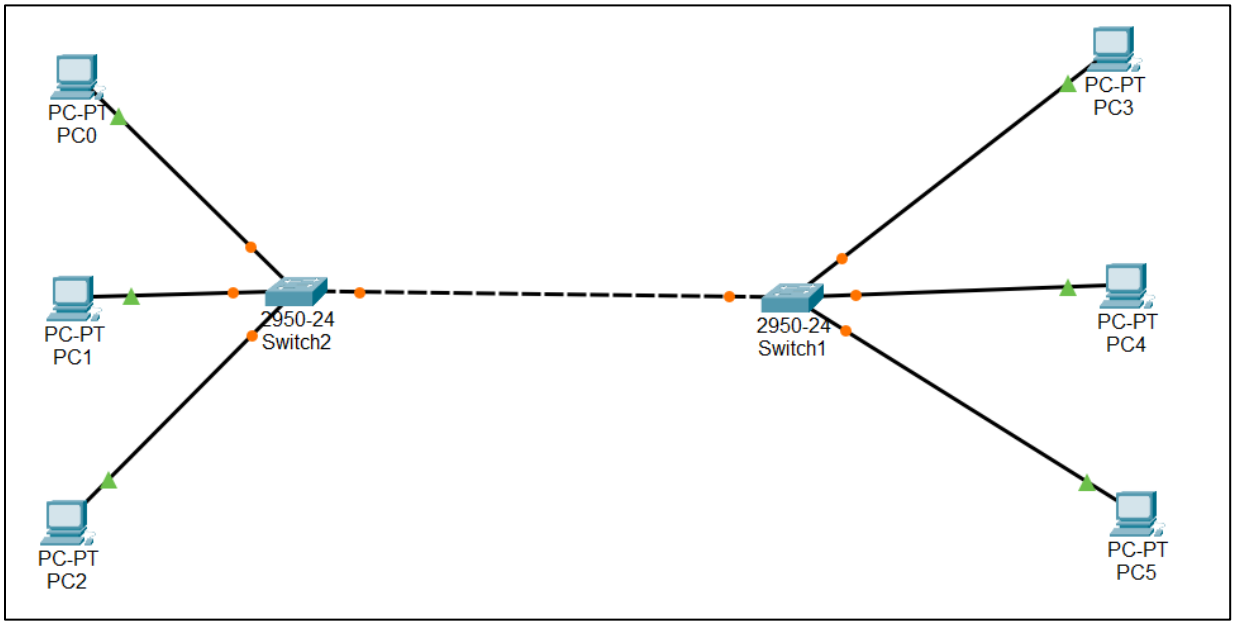
```
# enable
# config t
# router ospf 1
# network ~~192.168.55.~~ 172.16.0.0 0.0.255.255 area 0
# network 10.0.0.0 0.255.255.255 area 0
# exit
```

* Router 0 (Select 2901 Router)
  - Config
    - Interface
      - GigabitEthernet 0/0 (Ge 0/0)
        - on
        - IP address: 192.168.55.1
        - Subnet Mask: 255.255.255.0
      - GigabitEthernet 0/1 (Ge 0/1)
        - on
        - IP address: 172.16.10.1
        - Subnet Mask: 255.255.0.0
  - CLI
    - [Enter following commands one-by-one]

```
# enable
# config t
# router ospf 1
# network 192.168.55.0 0.0.0.255 area 0
# network 172.16.0.0 0.0.255.255 area 0
# exit
```

Program 8: To construct a VLAN and make the PC's communicate among a VLAN.

Network diagram:



Configuration:

14/10/25

(Virtual LAN)

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

Connections:

* PC0: (Select PC-PT)
  - Desktop
    - IP Configuration
      - IP address: 192.168.1.2
        Subnet mask: 255.255.255.0

* PC1:
  - Desktop
    - IP configuration
      - IP address: 192.168.1.3
        Subnet mask: 255.255.255.0

* PC2:
  - Desktop
    - IP configuration
      - IP address: 192.168.1.4
        Subnet mask: 255.255.255.0

* PC3:
  - Desktop
    - IP configuration
      - IP address: 192.168.1.5
        Subnet mask: 255.255.255.0

* PC4:
  - Desktop
    - IP configuration
      - IP address: 192.168.1.6
        Subnet mask: 255.255.255.0

* PC5:
  - Desktop
    - IP configuration
      - IP address: 192.168.1.7
        Subnet mask: 255.255.255.255.

