VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Computer Networks

Submitted by

AMRUTHA MURALIDHAR

(1BM21CS257)

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

June-2023 to September-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 **Amrutha Muralidhar (1BM21CS257),** 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 academic semester June-2023 to September-2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks (22CS4PCCON)** work prescribed for the said degree.

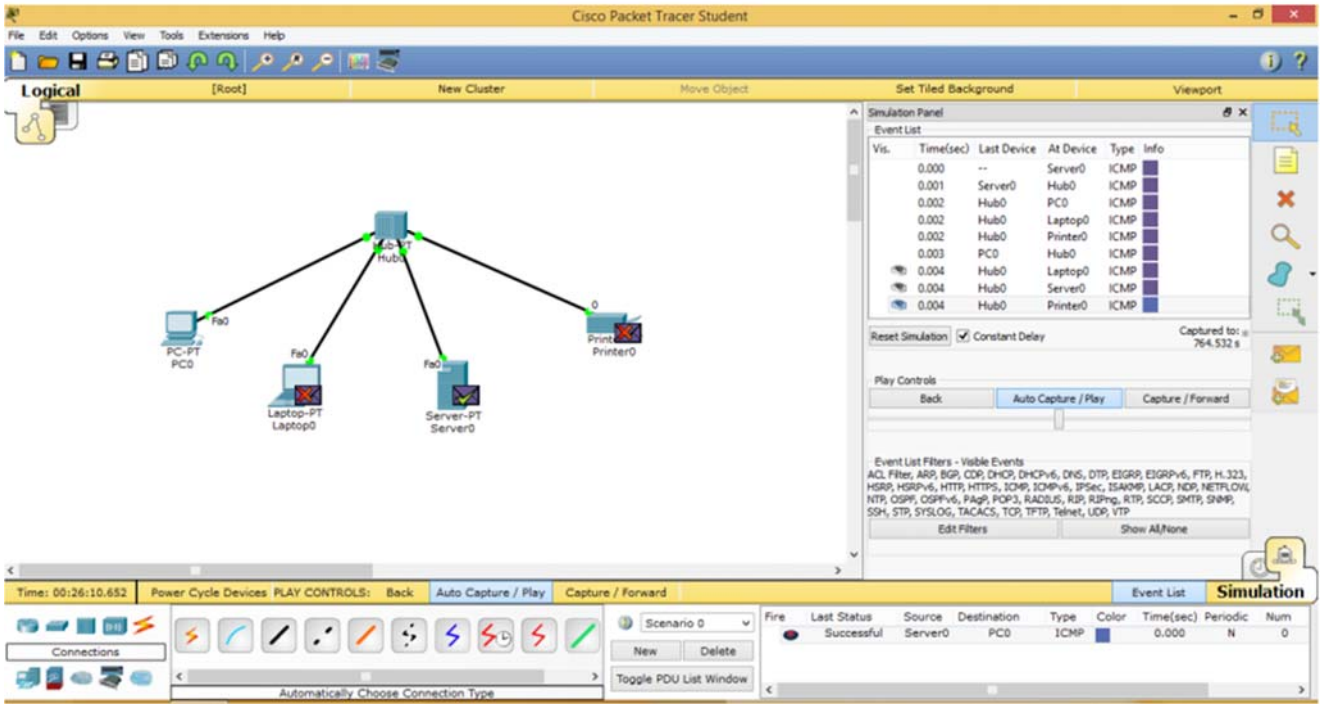
Dr. Shyamala G	Dr. Jyothi S Nayak
Assistant Professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

Index Sheet

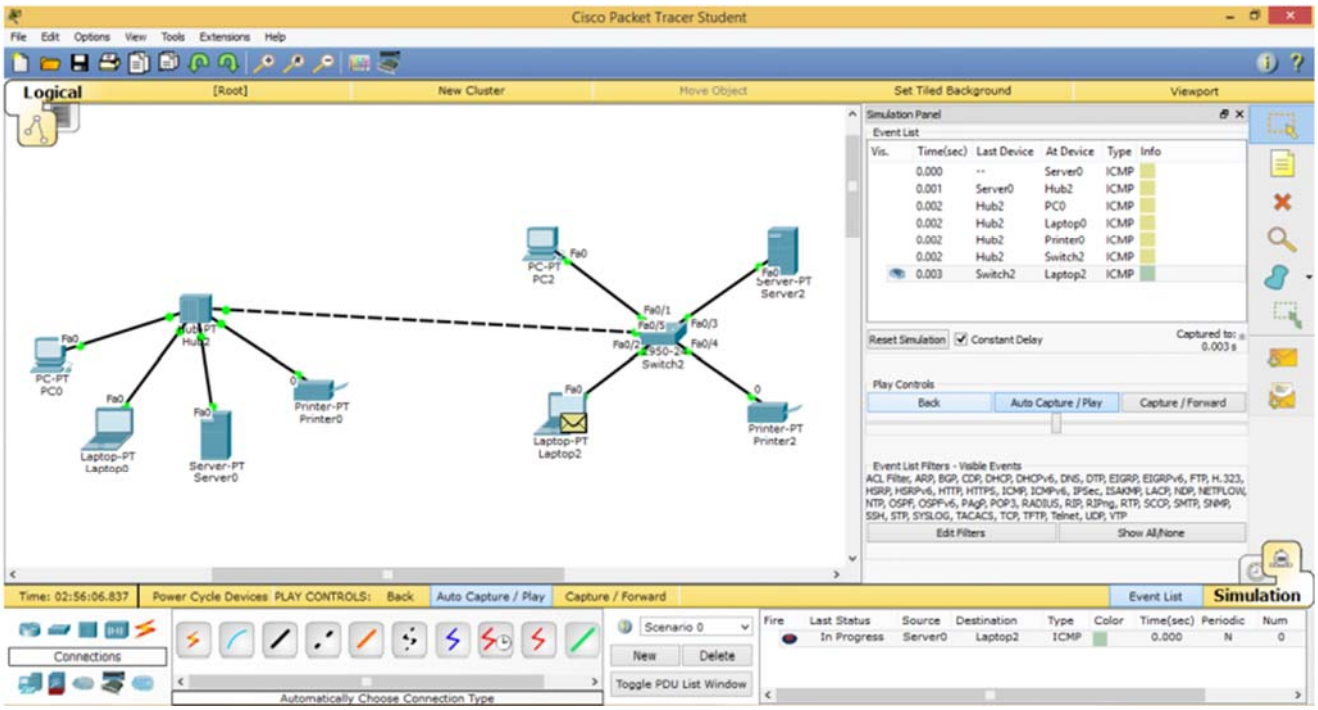
Lab Program No.	Program Details
1	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.
2	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply
3	Configure default route, static route to the Router
4	Configure DHCP within a LAN and outside LAN.
5	Configure RIP routing Protocol in Routers
6	Configure OSPF routing protocol
7	Demonstrate the TTL/ Life of a Packet
8	Configure Web Server, DNS within a LAN.
9	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)
10	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.
11	To construct a VLAN and make the PC's communicate among a VLAN
12	To construct a WLAN and make the nodes communicate wirelessly
13	Write a program for error detecting code using CRCCCITT (16-bits).
14	Write a program for congestion control using Leaky bucket algorithm
15	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
16	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

