VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

ADITYA BASAVARAJ NAGATHAN (1BM20CS193)

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

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**COMPUTER NETWORKS**" carried out by **ADITYA BASAVARAJ NAGATHAN (1BM20CS193),** who is bonafide student of B.M. **S. College of Engineering.** It isin 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.

Dr. Nandini vineeth
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

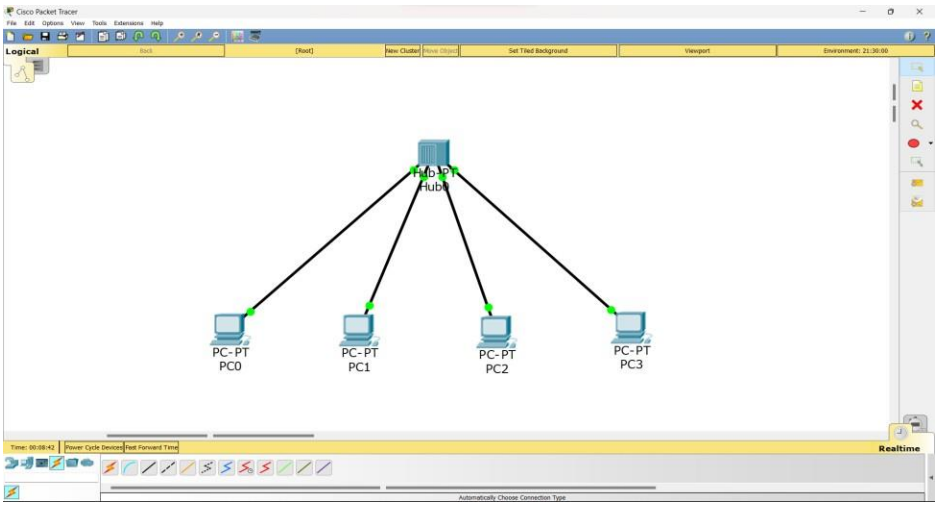
Cycle-1

Experiment No 1

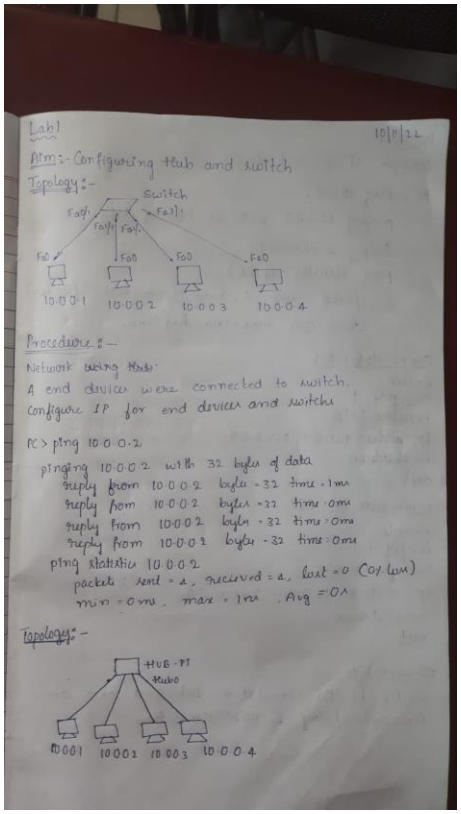
Aim of the program

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

Hub
Topology



Procedure



Lab1 10/8/22

Aim:- Configuring Hub and switch

Topology:-

Procedure:-

Network using Hub:

A end devices were connected to switch.

Configure IP for end devices and switch.

PC > ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data

reply from 10.0.0.2 bytes = 32 time = 1ms

reply from 10.0.0.2 bytes = 32 time = 0ms

reply from 10.0.0.2 bytes = 32 time = 0ms

reply from 10.0.0.2 bytes = 32 time = 0ms

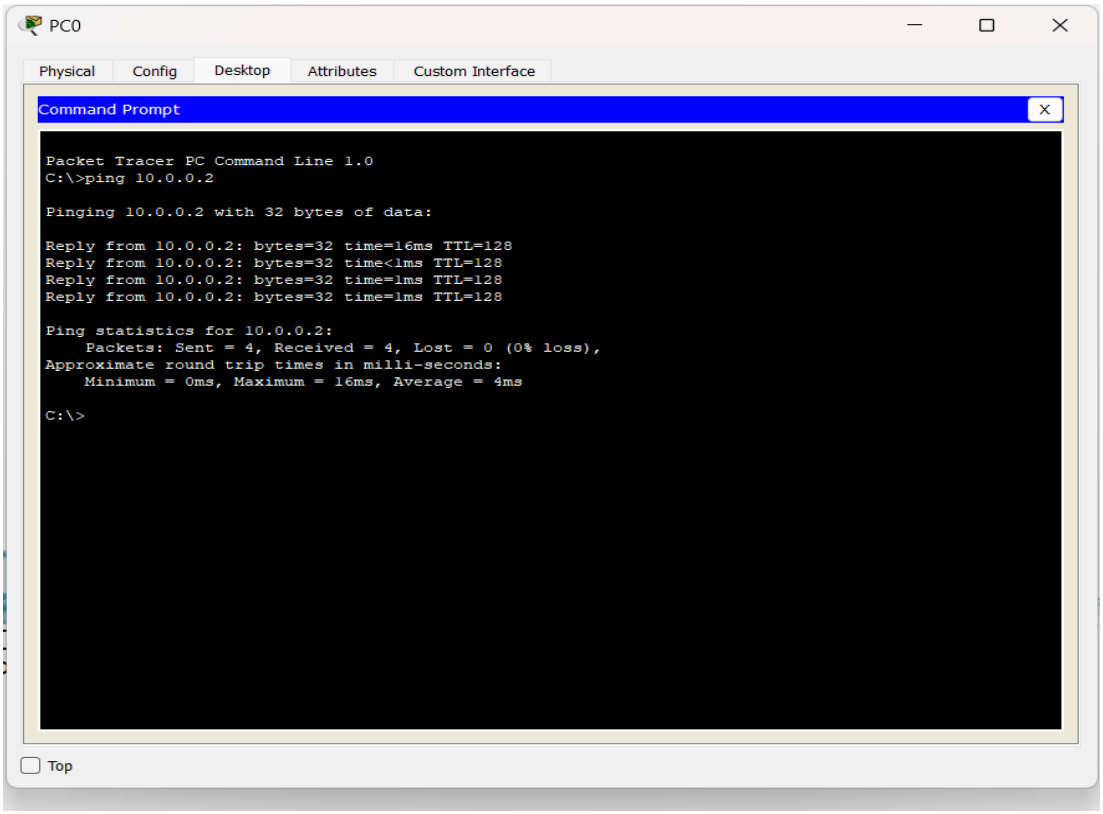
ping statistics 10.0.0.2

packets sent = 4, received = 4, lost = 0 (0% loss)

min = 0ms, max = 1ms, Avg = 0ms

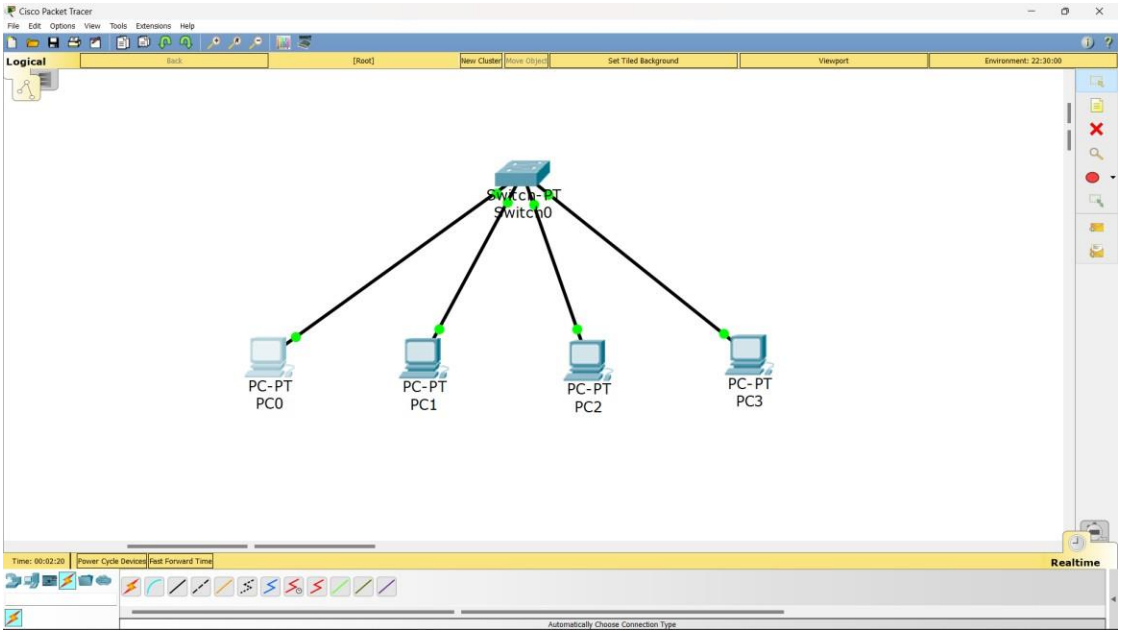
Topology:-

Output

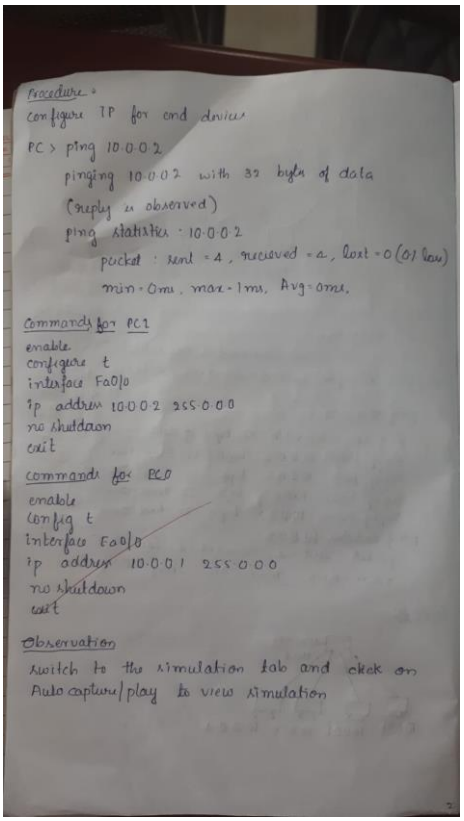


Switch

Topology



Procedure



Procedure:

configure IP for end devices

PC > ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data

(reply is observed)

ping statistics : 10.0.0.2

packet : sent = 4, received = 4, lost = 0 (0% loss)

min = 0ms, max = 1ms, Avg = 0ms.

