VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

ANANYA SETTY B A (1BM21CS400)

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 **ANANYA SETTY B A (1BM21CS400),** 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
Associate Professor
Department of CSE
BMSCE, Bengaluru

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

`

Index

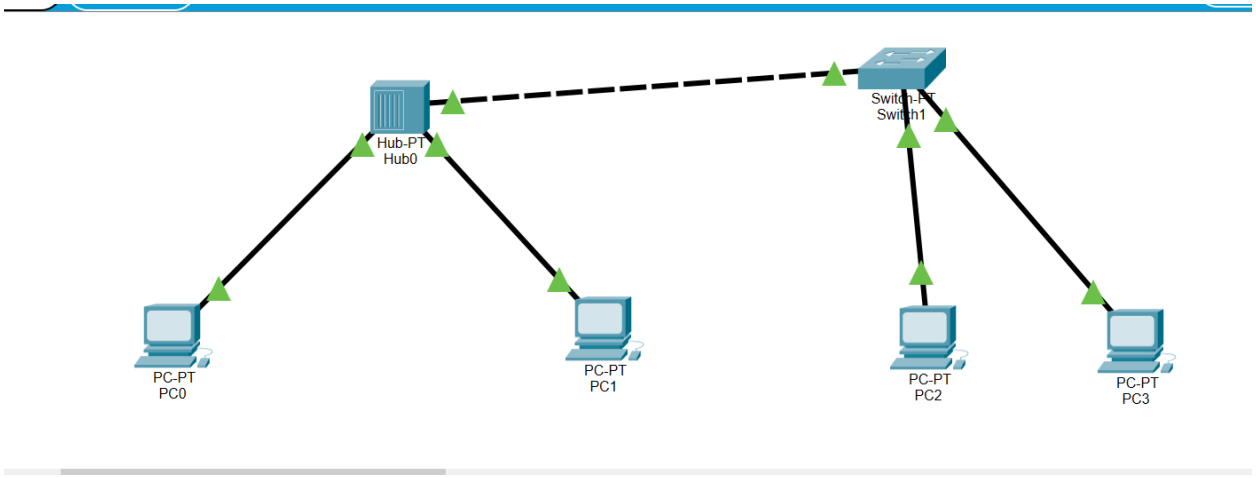
Sl. No.	Date	Experiment Title	Page No.
1		Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.	
4		Write a program for congestion control using Leaky bucket algorithm.	
5		Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	
6		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.	

Experiment No

Cycle-1

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

Topology



Experiment 1:- Creating a topology & Simulate sending a simple PDU from source to destination

Aim: using hub & switch as connecting devices

Date / /

Page 3 LAB2

TTL - Time To Live

Procedure: Hub

1. Hubs → Generic (1)
2. End Devices → Generic (desktop) (4 devices
3. Connect the device using black wire
4. Configure the IP address as 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4 so on.
5. Next in simulation mode add PDU from Src → dest & Click on auto capture/pl
6. we see that PDU's a being sent to all the devices but only to the correct destination. Only particular device accept it rest all reject it.

Switch

1. From device type selection box select devices by clicking end devices.
2. Connect in the similar manner as you did for the Hub.

Results

>> ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of da
Reply from 10.0.0.4 bytes: 32 times=
Statistics

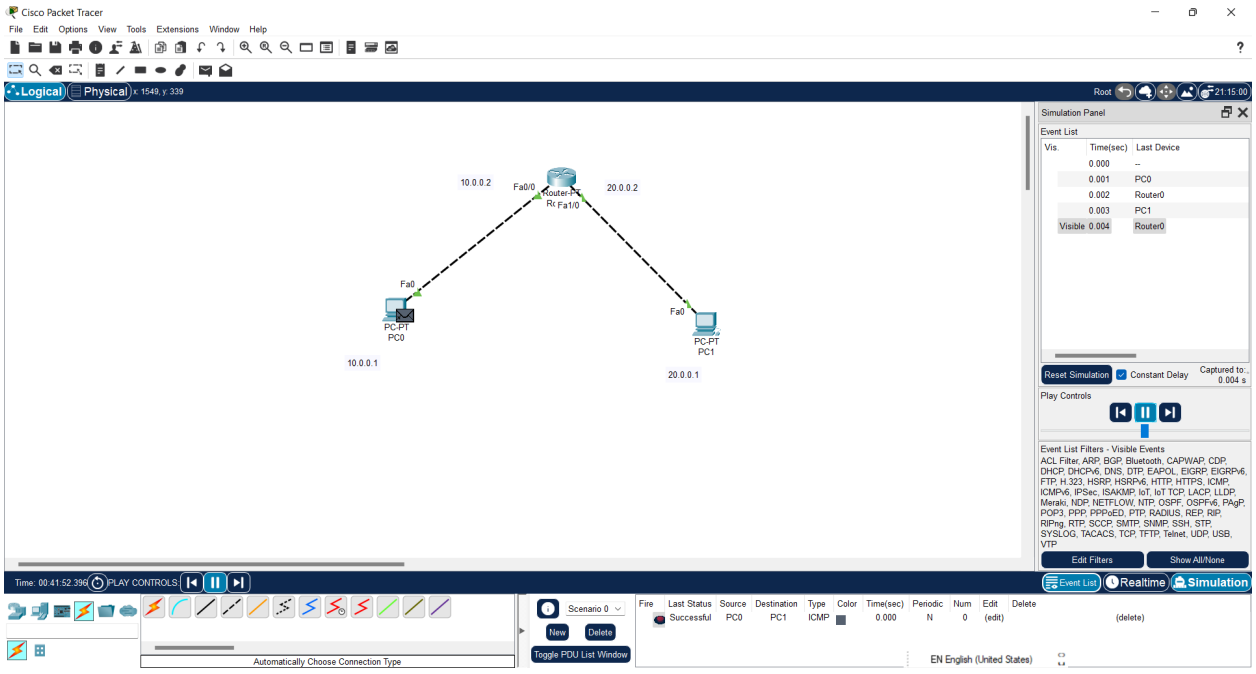
Packets: Sent = 4 Received = 4,

Round trip times

Minimum = 0ms, Max = 0ms, Ave

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

Topology:



PROCEDURE:

LAB-3

Date 17/11/22
Page 5

2. Aim: Configuring IP address to Routers in Packet Tracer.

Explore the following messages:
1) Ping Responses
2) Destination Unreachable
3) Request timed Out
4) Reply.

Procedure

1. Go to Routers → Router Generic → drag
2. Go to End device → Laptop-PT → drag & drop 2 laptops
3. Connections → Automatically Choose Connection type
4. Label the laptops → Config → IP Laptop as 10.0.0.1
6. Go to Router → CLI type the Commands - No dialog

```
Router > enable
Router # Configure terminal
Enter Configure Commands, one per line.
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.2
255.0.0.0
Router (config-if) # no shutdown
```

5. Configure for Laptop-PT 1 as 20.0.0.1 & subnet mask as well.

6. Coil. By clicking on Config I/O get to know the fast ethernet enter following Commands

interface fastethernet 1/0
ip address 20.0.0.2 255.0.0.0
no shutdown

You will get:- Interface FastEthernet 1/0, Changed State to up

7. Later, Laptop0 → Desktop → Command p -pt → Enter as follows

> ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of
Reply from 10.0.0.1: bytes=32 time=6m
TTL=128
Reply from 10.0.0.1: bytes=32 time
Ping Statistics for 10.0.0.1:
Packets: Sent=4, Received=4, Lost=0(0
Approximate round trip times in ms:
Minimum=6ms, Maximum=13ms, Avg

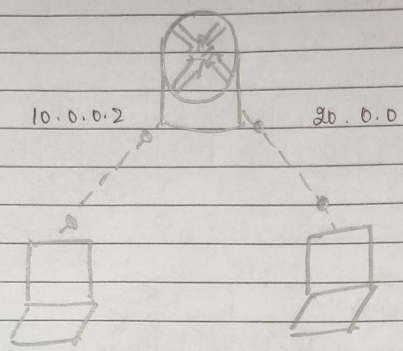
Try to ping for 20.0.0.1
Laptop1 → Desktop → Cmd prompt → pin
20.0.0.1
You will get the reply.

8. Laptop0 → Config → Gateway 20.0.0.0
Laptop1 → Config → Gateway 20.0.0.0

9. Simulation → Drop PDU Laptop0 to Lapt
until you get successful → Capture/
Forward

Date / /
Page 7

Topology



Laptop-PT	Laptop-PT
Laptop0	Laptop1
10.0.0.1	20.0.0.1

You will be getting request timed out when you try to ping 20.0.0.1 before Configuring other side of the router

PC> Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.

Request timed out.