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

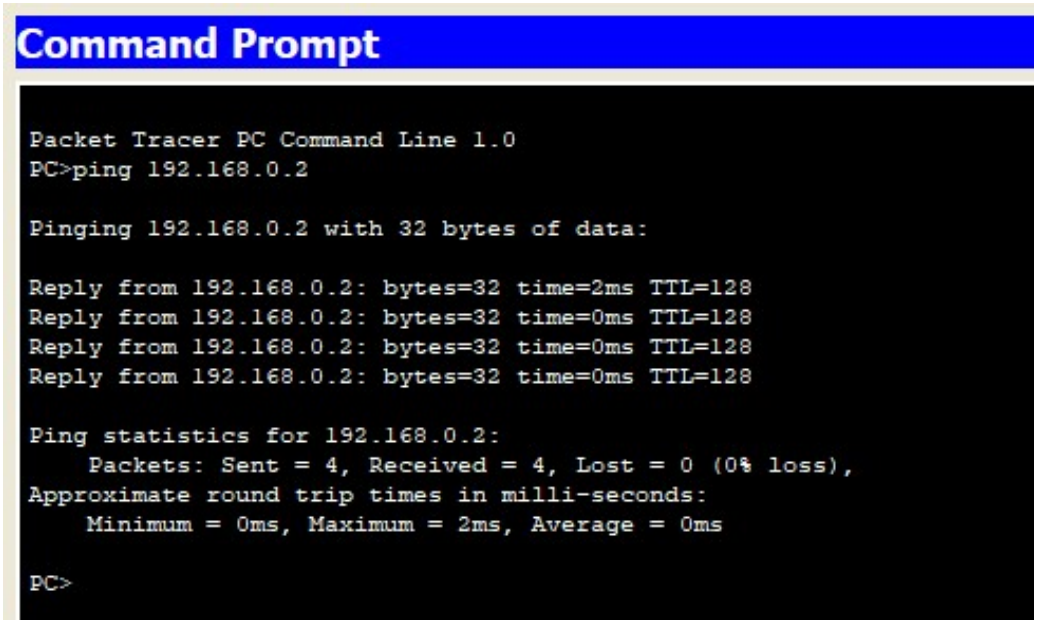
Hub Topology



Switch Topology

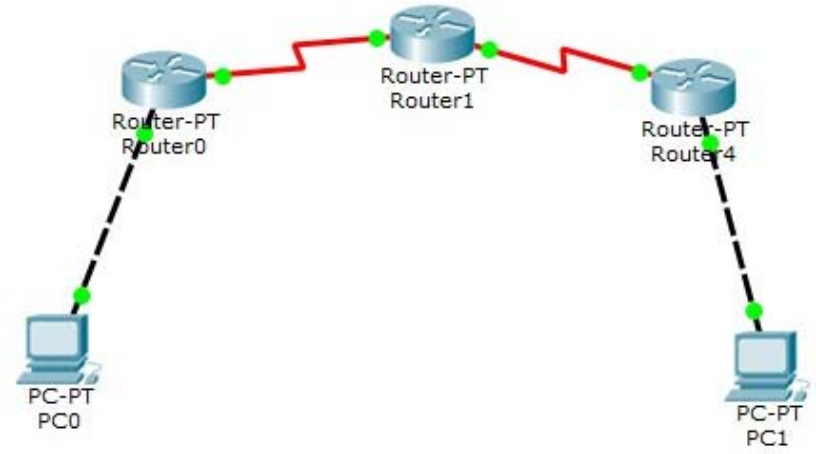


Output

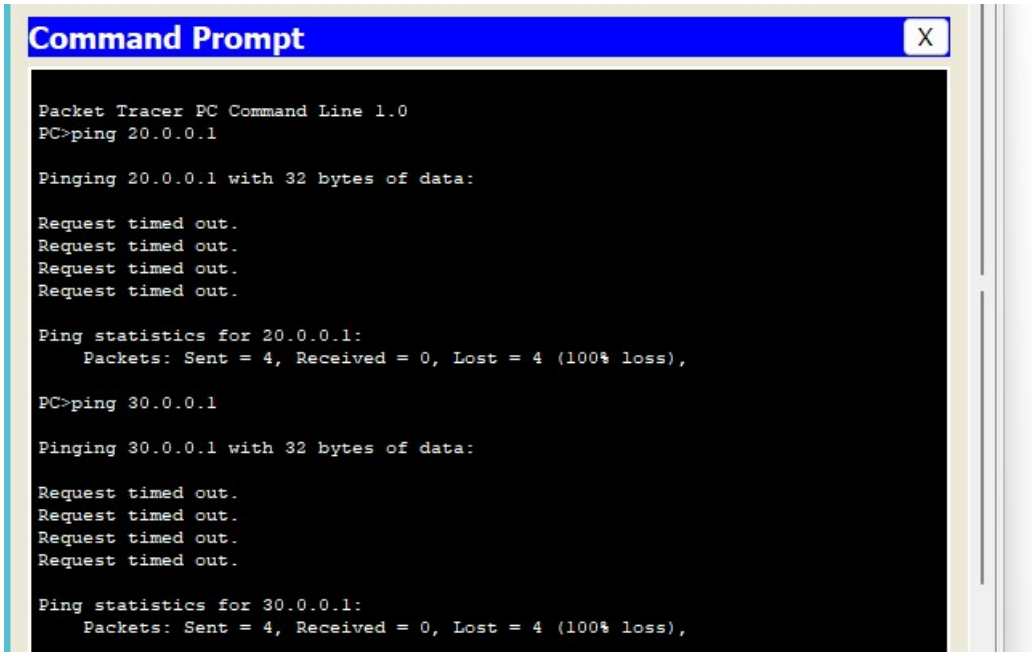

Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.0.2
Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time=2ms TTL=128
Reply from 192.168.0.2: bytes=32 time=0ms TTL=128
Reply from 192.168.0.2: bytes=32 time=0ms TTL=128
Reply from 192.168.0.2: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms
PC>

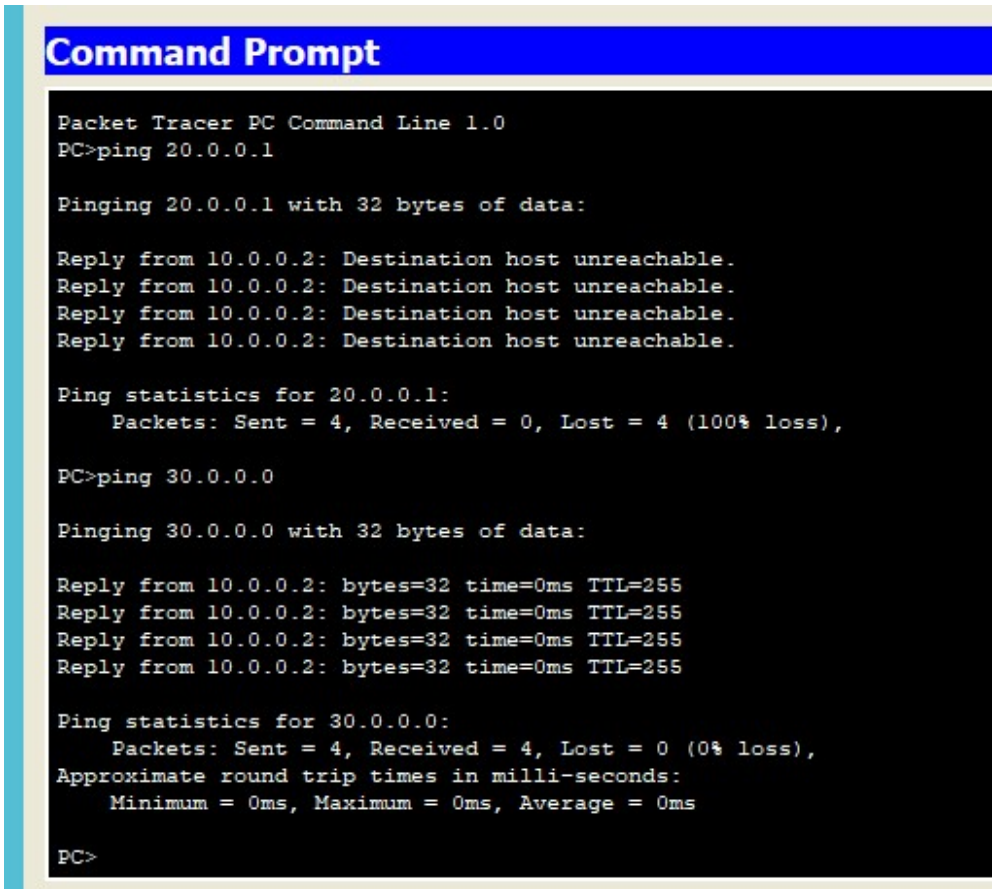
2. **Configuring IP address to Routers in Packet Tracer. Exploring the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.**

Topology



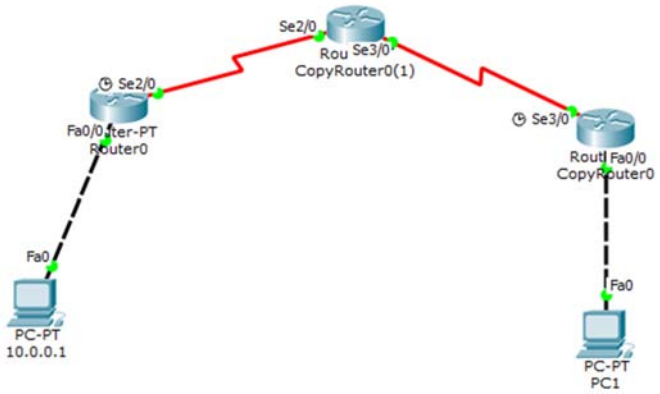
Output




Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes 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 for 20.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 30.0.0.0
Pinging 30.0.0.0 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Ping statistics for 30.0.0.0:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

3. **Configure default route, static route to the Router.**

Topology



Output

Router0

```
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 30.0.0.2 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    30.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 30.0.0.2
Router>
```

Router1

```
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

S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 40.0.0.2
C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, Serial3/0
Router>
```

Router2

```
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 40.0.0.1 to network 0.0.0.0

C    20.0.0.0/8 is directly connected, FastEthernet0/0
C    40.0.0.0/8 is directly connected, Serial3/0
S*   0.0.0.0/0 [1/0] via 40.0.0.1
Router>
```