Commands for PC1

enable
configure t
interface Fa0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
exit

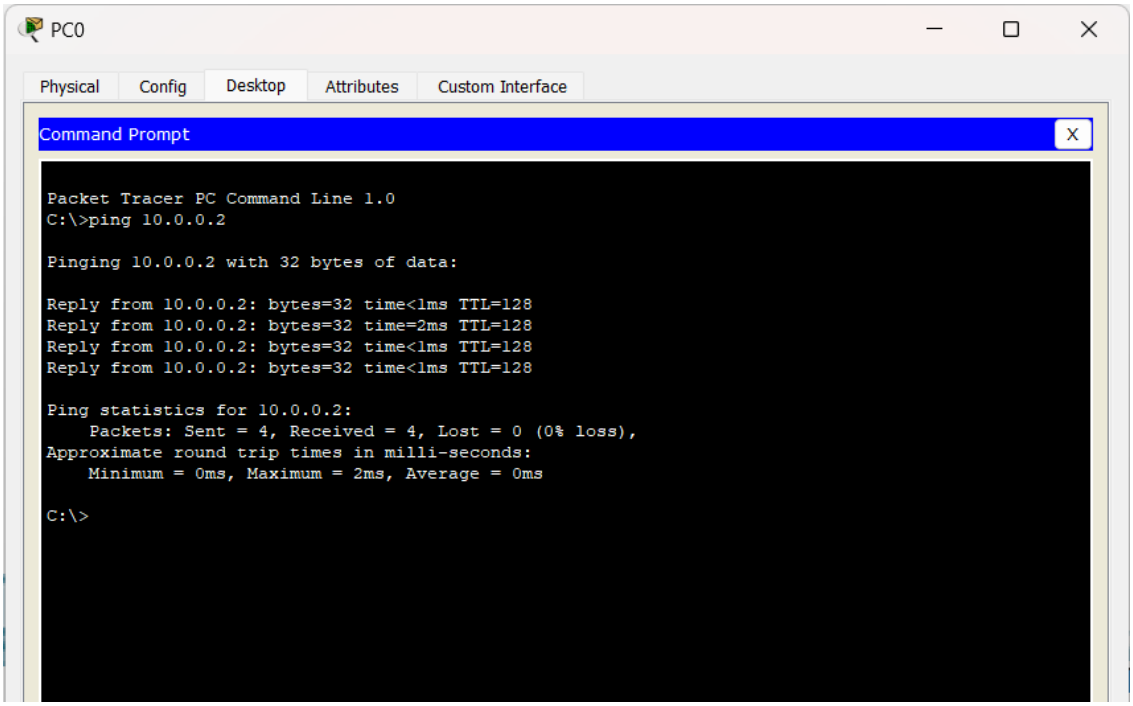
commands for PC0

enable
config t
interface Fa0/0
ip address 10.0.0.1 255.0.0.0
no shutdown
exit

Observation

switch to the simulation tab and click on Auto capture/play to view simulation

Output

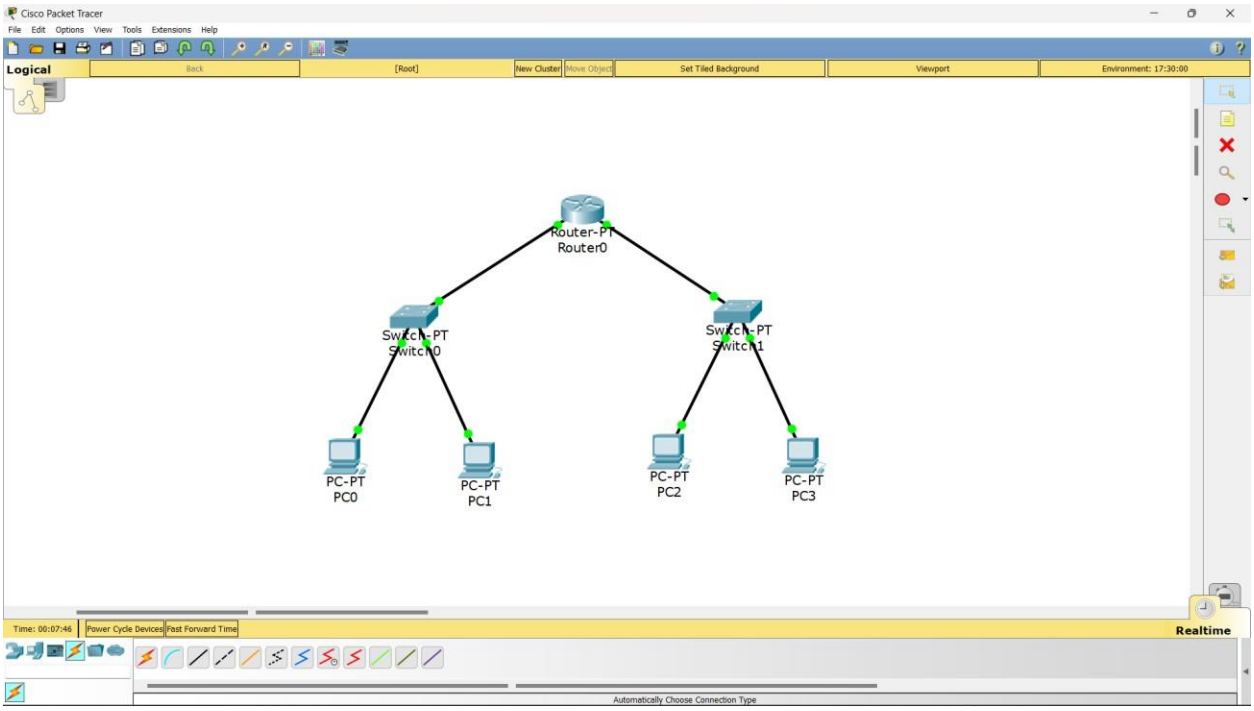


Experiment No 2

Aim of the program

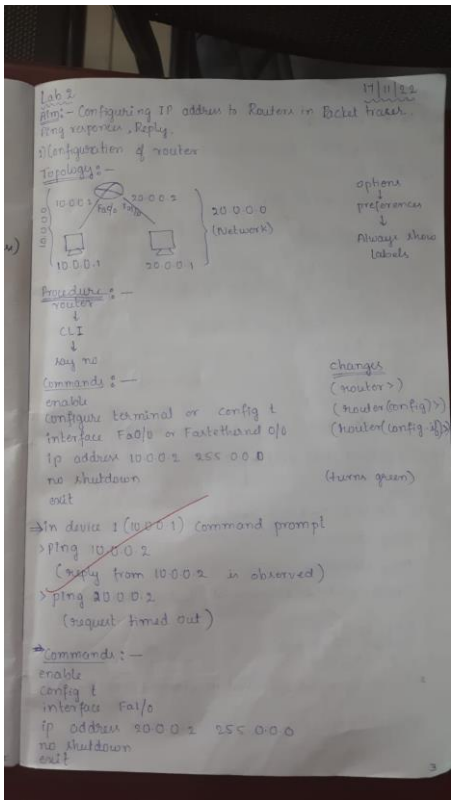
Configuring IP address to Routers in Packet Tracer. Exploring the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.