⋮

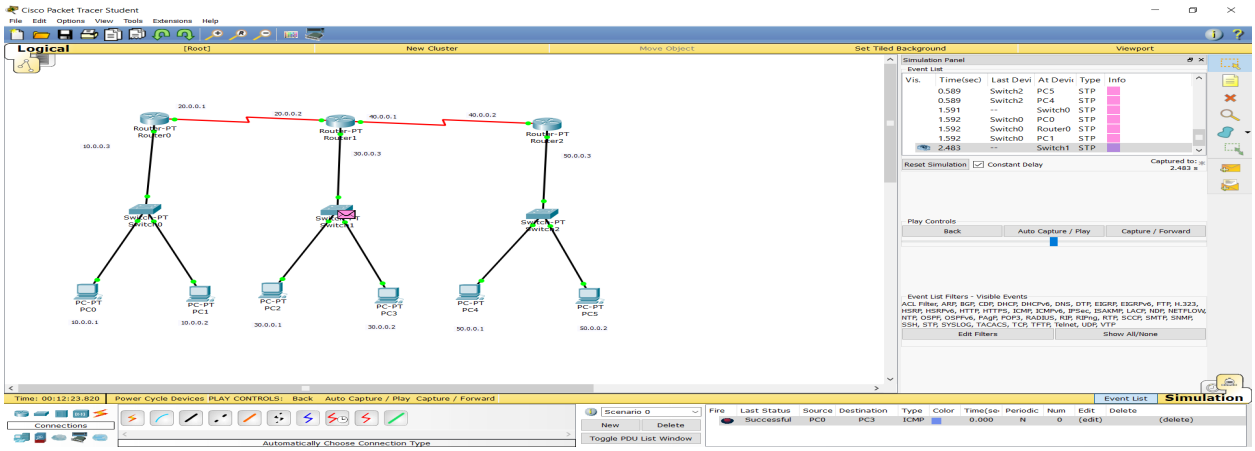
Ping Statistics for 20.0.0.1:

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

0.000	-	PC0	ICMP	☐
31.944		Router0	CDP	☐
31.944		Router0	CDP	☐
31.945	Router0	PC0	CDP	☐
31.945	Router0	PC1	CDP	☐

3. Configuring default route to the Router

Topology:



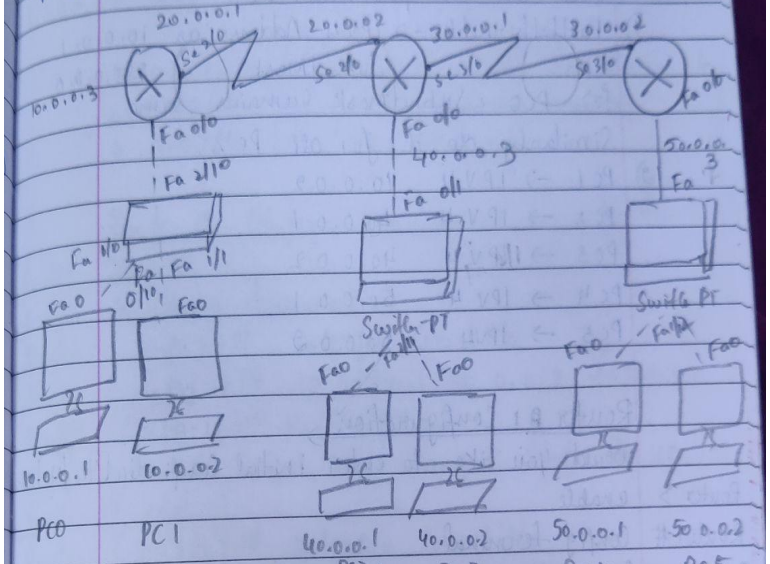
PROCEDURE:

LAB-5

Date 1/12/30
Page 13

Aim: Configuration of default route.

Topology:



Setting up Components →

1) End Routers → Router PT [3 no's]
2) Switches → Switch-PT [3 no's]
3) End devices → PC-PT [6 no's]

Connections →

1) Connections → Automatically choose Connection type
   [blw router → switches → PC]
2) Connections → Serial DCE
   [blw the routers]

Naming →

1) Routers - Router-PT Router0, Router-PT Router1, Router-PT Router2
2) Switches - Switch-PT Switch0, Switch-PT Switch1, Switch-PT Switch2

PC-PT PC-PT PC-PT PC-PT PC-PT

3) PC - PC0 PC1 PC2 PC3 PC4

Assigning IP to PC's →
click on the particular PC
1) FastEthernet0 → IPv4 Address as 10.0.0.1
SubnetMask 255.0.0.0
for PC0, SubnetMask remains same.
Similarly do it for all PC's.

2) PC1 → IPV4 10.0.0.2
PC2 → IPV4 40.0.0.1
PC3 → IPV4 40.0.0.2
PC4 → IPV4 50.0.0.1
PC5 → IPV4 50.0.0.2

Router 0 Configuration →
would you like to enter initial Config dia [Yes/N

Router > enable
Router # Config terminal
outer (Config)# interface fastEthernet 0/0
outer (Config-if)# ip address 10.0.0.3 255.0.0.0
outer (Config-if)# no Shutdown

Alternatively →
Select particular Router → Choose the type of
interface → IPv4 address type it & Sel
the Subnet mask.

Interface FastEthernet 0/0, Changed State to U
uter (Config-if)# exit
uter (Config)# interface Serial 2/0
uter (Config-if)# ip address 20.0.0.1 255.0.0
ter (Config-if)# no Shutdown

To check for Router 0 [enabled routes] →

Date
Page 15

Router > enable
Router # Config terminal
Router (Config)# ip route 0.0.00 0.0.0.0 20.0.0.2
Router # Show ip route

C 10.0.0.0/8 is directly Connected, FastEthernet 0/0
C 20.0.0.0/8 is directly Connected, Serial 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2

Router 2 Configuration →

Router >enable
Router # Config terminal
Router (config)# interface Serial 2/0
Router (Config-if)# ip address 20.0.0.2 255.0.0.0
Router (config-if)# no Shoutdown
Interface Serial 2/0, Changed State to up
Router # exit
Router (Config)# interface Serial 3/0
Router (Config-if)# ip address 30.0.0.1 255.0.0.0
Router (Config-if)# no Shutdown
Interface Serial 3/0, Changed State to down
Router # exit
Router (Config)# interface fastEthernet 0/0
Router (Config-if)# ip address 40.0.0.3 255.0.0.0
" # no Shutdown
Interface FastEthernet 0/0, Changed State to up
exit

Router> enable
Config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.2
exit
Show ip route

Set the gate to each ... 10.0.0.
say for PCO - Gateway

C 20.0.0.0/8 is dir conn, Serial 2/0
C 30.0.0.0/8 " " " " 3/0
C 40.0.0.0/8 " " " FastEthernet 0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.2

> enable
Config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.1
exit
Show ip route
C 20.0.0.0/8 is dir conn, Serial 2/0
C 30.0.0.0/8 " " " " 3/0
C 40.0.0.0/8 " " " FastEthernet
S* - [1/0] via 20.0.0.1

Router 3 Configuration >

Router> enable
Configure terminal
interface Serial 3/0
ip address 30.0.0.2 255.0.0.0
no Shutdown
Interface Serial 3/0, Changed State to up
interface fastethernet 0/0
ip address 50.0.0.3 255.0.0.0
no Shutdown
Interface FastEthernet 0/0, Changed State to
exit
> enable
Config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1
ip route 0.0.0.0 0.0.0.0 40.0.0.3
exit
uter # Show ip route

C 30.0.0.0/8 is directly connected, Serial3/0
C 50.0.0.0/8 is " " FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Pinging →

PC0 →

ping 50.0.0.2

pinging 50.0.0.2 with 32 bytes of data:
Reply from 50.0.0.2: bytes=32 time=33ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125

Ping Statistics for 50.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 33ms, Average = 9ms