Command Prompt

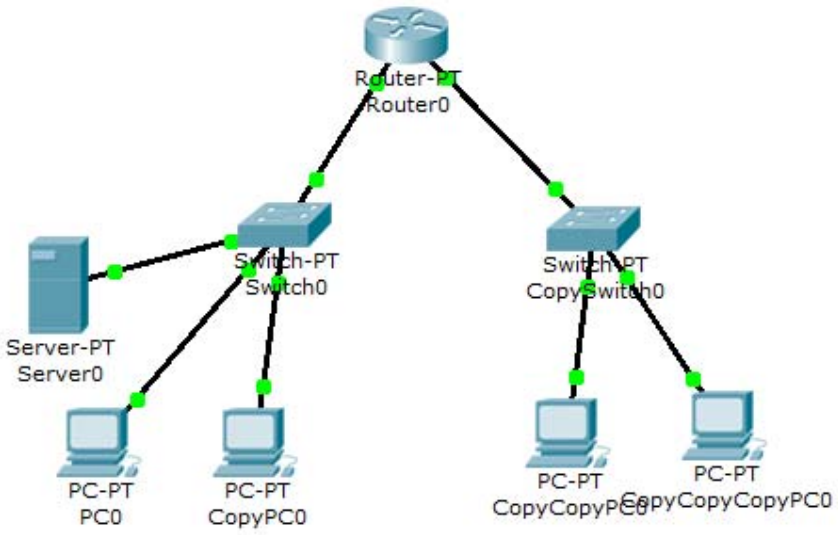
```
PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=10ms TTL=253
Reply from 20.0.0.1: bytes=32 time=9ms TTL=253
Reply from 20.0.0.1: bytes=32 time=6ms TTL=253
Reply from 20.0.0.1: bytes=32 time=7ms TTL=253

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 10ms, Average = 8ms

PC>ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

Reply from 30.0.0.1: bytes=32 time=0ms TTL=255
Reply from 30.0.0.1: bytes=32 time=0ms TTL=255
Reply from 30.0.0.1: bytes=32 time=0ms TTL=255
Reply from 30.0.0.1: bytes=32 time=0ms TTL=255

Ping statistics for 30.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

Command Prompt

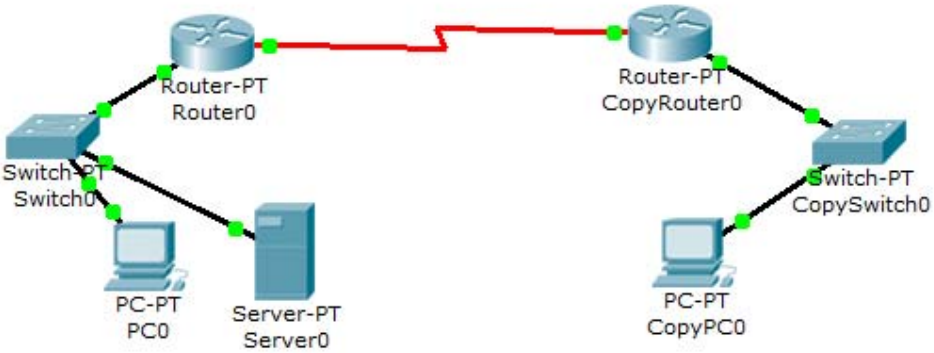
```
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=7ms TTL=253
Reply from 10.0.0.1: bytes=32 time=10ms TTL=253
Reply from 10.0.0.1: bytes=32 time=9ms TTL=253
Reply from 10.0.0.1: bytes=32 time=6ms TTL=253

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 10ms, Average = 8ms

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time=9ms TTL=254
Reply from 40.0.0.1: bytes=32 time=3ms TTL=254
Reply from 40.0.0.1: bytes=32 time=5ms TTL=254
Reply from 40.0.0.1: bytes=32 time=3ms TTL=254