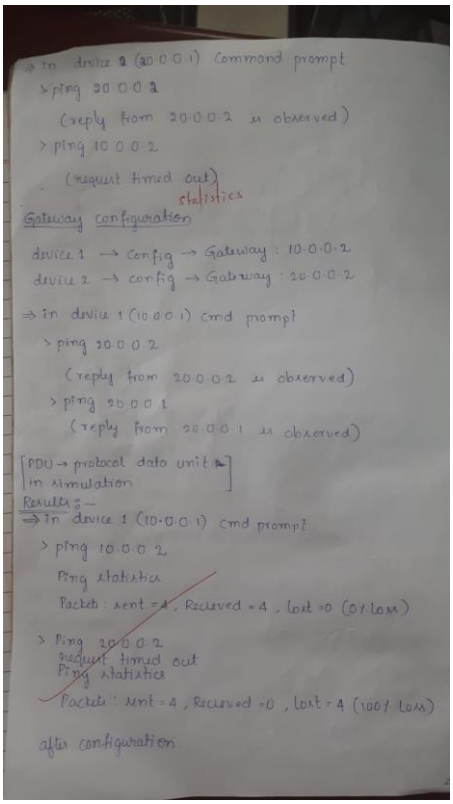
Topology

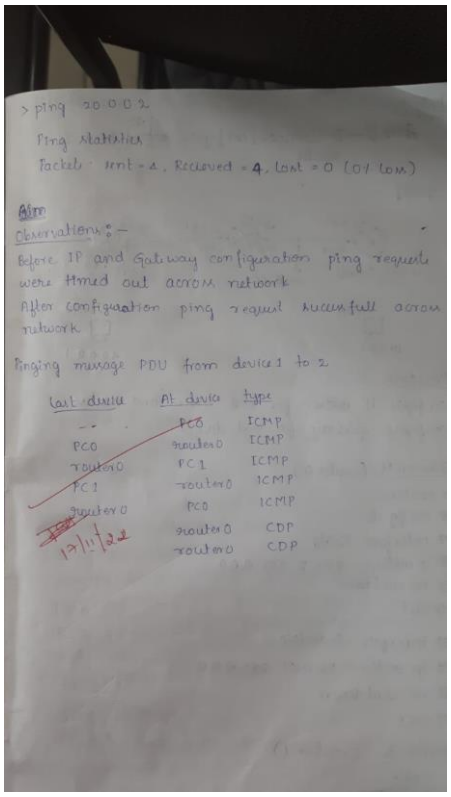


Procedure

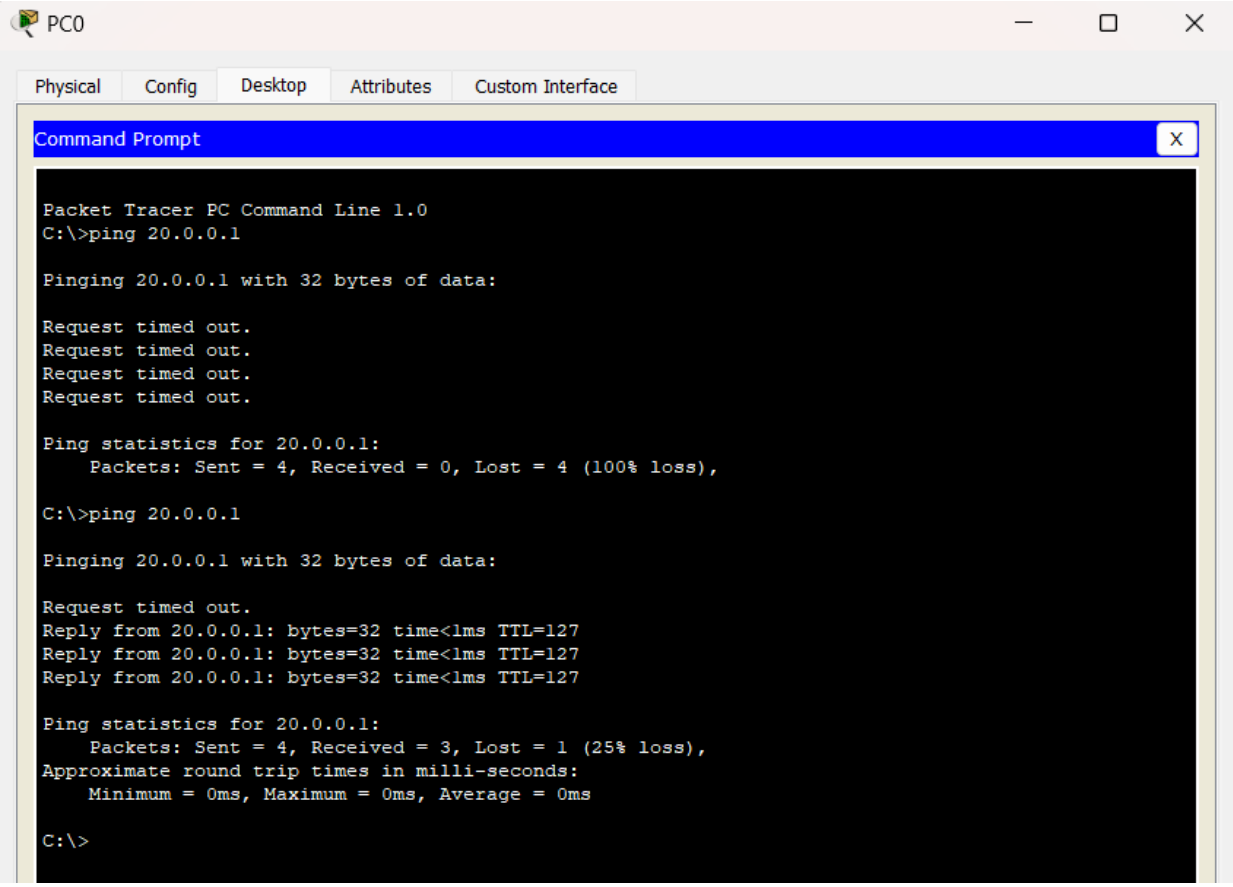
```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shutdown

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)#
```


Lab 2
17/11/22
Aim:- Configuring IP address to Routers in Packet Tracer.
Ping responses, Reply.
i) Configuration of router
Topology:-
10.0.0.2
20.0.0.2
Fa0/0
Fa0/0
10.0.0.0
20.0.0.0
(Network)
10.0.0.1
20.0.0.1
options
preferences
Always show labels
Procedure:-
router
CLI
say no
Commands:-
enable
configure terminal or config t
interface Fa0/0 or FastEthernet 0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
exit
changes
(router>)
(router(config)>)
(router(config-if)>)
(turns green)
In device 1 (10.0.0.1) Command prompt
>ping 10.0.0.2
(reply from 10.0.0.2 is observed)
>ping 20.0.0.2
(request timed out)
Commands:-
enable
config t
interface Fa1/0
ip address 20.0.0.2 255.0.0.0
no shutdown
exit


In device 2 (20.0.0.1) Command prompt
>ping 20.0.0.2
(reply from 20.0.0.2 is observed)
>ping 10.0.0.2
(request timed out)
statistics
Gateway configuration
device 1 → Config → Gateway : 10.0.0.2
device 2 → config → Gateway : 20.0.0.2
In device 1 (10.0.0.1) cmd prompt
>ping 20.0.0.2
(reply from 20.0.0.2 is observed)
>ping 20.0.0.1
(reply from 20.0.0.1 is observed)
[PDU → protocol data unit in simulation]
Results:-
In device 1 (10.0.0.1) cmd prompt
>ping 10.0.0.2
Ping statistics
Packets: sent = 4, Received = 4, Lost = 0 (0% loss)
>Ping 20.0.0.2
request timed out
Ping statistics
Packets: sent = 4, Received = 0, Lost = 4 (100% loss)
after configuration


>ping 20.0.0.2
Ping statistics
Packets: sent = 4, Received = 4, Lost = 0 (0% loss)
Aim
Observations:-
Before IP and Gateway configuration ping requests were timed out across network
After configuration ping request successfull across network
Pinging message PDU from device 1 to 2
Last device
At device
Type
PC0
ICMP
PC0
router0
ICMP
router0
PC1
ICMP
PC1
router0
ICMP
router0
PC0
ICMP
router0
CDP
router0
CDP
17/11/22

Output

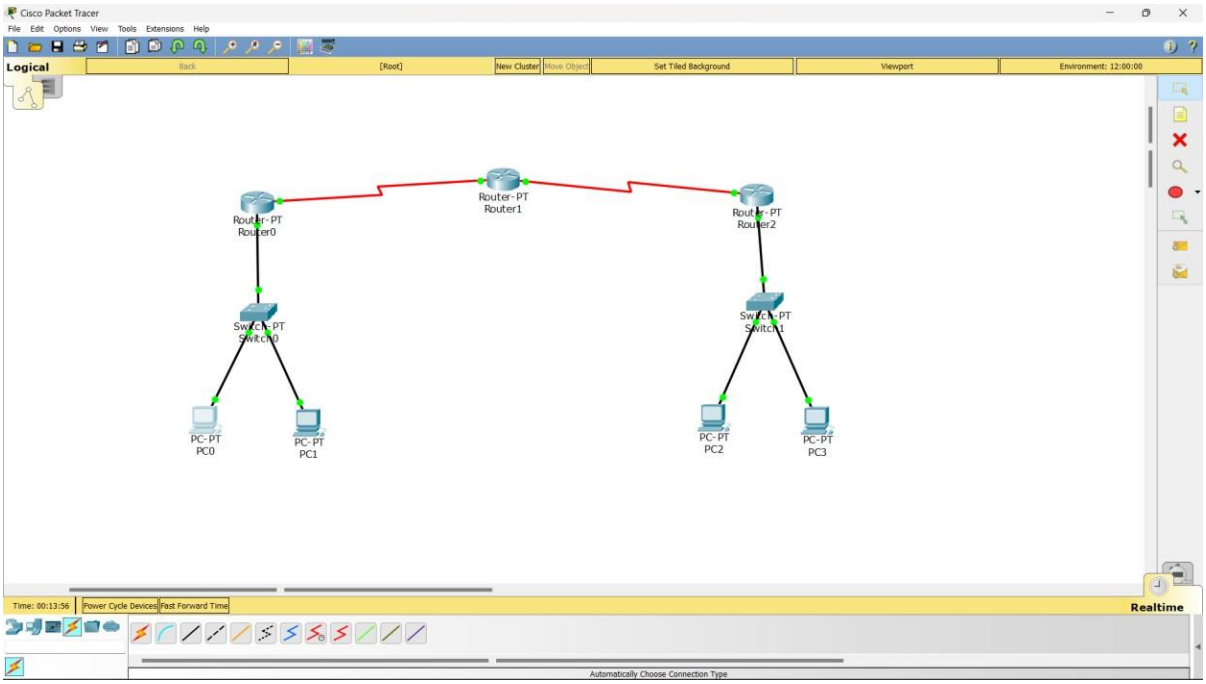

PC0
Physical
Config
Desktop
Attributes
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 0, 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

