VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

HARISHA G (1BM20CS054)

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
October-2022 to Feb-2023

CERTIFICATE

This is to certify that the Lab work entitled "**COMPUTER NETWORKS**" carried out by **HARISHA G (1BM20CS054),** 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 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks- (20CS5PCCON)** work prescribed for the said degree.

Rekha G S   **Dr. Jyothi S Nayak**
Assistant Professor   Professor and Head
Department of CSE   Department of CSE
BMSCE, Bengaluru   BMSCE, Bengaluru
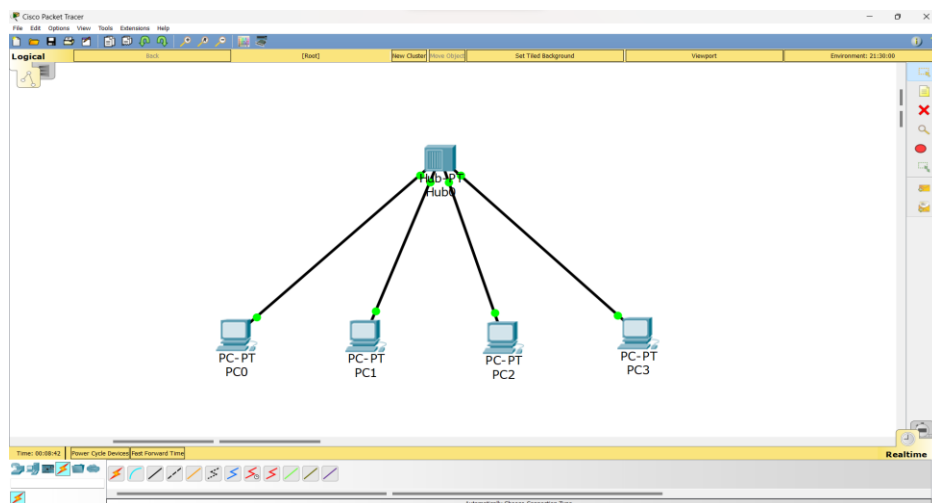
`

Index

Cycle-1

Experiment No 1

Aim of the program

Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.

Hub
Topology

Procedure

Output

Switch

Topology

Procedure

Output


```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=2ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms 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 = 0ms, Maximum = 2ms, Average = 0ms

C:\>
```

Experiment No 2

Aim of the program

Configuring IP address to Routers in Packet Tracer. Exploring the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.
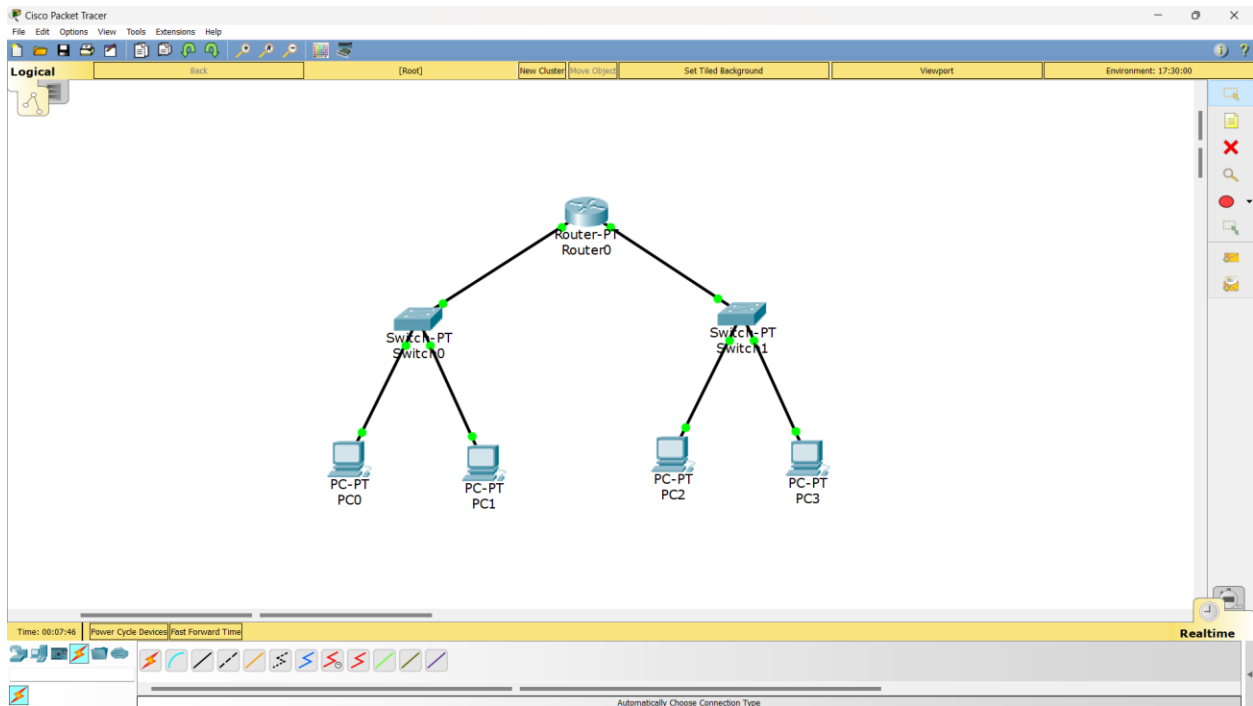
Topology

Procedure

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shutdown

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

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

Router(config-if)#exit
Router(config)#
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up

Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
```

Aim: To configure IP address to Routers in Packet tracer. Explore following messages: Ping response, Destination un reachable, Request timed out, Reply.

* Topology:

Router - 0
10.0.0.2 Fa0/0   20.0.0.2 Fa1/0
copper cross

PC - 0
10.0.0.1

PC - 1
20.0.0.1

* Procedure:
- select two end devices & configure ip address (different network)
- select a router & connect these devices as shown above
- configure the ip for router as per the commands
- configure the gateway for a device (gateway = ip address of router for same network)
- send a simple PDU b/w 2 devices.
  ping

* Commands: (In router CLI)
⮡ Router> enable
  > Router # configure terminal
        config t
  > Router(config)# interface Fa 0/0
  > Router(config-ip)# ip address [ip] [subnet mask]
  > Router(config-ip)# no shutdown
  > exit

* Results:
  → The PDU is sent & received appropriately.

* Observation:
  → when the device is disconnected from the router until it is ip-configed.
  → Request timeout b/w 2 devices occur when the network are isolated to connect the network, also configure the gateways

* Ping: (Request timed out)
  Ping statistics for 20.0.0.2
  Packets: Sent = 4, Received = 0, Lost = 4 (100% lost)

Output