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 9ms, Average = 5ms

PC>
```

4. **Configure DHCP within a LAN and outside LAN.**

Topology



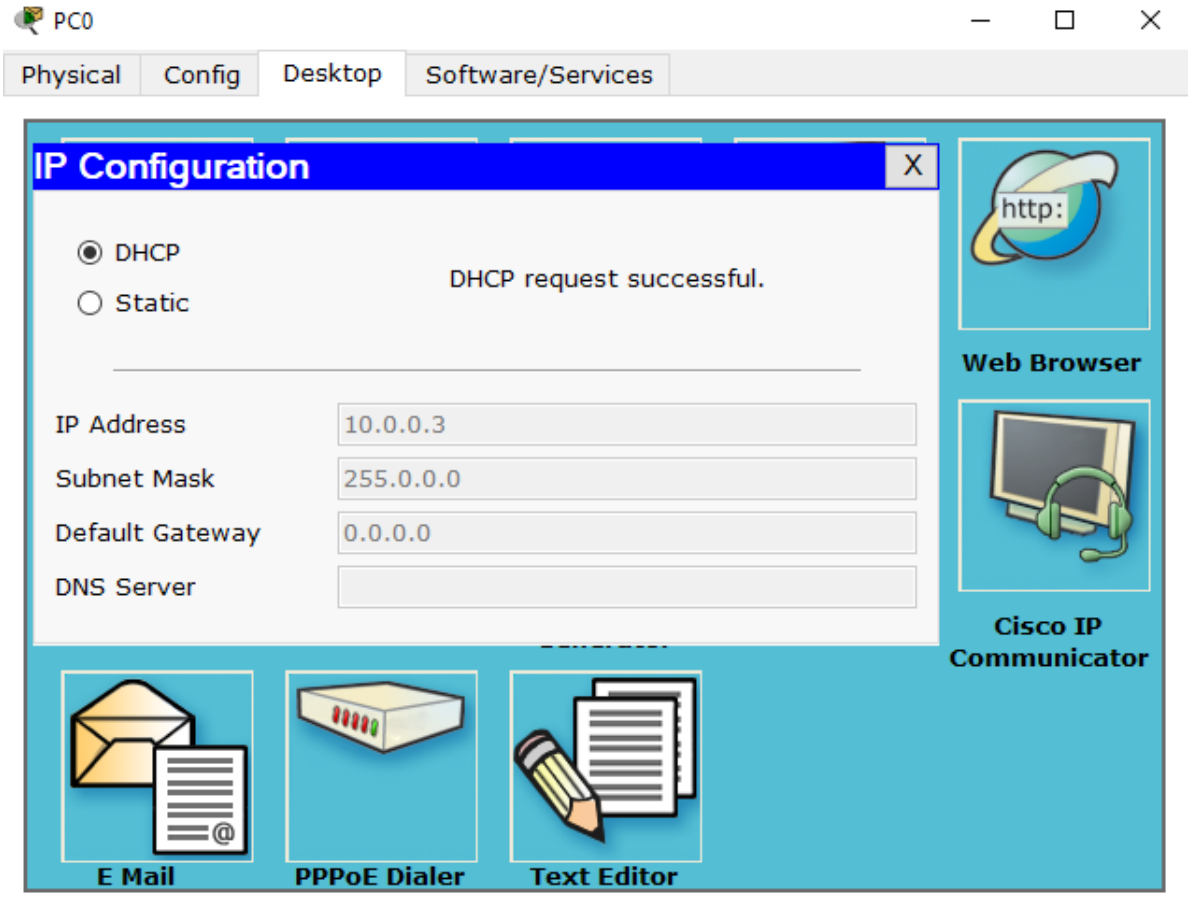
Within a LAN

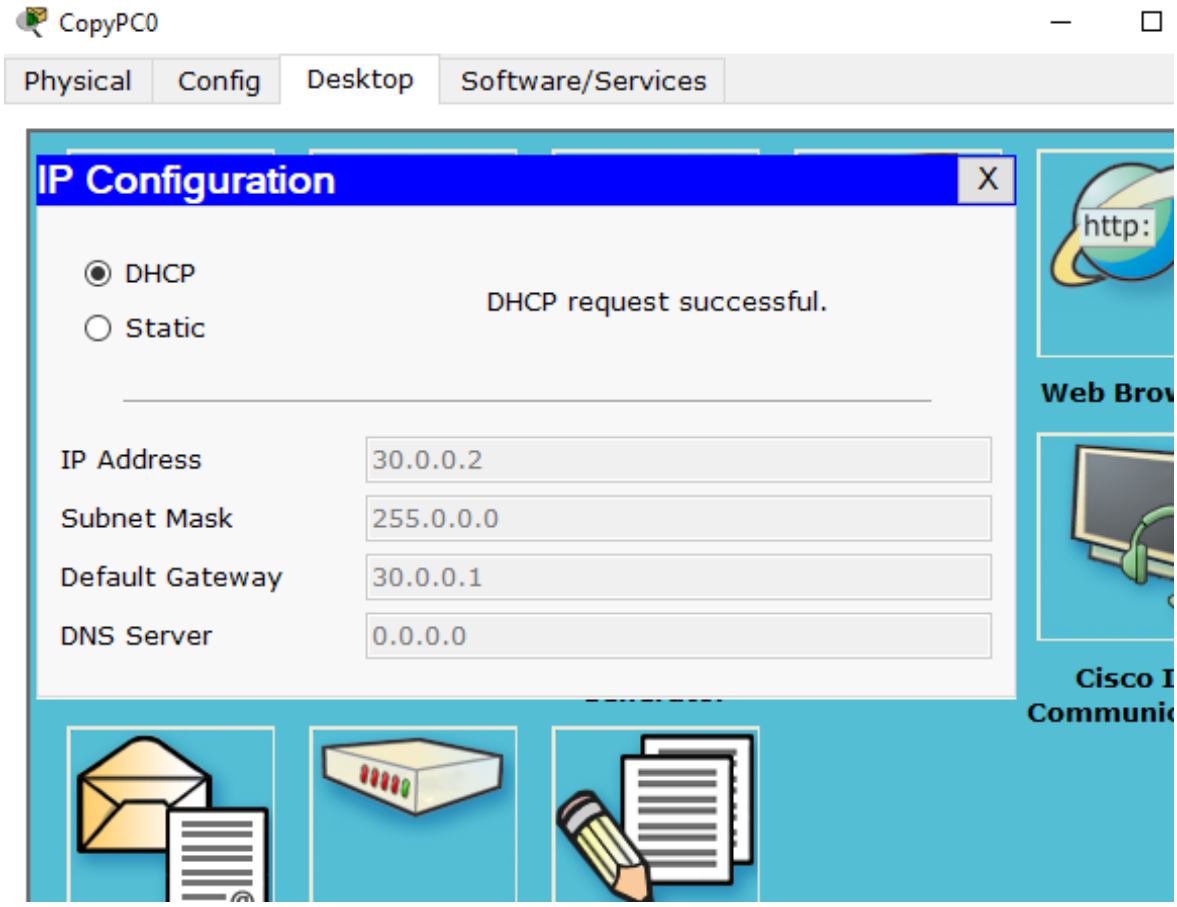


Outside LAN

Router#configure terminal
Router(config)#int fa4/0 ->>> this is the router ip address
Router(config-if)#ip helper-address 10.0.0.2 ->>> this is the server ip address
Router(config-if)#exit

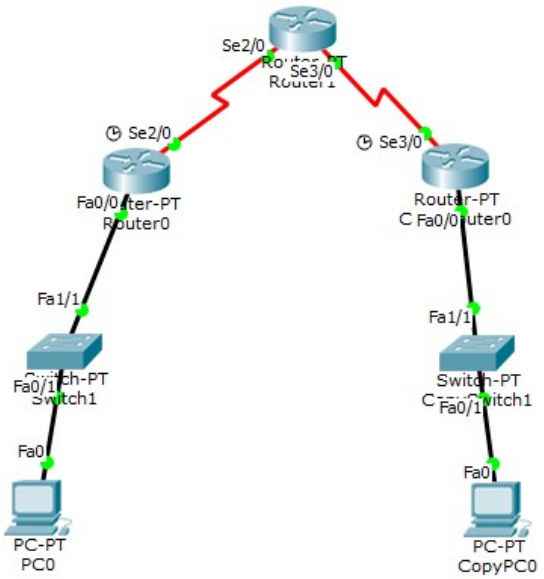
Output


PC0
Physical
Config
Desktop
Software/Services
IP Configuration
X
DHCP
Static
DHCP request successful.
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
http:
Web Browser
Cisco IP Communicator
E Mail
PPPoE Dialer
Text Editor

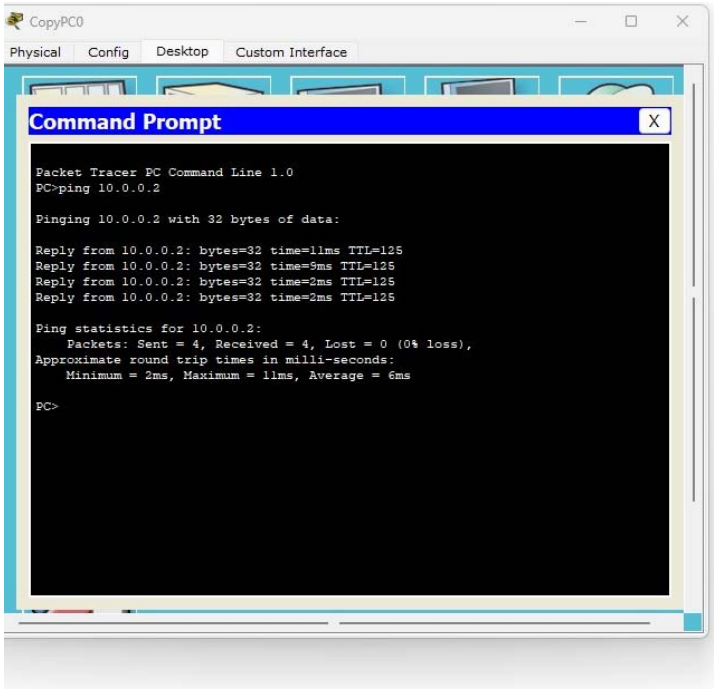

CopyPC0
Physical
Config
Desktop
Software/Services
IP Configuration
X
DHCP
Static
DHCP request successful.
IP Address
30.0.0.2
Subnet Mask
255.0.0.0
Default Gateway
30.0.0.1
DNS Server
0.0.0.0
http:
Web Brov
Cisco I
Communic

5. Configure RIP routing Protocol in Routers.

Topology

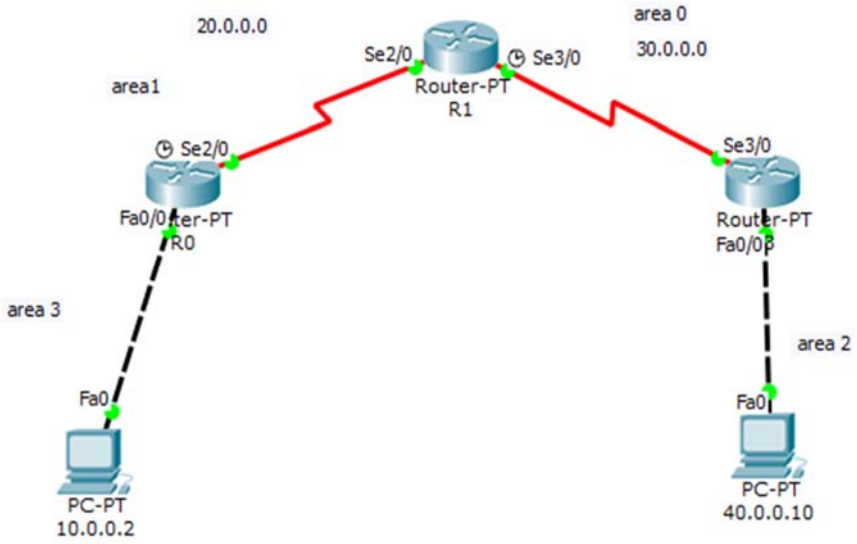


Output

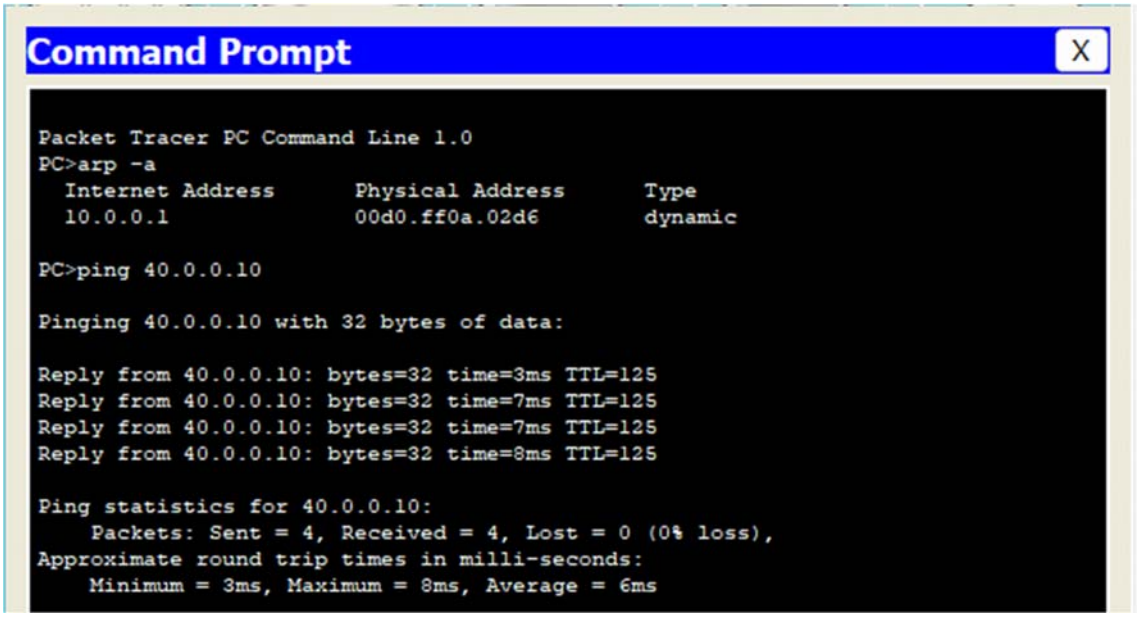


6. **Configure OSPF routing protocol.**

Topology

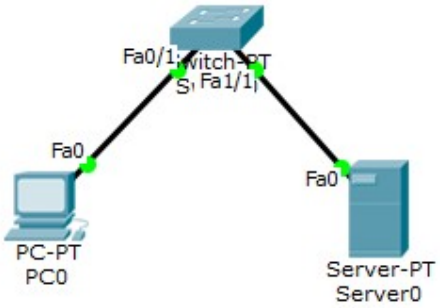


Output

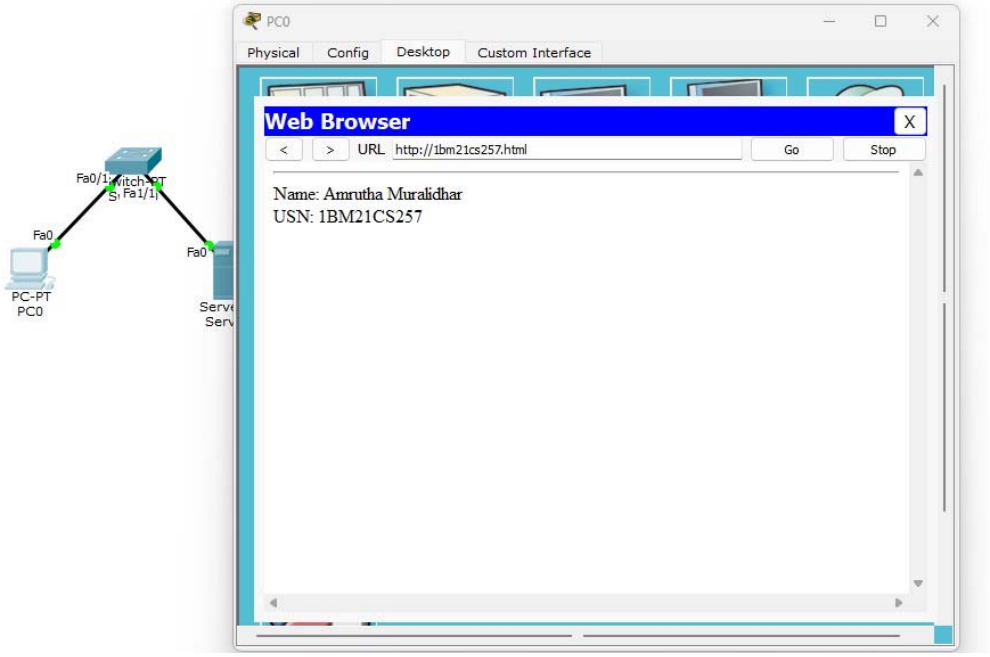


7. Configure Web Server, DNS within a LAN.

Topology

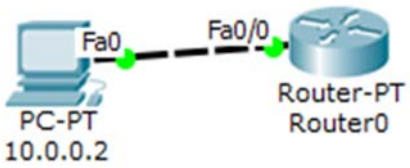


Output



8. **To understand the operation of TELNET by accessing the router in server room from a PC in IT office.**

Topology



Router

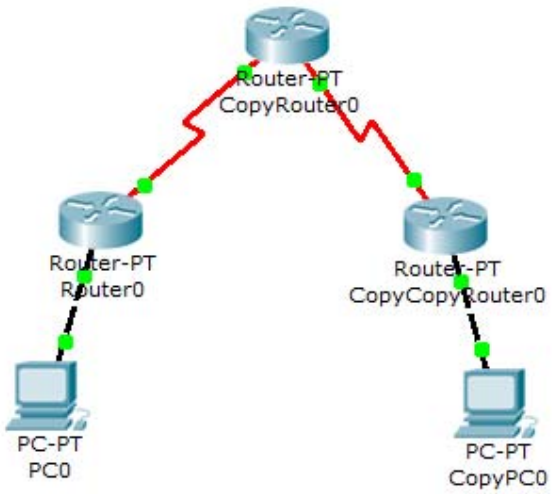
```
         --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n