VLAN 10		Vlan 20		Vlan 30	
fa0/1	PC0	fa0/2	PC1	fa0/3	PC2
fa0/2	PC3	fa0/3	PC4	fa 0/4	PC5

* Switch0:
  - CLI (Command Line Interface)
    - [type this commands one by one]

```
enable
config t
int fa 0/1
switchport access vlan 10
int fa 0/2
switchport access vlan 20
int fa 0/3
switchport access vlan 30
int fa 0/4
switchport mode trunk
exit
```

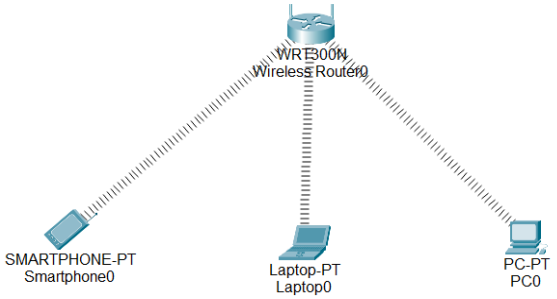
* Switch1:
  - CLI

```
enable
config t
int fa 0/2
switchport access vlan 10
int fa 0/3
switchport access vlan 20
int fa 0/4
switchport access vlan 30
int fa 0/1
switchport mode trunk
exit
```

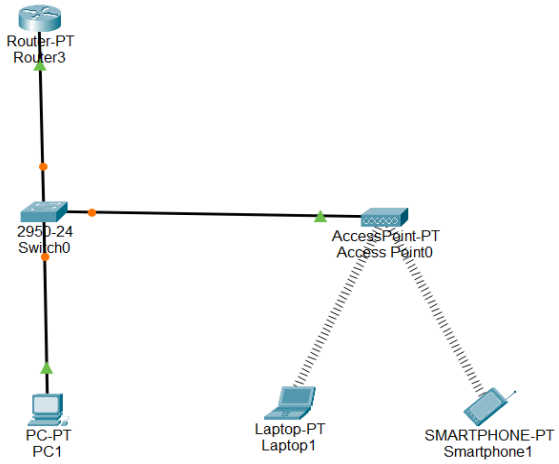
PC-PT

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

Network diagram:

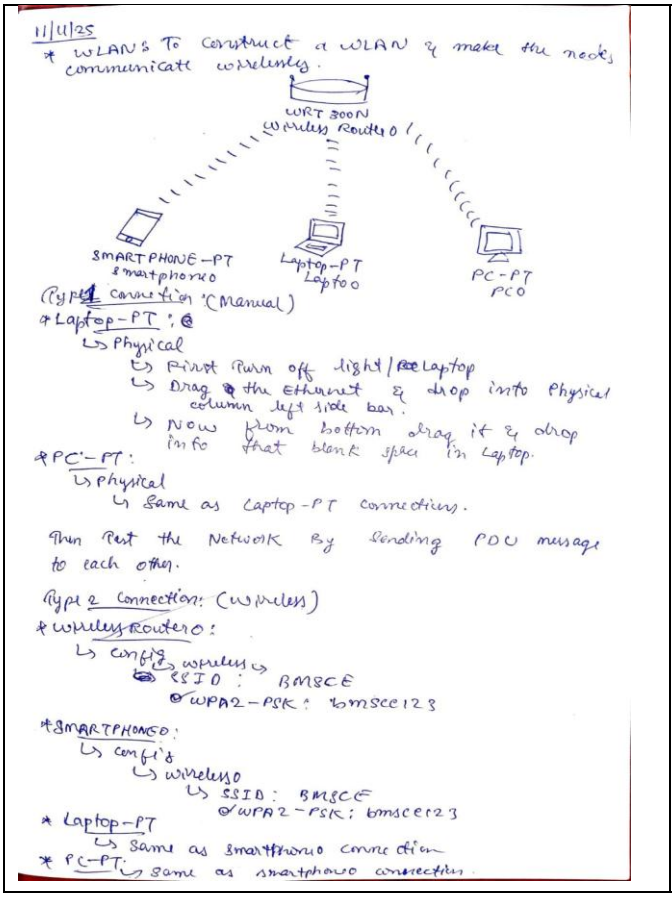


Configuration 1



Configuration 2

Configuration:


11/11/25
* WLAN's To construct a WLAN & make the nodes communicate wirelessly.

WRT 300N
Wireless Router0

SMARTPHONE-PT
Smartphone0

Laptop-PT
Laptop0

PC-PT
PC0

(Type 1 connection (Manual)

* Laptop-PT :
↳ Physical
↳ First Turn off light / Laptop
↳ Drag the Ethernet & drop into Physical column left side bar.
↳ Now from bottom drag it & drop in to that blank space in Laptop.

* PC-PT :
↳ Physical
↳ Same as Laptop-PT connections.

Then Test the Network By Sending PDU message to each other.