```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.1:
    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:

Request timed out.
Reply from 20.0.0.1: bytes=32 time<1ms TTL=127
Reply from 20.0.0.1: bytes=32 time<1ms TTL=127
Reply from 20.0.0.1: bytes=32 time<1ms TTL=127

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```
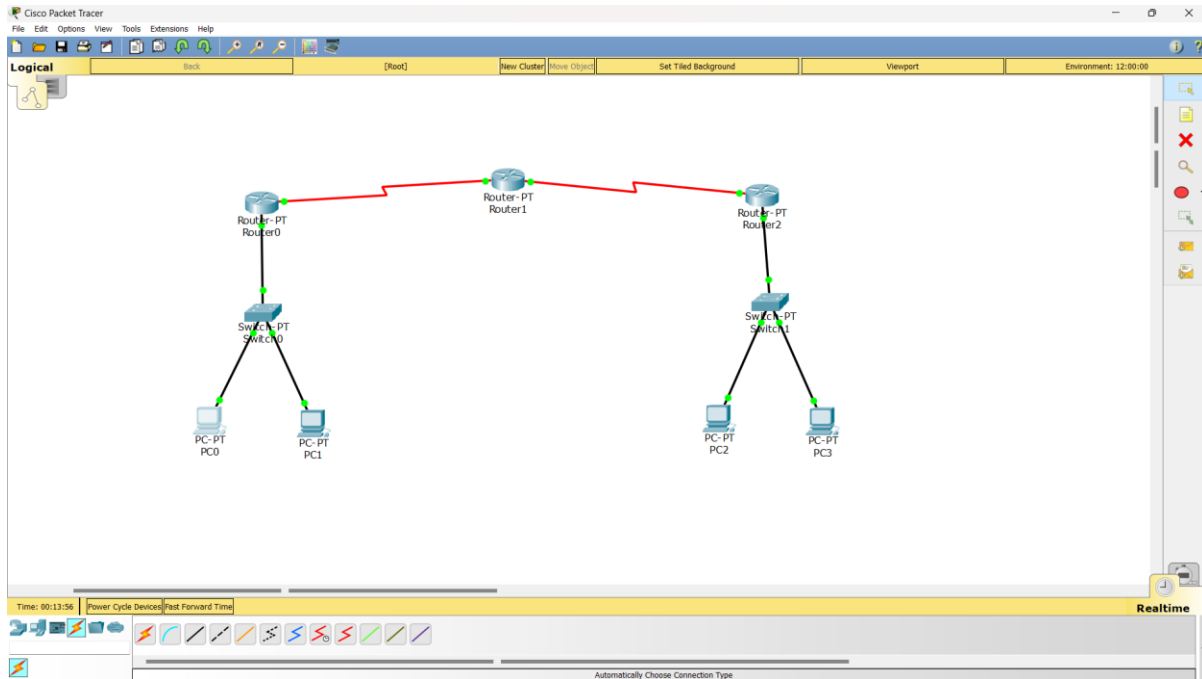
Experiment No 3

Aim of the program

Configuring default route to the Router

Topology

Procedure

```
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    20.0.0.0/8 is directly connected, Serial2/0

Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 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
```
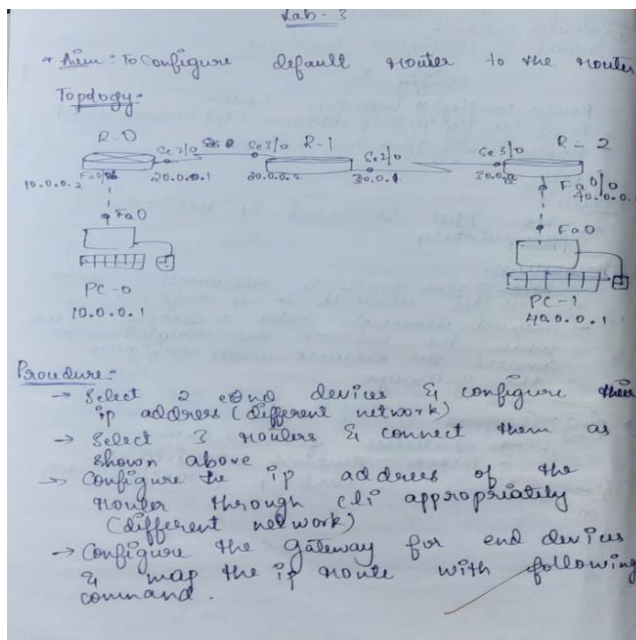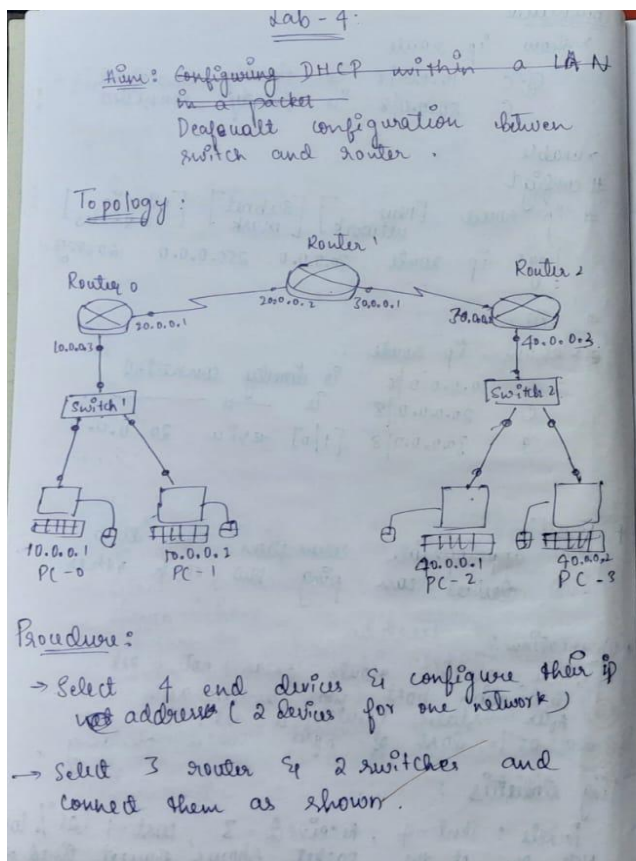
Lab - 3

* **Aim:** To Configure default router to the router

Topology:

Procedure:
→ Select 2 end devices & configure their ip address (different network)
→ Select 3 routers & connect them as shown above
→ Configure the ip address of the router through CLI appropriately (different network)
→ Configure the gateway for end devices & map the ip route with following command.