Press RETURN to get started!



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 10.0.0.1 255.0.0.0
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)#hostname r1
r1(config)#enable secret p1
r1(config)#int fa
% Incomplete command.
r1(config)#interface fastethernet 0/0
r1(config-if)#ip add 10.0.0.1 255.0.0.0
r1(config-if)#no shutdown
r1(config-if)#line vty 0 5
r1(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
% Login disabled on line 136, until 'password' is set
% Login disabled on line 137, until 'password' is set
r1(config-line)#password p0
r1(config-line)#exit
r1(config)#exit
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#wr
Building configuration...
[OK]
r1#
```

Output

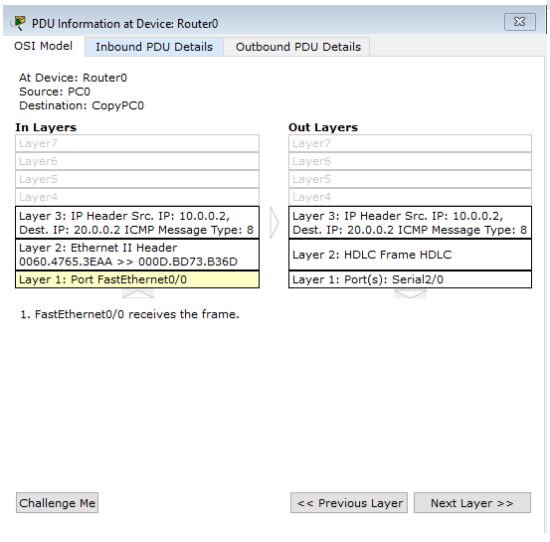
```
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open


User Access Verification

Password:
r1>enable
Password:
Password:
r1#how ip route
       ^
% Invalid input detected at '^' marker.