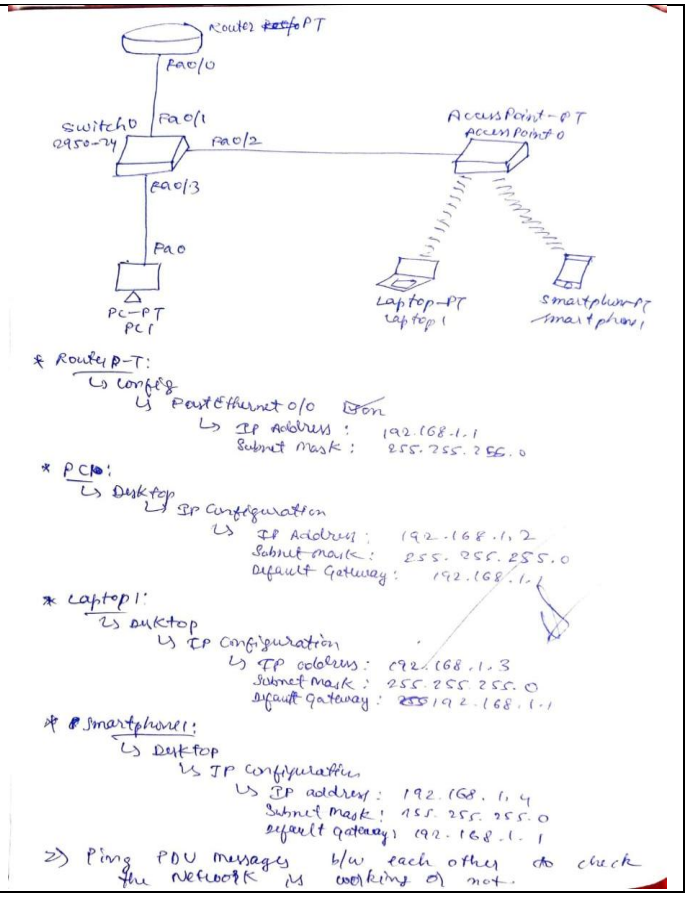
Type 2 Connection: (Wireless)

* Wireless Router0 :
↳ config ↳ wireless ↳
SSID : BMSCE
WPA2-PSK : bmscel23

* SMARTPHONE0 :
↳ config ↳ wireless0 ↳ SSID : BMSCE
WPA2-PSK : bmscel23

* Laptop-PT
↳ Same as Smartphone0 connection

* PC-PT : ↳ Same as Smartphone0 connection


Router PT
Fa0/0
Switch0
2950-24
Fa0/1
Fa0/2
Fa0/3
Access Point-PT
Access Point0
Fa0
PC-PT
PC1
Laptop-PT
Laptop1
Smartphone-PT
Smartphone1

* Router-PT:
↳ config
↳ FastEthernet 0/0
↳ IP Address : 192.168.1.1
Subnet Mask : 255.255.255.0

* PC0 :
↳ Desktop
↳ IP Configuration
↳ IP Address : 192.168.1.2
Subnet mask : 255.255.255.0
Default Gateway : 192.168.1.1

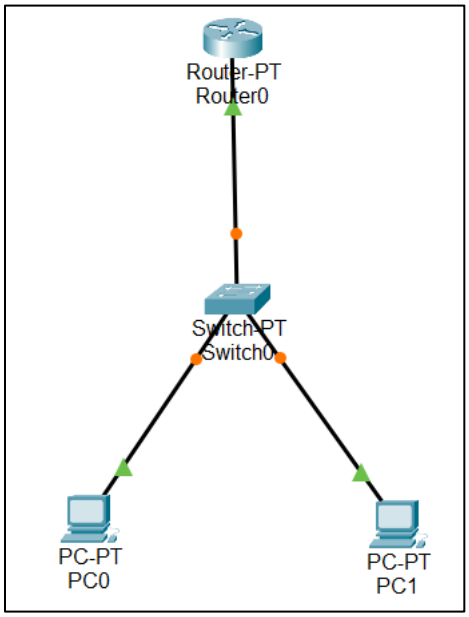
* Laptop1 :
↳ Desktop
↳ IP Configuration
↳ IP address : 192.168.1.3
Subnet Mask : 255.255.255.0
Default Gateway : 192.168.1.1

* Smartphone1 :
↳ Desktop
↳ IP Configuration
↳ IP address : 192.168.1.4
Subnet mask : 255.255.255.0
Default Gateway : 192.168.1.1

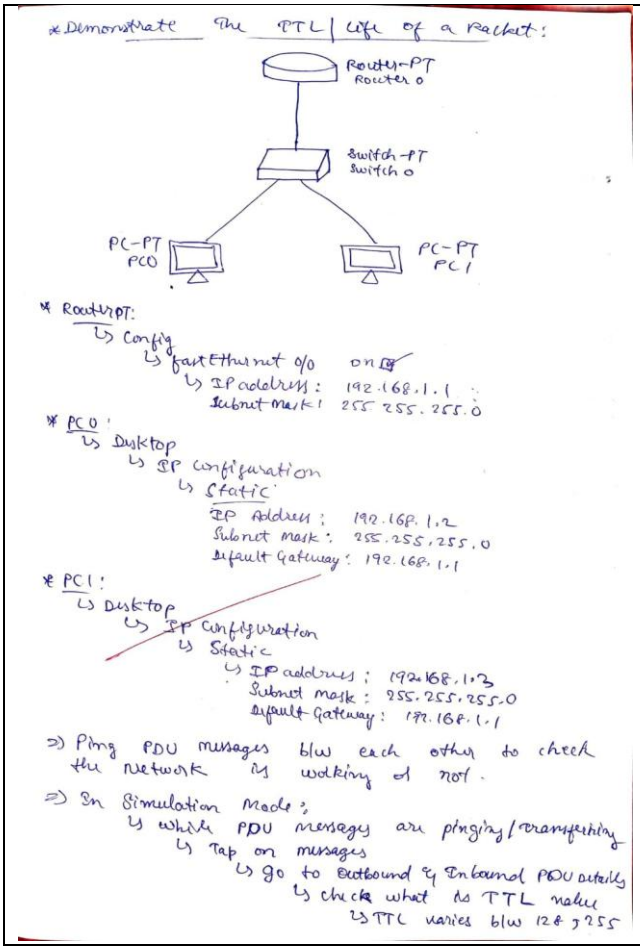
2) Ping PDU message b/w each other to check the Network is working or not.