Command
> show ip route
  eg: C 10.0.0.0|8 is directly connected
      C 20.0.0|8 is directly connected

> enable
config t
ip route [new network] [subnet mask] [via ip address]
  [eg: ip route 20.0.0.0 255.0.0.0 20.0.0.9]

exit
> show ip route
  C 10.0.0.0|8 is directly connected
  C 20.0.0.0|8 is directly connected
  S 30.0.0.0|8 [1|0] via 20.0.0.2

* **Result:**
After appropriate connection, the two end devices can ping to each other

* **Observation:**
① Since static route was not set, destination host was unreachable
② After static route is set, there is a 25% loss of bytes when pinging

Ping statistics:
Packets : Sent = 4, Received = 3, Lost = 1 (25%)
          packet shows request timed...

Lab - 4:

Aim: ~~Configuring DHCP within a LAN in a packet~~
Default configuration between switch and router.

Topology:

Procedure:
→ Select 4 end devices & configure their ip addresses (2 devices for one network)
→ Select 3 router & 2 switches and connect them as shown.
→ Configure the ip address of routers through CLI appropriately (different networks)
→ Configure the gateway for end devices & map the ip route with following command.

Commands:
Case - 1: For Router with a switch connection
[Router - 0 & Router - 2]

> show ip route
  C 10.0.0.0|8 is directly connected.
  C 20.0.0.0|8 is directly connected.

> enable
config t
ip route 0.0.0.0 0.0.0.0 [via ip address]
  [eg: ip route 0.0.0.0 0.0.0.0 {20.0.0.2]

exit
> show ip route
  C 10.0.0.0|8 is directly connected
  C 20.0.0.0|8 "
  S* 0.0.0.0|0 [1|0] via 20.0.0.2

Case - 2: For Router with Router connection [Router - 1]
> show ip route:

Output

```
Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 40.0.0.1

Pinging 40.0.0.1 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 40.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 10ms, Average = 10ms

C:\>
```
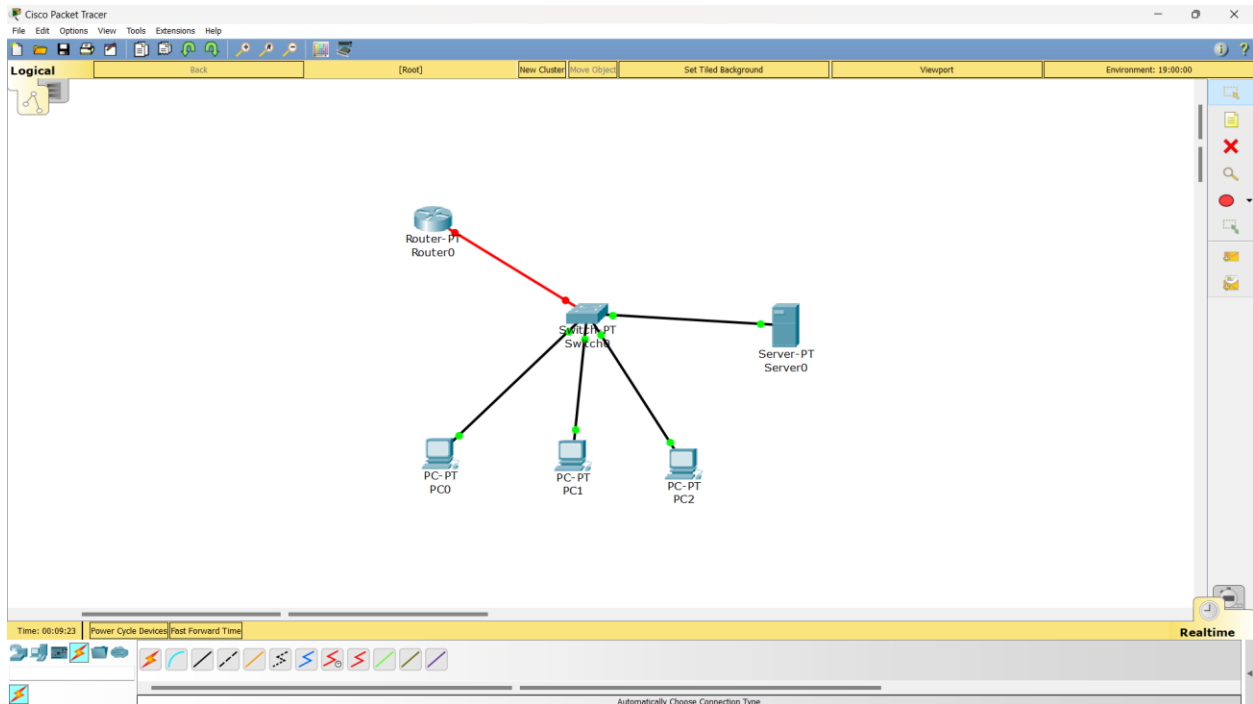
Experiment No 4

Aim of the program

Configuring DHCP within a LAN in a packet Tracer

Topology

Procedure

Aim: Configuring DHCP within a LAN

Topology:

10.0.0.1

Router-PT

10.0.0.2

Switch PT

Server PT

PC-0   PC-1   PC-2

Should be Dynamically allocated.
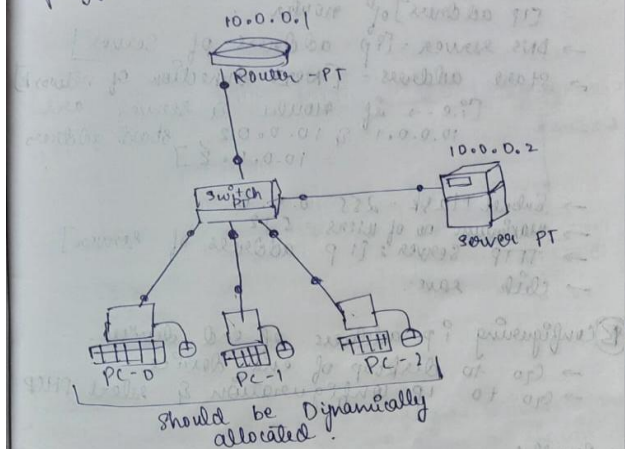
Procedure:

→ Select 3 end devices, do not assign ip address to them

→ Select a switch, a server & a router & connect them as shown.