r1#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
r1#
```

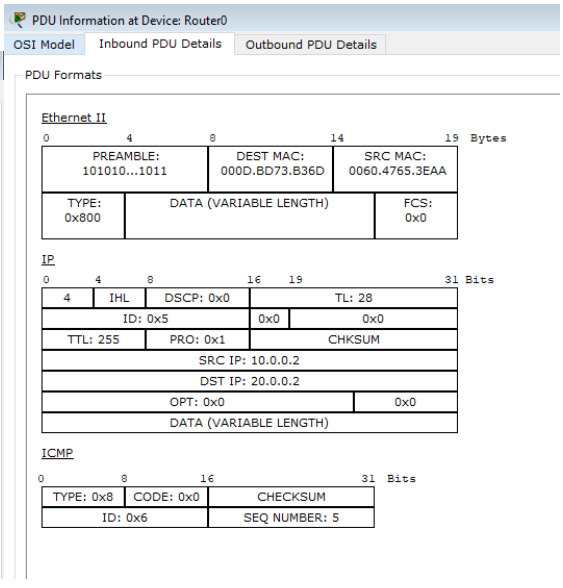
9. Demonstrate the TTL/ Life of a Packet

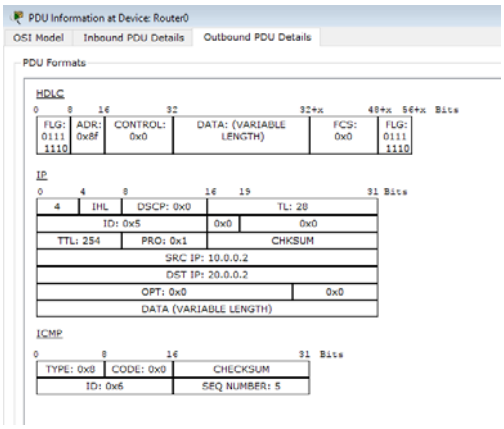
Topology



Output

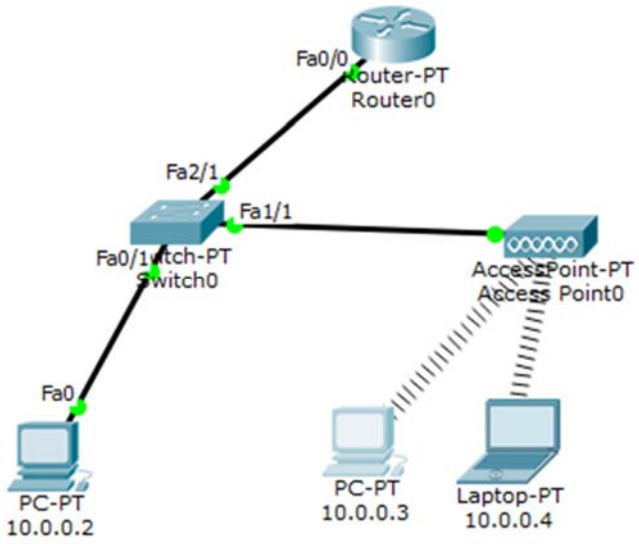




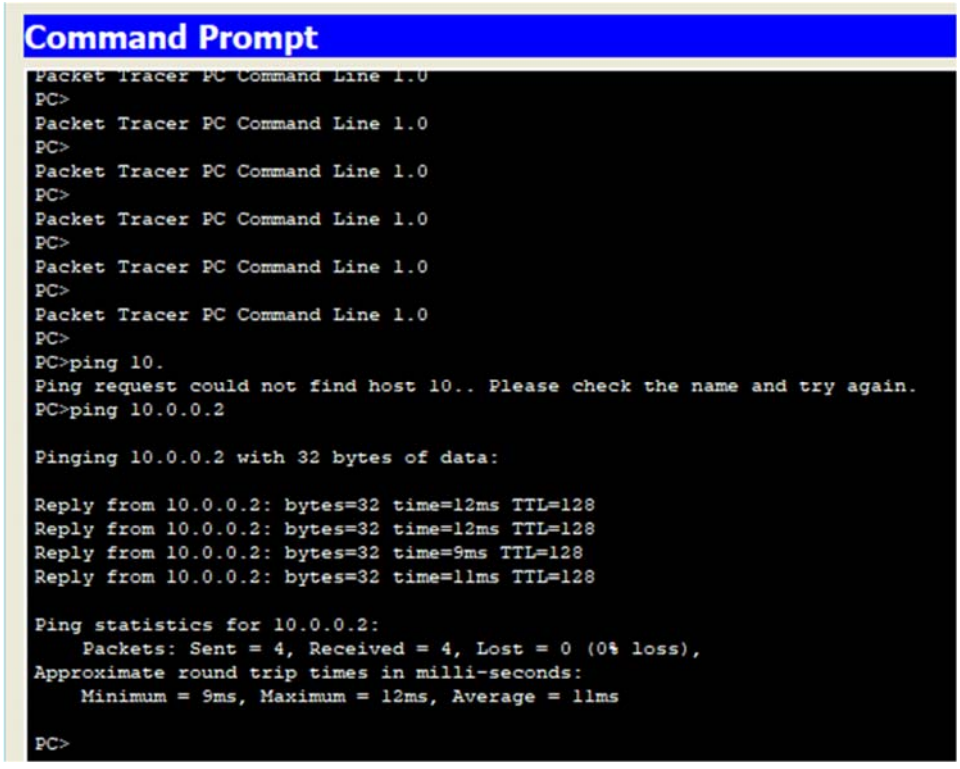


10. To construct a WLAN and make the nodes communicate wirelessly

Topology

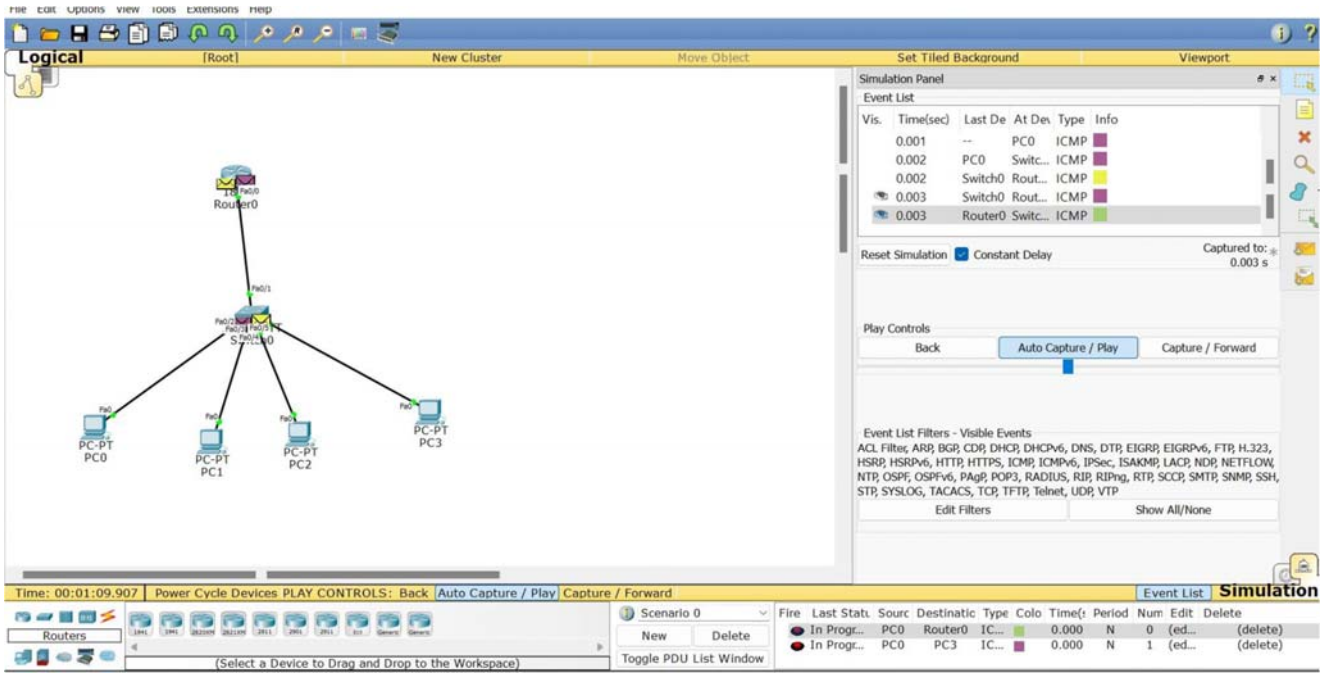


Output

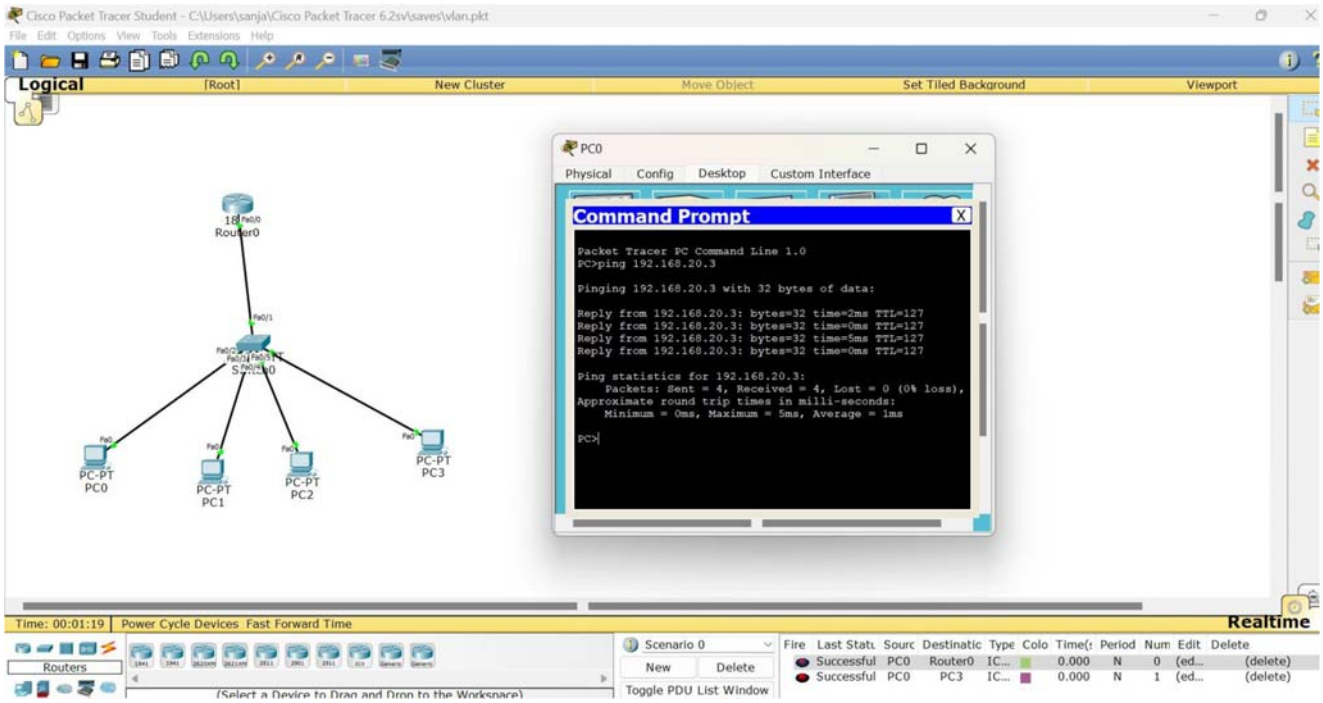


11. To construct a VLAN and make the PCs communicate among a VLAN

Topology

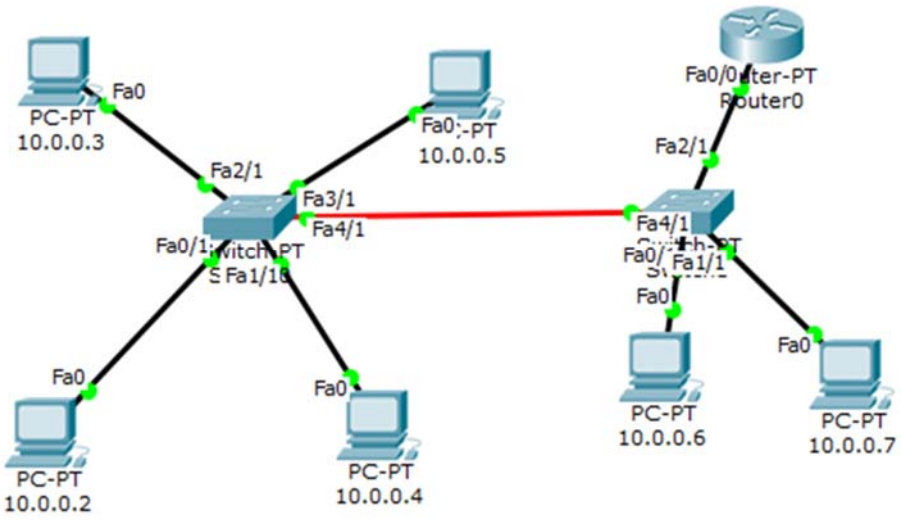


Output



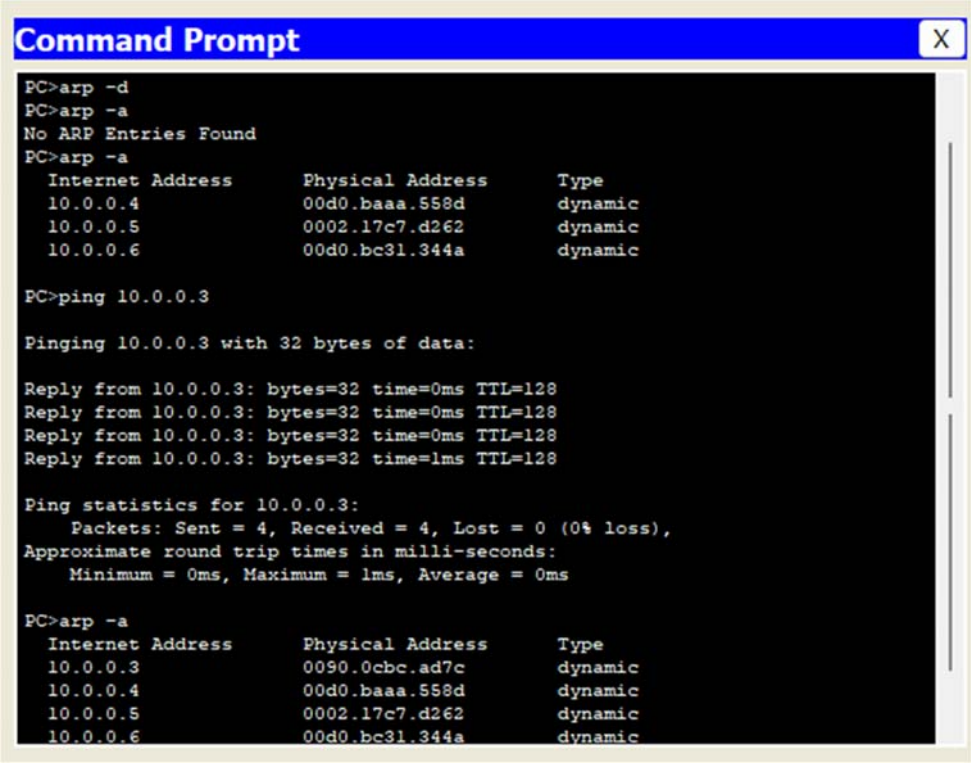
12. To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Topology



Output

1) Using PC command Prompt



2) Using inspect tool

ARP Table for 10.0.0.2

IP Address	Hardware Address	Interface
10.0.0.3	0090.0CBC.AD7C	FastEthernet0
10.0.0.4	00D0.BAAA.558D	FastEthernet0
10.0.0.5	0002.17C7.D262	FastEthernet0
10.0.0.6	00D0.BC31.344A	FastEthernet0
10.0.0.7	0001.638D.409B	FastEthernet0

3) Using switch CLI