PC3 →

(i) ping 40.0.0.3

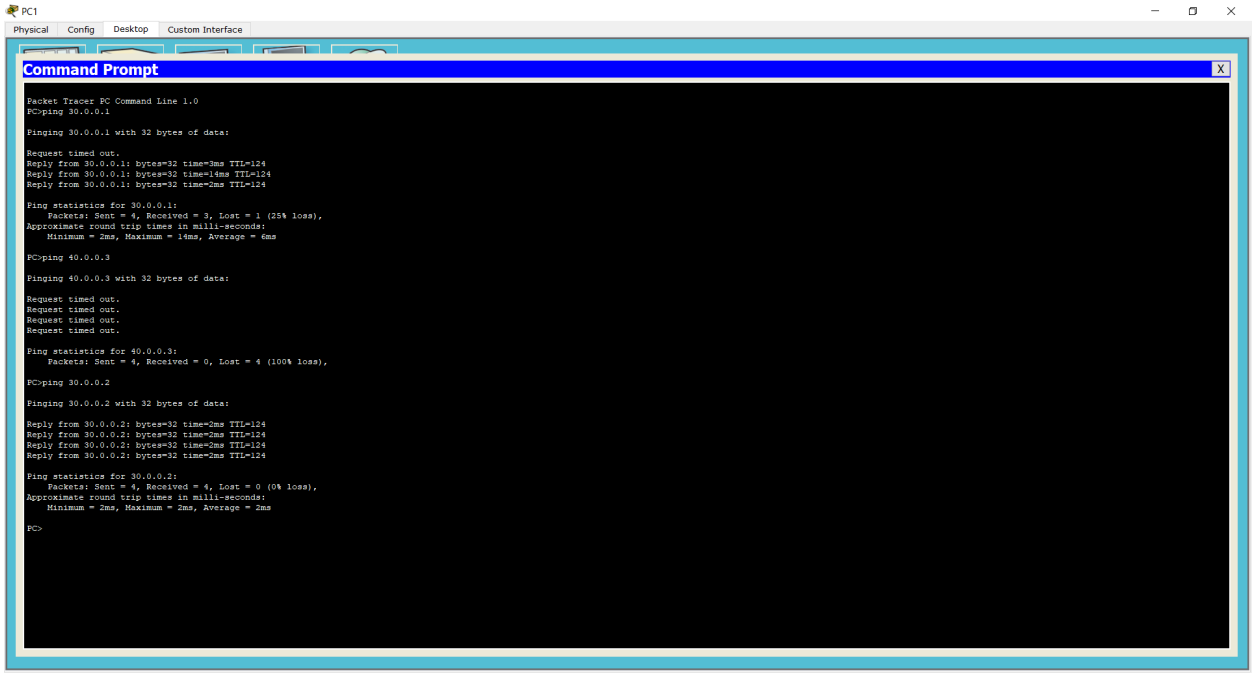
Pinging 40.0.0.3 with 32 bytes of data:
Reply from 40.0.0.3: bytes=32 time=33ms TTL=125
Reply " 40.0.0.3: bytes=32 time=2ms TTL=125
Reply " 40.0.0.3: bytes=32 time=2ms TTL=125
Reply " 40.0.0.3: bytes=32 time=2ms TTL=125

ping Statistics for 40.0.0.3
Packets Sent = 4, Received = 4, Lost = 0 (0% Loss)

(ii) ping 50.0.0.3

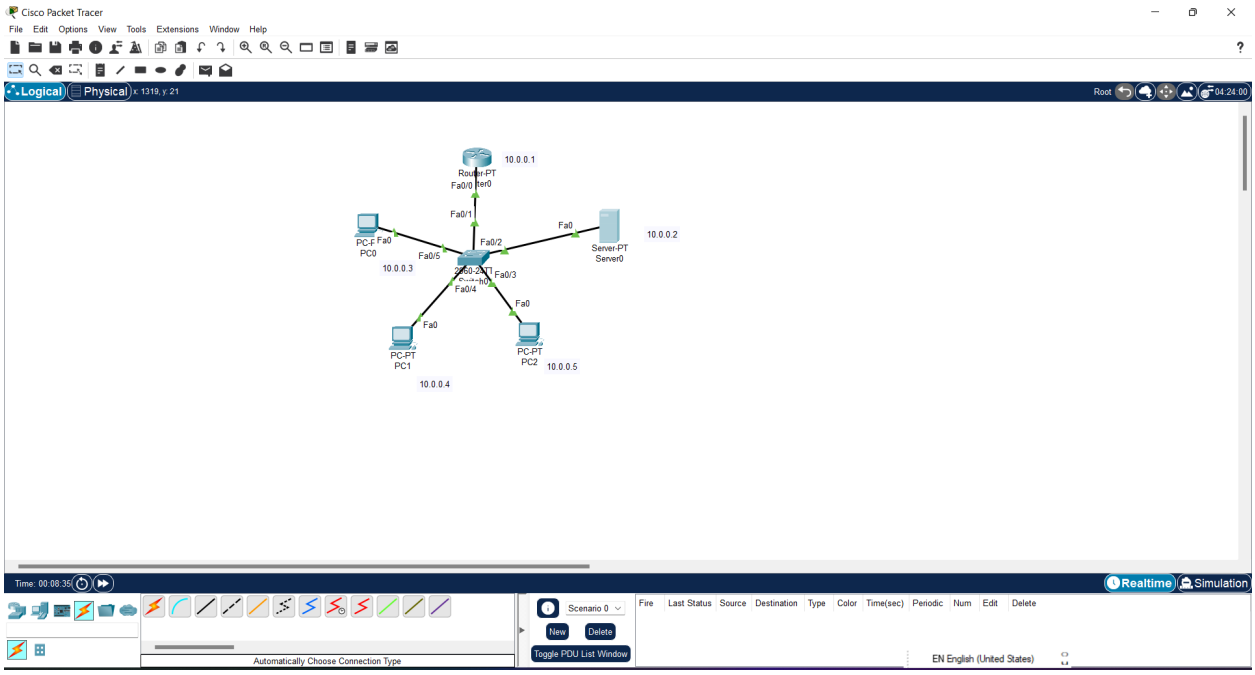
pinging 50.0.0.3 with 32 bytes of data:
Reply from 50.0.0.3: bytes=32 time=33ms TTL=125
Reply from " : bytes=32 time=33ms TTL=125
Reply from " : bytes=32 time=33ms TTL=125
Reply from " : bytes=32 time=33ms TTL=125

Observation: Packets are sent to any unknown destination to a single next hop address


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 30.0.0.1: bytes=32 time=3ms TTL=124
Reply from 30.0.0.1: bytes=32 time=14ms TTL=124
Reply from 30.0.0.1: bytes=32 time=2ms TTL=124
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 14ms, Average = 6ms
PC>ping 40.0.0.3
Pinging 40.0.0.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 40.0.0.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=2ms TTL=124
Reply from 30.0.0.2: bytes=32 time=2ms TTL=124
Reply from 30.0.0.2: bytes=32 time=2ms TTL=124
Reply from 30.0.0.2: bytes=32 time=2ms TTL=124
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 2ms, Average = 2ms
PC>

4. Configuring DHCP within a LAN in a packet Tracer

TOPOLOGY:



PROCEDURE:

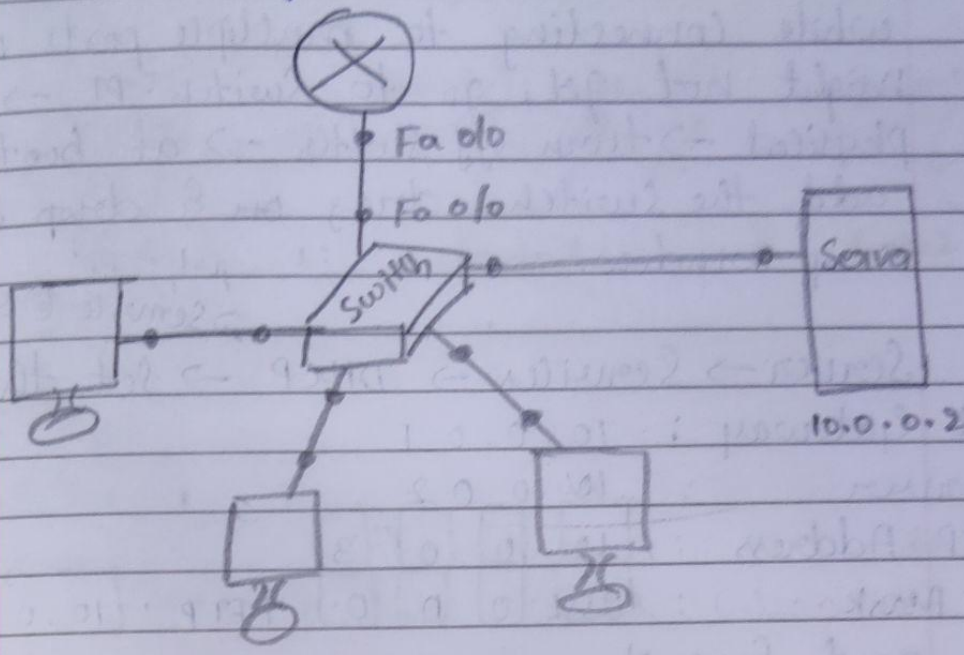
LAB-6

Configuring DHCP within a LAN in packet Tracer

Date 8/12/22
Page 19

Aim: Students will design n/w based on topology of nodes & req given 10.0.0.1 & Simulate the topology.