→ Configure the ip address for the server (Static) & the Router (through cli) for same network.

A) Configuring DHCP of server.

→ Go to services in server

→ Select DHCP & turn ON the service

→ Set the Default Gateway as the [ip address] of router.

→ DNS server = [ip address of server]

→ Start address = [Next connection of network [i.e., if router & server are 10.0.0.1 & 10.0.0.2, start address = 10.0.0.3]

→ Subnet Mask = 255.0.0.0

→ Maximum no of users = 512

→ TFTP Server: [ip address of server]

→ Click save.

B) Configuring ip address of end devices.

→ Go to desktop of end device

→ Go to ip configuration & select DHCP

Result:

The ip addresses of end devices are connected allotted dynamically.

Observations:

DHCP (Dynamic Host Configuration Protocol) is a client/server protocol that automatically provides an ip (internet protocol) with its ip address network as base & other related configuration. DHCP application is DORA (Discover, offer, Request, Acknowledgement)

Output


```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.6

Pinging 10.0.0.6 with 32 bytes of data:

Reply from 10.0.0.6: bytes=32 time=1ms TTL=128
Reply from 10.0.0.6: bytes=32 time<1ms TTL=128
Reply from 10.0.0.6: bytes=32 time<1ms TTL=128
Reply from 10.0.0.6: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```
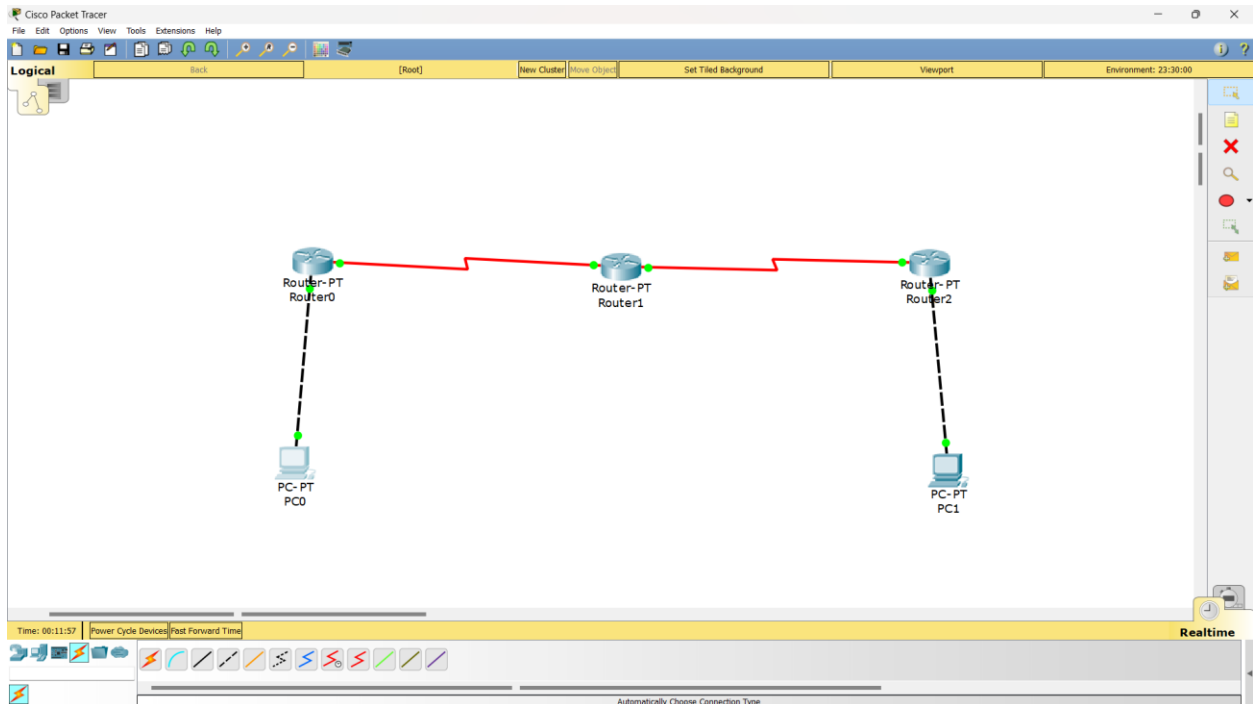
Experiment No 5

Aim of the program

Configuring RIP Routing Protocol in Routers

Topology

Procedure

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shutdown

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

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

Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#no shutdown

Router(config-if)#
```

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
This command applies only to DCE interfaces
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
Router(config-if)#exit
Router(config)#interface serial3/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
```
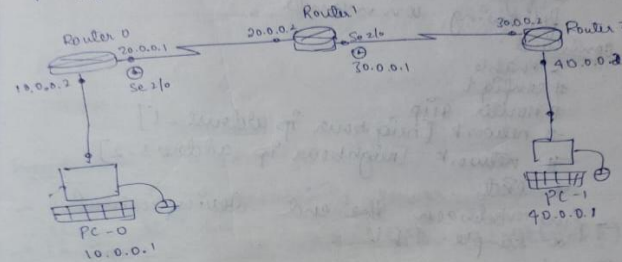
Lab – 6.

Aim: Configuring RIP routing Protocol in Routers.

Topology:

Procedure:
→ Select 2 end devices & configure their ip address
→ Select 3 routers & connect them as shown above. Configure the Fast ethernet with appropriate cli commands & gateway.
→ For the Serial ports, to configure them with the following commands

Commands:
> enable
config t
interface se[port]
ip address [ip] [subnet mask]
encapsulation PPP
clock rate 64000 [Only for serial ports with clock (&) symbol]

#no shutdown
exit.
→ Configure the RIP for routers with following commands
Command:
> enable
config t
router rip
network [neighbour ip address - 1]
network [neighbour ip address - 2]
→ exit.
→ ping between the end device & send a simple PDU.

Result:
The devices are able to ping each other a simple PDU is passed through.

Observation:
RIP is a dynamic routing protocol that uses hop count as a routing metric to find the best path between source & destination network routing. The end devices are able to ping with each other [but the first ping had 85% loss]

Ping statistics:
Packets sent = 4, Received = 4, Cost - 0 6% loss
Minimum = 4ms, Maximum 12ms, Average = 5ms

12

Output:

```
C:\>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125
Reply from 40.0.0.1: bytes=32 time=3ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 4ms, Average = 3ms

C:\>
```
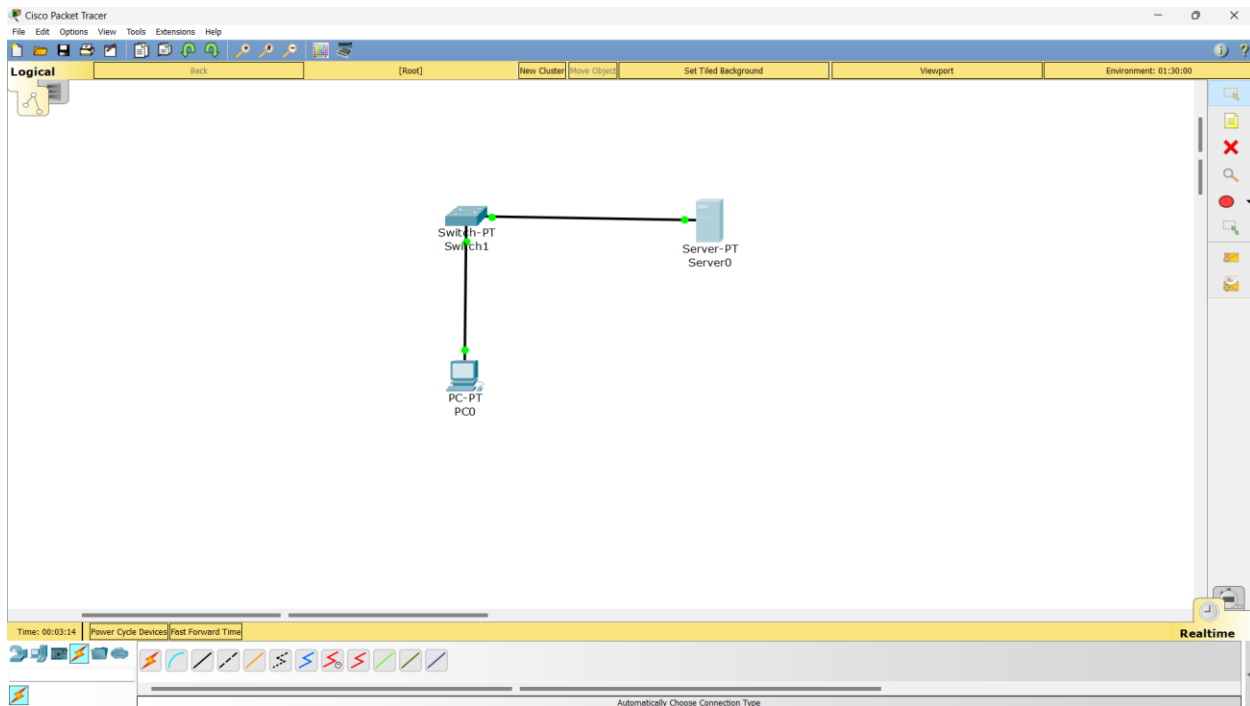
Experiment No 6

Aim of the program

Demonstration of WEB server and DNS using Packet Tracer

Topology
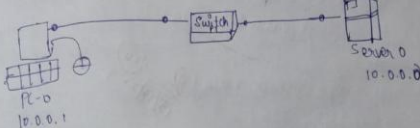
Procedure

lab - 6

Aim: Demonstration of WEB server &
DNS.

Topology:

Procedure:
→ Select a end device & server & connect
  them through a switch as shown.
→ Configure the ip address (same network)
→ Go to service in server
→ enable HTTP & DNS.
→ Add a domain name [eg. www.xyz.com]
  Address = [Server ip] [eg.10.0.0.2]
→ Add & save.
→ Go to web browser in end device.
  & search for the added Domain
  name.
→ Try to search for Domain name/.html

Result:
The added domain name is accessable

Observation:
Consider an HTTP file name helloworld
in server. When going through
www.xyz.com/helloworld.html in web
browser, the changes done to helloworld
html is reflected in the browser

Output
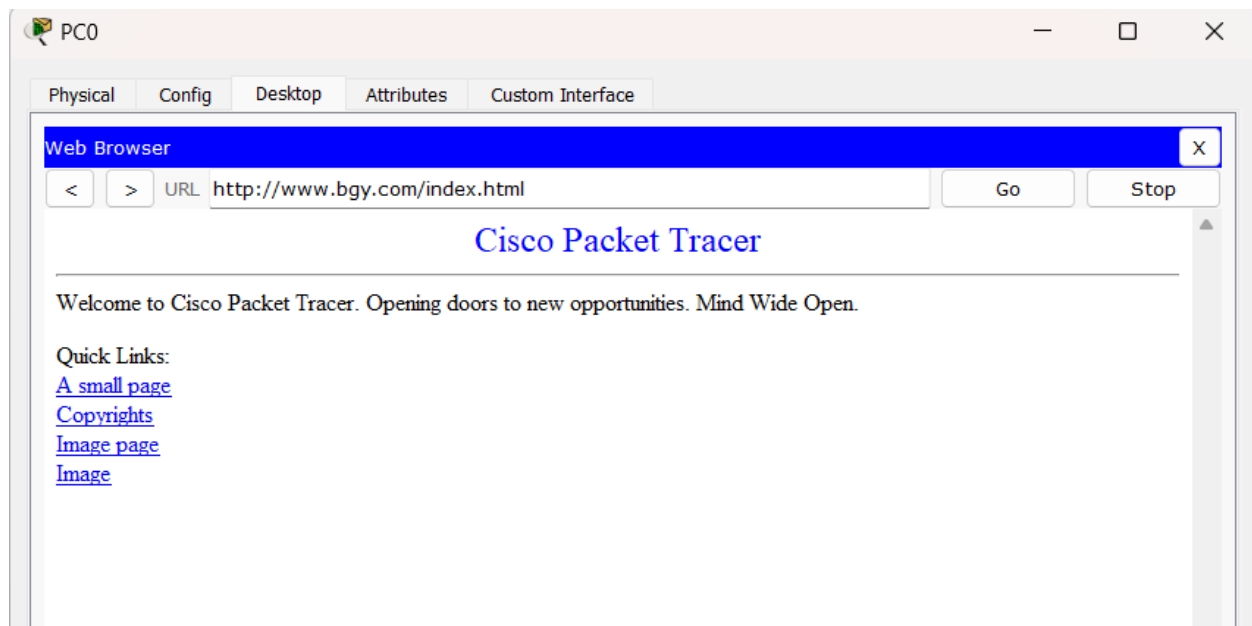
Experiment No 1

Aim of the Experiment

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

Code

```cpp
#include<bits/stdc++.h>

using namespace std;

void receiver(string data, string key);


string xor1(string a, string b)
{

        string result = "";

        int n = b.length();

        for(int i = 1; i < n; i++)
        {
                if (a[i] == b[i])
                        result += "0";
                else
                        result += "1";
        }
        return result;

}


string mod2div(string dividend, string divisor)
{
```

```cpp
        int pick = divisor.length();

        string tmp = dividend.substr(0, pick);

        int n = dividend.length();

        while (pick < n)
        {
                if (tmp[0] == '1')
                        tmp = xor1(divisor, tmp) + dividend[pick];
                else
                        tmp = xor1(std::string(pick, '0'), tmp) +
                                dividend[pick];

                pick += 1;
        }
        if (tmp[0] == '1')
                tmp = xor1(divisor, tmp);
        else
                tmp = xor1(std::string(pick, '0'), tmp);

        return tmp;
}

void encodeData(string data, string key)
{
        int l_key = key.length();
```

```cpp
        string appended_data = (data +std::string(l_key - 1, '0'));


        string remainder = mod2div(appended_data, key);


        string codeword = data + remainder;
        cout << "Remainder : "
                << remainder << "\n";
        cout << "Encoded Data (Data + Remainder) :"
                << codeword << "\n";
        receiver(codeword, key);
}
void receiver(string data, string key)
{
        string currxor = mod2div(data.substr(0, key.size()), key);
        int curr = key.size();
        while (curr != data.size())
        {
                if (currxor.size() != key.size())
                {
                        currxor.push_back(data[curr++]);
                }
                else
                {
                        currxor = mod2div(currxor, key);
                }
        }
        if (currxor.size() == key.size())
        {
```

```cpp
            currxor = mod2div(currxor, key);

        }
        if (currxor.find('1') != string::npos)

        {
                cout << "there is some error in data" << endl;

        }
        else

        {
                cout << "correct message recieved" << endl;

        }
}
int main()
{

        string data = "1011101";
        string key = "100010000001";


        encodeData(data, key);


        return 0;
}
```

Observation:
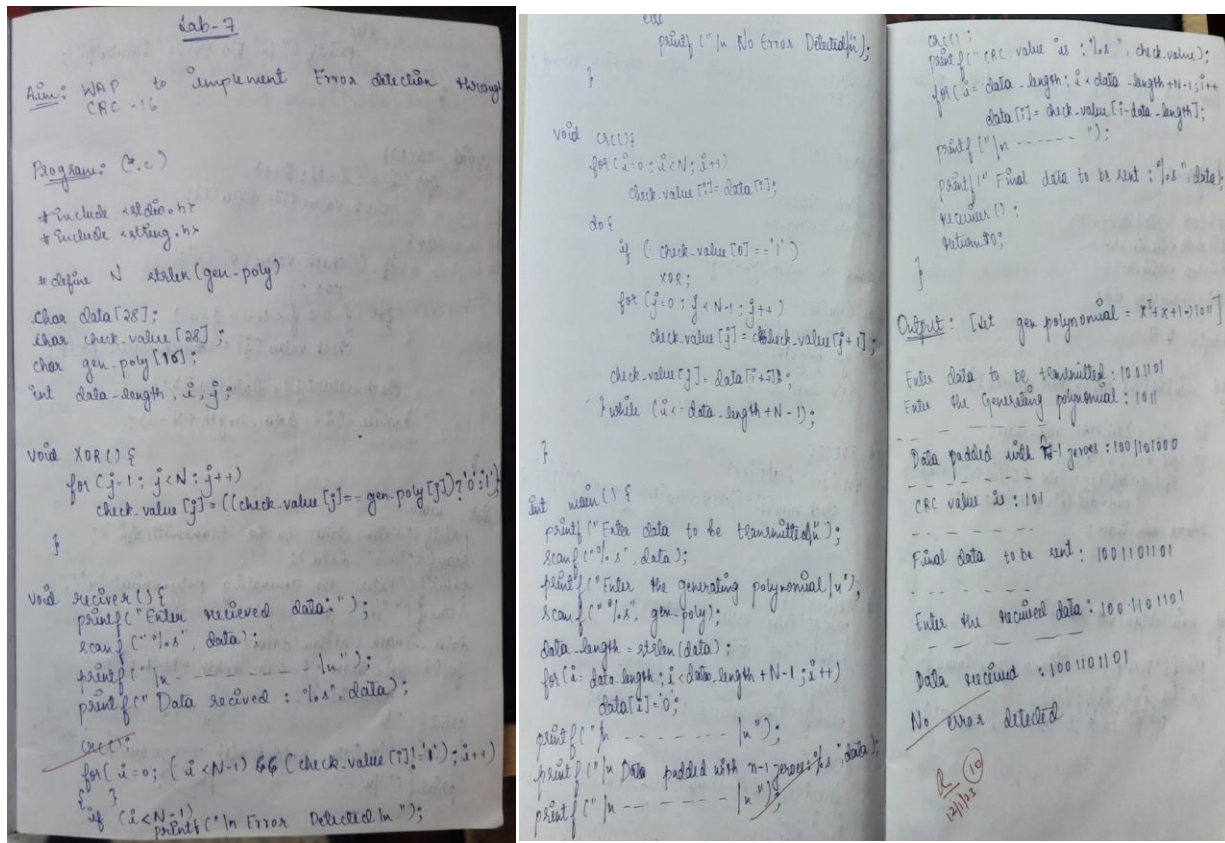
Output

```
Remainder : 10001011000
Encoded Data (Data + Remainder) :101110110001011000
correct message recieved


...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No 2

Aim of the Experiment

Write a program for distance vector algorithm to find suitable path for transmission.

Code

```
#include<stdio.h>


#define INF  99999
#define n 5


void printSolution(int g[n])
{
   printf("Hop count       : ");
   for(int j=0;j<n;j++)
   {
      if(g[j] == INF)
         printf("INF\t");
      else
         printf("%d\t",g[j]);
   }
   printf("\n");
}


void findShortestPath(int dist[][n])
{


   for(int k=0;k<n;k++)
   {
      for(int i=0;i<n;i++)
```

```c
    {
        for(int j=0;j<n;j++)
        {
            if(dist[i][j] > dist[i][k] + dist[k][j]
            &&(dist[i][k] != INF && dist[k][j] != INF))
            {
                dist[i][j] = dist[i][k] + dist[k][j];
            }
        }
    }
}

    char c = 'A';
    for(int i=0; i<n; i++ )
    {
        printf("Router table entries for router %c:\n", c);
        printf("Destination router: A\tB\tC\tD\tE\n");
        printSolution(dist[i]);
        c++;
    }
}

int main()
{
    int graph[][n] = { {0,   1,  1, INF, INF},
            {1,  0,  INF,  INF, INF},
            {1, INF, 0, 1,  1},
            {INF, INF, 1, 0, INF},
```

```
        {INF, INF, 1, INF, 0}};


    findShortestPath(graph);

    return 0;

}
```

Observation:

Output:

```
Router table entries for router A:
Destination router: A    B         C         D         E
Hop count          : 0   1         1         2         2
Router table entries for router B:
Destination router: A    B         C         D         E
Hop count          : 1   0         2         3         3
Router table entries for router C:
Destination router: A    B         C         D         E
Hop count          : 1   2         0         1         1
Router table entries for router D:
Destination router: A    B         C         D         E
Hop count          : 2   3         1         0         2
Router table entries for router E:
Destination router: A    B         C         D         E
Hop count          : 2   3         1         2         0


...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No 3

Aim of the Experiment

Implement Dijkstra's algorithm to compute the shortest path for a given topology.

Code

```c
#include <stdio.h>

#include <stdlib.h>



void dijkstra(int graph[10][10],int V)

{

    int distance[V], predefine[V], visited[V];

    int startnode, count, min_distance, nextnode, i, j;

    printf("\nEnter the start node: ");

    scanf("%d", &startnode);

    for(i=0; i<V; i++) {

        distance[i] = graph[startnode][i];

        predefine[i] = startnode;

        visited[i] = 0;

    }

    distance[startnode] = 0;

    visited[startnode] = 1;

    count = 1;

    while(count<V-1) {

        min_distance = 99;

        for(i=0; i<V; i++) {

            if(distance[i] < min_distance && visited[i]==0)

            {

                min_distance = distance[i];
```

```c
        nextnode = i;
     }
  }
  visited[nextnode] = 1;
  for(i=0;i<V;i++)
  {
     if(visited[i] == 0)
     {
        if((min_distance + graph[nextnode][i]) < distance[i])
        {
           distance[i] = min_distance + graph[nextnode][i];
           predefine[i] = nextnode;
        }
     }
  }
  count = count + 1;
}
for(i=0;i<V;i++) {
  if(i!=startnode) {
     printf("\nDistance of node %d = %d", i, distance[i]);
     printf("\nPath = %d",i);
     j = i;
     do
     {
        j = predefine[j];
        printf(" <- %d",j);
     } while (j != startnode);
  }
```

```c
    }
}
int main()
{
    int i, j;
    int V;
    printf("Enter the number of vertices: ");
    scanf("%d", &V);
    int graph[V][V];
    printf("\nEnter the cost/weight matrix: \n");
    for(i=0; i<V; i++) {
        for(j=0;j<V;j++) {
            scanf("%d", &graph[i][j]);
        }
    }
    dijkstra(graph, V);
    return 0;
}
```

Observation:

Output:

```
Enter the number of vertices: 5

Enter the cost/weight matrix:
0 10 99 5 7
10 0 1 2 99
99 1 0 9 4
5 2 9 0 99
7 99 4 99 0

Enter the start node: 0

Distance of node 1 = 5
Path = 1 <- 4 <- 3 <- 0
Distance of node 2 = 5
Path = 2 <- 4 <- 3 <- 0
Distance of node 3 = 5
Path = 3 <- 0
Distance of node 4 = 5
Path = 4 <- 3 <- 0

...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No 4

Aim of the Experiment

Write a program for congestion control using Leaky bucket algorithm

Code

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

```
            storage -= output_pkt_size;

    }

    return 0;}
```

Observation:
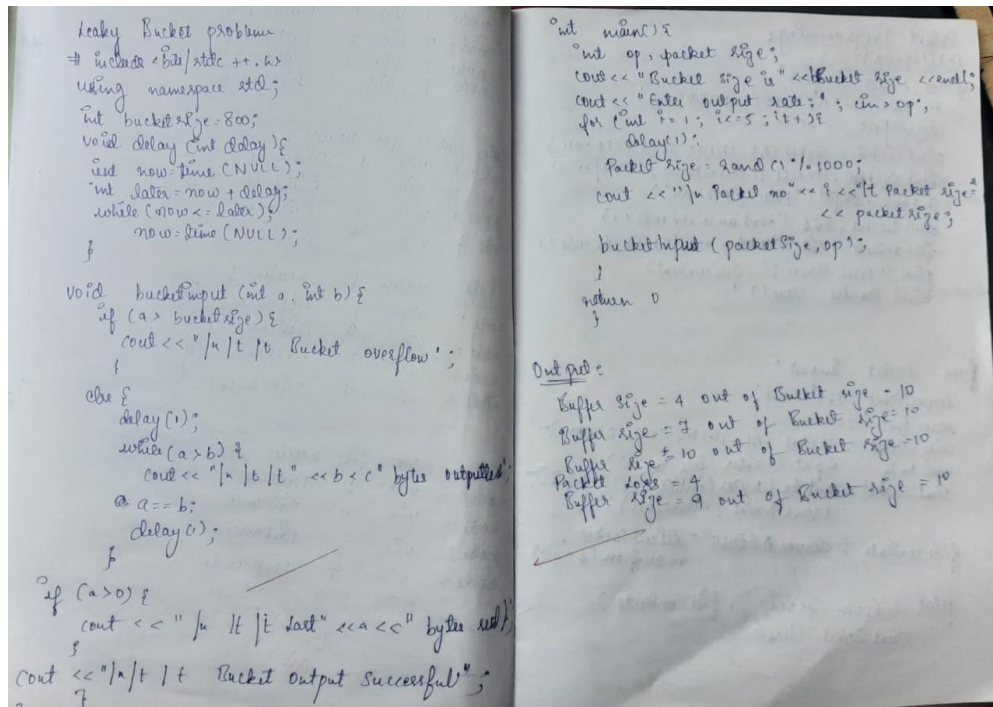
Output:


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

Experiment No 5

Aim of the Experiment

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

Server:

```
from socket import *

serverName = ''

serverPort = 12530

serverSocket = socket(AF_INET,SOCK_STREAM)

serverSocket.bind((serverName,serverPort))

serverSocket.listen(1)

print("The server is ready to receive")

while 1:

    connectionSocket, addr = serverSocket.accept()

    sentence = connectionSocket.recv(1024).decode()

    try:

        file = open(sentence,"r")

        l = file.read(1024)

        connectionSocket.send(l.encode())

        file.close()

    except Exception as e:

        message = "No such file exist"

        connectionSocket.send(message.encode())

    connectionSocket.close()
```
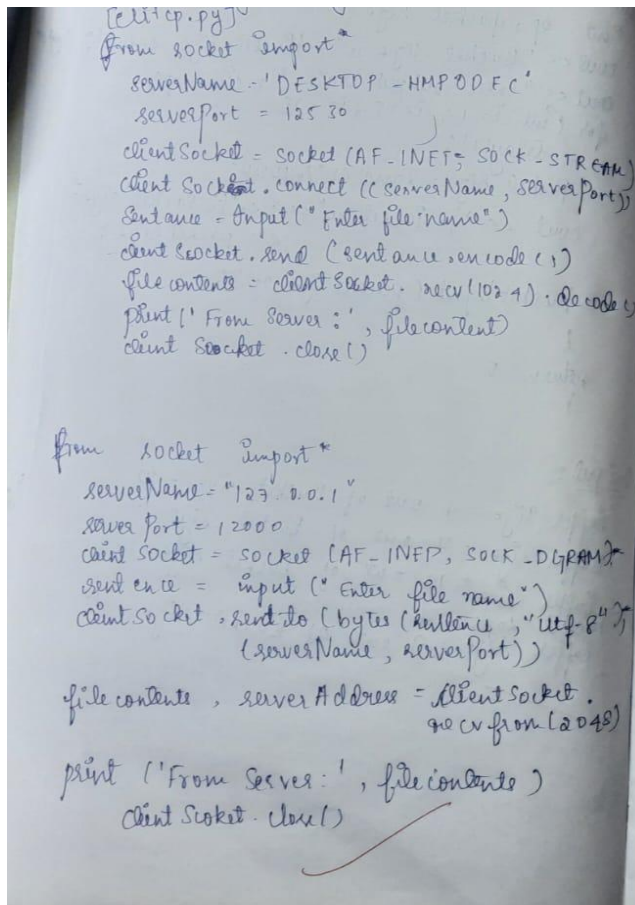
Client:

```
from socket import *

serverName = '192.168.1.104'

serverPort = 12530

clientSocket = socket(AF_INET, SOCK_STREAM)

clientSocket.connect((serverName,serverPort))

sentence = input("Enter file name")

clientSocket.send(sentence.encode())

filecontents = clientSocket.recv(1024).decode()

print ('From Server:', filecontents)

clientSocket.close()
```

Observation:

Output

Experiment No 5

Aim of the Experiment

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

Server:

```
from socket import *

serverPort = 12000

serverSocket = socket(AF_INET, SOCK_DGRAM)

serverSocket.bind(("127.0.0.1", serverPort))

print("The server is ready to receive")

while 1:

    sentence,clientAddress = serverSocket.recvfrom(2048)


    file=open(sentence,"r")

    l=file.read(2048)


    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)

    print("sent back to client",l)

    file.close()
```

Client:

```
from socket import *

serverName = "127.0.0.1"

serverPort = 12000

clientSocket = socket(AF_INET, SOCK_DGRAM)


sentence = input("Enter file name")

clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
```

filecontents,serverAddress = clientSocket.recvfrom(2048)

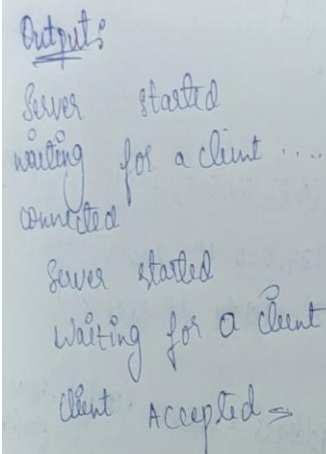print ('From Server:', filecontents)

clientSocket.close()

Observation:

Output

```
C:\Windows\System32\cmd.exe                                          —    ☐    ✕

Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

D:\con054-main\CON_LAB\lab10>py uclient.py
Enter file name: try.txt
From Server: b'HELLO WORLD\n\n'

D:\con054-main\CON_LAB\lab10>
```

```
C:\Windows\System32\cmd.exe - py  userver.py                         —    ☐    ✕

Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

D:\con054-main\CON_LAB\lab10>py userver.py
The server is ready to receive
sent back to client HELLO WORLD
```