```
Switch>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.423a.e612    DYNAMIC     Fa4/1
Switch>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.423a.e612    DYNAMIC     Fa4/1
   1    0001.638d.409b    DYNAMIC     Fa4/1
   1    0002.17c7.d262    DYNAMIC     Fa3/1
   1    0009.7c9b.70dc    DYNAMIC     Fa0/1
   1    0090.0cbc.ad7c    DYNAMIC     Fa2/1
   1    00d0.976e.e57b    DYNAMIC     Fa4/1
   1    00d0.baaa.558d    DYNAMIC     Fa1/1
   1    00d0.bc31.344a    DYNAMIC     Fa4/1
```

CYCLE - 2

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

```
#include<stdio.h>
#include<string.h>
#define N strlen(gen_poly)
char data[28];
char check_value[28];
char gen_poly[10];
int data_length,i,j;
void XOR(){
   for(j = 1;j < N; j++)
   check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');
}
void receiver(){
   printf("Enter the received data: ");
   scanf("%s", data);
   printf("\n-----------------------------\n");
   printf("Data received: %s", data);
   crc();
   for(i=0;(i<N-1) && (check_value[i]!='1');i++);
      if(i<N-1)
         printf("\nError detected\n\n");
      else
         printf("\nNo error detected\n\n");
}
void crc(){
   for(i=0;i<N;i++)
      check_value[i]=data[i];
   do{
      if(check_value[0]=='1')
         XOR();
      for(j=0;j<N-1;j++)
      check_value[j]=data[i++];
   }while(i<=data_length+N-1);
}
int main()  {
   printf("\nEnter data to be transmitted: ");
   scanf("%s",data);
   printf("\n Enter the Generating polynomial: ");
   scanf("%s",gen_poly);
   data_length=strlen(data);
```

```
    for(i=data_length;i<data_length+N-1;i++)
       data[i]='0';
    printf("\n----------------------------------------");
    printf("\n Data padded with n-1 zeros : %s",data);
    printf("\n----------------------------------------");
    crc();
    printf("\nCRC or Check value is : %s",check_value);
    for(i=data_length;i<data_length+N-1;i++)
       data[i]=check_value[i-data_length];
    printf("\n----------------------------------------");
    printf("\n Final data to be sent : %s",data);
    printf("\n----------------------------------------\n");
    receiver();
       return 0;  }
```

Output

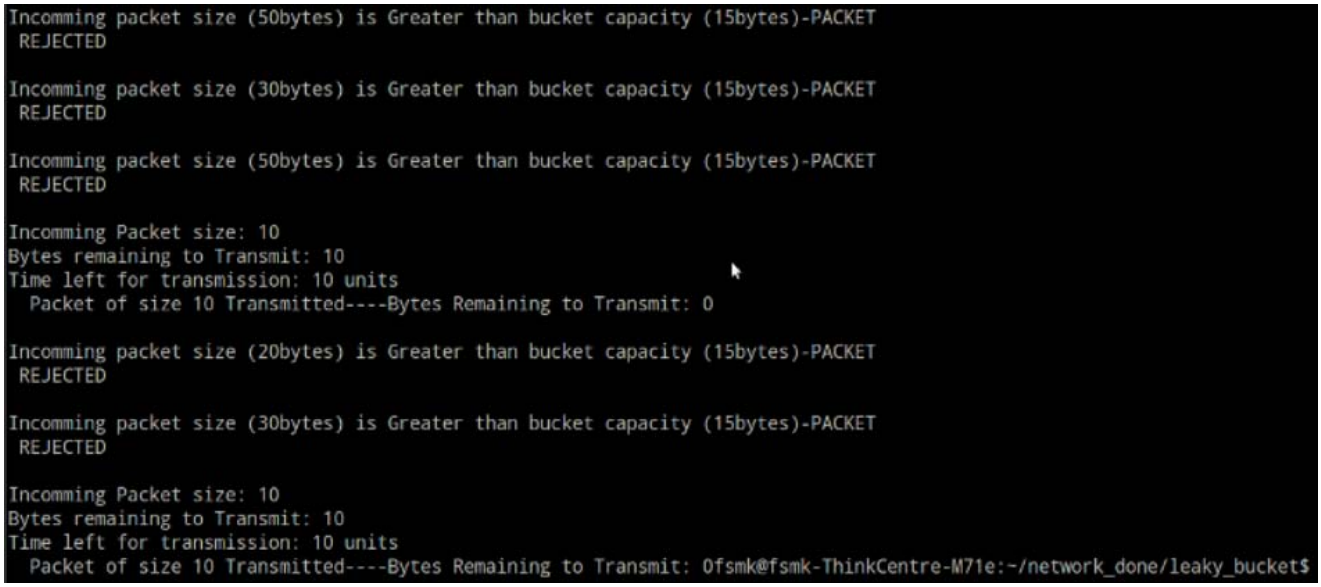
```
Enter data to be transmitted: 1001101
Enter the Generating polynomial: 1011
---------------------------------------
Data padded with n-1 zeros : 1001101000
---------------------------------------
CRC or Check value is : 101
---------------------------------------
Final data to be sent : 1001101101
---------------------------------------
Enter the received data: 1001101101
-----------------------------
Data received: 1001101101
No error detected
```

14. Write a program for congestion control using Leaky bucket algorithm.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#define NOF_PACKETS 10
int rand(int a){
   int rn = (random() % 10) % a;
   return  rn == 0 ? 1 : rn;}
int main(){
   int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate, p_sz_rm=0, p_sz, p_time, op;
   for(i = 0; i<NOF_PACKETS; ++i)
      packet_sz[i] = rand(6) * 10;
   for(i = 0; i<NOF_PACKETS; ++i)
      printf("\npacket[%d]:%d bytes\t", i, packet_sz[i]);
   printf("\nEnter the Output rate:");
   scanf("%d", &o_rate);
   printf("Enter the Bucket Size:");
   scanf("%d", &b_size);
   for(i = 0; i<NOF_PACKETS; ++i)    {
      if( (packet_sz[i] + p_sz_rm) > b_size)
         if(packet_sz[i] > b_size)/*compare the packet siz with bucket size*/
            printf("\n\nIncoming packet size (%dbytes) is Greater than bucket capacity (%dbytes)-PACKET
REJECTED", packet_sz[i], b_size);
         else
            printf("\n\nBucket capacity exceeded-PACKETS REJECTED!!");
      else {
         p_sz_rm += packet_sz[i];
         printf("\n\nIncoming Packet size: %d", packet_sz[i]);
         printf("\nBytes remaining to Transmit: %d", p_sz_rm);
         p_time = rand(4) * 10;
         printf("\nTime left for transmission: %d units", p_time);
         for(clk = 10; clk <= p_time; clk += 10) {
```

```
sleep(1);
if(p_sz_rm) {
   if(p_sz_rm <= o_rate)/*packet size remaining comparing with output rate*/
      op = p_sz_rm, p_sz_rm = 0;
   else
      op = o_rate, p_sz_rm -= o_rate;
   printf("\nPacket of size %d Transmitted", op);
   printf("----Bytes Remaining to Transmit: %d", p_sz_rm);              }
else              {
   printf("\nTime left for transmission: %d units", p_time-clk);
   printf("\nNo packets to transmit!!"); } } } }}
```