Topology:



Setting up Components →

1) Routers → Router-PT [1 no]
2) Switches → Switch-PT [1 no]
3) End Devices → Generic [3 no's]
4) End Devices → Server-PT [1 no's]

Connections

1) Connections → Automatically Choose Connection Typ
   [for all the devices]

IP addresses & Default Gateway of Server →

1) Server → Config → Gateway as 10.0.0.1
   Fast Ethernet0 → IP Address 10.0.0.2
   Subnet Mask 255.0.0.0

2) Router → CLI

Router →

Interface FastEthernet 0/0, Changed State to

Note: while connecting to multiple ports you might not get, go to Switch-PT → physical → turn off Switch → at bottom add the Switch, drag & drop on any window.

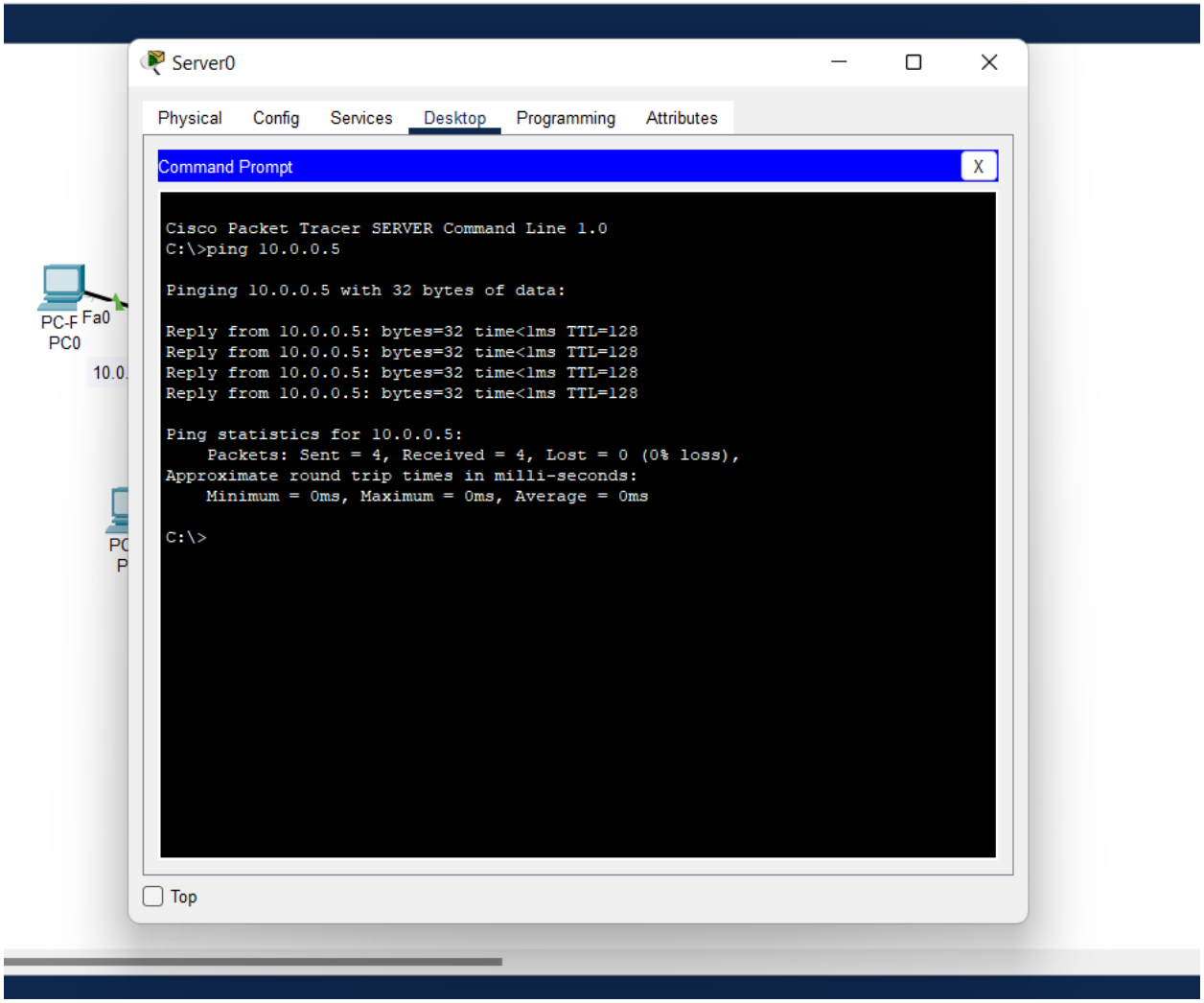
Server → Services → DHCP → Set the → Servile On

Default Gateway : 10.0.0.1
DNS Server : 10.0.0.2
Start IP Address : 10 0 0 3
Subnet Mask : 255 0 0 0
TFTP : 10.0.0.2
and Save it

Turn on the Switch, to turn the links green.

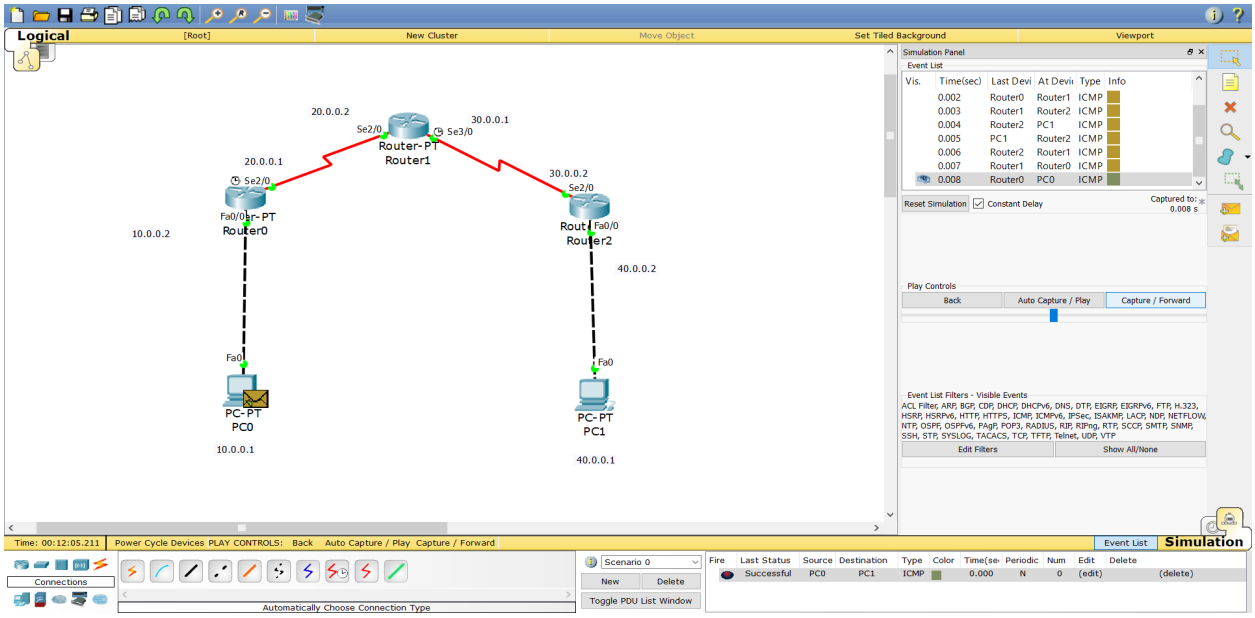
More About DHCP →

Dynamic Host Configuration Protocol is a client/server protocol that automatically provides an IP host with its IP # & other related configuration information such as the subnet mask and default gateway


Server0
Physical
Config
Services
Desktop
Programming
Attributes
Command Prompt
X
Cisco Packet Tracer SERVER Command Line 1.0
C:\>ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128
Ping statistics for 10.0.0.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
Top
PC-F
PC0
Fa0

5. Configuring RIP Routing Protocol in Routers.

TOPOLOGY:



PROCEDURE:

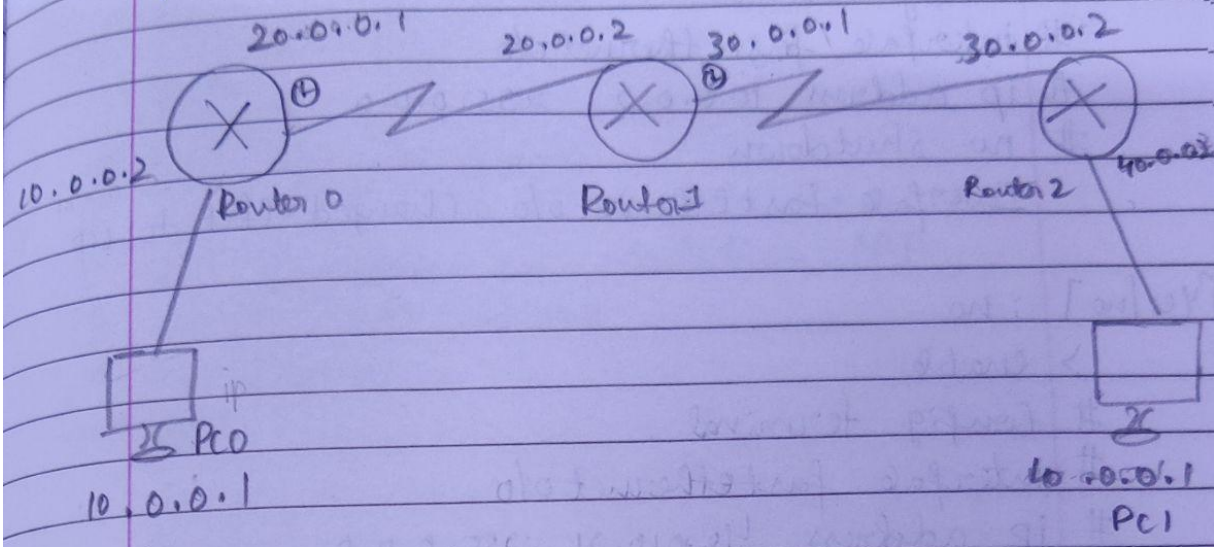
LAB - 7

Aim: Configuring RIP Routing Protocol in Routers

Gate PC0 10.0.0.2
PC1 40.0.0.2

Date 15/12/22
Page 21

Aim: Should design a n/w, apply the learnd prog tools and justify the usage



Setting up components →

1) Routers → Router-PT [3 no's]
2) End Devices → Generic-PT [2 no's]

Connections;

1) B/w Routers & PC, Select Automatic Connection type.
2) B/w the Routers, Select Serial DCE

Gateways for PC's →

1) PC0 → Config → Gateway 10.0.0.2
2) PC1 → Config → Gateway 40.0.0.2

IP address for PC's →

1) PC0 → config → FastEthernet0 → IP 10.0.0.1
Subnet 255.0.0

2) PC1 → Config → FastEthernet0 → IP 40.0.0.1

Configuring Routers for FastEthernet → Router0

[Yes/no] : no
> enable
Config terminal
interface fastethernet 0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
Interface FastEthernet 0/0, Changed State to up

[Yes/no] : no
> enable
Config terminal
interface fastethernet 0/0
ip address 40.0.0.2 255.0.0.0
no shutdown
Interface FastEthernet 0/0, Changed State to up

Configuring Router for Serial

NOTE: For Serial Conn with ① we need clock rate 64000, no need for other end.

Router0

interface Serial 2/0
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
Clock rate 64000
no shutdown
Interface Serial 2/0, Changed State to up

Router1

Yes/NO] : no
>enable

encapsulation PPP
no Shutdown
Interface Serial 2/0, Changed State to up

Router 1

interface Serial 3/0
ip address 30.0.0.1 255.0.0.0
no Shutdown encapsulation PPP
Clock rate 64000
no Shutdown
exit

Router 2

interface Serial 3/0
ip address 30.0.0.2 255
encapsulation PPP
no Shutdown
exit

Router 0 >

># enable
Config terminal
router rip
network 10.0.0.0
network 20.0.0.0
exit

Router 1

router rip

Router 2

router rip
network 30.0.0.0
network 40.0.0.0
exit

Ping> PC0>

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 =1ms
Reply from 40.0.0.1 : bytes =32 time =2ms TTL
Reply from 40.0.0.1 : bytes =32 time =21 ms TTL

Ping Stastistics for 40.0.0.1:
Packets : Sent =4, Received = 3, Lost =1

PC> ping 40.0.0.1
Reply from 40.0.0.1 bytes =32 time
Reply from 40.0.0.1 bytes =32 time =
Reply from 40.0.0.1 bytes =32 time =
Reply from 40.0.0.1 bytes =32 time =

Ping Stastistics for 40.0.0.1
Packets : Sent =4, Received = 4, Lost =0
Approximate round trip times in milli-seconds
Minimum =14 ms, Maximum = 27 ms, Average

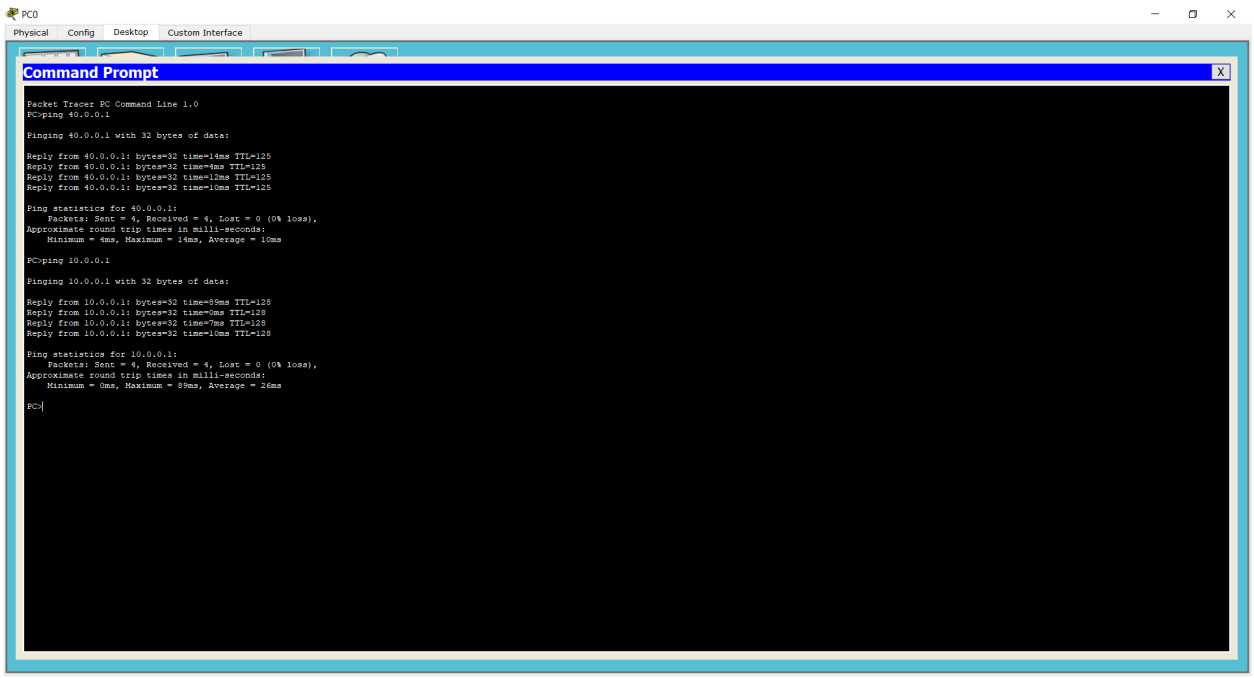
PC 1 >

PC> Ping 10.0.0.1
pinging 10.0.0.1 with 32 bytes :
Reply from 10.0.0.1 bytes =32 Time