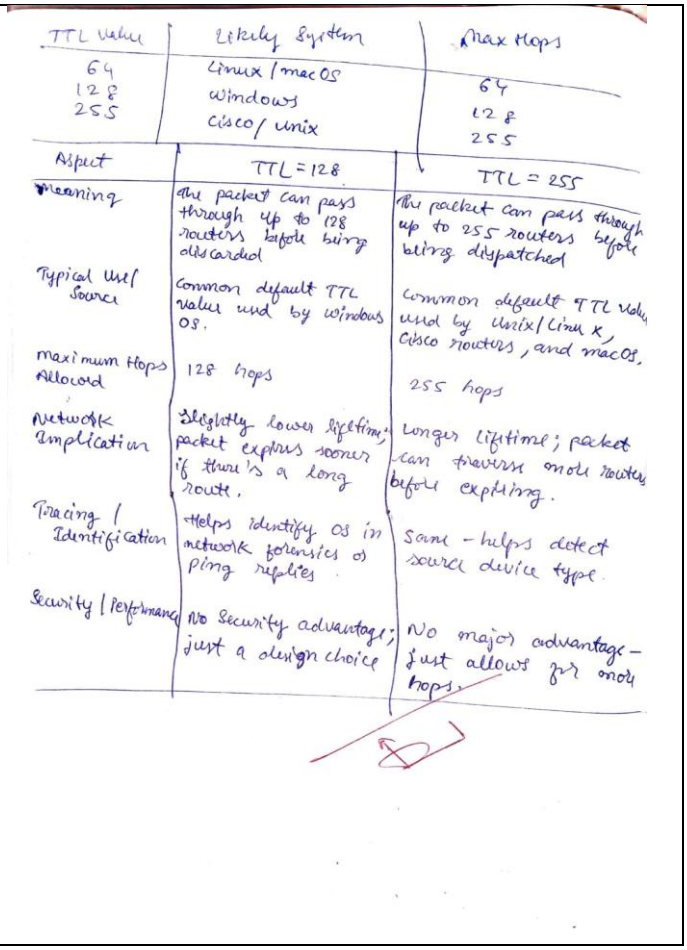
Program 10: Demonstrate the TTL/ Life of a Packet.

Network diagram:



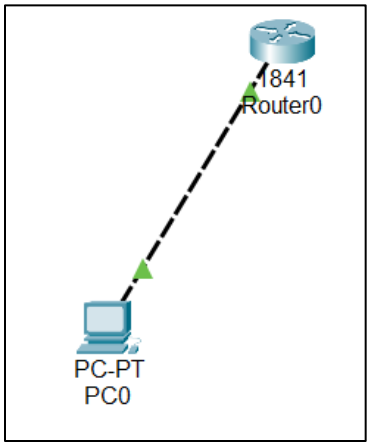
Configuration:





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

Network diagram:



Configuration:

18/4/25

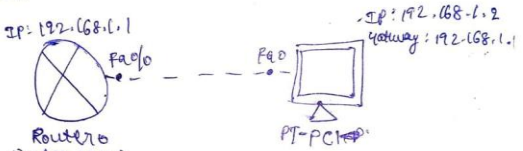
* Construct a topology to demonstrate concept of Telnet.

TELNET:

→ It is used to access remote server.

→ It's a simple command line tool that runs on your computer and it will allow you to send a command remotively to a server.

→ It is ~~also~~ also used managed devices like switches, routers ports are open or close on a server.



Configuration

* Router0:

↳ Go to CLI

↳ [yes/No]: No

↳ enter

↳ enter

↳ enable

↳ config t

↳ hostname R1

↳ enable secret ~~rp~~ rp

↳ int ~~Fa~~ Fa 0/0

↳ ip add 192.168.1.1 255.255.255.0

↳ no shutdown

↳ enter

↳ enter

↳ line vty 0 5

↳ login

↳ password ~~tp~~ tp

↳ exit

↳ exit

↳ wr

↳ show ip interface brief.