Output



15. **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.**

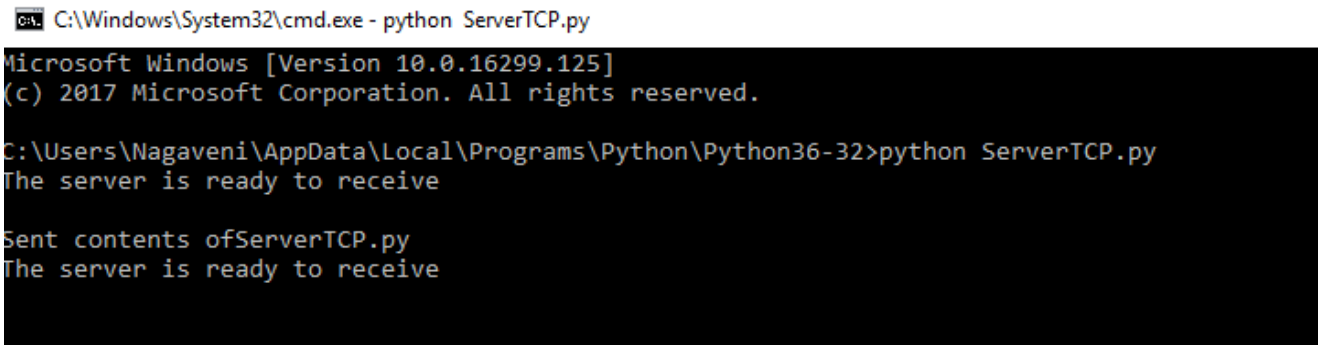
ClientTCP.py

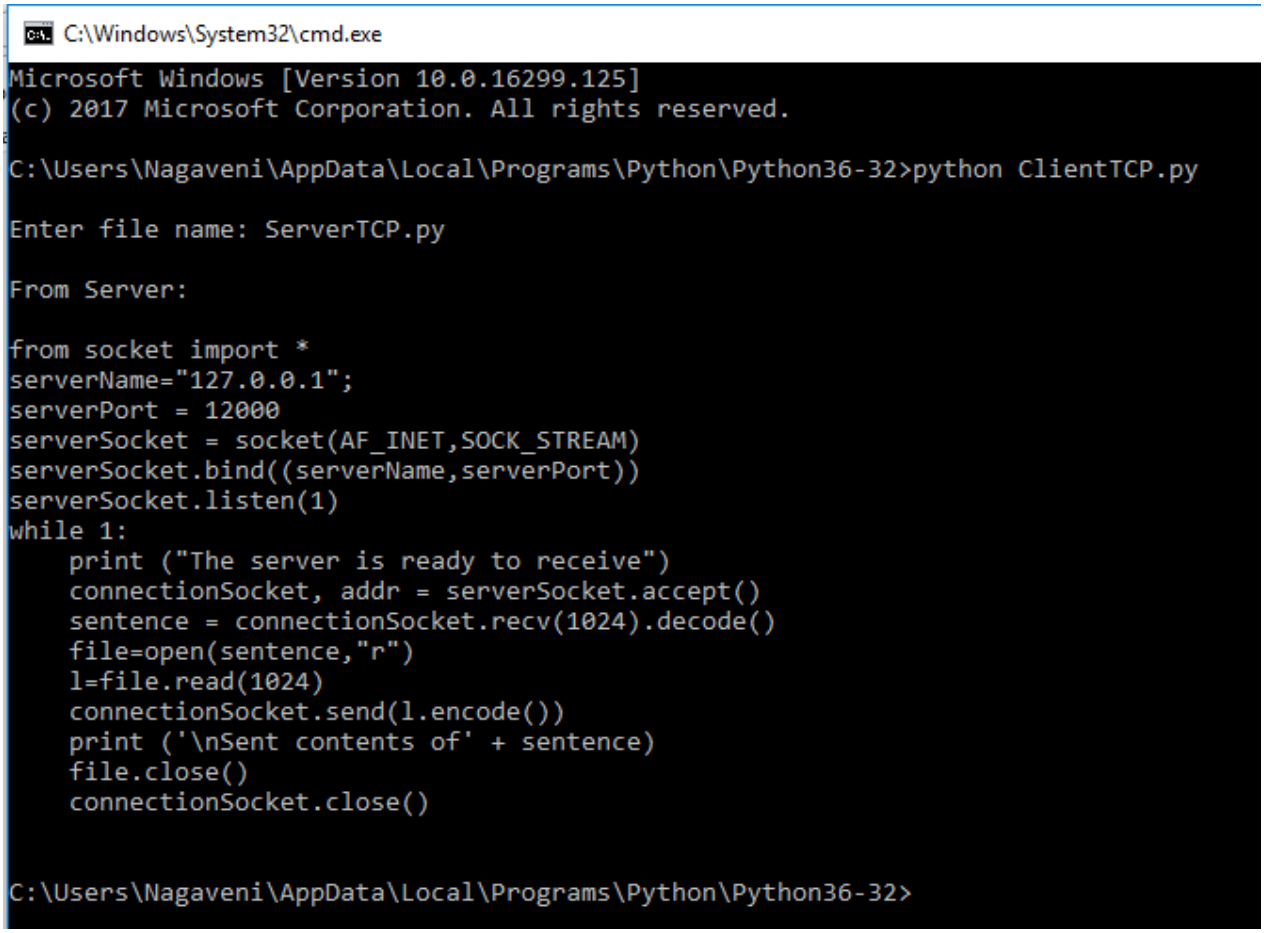
```
from socket import *
serverName = &#39;127.0.0.1&#39;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input(&quot;\nEnter file name: &quot;)
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print (&#39;\nFrom Server:\n&#39;)
print(filecontents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName=&quot;127.0.0.1&quot;
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print (&quot;The server is ready to receive&quot;)
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,&quot;r&quot;)
l=file.read(1024)
connectionSocket.send(l.encode())
print (&#39;\nSent contents of &#39; + sentence)
file.close()
connectionSocket.close()
```

Output




C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\Nagaveni\AppData\Local\Programs\Python\Python36-32>python ClientTCP.py
Enter file name: ServerTCP.py
From Server:
from socket import *
serverName="127.0.0.1";
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print ("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,"r")
l=file.read(1024)
connectionSocket.send(l.encode())
print ('\nSent contents of' + sentence)
file.close()
connectionSocket.close()
C:\Users\Nagaveni\AppData\Local\Programs\Python\Python36-32>

16. **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.**

ClientUDP.py

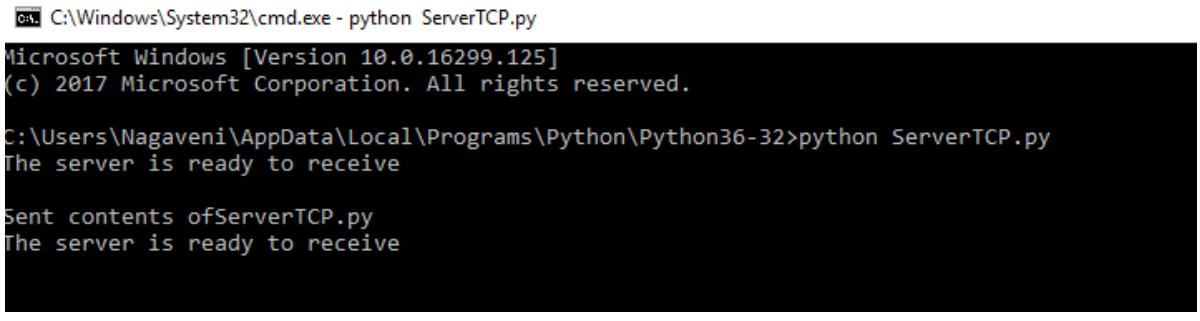
```
from socket import *
serverName = &amp;quot;127.0.0.1&amp;quot;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input(&amp;quot;\nEnter file name: &amp;quot;)
clientSocket.sendto(bytes(sentence,&amp;quot;utf-8&amp;quot;),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print (&amp;#39;\nReply from Server:\n&amp;#39;)
print (filecontents.decode(&amp;quot;utf-8&amp;quot;))
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((&amp;quot;127.0.0.1&amp;quot;, serverPort))
print (&amp;quot;The server is ready to receive&amp;quot;)
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode(&amp;quot;utf-8&amp;quot;)
file=open(sentence,&amp;quot;r&amp;quot;)
con=file.read(2048)
serverSocket.sendto(bytes(con,&amp;quot;utf-8&amp;quot;),clientAddress)
print (&amp;#39;\nSent contents of &amp;#39;, end = &amp;#39; &amp;#39;)
print (sentence)
file.close()
```

Output

Server



Client

From Server:

```
from socket import *
serverName="127.0.0.1";
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print ('\nSent contents of' + sentence)
    file.close()
    connectionSocket.close()


C:\Users\Nagaveni\AppData\Local\Programs\Python\Python36-32>
```