Reply from 10.0.0.1 : bytes = 32 time = 2ms
TTL = 125

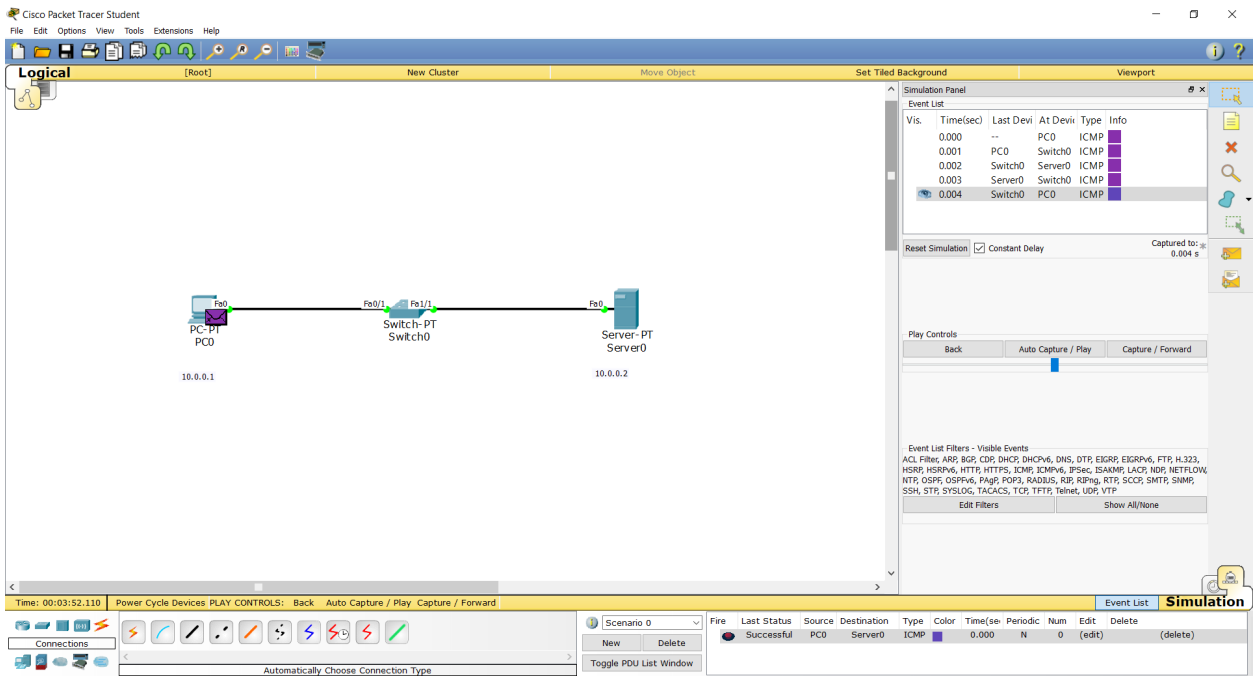
Ping statistics for 10.0.0.1:
Packets : Sent = 4, Received = 4, Lost = 0
(0% loss)

Observation: Ping message is sent from PC0 → PC1, eventhough hop address is not set.



6. Demonstration of WEB server and DNS using Packet Tracer.

TOPOLOGY:

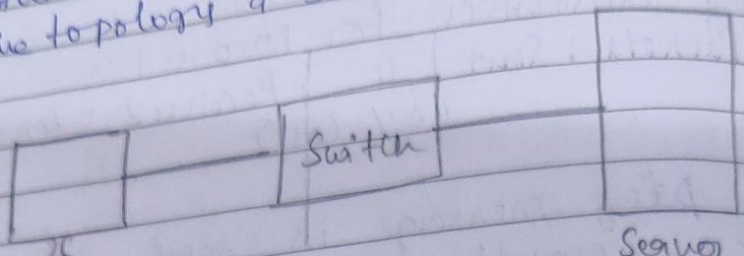


LAB - 7

Aim: Demonstration of WEB Server and DNS using Packet Tracer

7.

Aim: Students should design a network based the topology & Simulate the topology



1) End Devices → Generic [1 no's]
2) Switches → Generic [1 no]
3) End Devices → Server-PT → Generic [1 no]

IP to PC0 →

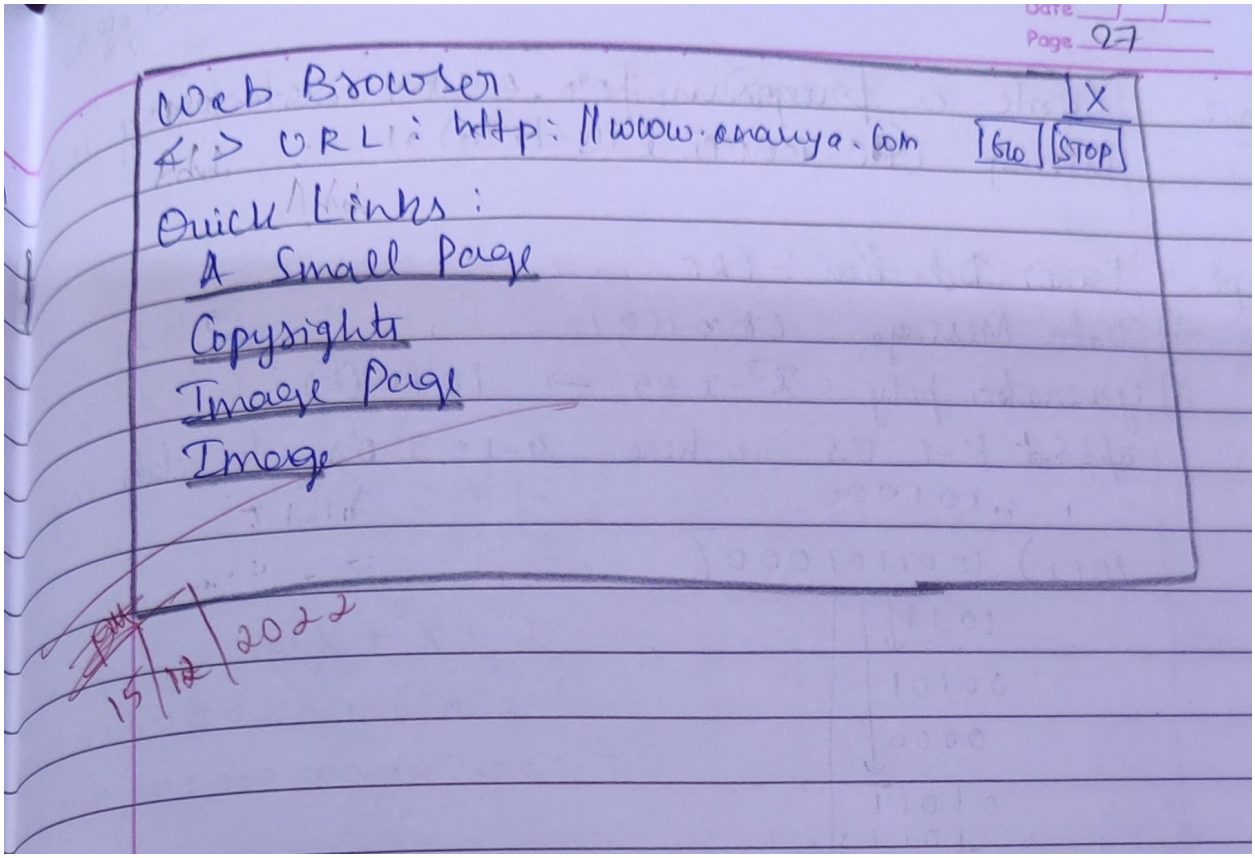
1) PC0 → Config → FastEthernet0 →
   IP address 10.0.0.1
   Subnet Mask 255.0.0.0

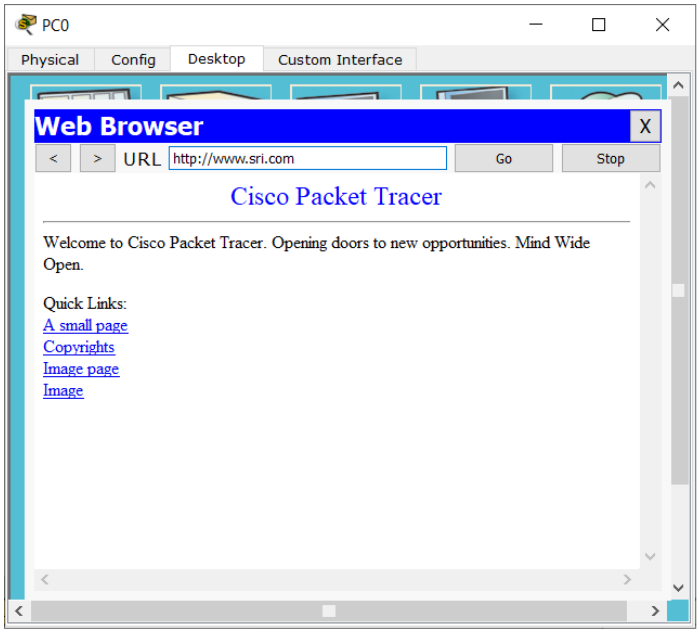
IP to Server0 →

1) Server0 → FastEthernet0 → IP 10.0.0.2
   Subnet 255.0.0.0

1) Server0 → Services → HTTP ⊙ on [Should check with DNS also,
2) DNS → DNS Service ⊙ on
3) Name: www.ananya.com
   Address: 10.0.0.2 [Server IP]
4) Add → Save
5) PC0 → Desktop → Web Browser →
   URL www.ananya.com → Go

links will be displayed


Date
Page 27
Web Browser
X
URL : http://www.anaya.com
Go
STOP
Quick Links :
A Small Page
Copysights
Image Page
Image
15/12/2022


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL
http://www.sri.com
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