*PC:

PC > ping 192.168.1.1

PC > telnet 192.168.1.1

Trying 192.168.1.1 open

User Access verification

Password: tp

R1 > enable

Password: rp

R1# show ip interface brief

R1# en

R1# config t

R1(config)# int Fa 0/1

R1(config)# ip add 192.168.1.2 255.255.255.0

% 192.168.1.0 overlaps with FastEthernet0/0

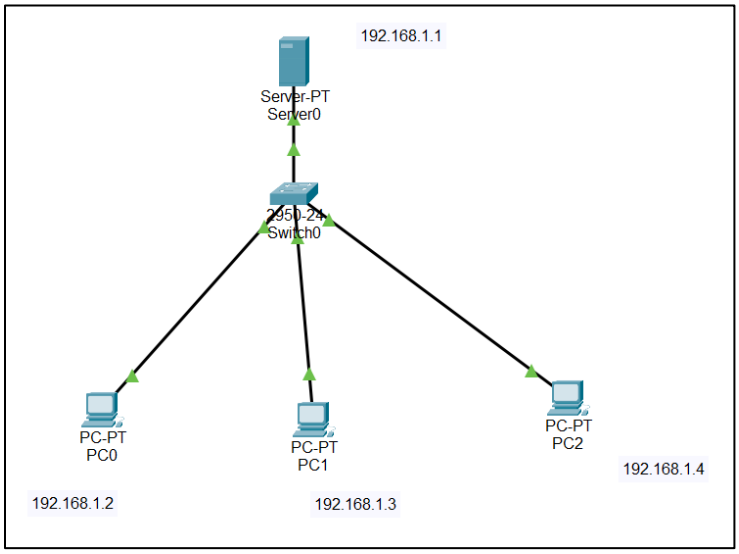
→ Then again go to Router 2

R1 > show ip interface brief

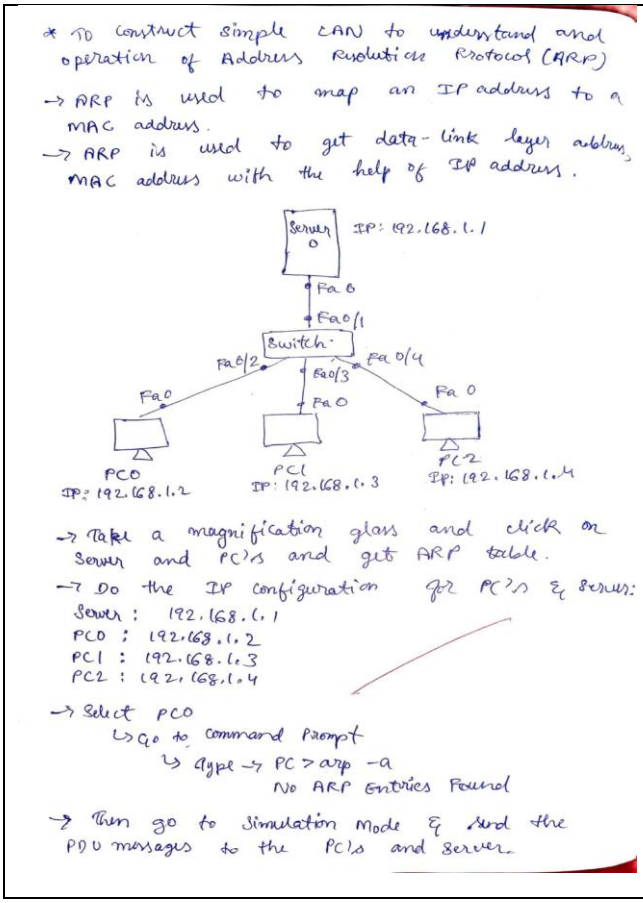
changes are made.

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

Network diagram:



Configuration:



* To construct simple LAN to understand and operation of Address Resolution Protocol (ARP)

→ ARP is used to map an IP address to a MAC address.

→ ARP is used to get data-link layer address, MAC address with the help of IP address.

→ Take a magnification glass and click on Server and PC's and get ARP table.

→ Do the IP configuration for PC's & Server:

Server : 192.168.1.1
PC0 : 192.168.1.2
PC1 : 192.168.1.3
PC2 : 192.168.1.4

→ Select PC0
↳ Go to Command Prompt
↳ Type → PC > arp -a
No ARP Entries Found

→ Then go to Simulation Mode & send the PDU messages to the PC's and Server.

→ Click on the mag and click on the outbound boundaries and check the ARP tables.