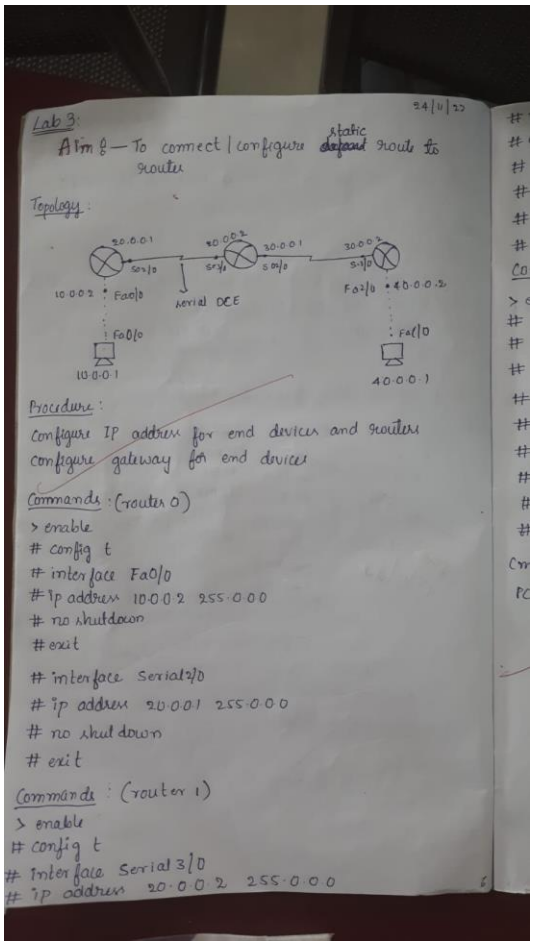
Configuring default route to the Router

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
```



24/4/22

Lab 3:

Aim :— To connect | configure static route to router

Topology:

Procedure:

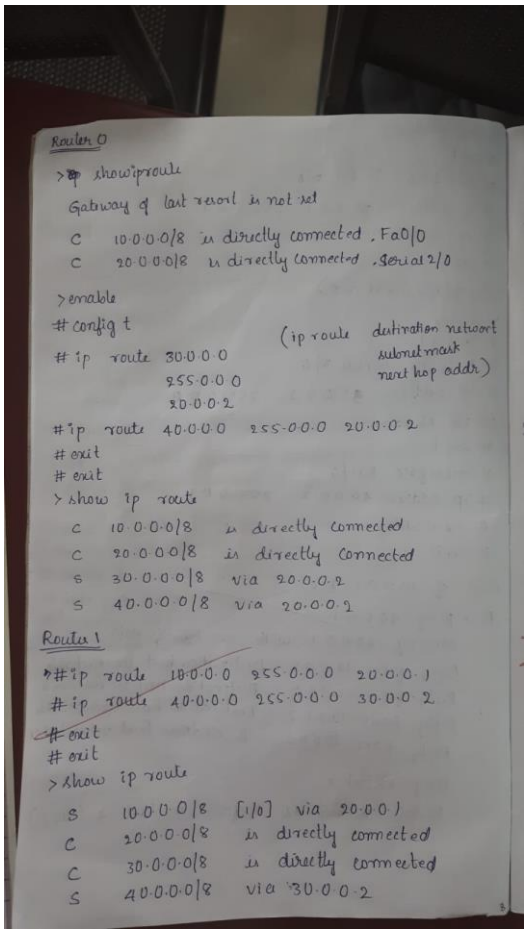
Configure IP address for end devices and routers
configure gateway for end devices

Commands: (router 0)

> enable
config t
interface Fa0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
exit
interface Serial2/0
ip address 20.0.0.1 255.0.0.0
no shut down
exit

Commands: (router 1)

> enable
config t
Interface Serial3/0
ip address 20.0.0.2 255.0.0.0



Router 0

> show ip route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, Fa0/0
C 20.0.0.0/8 is directly connected, Serial2/0

> enable
config t
ip route 30.0.0.0 255.0.0.0 20.0.0.2 (ip route destination network subnet mask next hop addr)
ip route 40.0.0.0 255.0.0.0 20.0.0.2
exit
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 via 20.0.0.2
S 40.0.0.0/8 via 20.0.0.2

Router 1

ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 40.0.0.0 255.0.0.0 30.0.0.2
exit
exit
> show ip route
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected
C 30.0.0.0/8 is directly connected
S 40.0.0.0/8 via 30.0.0.2

Output

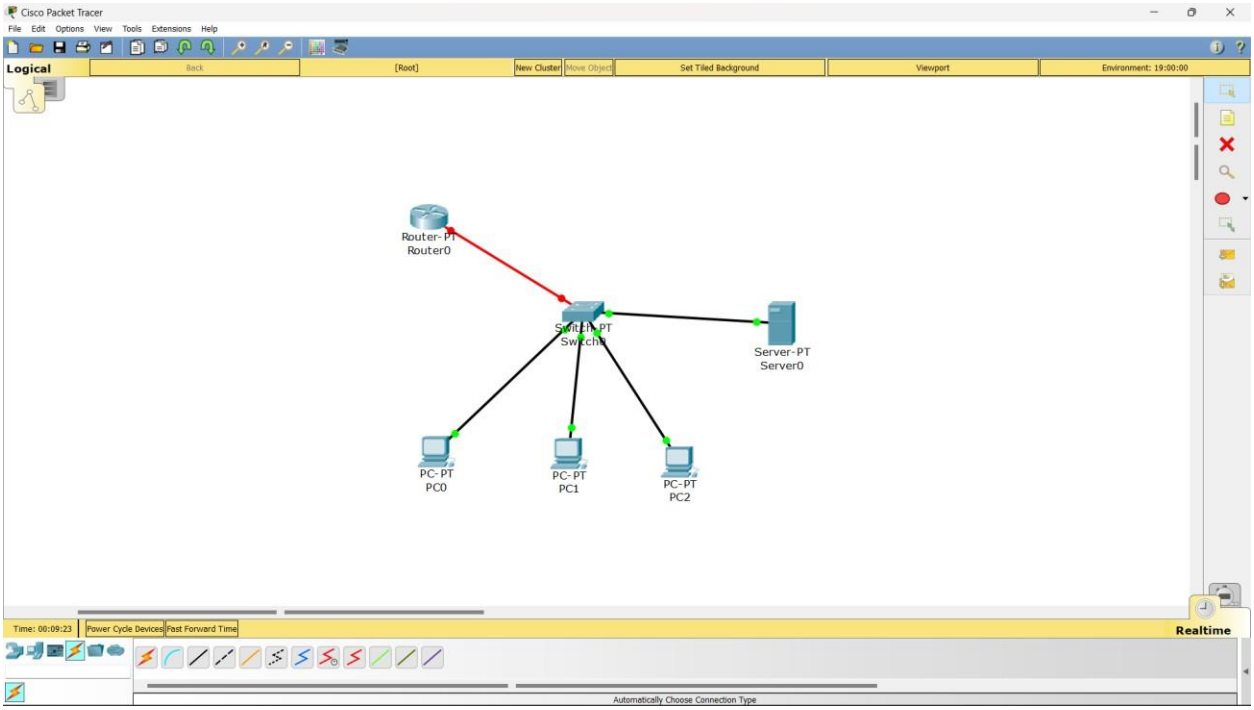
```
Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.1

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

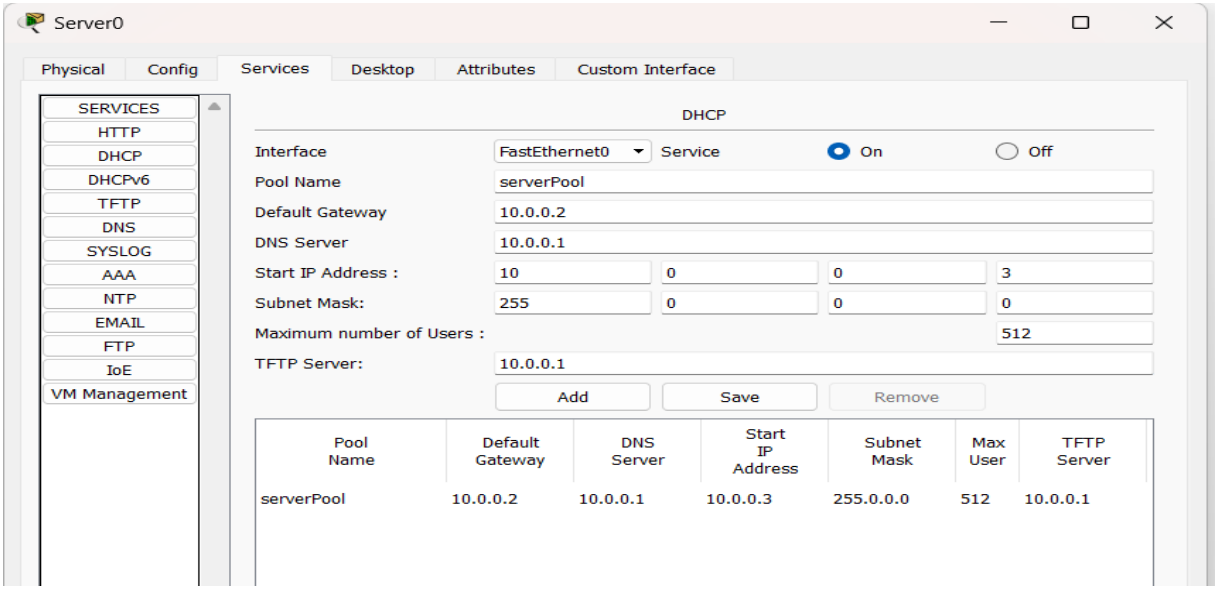
Aim of the program

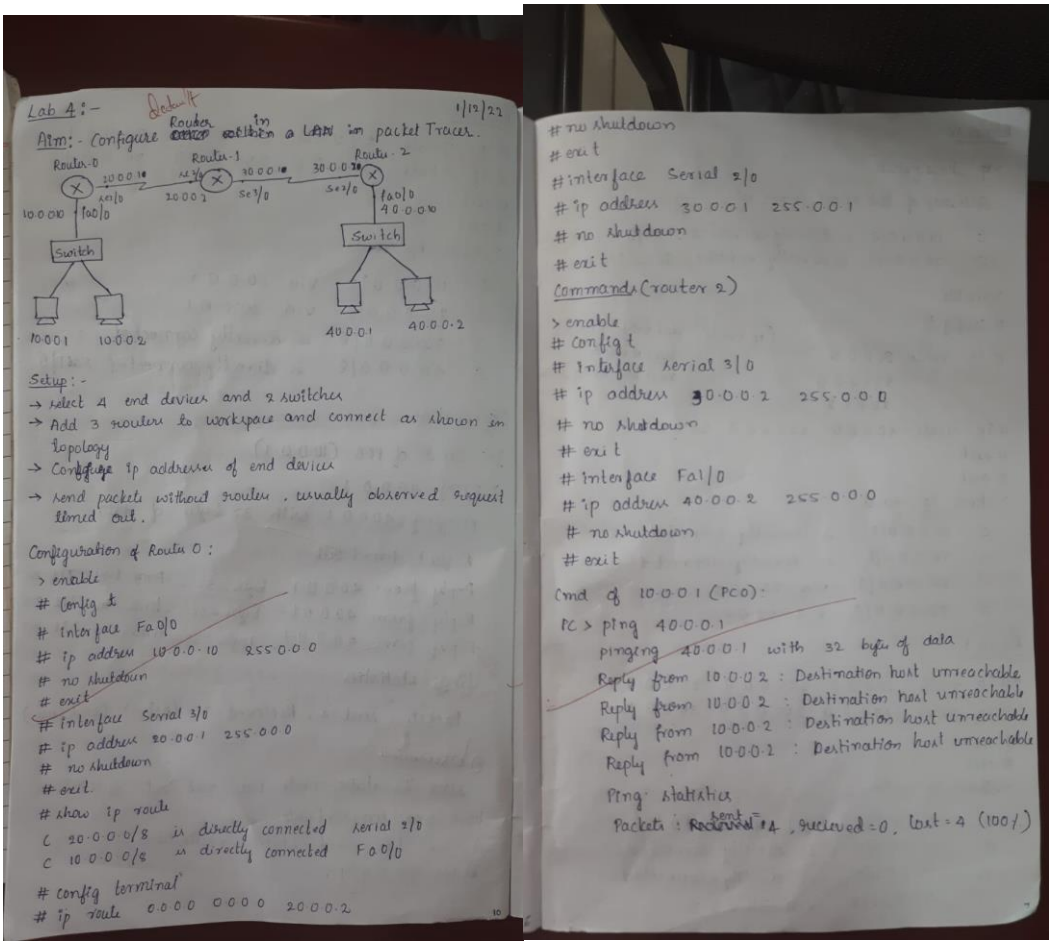
Configuring DHCP within a LAN in a packet Tracer

Topology



Procedure





Lab 4 :- default 1/12/22

Aim: - Configure Router in a LAN in packet Tracer.