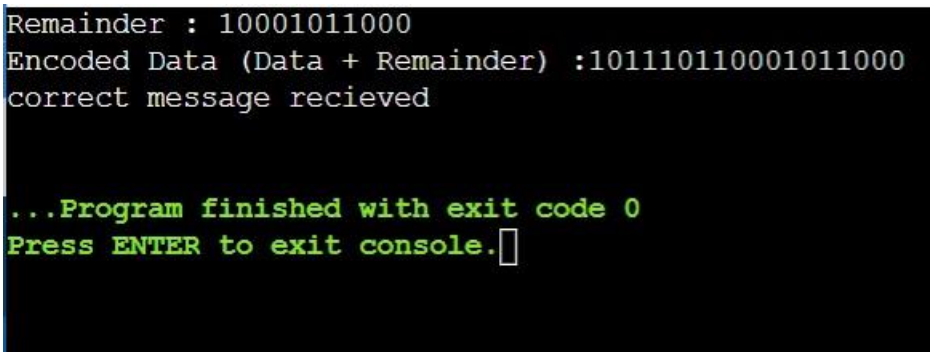
```
#include<bits/stdc++.h> using
namespace std; void receiver(string
data, string key);

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;
}
```

Output



Experiment No 2 Aim of the Experiment

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

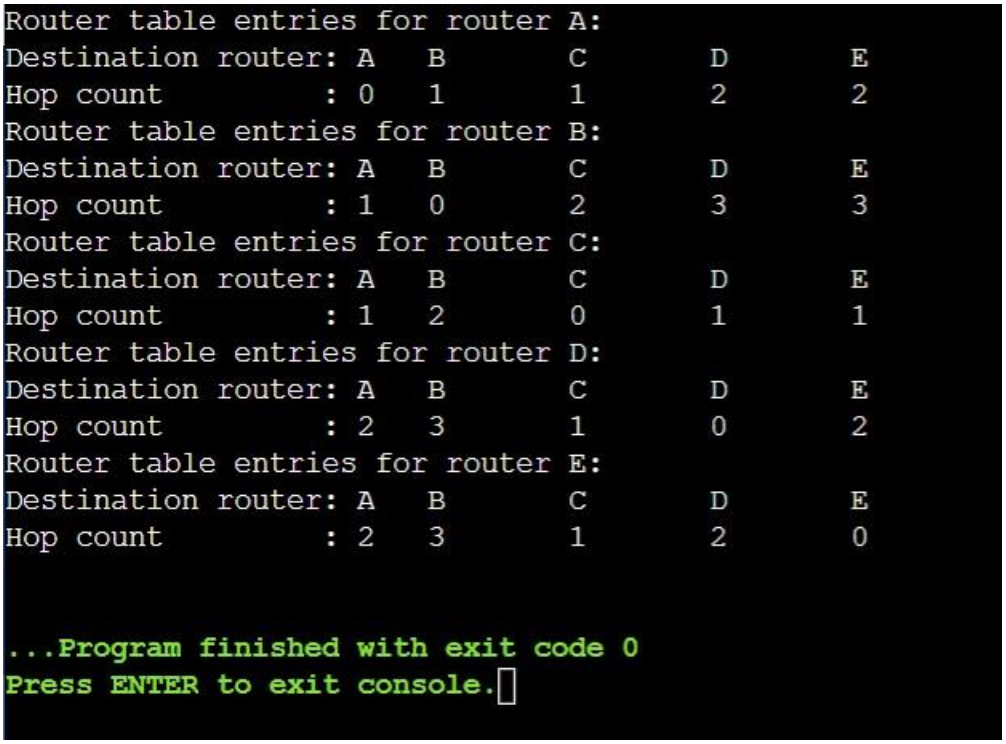
Code

```
#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;
}
```

Output:



Experiment No 3 Aim

of the Experiment

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

Code

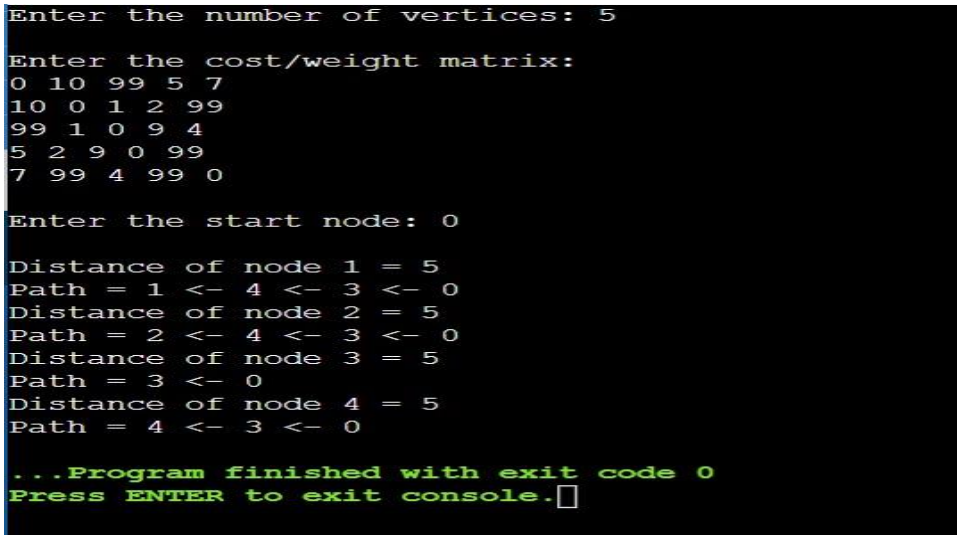
```
#include <stdio.h>
#include <stdlib.h>
```

```
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;
}
```

Output:



Experiment No 4 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()
```

Output

```
Enter file namemain.cpp
From Server: #include <bits/stdc++.h>
using namespace std

class Node{

        bool color = 0; // 1 -> black; 0 -> red
        Node *left = NULL;
        Node *right = NULL;
        Node *parent = NULL;
        int key;

        Node(int k)
        {
                key = k;
        }

};
```

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)
```

Program-5

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.

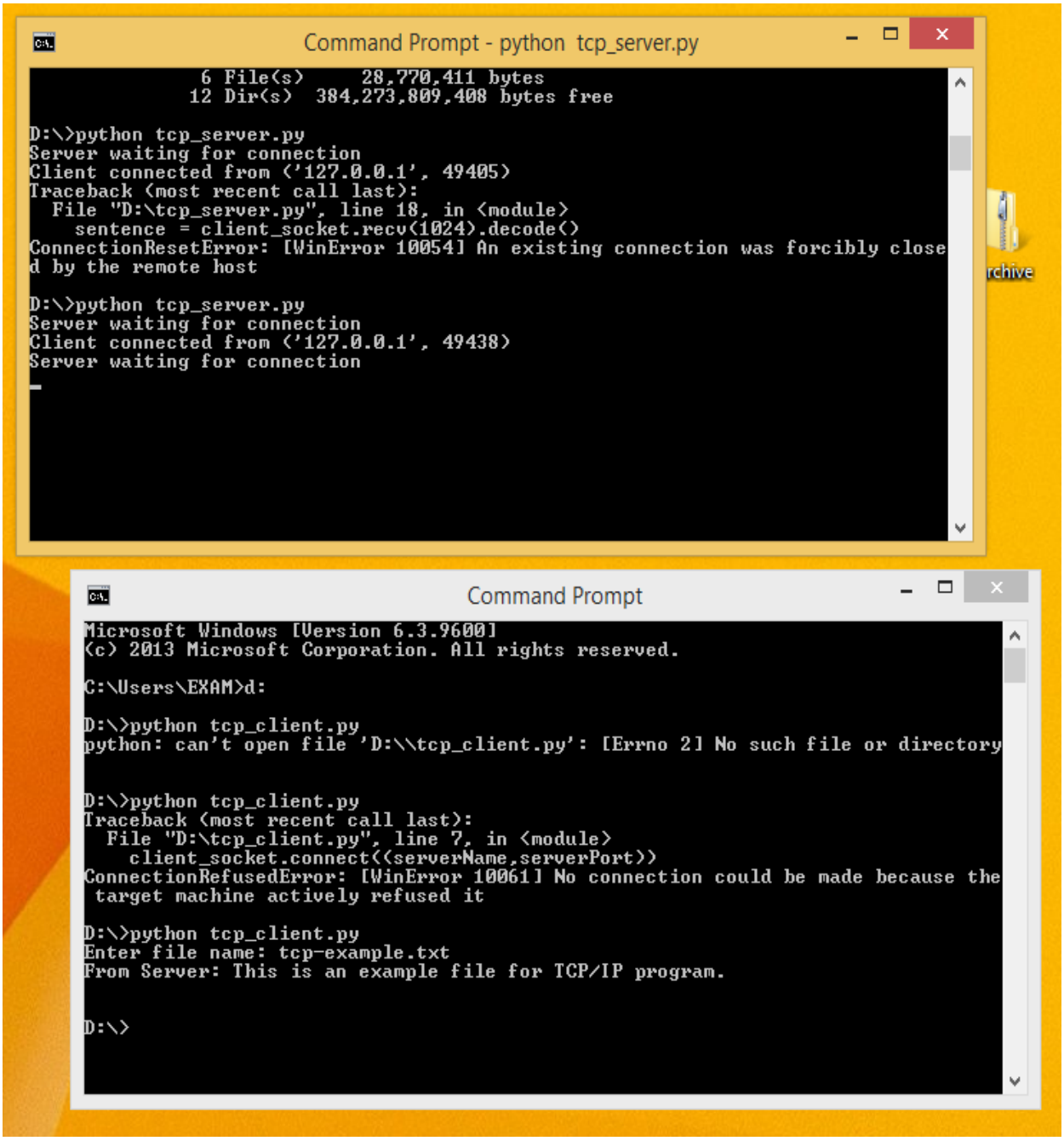
CODE:

SERVER

```
import socket

serverName = '127.0.0.1'
serverPort = 12345
#create
server_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

#bind
server_socket.bind((serverName, serverPort ))

#listen
server_socket.listen(5)

while True:
    print("Server waiting for connection")
    client_socket, addr = server_socket.accept()
    print("Client connected from",addr)
    sentence = client_socket.recv(1024).decode()

    file = open(sentence, "r")
    l = file.read(1024)

    client_socket.send(l.encode())
    file.close()
    client_socket.close()
```

CLIENT

```
import socket

serverName = '127.0.0.1'
serverPort = 12345

client_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
client_socket.connect((serverName,serverPort))
sentence = input("Enter file name: ")

client_socket.send(sentence.encode())
filecontents = client_socket.recv(1024).decode()
print ('From Server:', filecontents)
client_socket.close()
```


Command Prompt - python tcp_server.py
6 File(s) 28,770,411 bytes
12 Dir(s) 384,273,809,408 bytes free
D:\>python tcp_server.py
Server waiting for connection
Client connected from ('127.0.0.1', 49405)
Traceback (most recent call last):
File "D:\tcp_server.py", line 18, in <module>
sentence = client_socket.recv(1024).decode()
ConnectionResetError: [WinError 10054] An existing connection was forcibly close
d by the remote host
D:\>python tcp_server.py
Server waiting for connection
Client connected from ('127.0.0.1', 49438)
Server waiting for connection
rchive
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\EXAM>d:
D:\>python tcp_client.py
python: can't open file 'D:\\tcp_client.py': [Errno 2] No such file or directory
D:\>python tcp_client.py
Traceback (most recent call last):
File "D:\tcp_client.py", line 7, in <module>
client_socket.connect((serverName,serverPort))
ConnectionRefusedError: [WinError 10061] No connection could be made because the
target machine actively refused it
D:\>python tcp_client.py
Enter file name: tcp-example.txt
From Server: This is an example file for TCP/IP program.
D:\>

Program-6

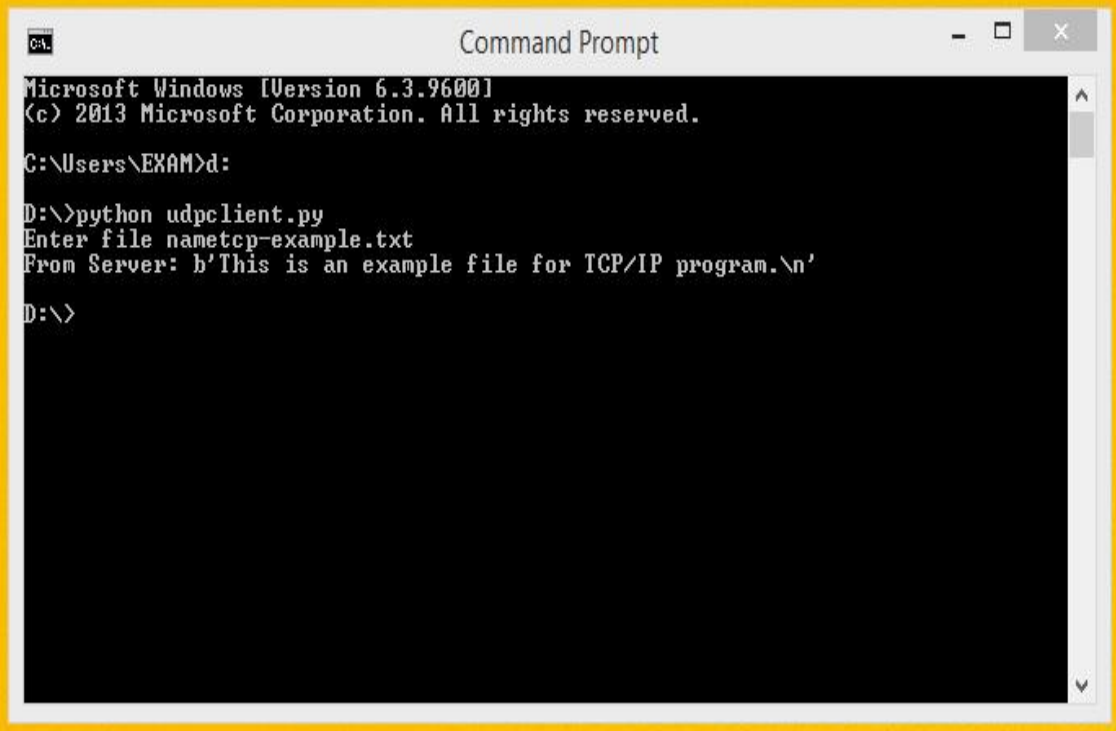
Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

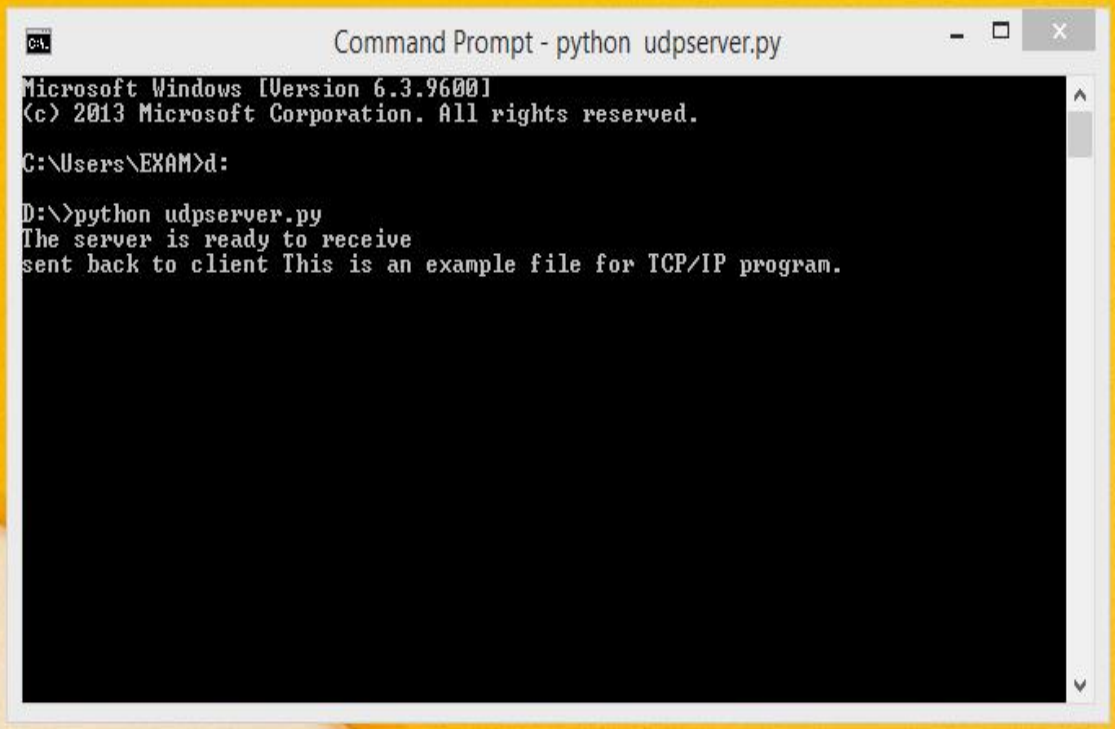
SERVER

```
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()
```


Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\EXAM>d:
D:\>python udpclient.py
Enter file nametcp-example.txt
From Server: b'This is an example file for TCP/IP program.\n'
D:\>


Command Prompt - python udpserver.py
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\EXAM>d:
D:\>python udpserver.py
The server is ready to receive
sent back to client This is an example file for TCP/IP program.