ARP Table for Server0:
IP: 192.168.1.4
Hlw add: 0060.3E88.4332
Interface: FastEthernet 0

ARP Table for PC0:
IP: 192.168.1.4
Hlw add: 0060.3E88.4332
Interface: FastEthernet 0

ARP Table for PC1:
IP: 192.168.1.4
Hlw add: 0060.3E88.4332
Interface: FastEthernet 0

ARP Table for PC2:
IP: 192.168.1.1, 192.168.1.2, 192.168.1.3
Hlw add: 0009.BD02.598E, 000C.CF4D.05BD, 0006.2A26.0033
Interface: FastEthernet 0, FastEthernet 0, FastEthernet 0

PART - B

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

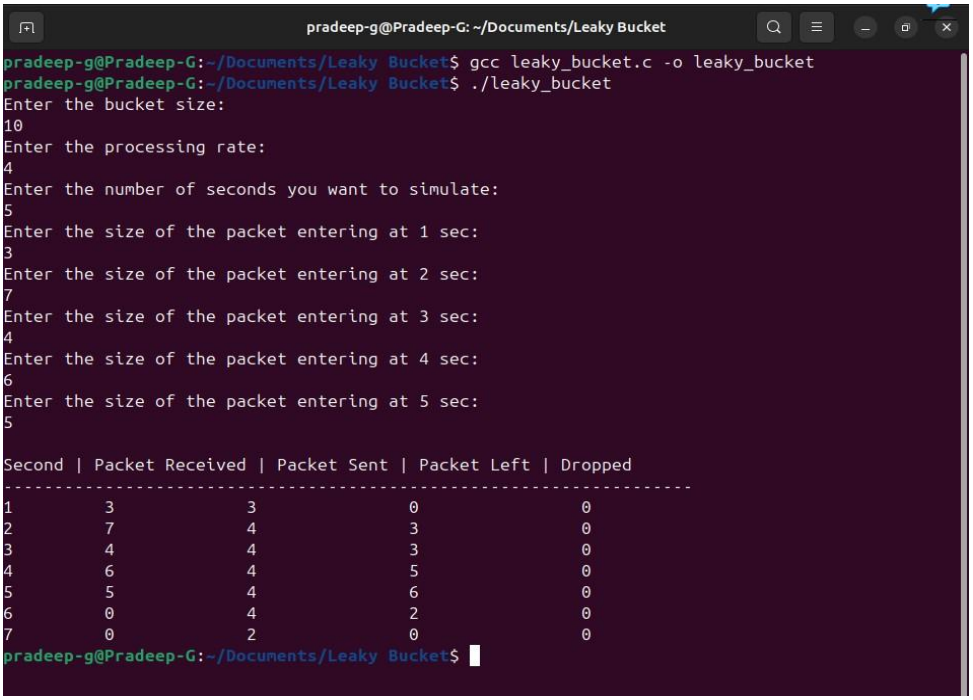
Code:

```
#include <stdio.h>


int min(int x, int y) {
    if (x < y)
        return x;
    else
        return 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("\nSecond | 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("%d\t  %d\t\t", i + 1, inp[i]);

        mini = min(count, process);
        printf("%d\t\t", mini);

        count = count - mini;
        printf("%d\t\t %d\n", count, drop);

        drop = 0;
    }

    // Remaining packets after time ends
    for (; count != 0; i++) {
        if (count > cap) {
            drop = count - cap;
```

```
            count = cap;
        }

        printf("%d\t  0\t\t", i + 1);

        mini = min(count, process);
        printf("%d\t\t", mini);

        count = count - mini;
        printf("%d\t\t %d\n", count, drop);

        drop = 0;
    }

    return 0;
}
```

Output:



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.

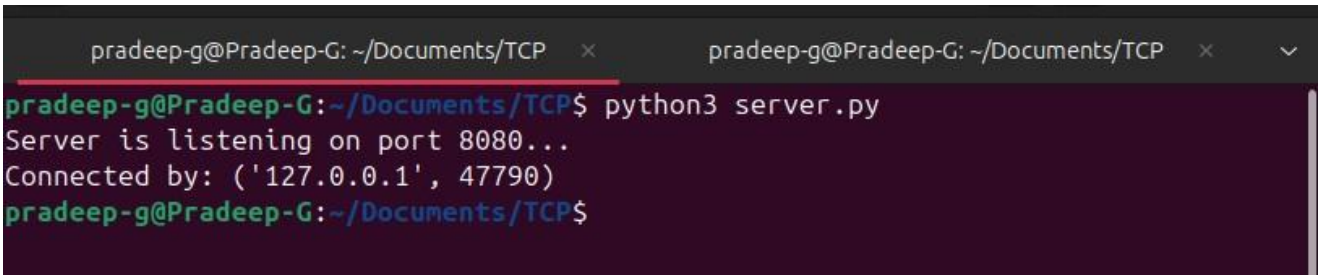
Code:

```
# tcp_client.py


import socket


# Step 1: Create TCP socket

client_socket =
socket.socket(socket.AF_INET,
socket.SOCK_STREAM)


# Step 2: Connect to server

client_socket.connect(('localhost',
8080))


# Step 3: Send filename

filename = input("Enter filename to
request: ")


client_socket.send(filename.encode())


# Step 4: Receive file contents

data =
client_socket.recv(4096).decode()


print("\n--- File Content ---\n")

print(data)


# Step 5: Close connection

client_socket.close()
```