Setup: -
→ select 4 end devices and 2 switches
→ Add 3 routers to workspace and connect as shown in topology
→ Configure ip addresses of end device
→ send packets without routes, usually observed request timed out.

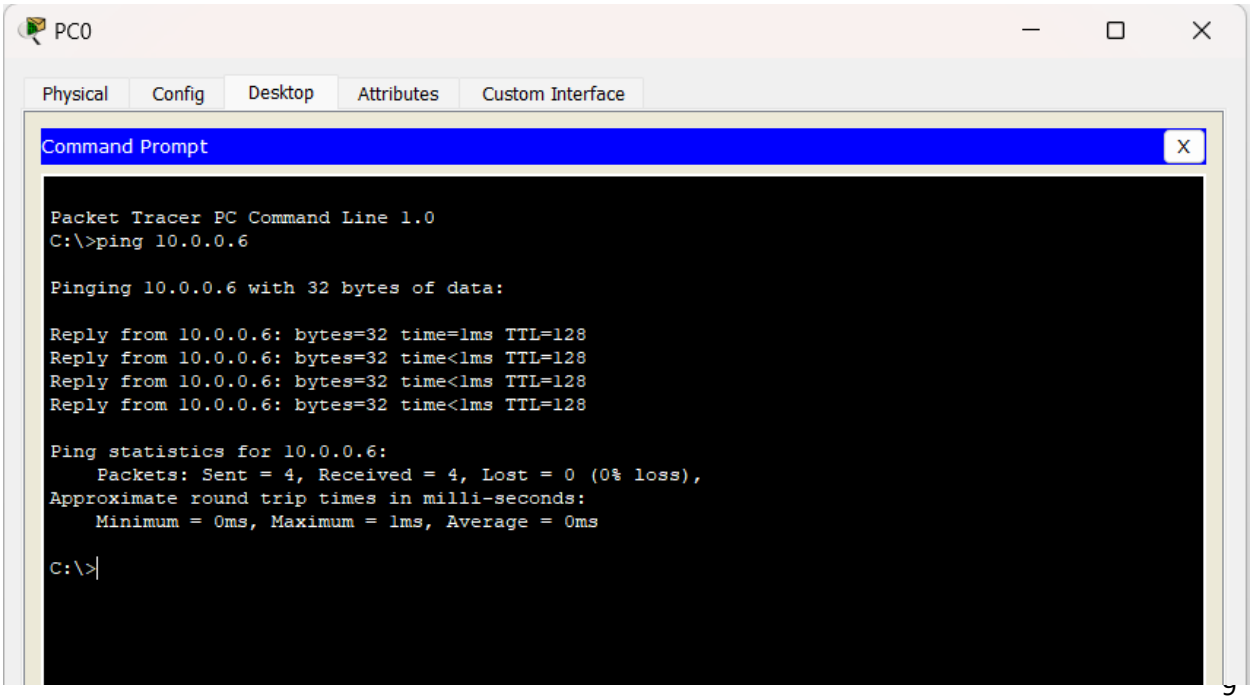
Configuration of Router 0 :
> enable
Config t
interface Fa0/0
ip address 10.0.0.10 255.0.0.0
no shutdown
exit
interface Serial 3/0
ip address 20.0.0.1 255.0.0.0
no shutdown
exit
show ip route
C 20.0.0.0/8 is directly connected Serial 2/0
C 10.0.0.0/8 is directly connected Fa0/0
config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.2

no shutdown
exit
interface Serial 2/0
ip address 30.0.0.1 255.0.0.1
no shutdown
exit

Commands (router 2)
> enable
Config t
Interface Serial 3/0
ip address 30.0.0.2 255.0.0.0
no shutdown
exit
interface Fa1/0
ip address 40.0.0.2 255.0.0.0
no shutdown
exit

Cmd of 10.0.0.1 (PC0):
PC > ping 40.0.0.1
pinging 40.0.0.1 with 32 byte of data
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable
Ping statistics
Packets : Sent = 4, Received = 0, Lost = 4 (100%)

Output

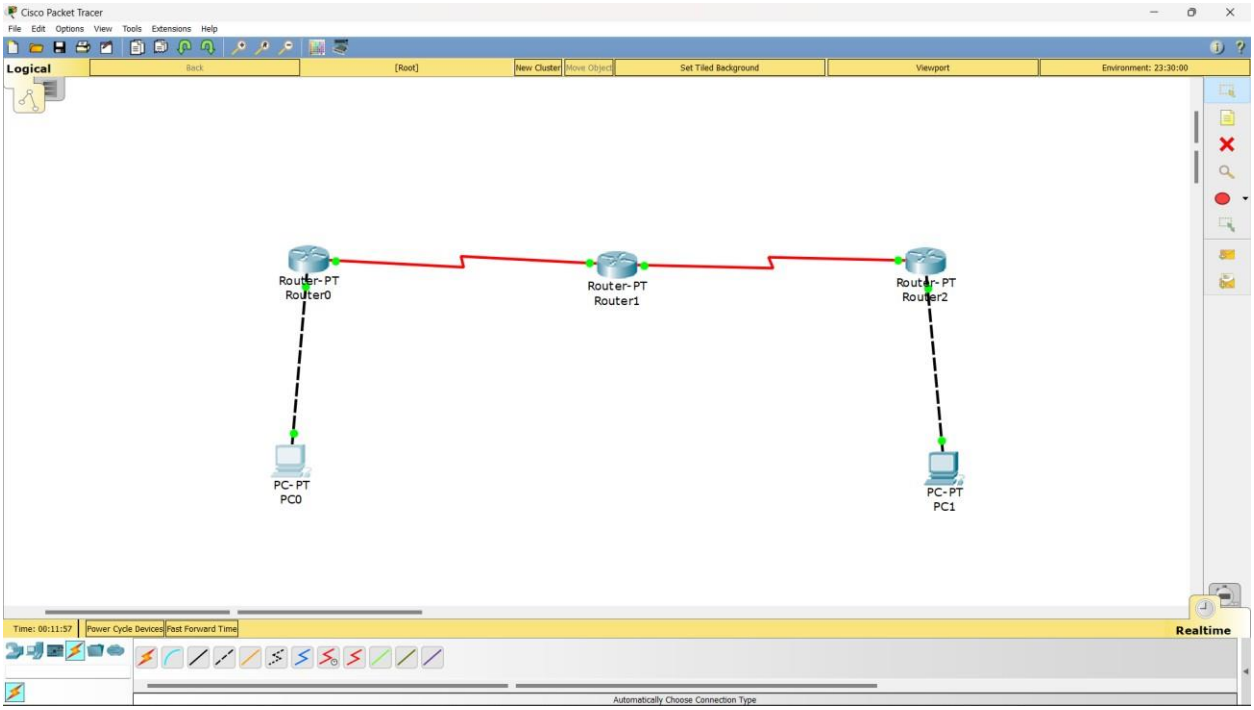


Experiment No 5

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)#
```

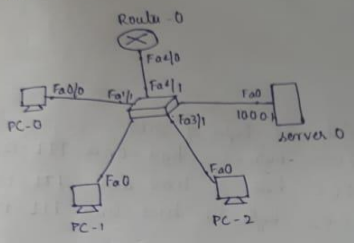
```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
This command applies only to DCE interfaces
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 5 :- 8/12/22.

Aim :- Configure DHCP within a LAN in packet

Topology :-

DORA
D - discover
O - Offer
R - Request
A - Acknowledgement



Setup :-

→ select 3 end devices, 1 switch, 1 router and a server and connect as shown in topology.

→ do not config static ip address

→ Configure the server by giving the IP address & gateway

Configure server :

services → DHCP → on service

Default Gateway	10.0.0.1
DNS server	10.0.0.2
Start IP addr	10 0 0 3
Subnet mask	255 0 0 0
TFTP server	10.0.0.2

→ save the changes

→ Open IP configuration in desktop change IP configuration from static to DHCP. Follow the same for all other PC's

→ IP addresses will be automatically assigned and now ping can be performed.

→ Procedure has to be followed to activate DHCP. (13)

Router > enable
Router # config t
Router (config-if) # ip address 10.0.0.2 255.0.0.0
no shutdown
exit

Result

> ping 10.0.0.5

pinging 10.0.0.5 with 32 bytes of data

Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.5
packets : sent = 4, Received = 4 lost = 0 (0% loss)

Observation :-

DHCP : Dynamic host configuration protocol provides IP address (IP) host automatically with IP with subnet mask and default gateway.

Output:

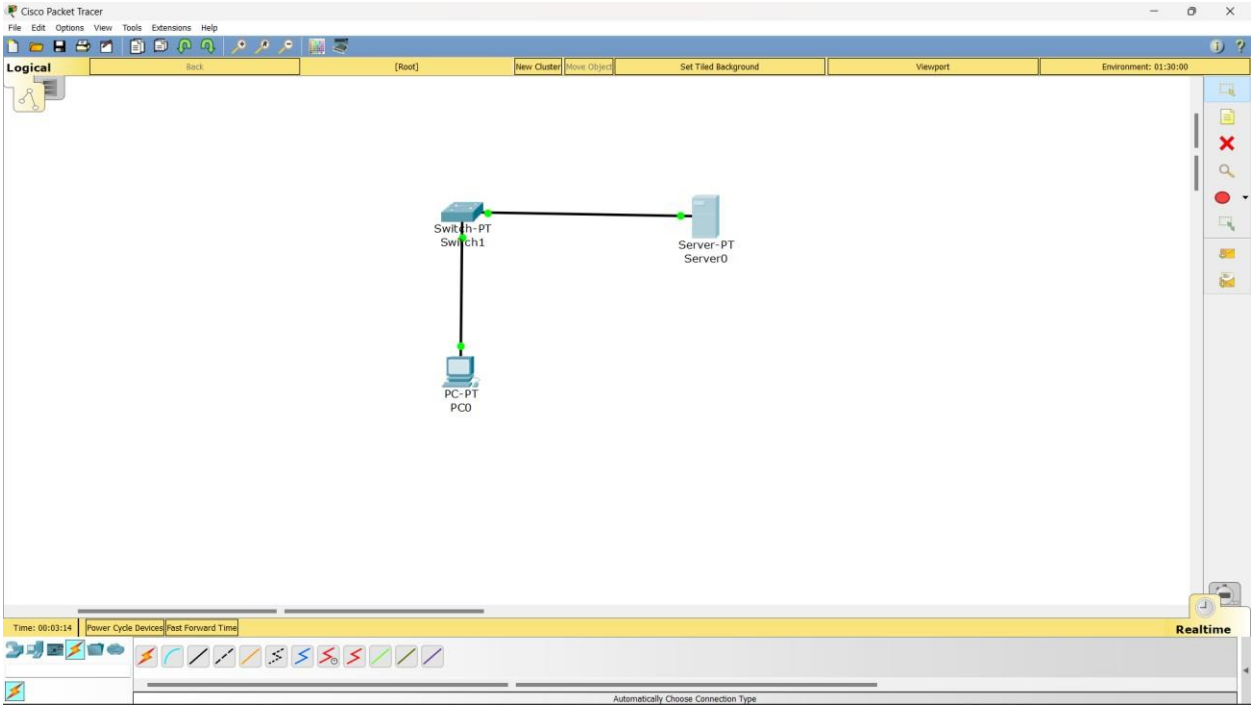
```
C:\>ping 40.0.0.1

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

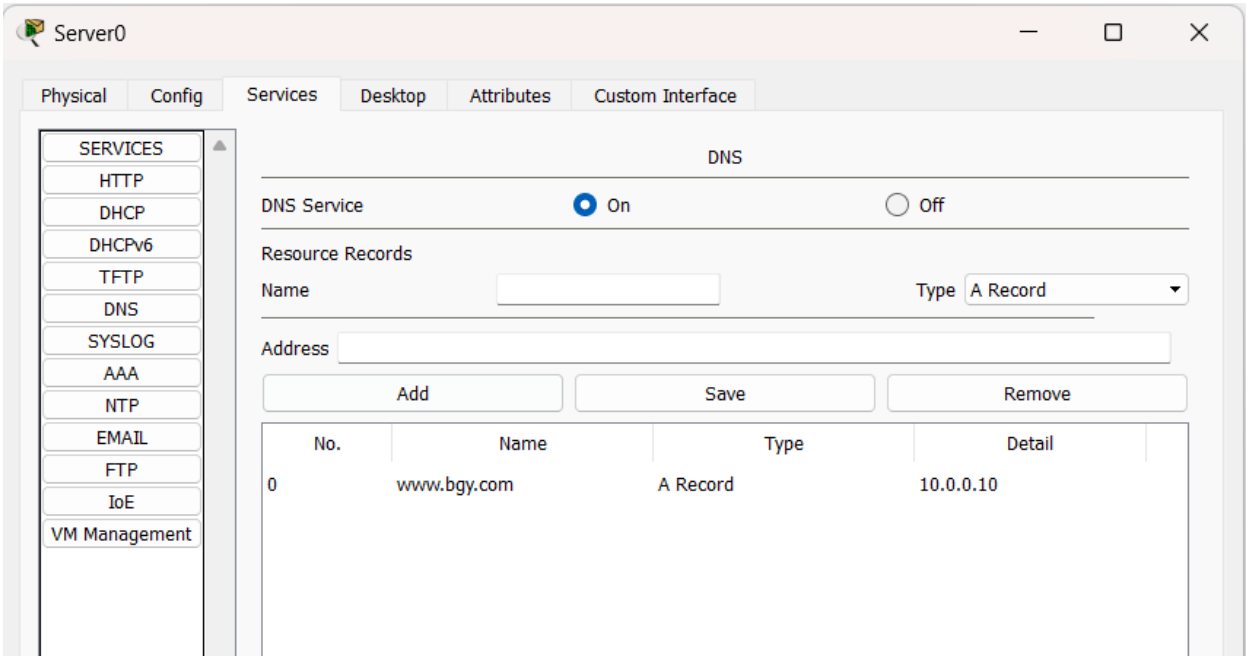
Aim of the program

Demonstration of WEB server and DNS using Packet Tracer

Topology



Procedure

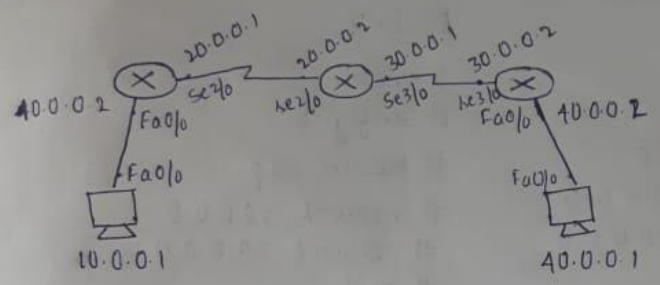


15/12/22

Lab 6 :—

Aim: Configure RIP (Routing ~~Interface~~ Information protocol)

Topology:



* To select best path in network based on hop count.
* margin is 15 hops. it rejects if hop count is greater than 15 and chooses best among the selected if hop count is less than 15

Configuration of Router 1:

```
# interface Serial 2/0
# ip address 20.0.0.1 255.0.0.0
# encapsulation PPP
# clock rate 64000
# no shutdown
```

Configure fastethernet also

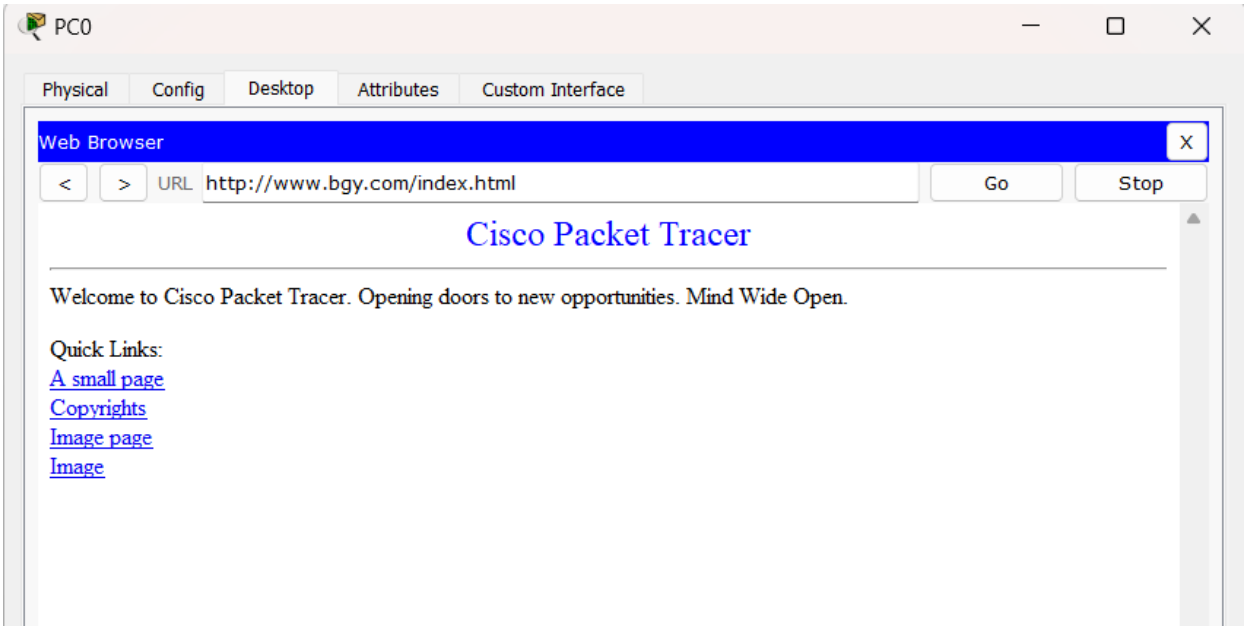
Configuration of Router 3:

```
# interface Serial 3/0
# ip address 30.0.0.2 255.0.0.0
# encapsulation PPP
# clock rate 64000
# no shut down
```

Configuration of Router 2:

```
# interface serial 2/0
# ip address 20.0.0.2 255.0.0.0
# no shut down
# interface serial 3/0
# ip address 30.0.0.1 255.0.0.0
```

Output


PC0
Physical
Config
Desktop
Attributes
Custom Interface
Web Browser
X
<
>
URL
http://www.bgy.com/index.html
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image

Cycle-2

Experiment No 1

Aim of the Experiment

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

Code

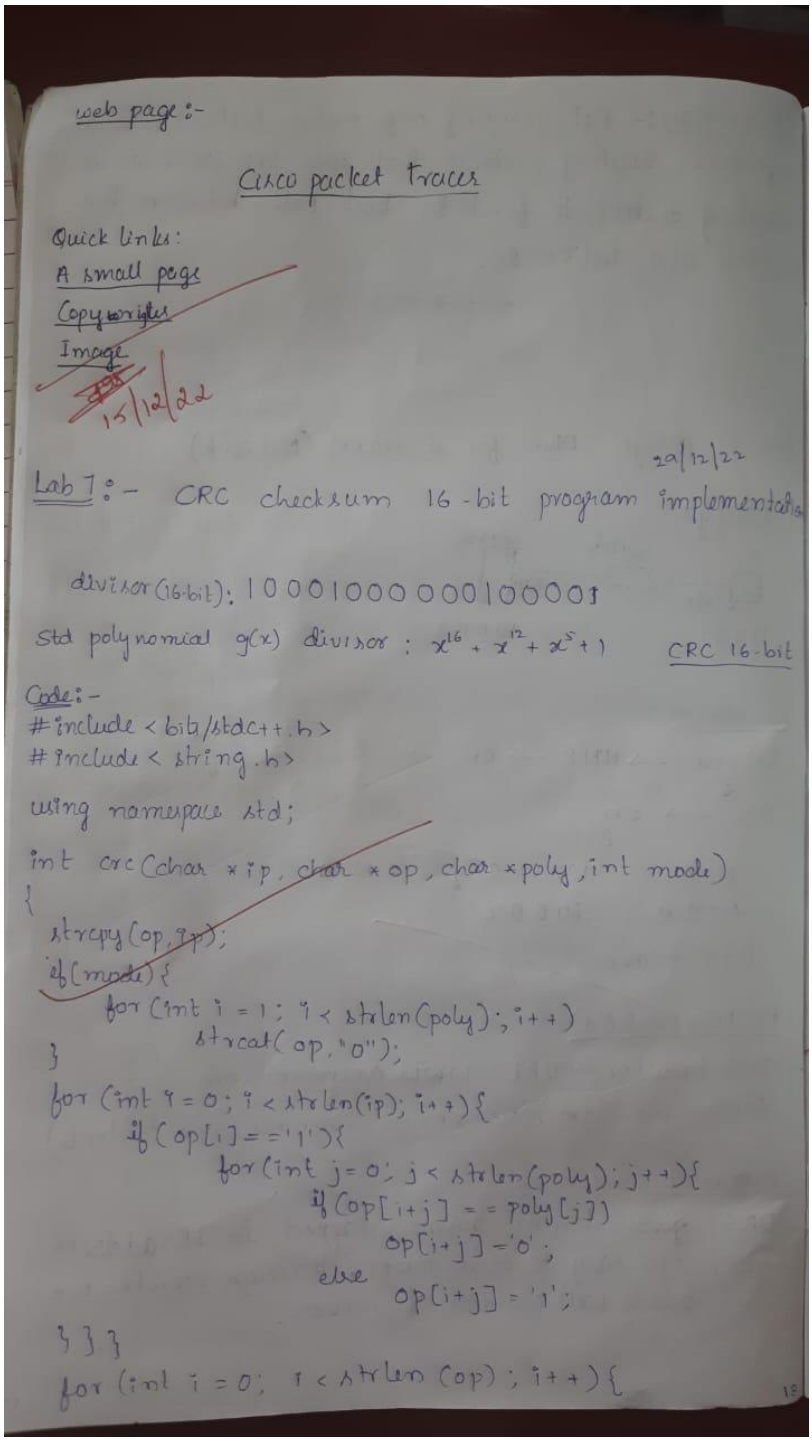
```
#include<bits/stdc++.h>
using namespace std;
void receiver(string data, string key);