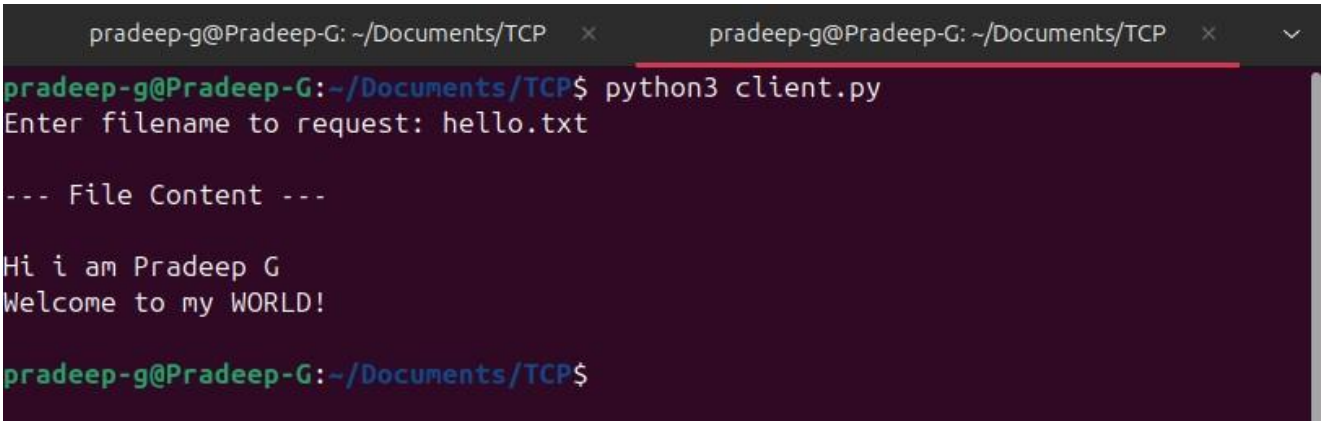
```
# tcp_server.py

import socket

# Step 1: Create a TCP socket
server_socket =
socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

# Step 2: Bind to address and port
server_socket.bind(('localhost',
8080))

# Step 3: Listen for client
connections
server_socket.listen(1)
print("Server is listening on port
8080...")

# Step 4: Accept connection
conn, addr = server_socket.accept()
print("Connected by:", addr)

# Step 5: Receive file name
filename =
conn.recv(1024).decode().strip()

try:
    # Step 6: Open and read file
    with open(filename, 'r') as f:
        data = f.read()

    conn.send(data.encode())  # Send
file contents

except FileNotFoundError:
    conn.send(b"File not found on
server.")

# Step 7: Close connection
conn.close()
server_socket.close()
```

Output:

Server side Terminal:



Client side Terminal:



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.

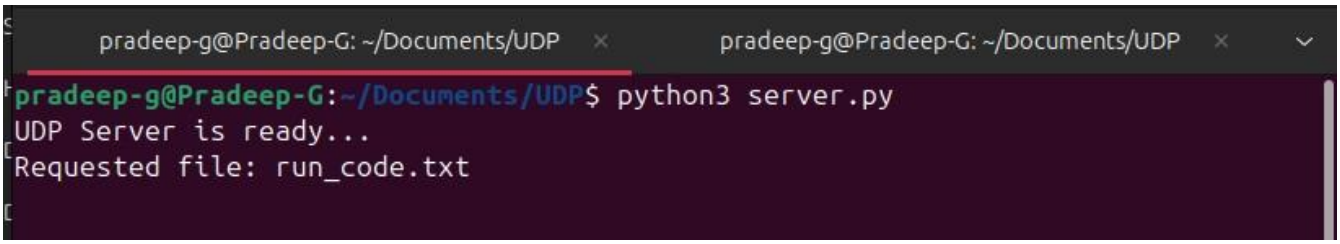
Code:

```
# udp_client.py


import socket


# Step 1: Create UDP socket

client_socket =
socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)


server_address = ('localhost',
8081)


filename = input("Enter filename
to request: ")


# Step 2: Send filename to
server

client_socket.sendto(filename.en
code(), server_address)


# Step 3: Receive response

data, addr =
client_socket.recvfrom(4096)


print("\n--- File Content ---
\n")

print(data.decode())


# Step 4: Close socket

client_socket.close()
```