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)
{
```

```
	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();
```

```
	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())
	{
```

```
            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:



web page :-

Cisco packet Tracer

Quick links:
A small page
Copywriter
Image
15/12/22

29/12/22

Lab 7 :- CRC checksum 16-bit program implementation

divisor (16-bit): 10001000000100001

Std polynomial g(x) divisor : $x^{16} + x^{12} + x^5 + 1$ CRC 16-bit

Code:-

```
#include <bits/stdc++.h>
#include <string.h>
using namespace std;
int crc(char *ip, char *op, char *poly, int mode)
{
  strcpy(op, ip);
  if(mode){
      for (int i = 1; i < strlen(poly); i++)
            strcat(op, "0");
  }
  for (int i = 0; i < strlen(ip); i++){
      if (op[i] == '1'){
            for (int j = 0; j < strlen(poly); j++){
                  if (op[i+j] == poly[j])
                        op[i+j] = '0';
                  else
                        op[i+j] = '1';
  }}}
  for (int i = 0; i < strlen (op); i++){
```

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.

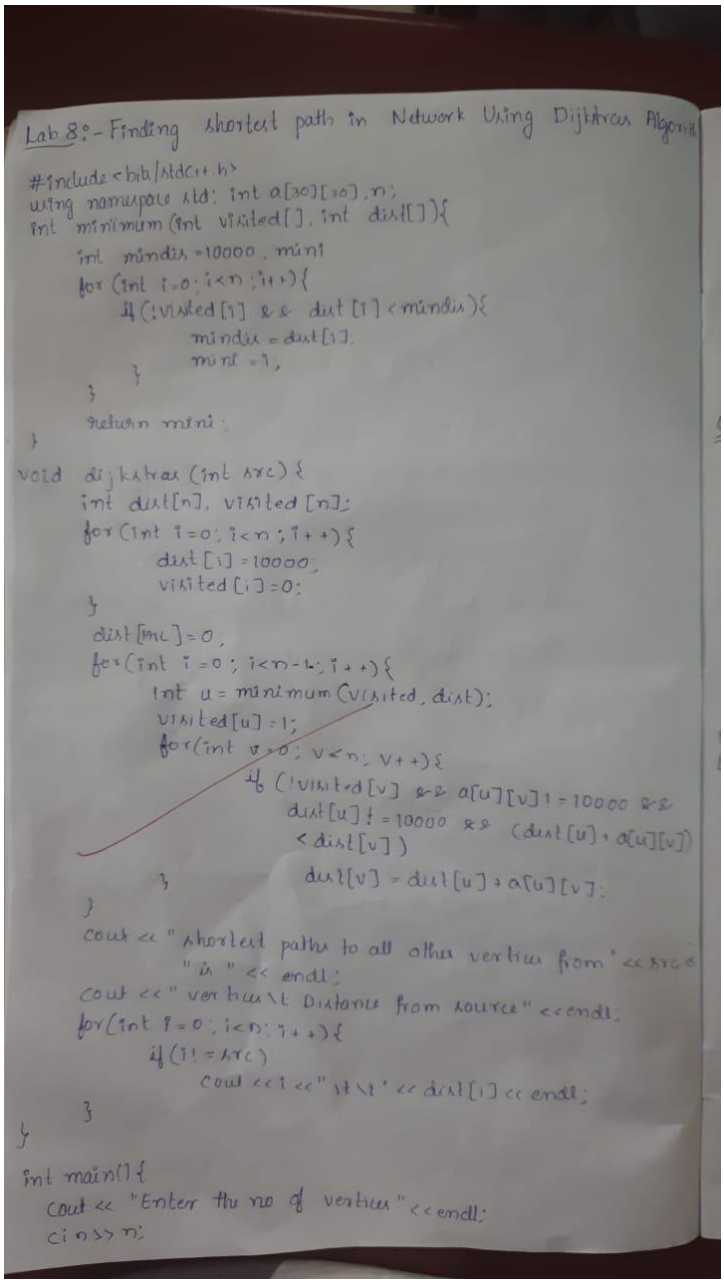
Code

```
#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++)
```

```
        {
            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:



Lab 8:- Finding shortest path in Network Using Dijkstras Algorith

```
#include <bits/stdc++.h>
using namespace std; int a[30][30],n;
int minimum (int visited[], int dist[]){
    int mindis =10000, mini
    for (int i=0; i<n ;i++){
        if (!visited[i] && dist[i] < mindis){
            mindis = dist[i];
            mini = i;
        }
    }
    return mini;
}
void dijkstras (int src){
    int dist[n], visited[n];
    for (int i=0; i<n ; i++){
        dist[i] = 10000;
        visited[i]=0;
    }
    dist[src]=0;
    for (int i =0 ; i<n-1; i++){
        int u = minimum (visited, dist);
        visited[u] = 1;
        for(int v=0; v<n; v++){
            if (!visited[v] && a[u][v]!=10000 &&
                dist[u]!=10000 && (dist[u]+a[u][v])
                < dist[v])
            dist[v] = dist[u]+a[u][v];
        }
    }
    cout << "shortest paths to all other vertices from" << src <<
        " is " << endl;
    cout << "vertices\t Distance from source" <<endl;
    for(int i=0; i<n; i++){
        if (i!=src)
            cout << i << " it \t" << dist[i] << endl;
    }
}
int main(){
    cout << "Enter the no of vertices" <<endl;
    cin>> n;
```

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.

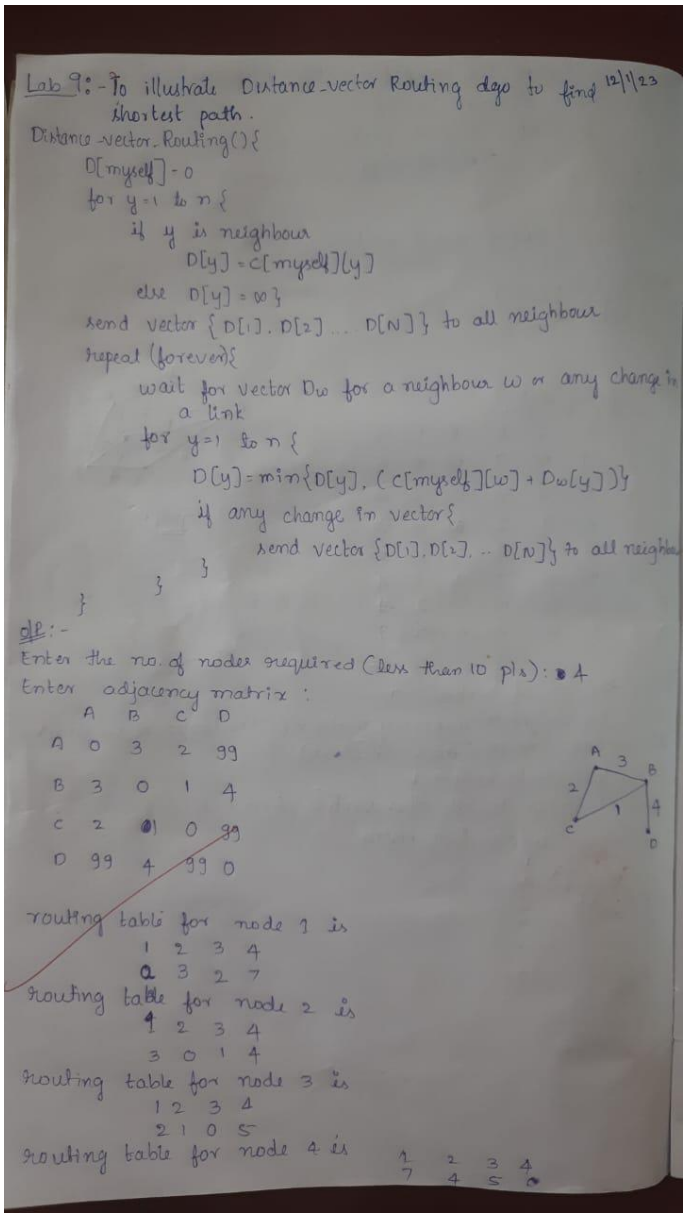
Code

```
#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];
```

```
            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);
    }
```

```
    }
}
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:



Lab 9:- To illustrate Distance-vector Routing algo to find shortest path. 12/1/23

```
Distance-vector-Routing(){
    D[myself] = 0
    for y = 1 to n {
        if y is neighbour
            D[y] = C[myself][y]
        else D[y] = ∞ }
    send vector { D[1], D[2] ... D[N] } to all neighbour
    repeat (forever){
        wait for vector Dw for a neighbour w or any change in
            a link
        for y = 1 to n {
            D[y] = min{D[y], (C[myself][w] + Dw[y])}
            if any change in vector{
                send vector {D[1], D[2], .. D[N]} to all neighbour
            }
        }
    }
```

Ole:-

Enter the no. of nodes required (less than 10 pls): 4

Enter adjacency matrix :

	A	B	C	D
A	0	3	2	99
B	3	0	1	4
C	2	1	0	99
D	99	4	99	0

routing table for node 1 is

1	2	3	4
0	3	2	7

routing table for node 2 is

1	2	3	4
3	0	1	4

routing table for node 3 is

1	2	3	4
2	1	0	5

routing table for node 4 is

1	2	3	4
7	4	5	0

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

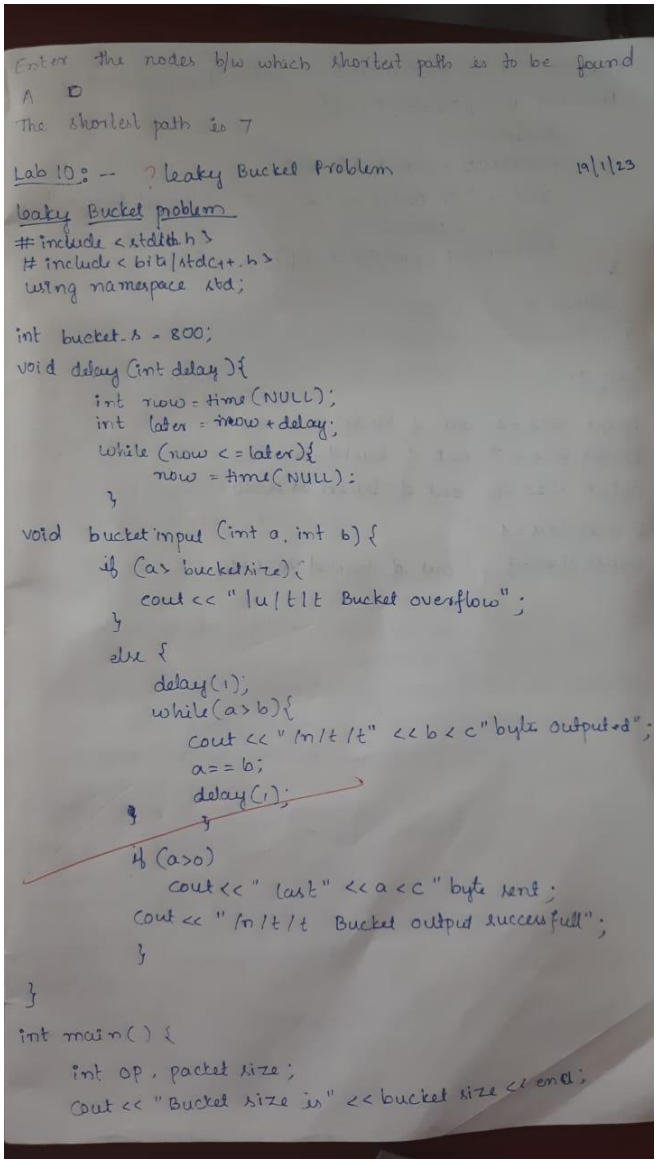
Write a program for congestion control using Leaky bucket algorithm

Code

```
#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:



```
Enter the nodes b/w which shortest path is to be found
A D
The shortest path is 7

Lab 10 :-  ? leaky Bucket Problem                    19/1/23

leaky Bucket problem
#include <stdlib.h>
#include <bits/stdc++.h>
using namespace std;

int bucket_s = 800;
void delay (int delay){
        int now = time(NULL);
        int later = now + delay;
        while (now <= later){
            now = time(NULL);
        }
void bucket input (int a, int b) {
        if (a> bucketsize){
            cout << "\n\t\t Bucket overflow";
        }
        else {
            delay(1);
            while(a>b){
                cout << "\n\t\t" << b << "byte outputed";
                a== b;
                delay(1);
            }
        if (a>0)
            cout << " last" << a << " byte sent;
        cout << "\n\t\t Bucket output successfull";
        }
}
int main() {
    int op, packet size;
    cout << "Bucket size is" << bucket size << endl;
```

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

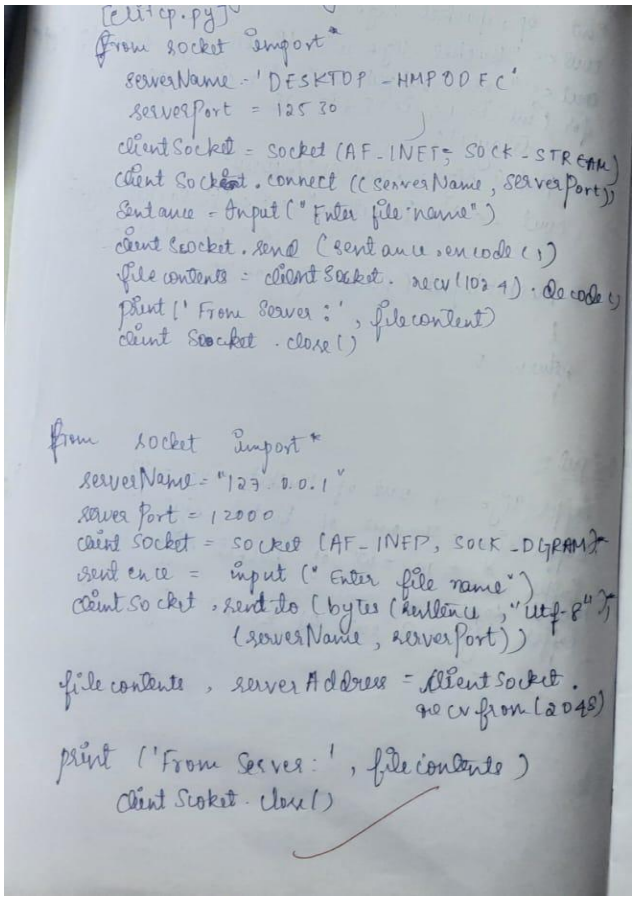
Server:

```
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:



[client.py]

```
from socket import*
serverName = 'DESKTOP-HMPODFC'
serverPort = 12530
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
Sentance = Input("Enter file name")
clientSocket.send(sentance.encode())
filecontents = clientSocket.recv(1024).decode()
print('From Server:', filecontent)
clientSocket.close()
```

```
from socket import*
serverName = "127.0.0.1"
server Port = 12000
client Socket = 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()
```

Output

```
C:\Windows\System32\cmd.exe - py server.py
Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

D:\con054-main\CON_LAB\lab10>py server.py
The server is ready to receive
```

```
C:\Windows\System32\cmd.exe
icrosoft Windows [Version 10.0.19045.2486]
c) Microsoft Corporation. All rights reserved.

:\con054-main\CON_LAB\lab10>py client.py
nter file name: try.txt
rom Server: HELLO WORLD
```

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:

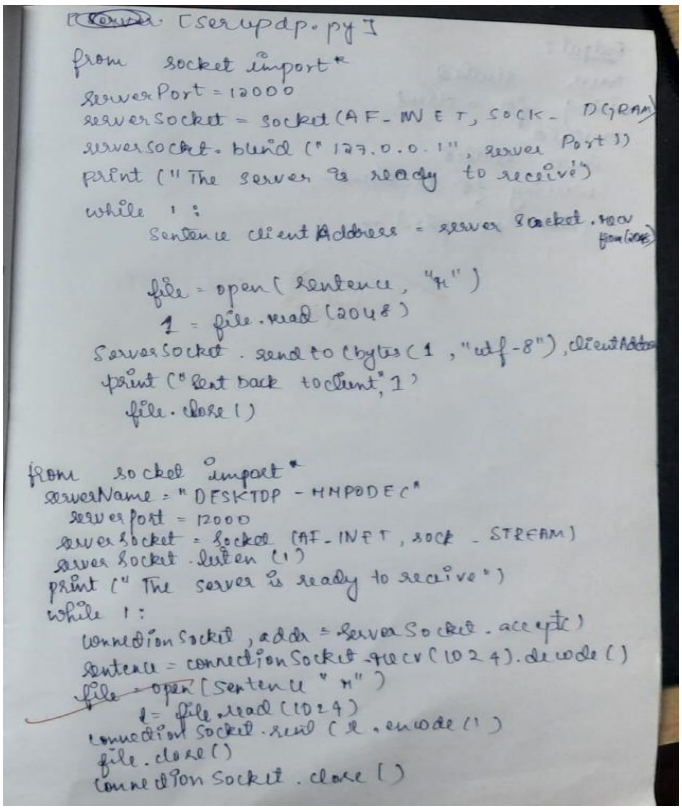
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence,clientAddress = serverSocket.recvfrom(2048)

    file=open(sentence,"r")
    l=file.read(2048)

    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print("sent back to client",l)
    file.close()
```

Client:

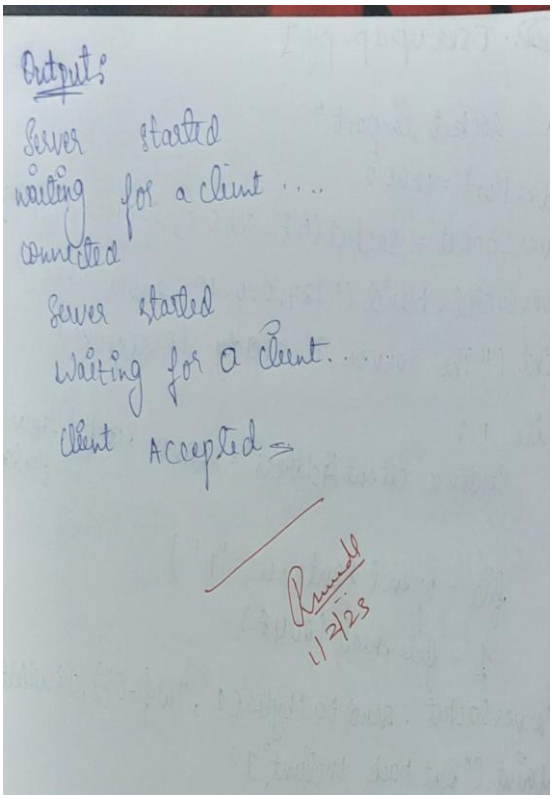
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

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:



```
[serupdp.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.recv
    file = open( sentence, "r" )
    1 = file.read (2048)
    ServerSocket . sendto (bytes (1 , "utf-8"), clientAddr
    print ("Sent back to client", 1)
    file.close ()

from socket import*
serverName = "DESKTOP - HMPODEC"
serverport = 12000
serverSocket = socket (AF_INET , sock_STREAM)
serversocket.listen (1)
print (" The server is ready to receive")
while 1:
    connectionSocket , addr = serverSocket . accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open [sentence " r" )
    l = file.read (1024)
    connectionSocket.send ( l . encode (1 )
    file.close()
    connection Socket . close ()
```



```
Output:
Server started
waiting for a client ...
connected
Server started
Waiting for a client...
Client Accepted =
```

Output

```
Select 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
```

```
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
```