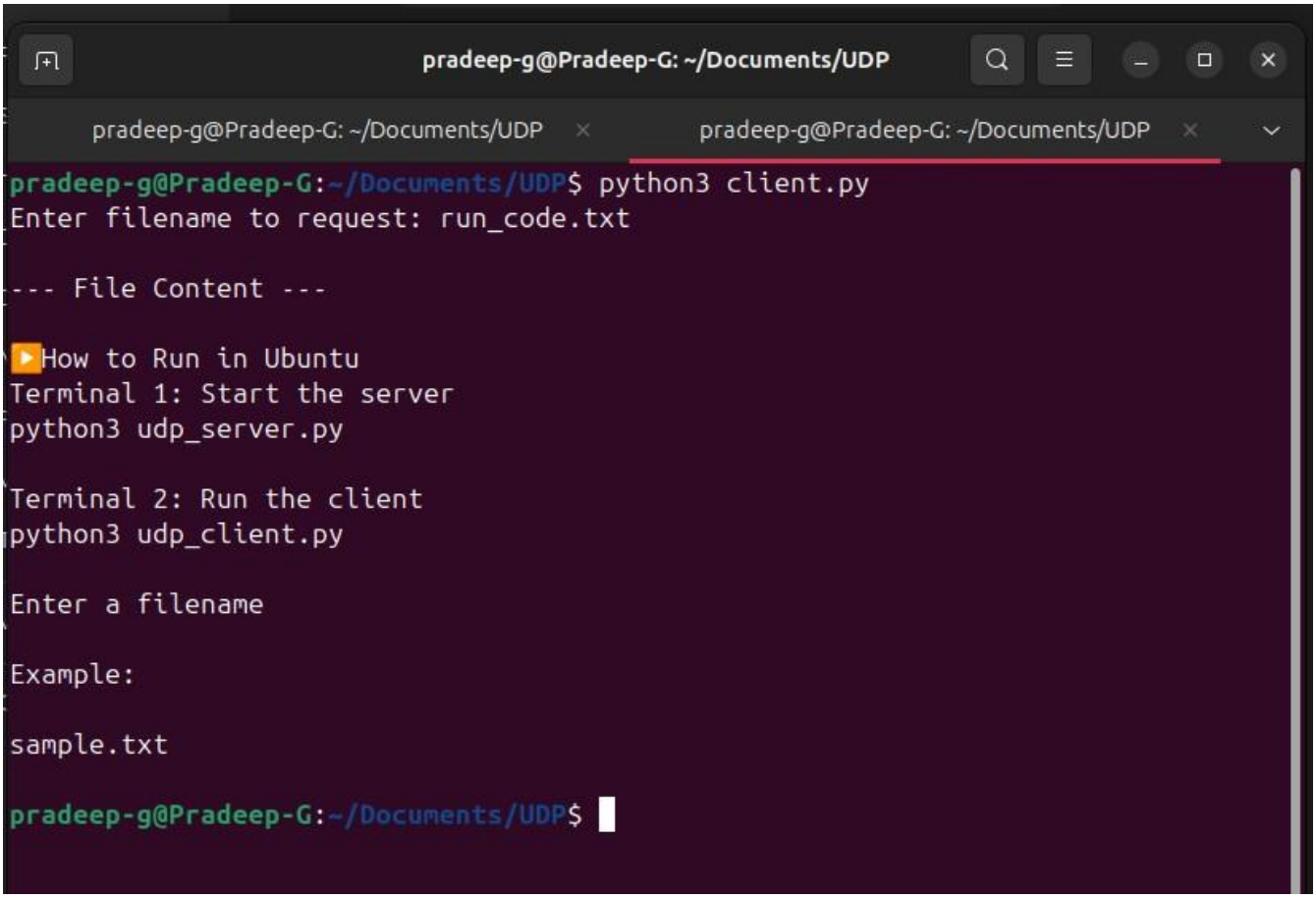
```
# udp_server.py

import socket

# Step 1: Create UDP socket
server_socket =
socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)

# Step 2: Bind to address and port
server_socket.bind(('localhost',
8081))

print("UDP Server is ready...")

while True:
    # Step 3: Receive filename
from client
    filename, addr =
server_socket.recvfrom(1024)
    filename =
filename.decode().strip()

    print(f"Requested file:
{filename}")

    try:
        # Step 4: Open file and
send content
        with open(filename, 'r')
as f:
            data = f.read()

        server_socket.sendto(data.
encode(), addr)

    except FileNotFoundError:
        server_socket.sendto(b"Fil
e not found on server.", addr)
```

Output:

Server side Terminal:



Client side Terminal:



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

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 generation polynomial:\n");
    gets(gen);
    printf("Generator polynomial is CRC-CCITT: %s\n", gen);

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

    printf("Enter the message:\n");
    n = 0;
    while ((c = getchar()) != '\n') {
        msj[n] = c;
        n++;
    }
    msj[n] = '\0';

    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);

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++;
        }
    }
}

for (i = 0; i < k; i++)
    rem[i] = a[n + i];
rem[k] = '\0';

printf("Checksum (remainder):\n");
puts(rem);

printf("\nMessage with checksum appended:\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);

n = 0;
printf("Enter the received message:\n");
while ((c = getchar()) != '\n') {
    s[n] = c;
    n++;
}
s[n] = '\0';

for (i = 0; i < n; 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 + i];
rem[k] = '\0';

for (i = 0; i < k; i++) {
    if (rem[i] == '1')
        flag = 1;
```

```
    }

    if (flag == 0)
        printf("Received polynomial is error-free \n");
    else
        printf("Received polynomial contains error \n");

    return 0;
}
```

Output: