Final Lab Report

Submitted by

Name: Md.Tanvir Hossain Saon

Roll: 388

Exam Roll: 202200

Course Code: CSE-360

Course Name: Computer Network Lab

Submitted to

Dr.Md.Imdadul Islam
Professor
Department of Computer Science & Engineering
Jahangirnagar University
Savar, Dhaka

Date of submission: 14 May 2024

Department of Computer Science & Engineering
Jahangirnagar University
Savar, Dhaka-1342

Table Of Content

Experiment No: 01

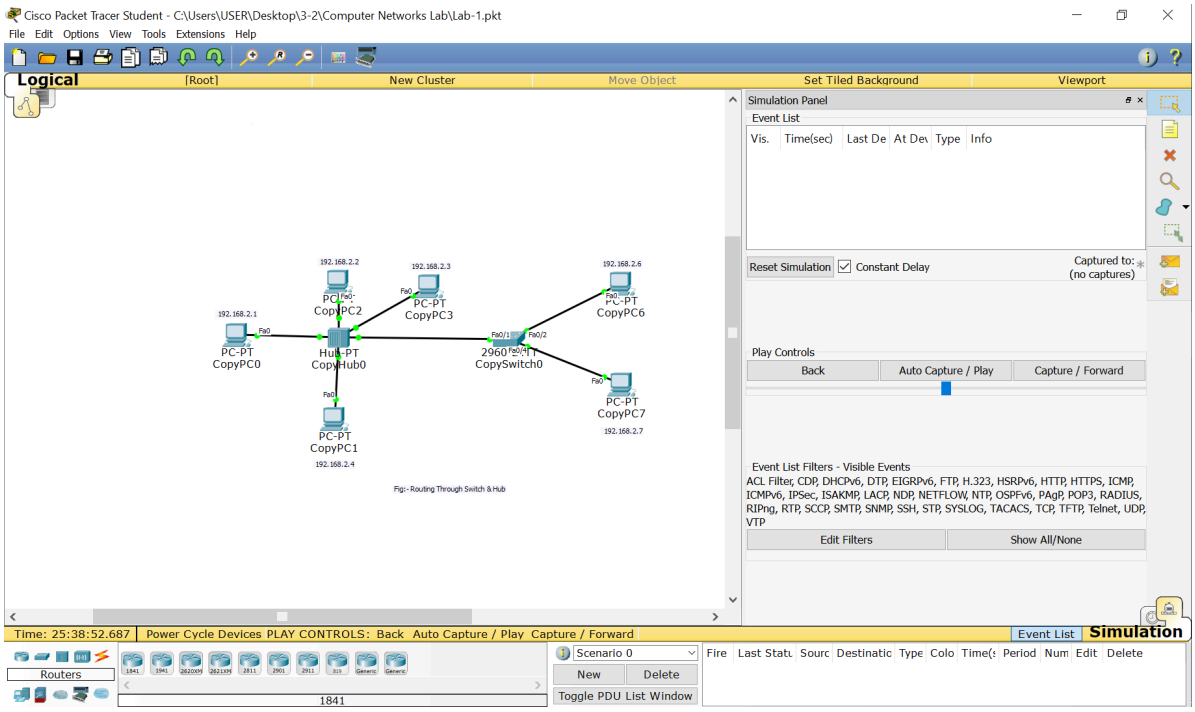
Name of experiment: A trace test from the command prompt and a hub and switch allow ICMP packets to flow.

Objective:

We will learn about ICMP packet passing utilizing a hub and switch in this project. Additionally, we'll learn about the trace test from cmd.

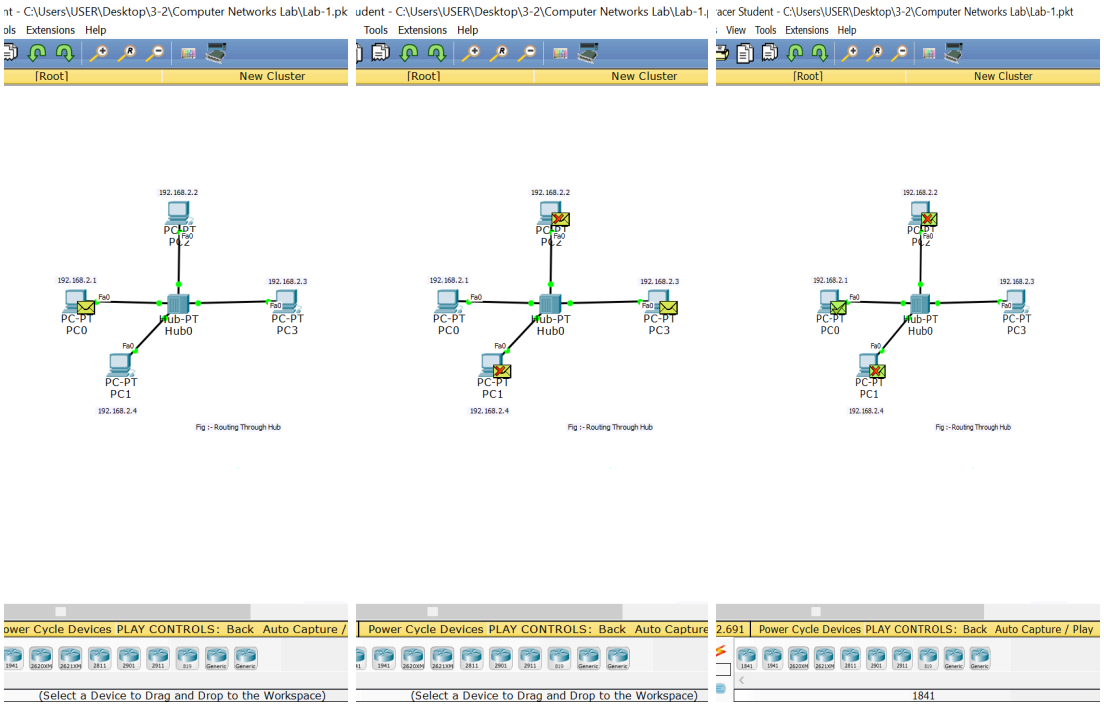
Apparatus: Cisco Packet tracer, PC

Circuit / Network diagram:

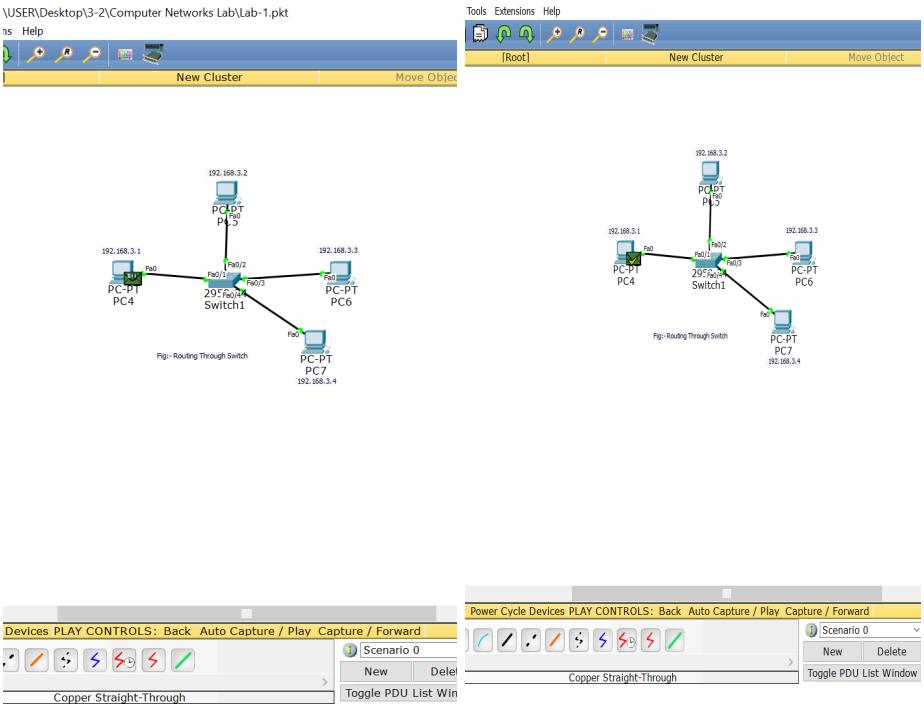


Result and discussion:

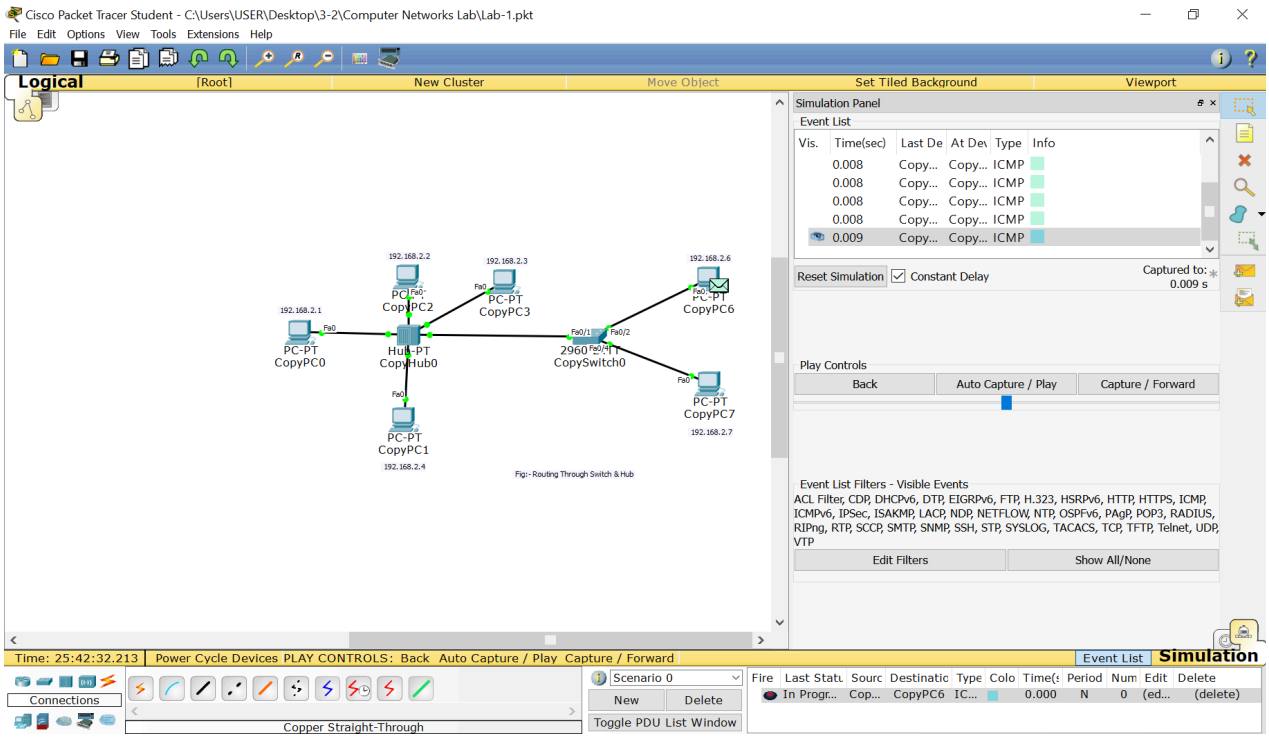
The ICMP packets are being traced here using Cisco Packet Tracer, which is connected to a hub and a switch. We will begin by connecting a few computers through a hub and then observe the messages being transmitted between them. We may observe a packet transfer from 191.168.2.1 to 191.168.2.3 in this case.

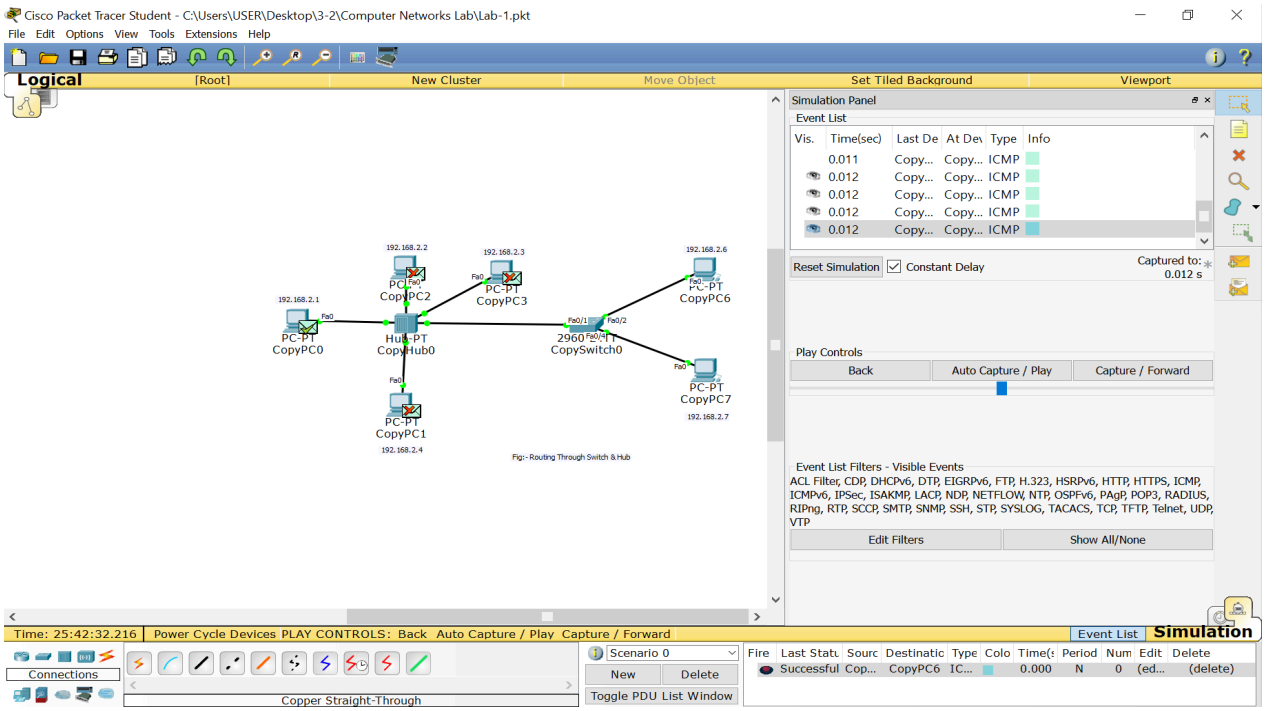


Then we will begin by connecting a few computers through a Switch and then observe the messages being transmitted between them. We may observe a packet transfer from 191.168.3.1 to 191.168.3.3 in this case.



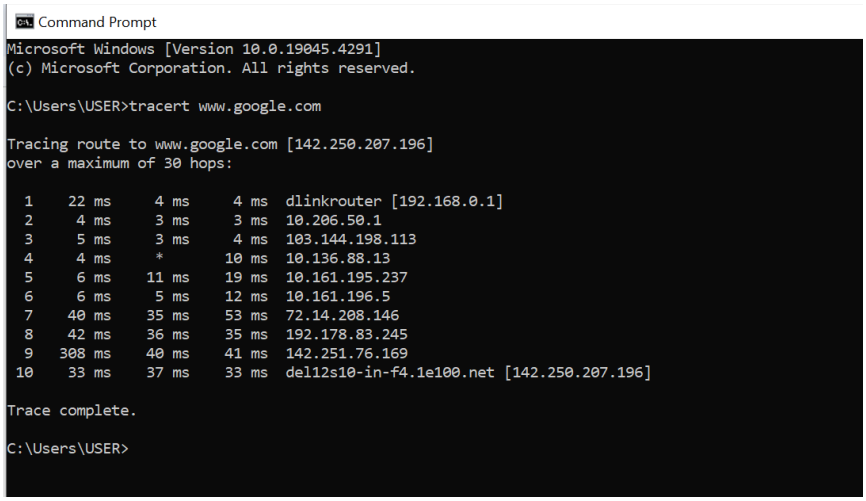
Next we are going to combine Hub,computers and Switch . Then we will see packets passing using hubs and switches. Here let's see packets passing from IP address 191.168.2.1 to IP address 191.168.2.5.



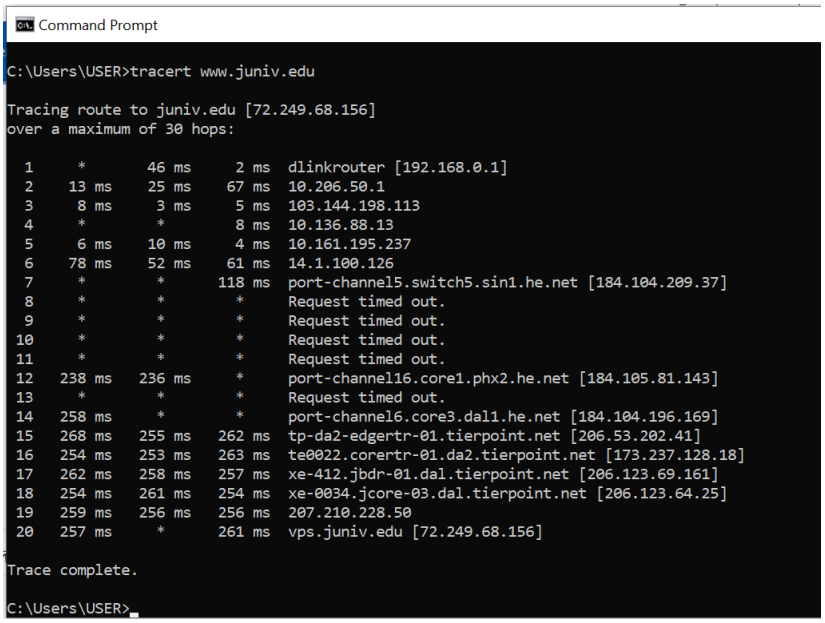


Now we will see the track test from cmd.

We will check www.google.com and also test the ping of the IP addresses.



We will also check for www.juniv.edu and test ping for IP addresses. Here we can see that some requests failed due to request time out.



Experiment No: 02

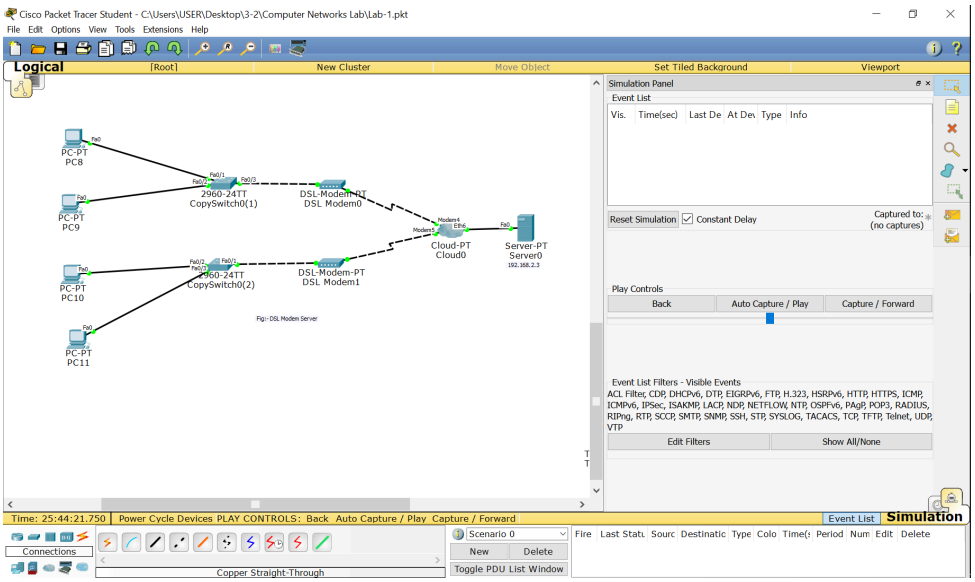
Name of experiment: Configuration of DSL modem and Router.

Objective:

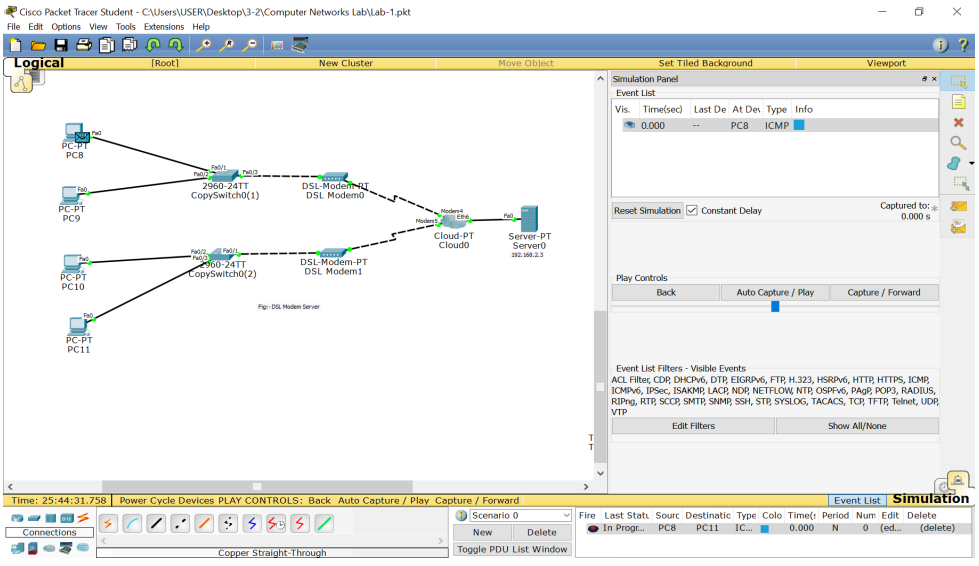
In this experiment, we will see how WAN routing and the Digital Subscriber Line (DSL) modem work together. After that, we'll use a packet tracer to mimic it.

Apparatus: Cisco Packet tracer, PC

Circuit / Network diagram:



Result and discussion:



We will be able to establish a Digital Subscriber Line (DSL) connection between an Internet service provider (ISP) and the phone line when we complete this experiment. A digital subscriber line (DSL) modem translates digital packets into analog signals that can travel unimpeded over a telephone line, and the computer is linked to this modem. On the opposite end, a Digital Subscriber Line Access Multiplexer (DSLAM) transforms signals into packets.

Experiment No: 03

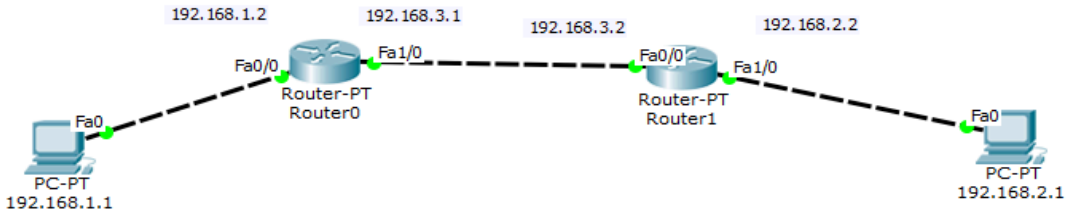
Name of experiment: Router configuration using CLI.

Objective:

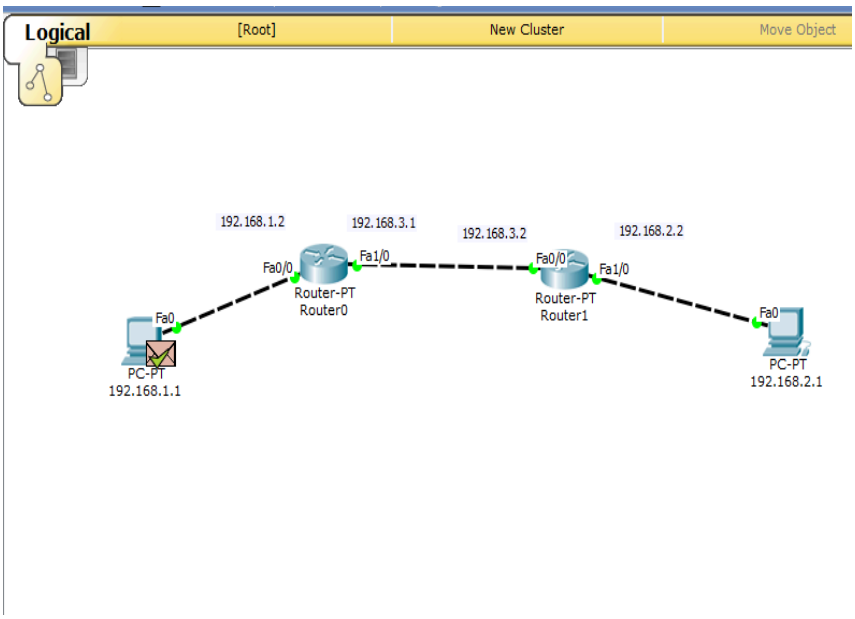
We will learn how to configure a router using the command line interface (CLI) in this experiment. We will connect routers with pc from command prompt and check their connection through cmd.

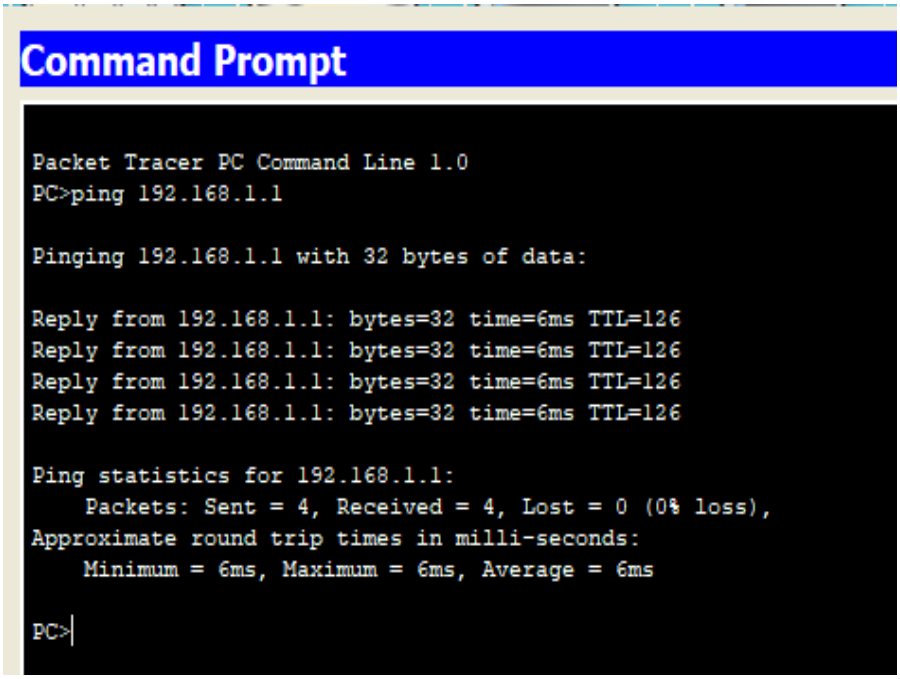
Apparatus: Cisco Packet tracer, PC

Circuit / Network diagram:



Result and discussion:





We will be able to route using CLI once we have completed this lab. The packet is successfully passed in both the command prompt and packet tracer. As IP addresses are used in routers rather than mac addresses, packets can be sent across several network IDs.

Experiment No: 04

Name of experiment: VLAN Configuration with Switch and Router.

Objective:

Using a switch and router, this experiment aims to implement VLAN. After that, we'll use packet tracing to replicate it. We will taste it and trace it in the command prompt to simulate real life.

Apparatus: Cisco Packet tracer, PC

Circuit / Network diagram:

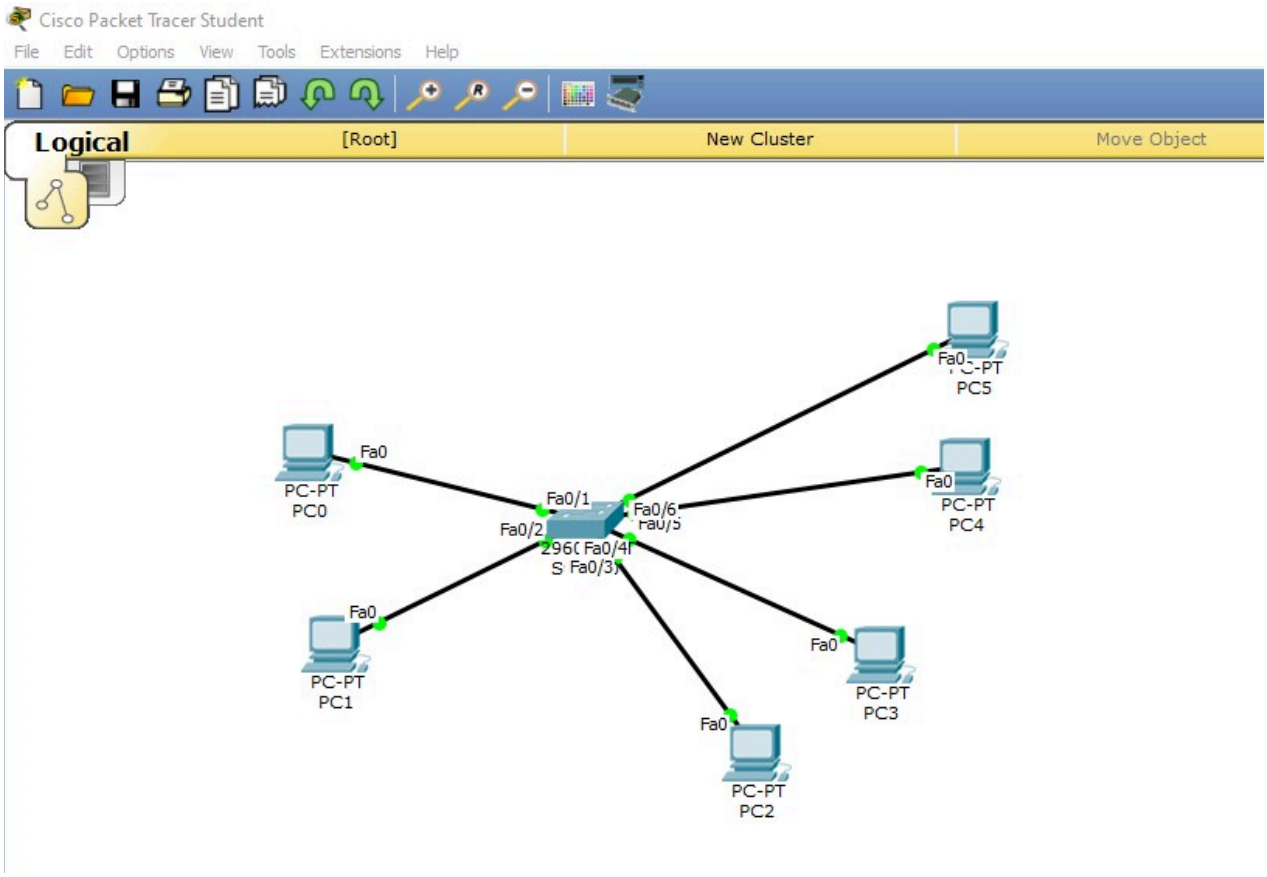


Fig: VLAN configuration using switch

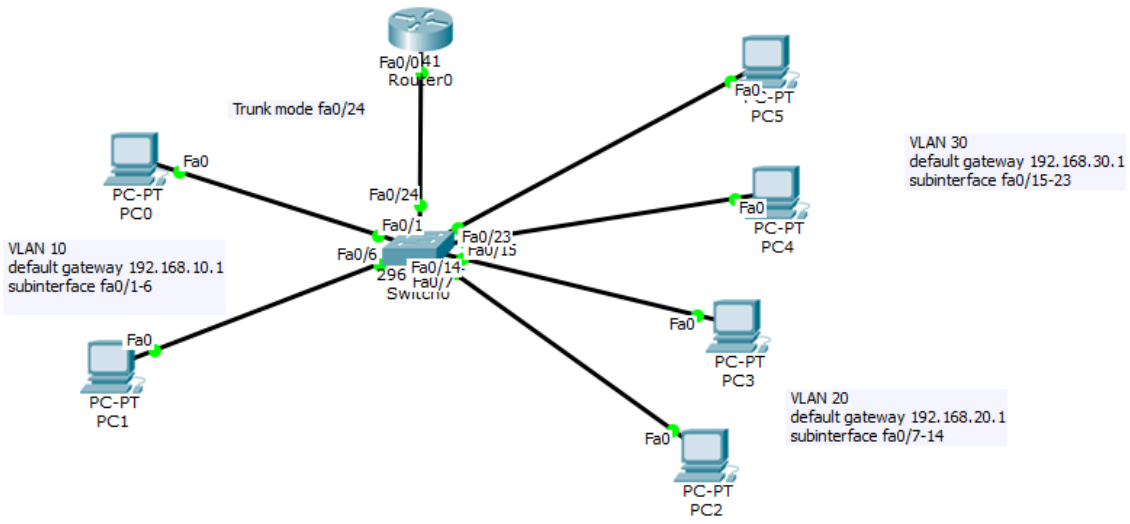
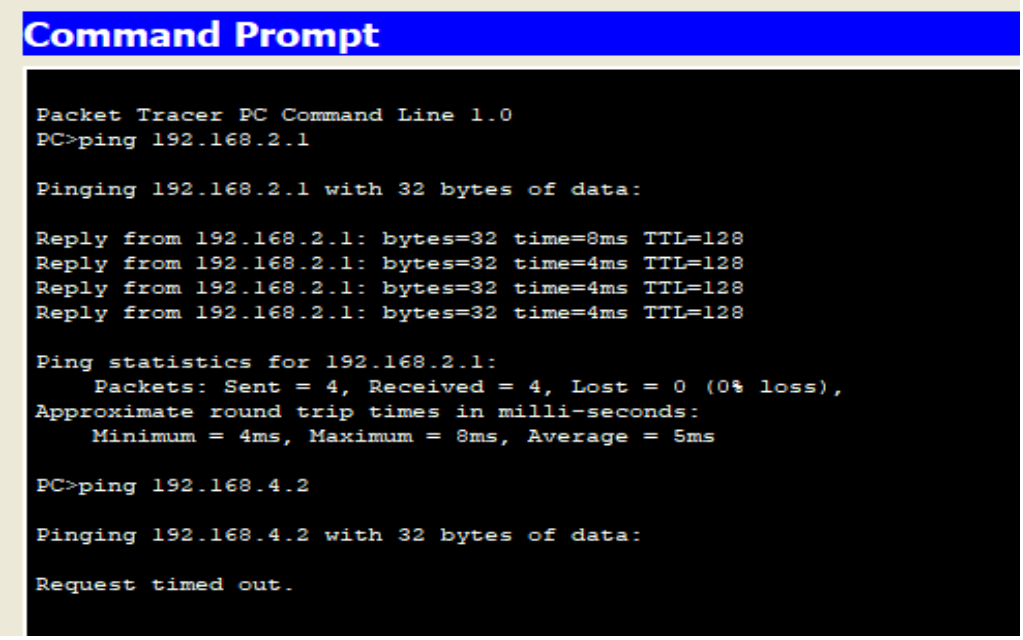


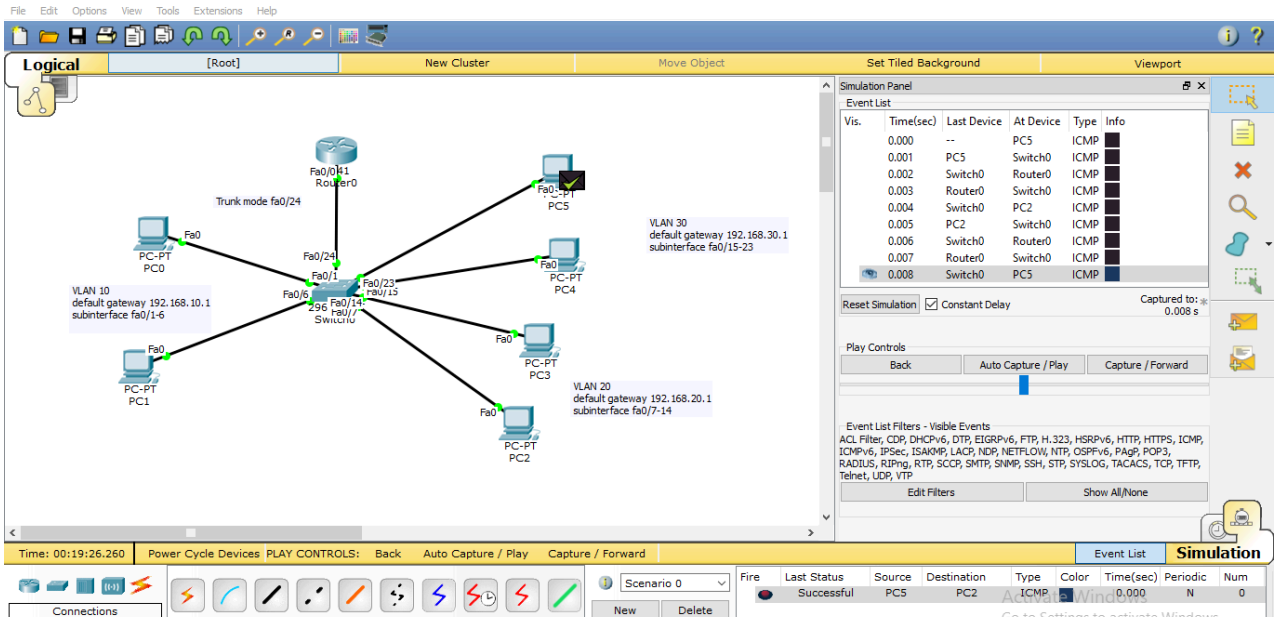
Fig: VLAN under sub interface

Result and discussion:

Pinging within the same VLAN sends the packet correctly. On the other hand, packet sending fails when the VLAN is different.



VLAN under sub interface: In this case we use a router to send packets within different VLAN.




```
PC>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:

Reply from 192.168.10.2: bytes=32 time=8ms TTL=127
Reply from 192.168.10.2: bytes=32 time=8ms TTL=127
Reply from 192.168.10.2: bytes=32 time=8ms TTL=127
Reply from 192.168.10.2: bytes=32 time=8ms TTL=127

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 8ms, Average = 8ms
```

Experiment No: 05

Name of experiment: Implementation of wireless LAN (wifi).

Objective:

Implementing wireless LAN in Cisco packet tracer is the aim of this exercise. We will taste it and trace it in the command prompt to simulate real life.

Apparatus: Cisco Packet tracer, PC

Circuit / Network diagram:

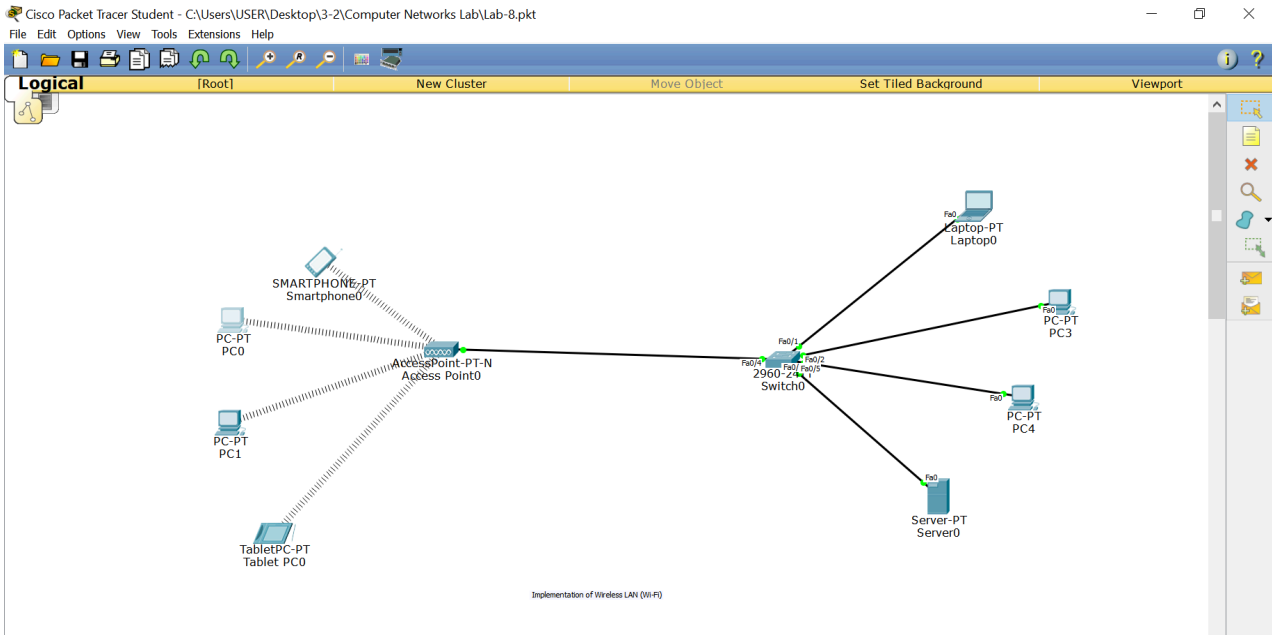
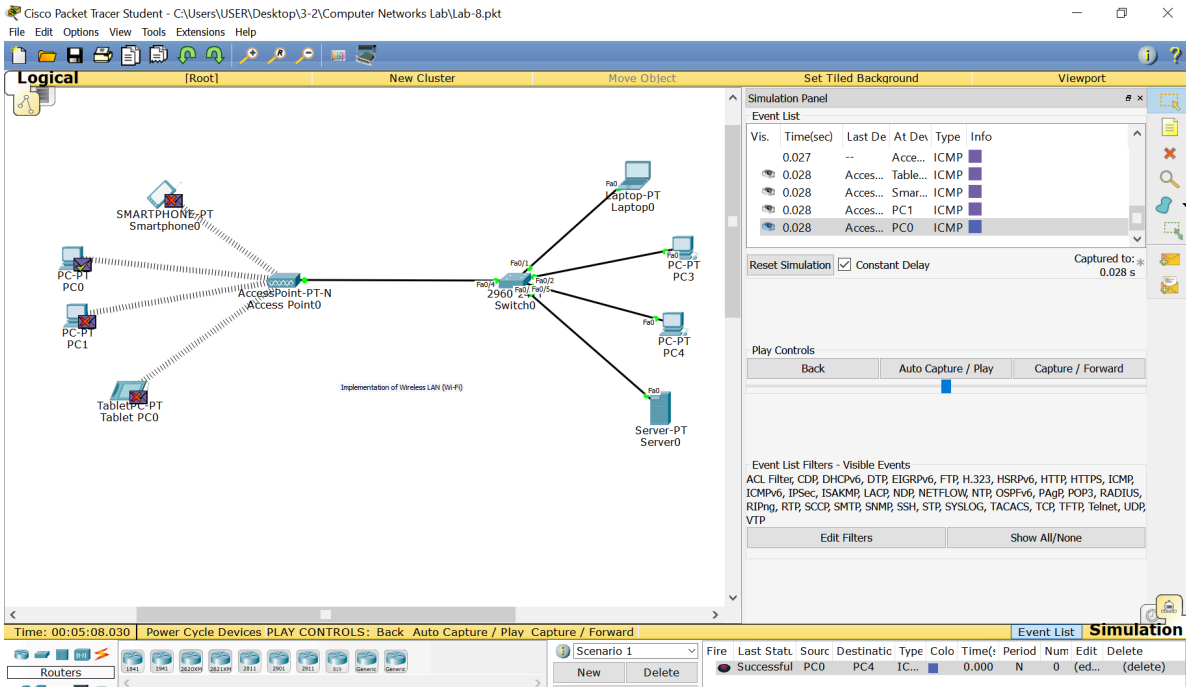


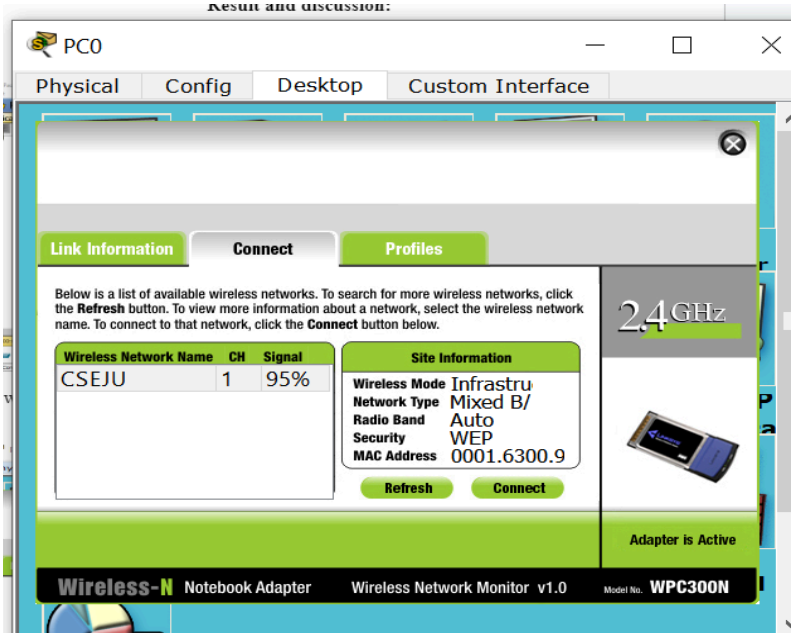
Fig: wireless LAN configuration

Result and discussion:

Wireless connections can be established without the need for access port authentication.



If we use authentication we cannot connect the pc without WEP key provided by the access port.



Experiment No: 06

Name of experiment: Implementation of IP telephony.

Objective:

The purpose of this project is to put in place a modest IP telephone network. Every phone will be authenticated using its IP address and associated phone number. Lastly, contacting each other's IP phones will test the network.

Apparatus: Cisco Packet tracer, PC

Circuit / Network diagram:

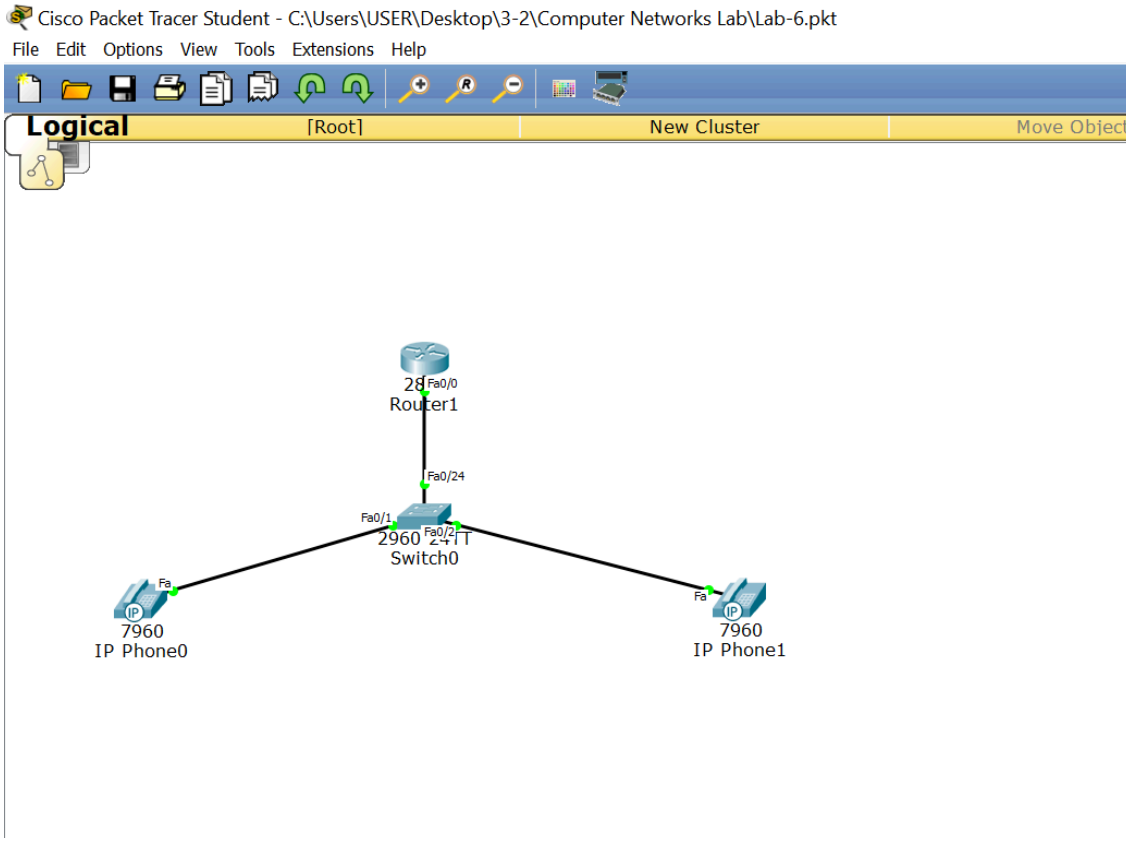
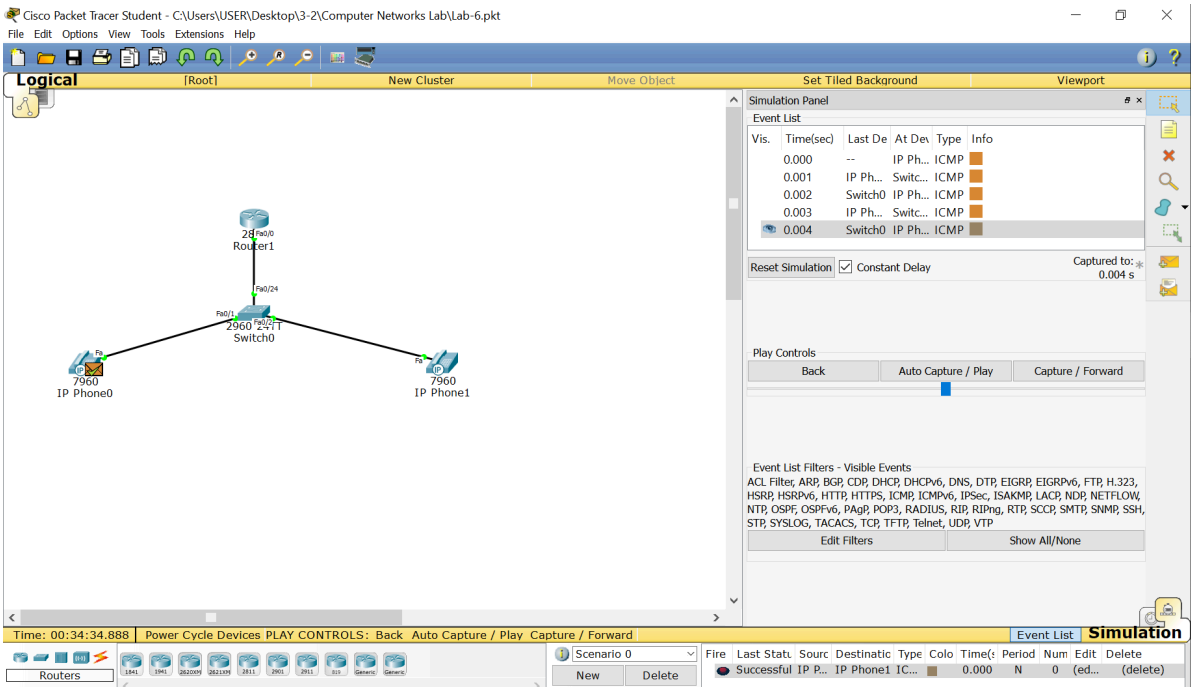


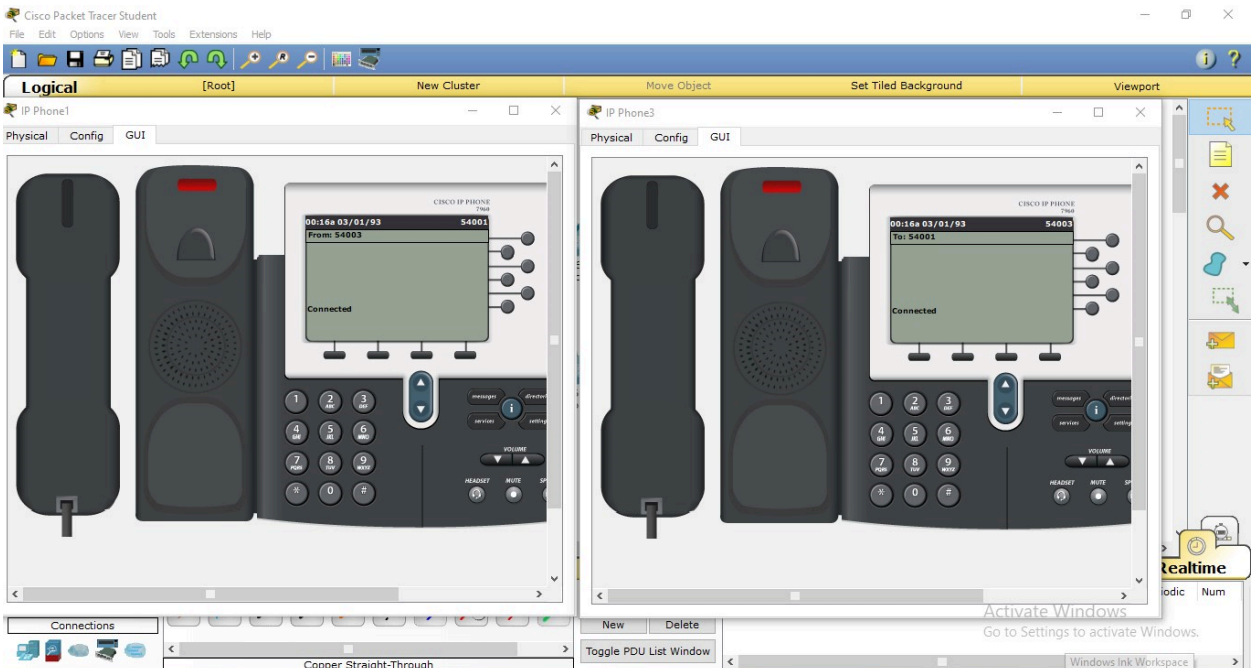
Fig: Implementation of IP telephony

Result and discussion:

After performing this lab, we will be able to implement IP telephony in packet tracer,



We verified each IP telephone by dialing each other.



Experiment No: 07

Name of experiment: Socket programming establishes connection between two nodes of a network.

Objective:

The goal of this exercise is to link two nodes—a client and a server—of a network via a process known as socket programming. This experiment will be carried out in Python.

Apparatus: Visual studio, PC

VS Codes:

Server Side:

```
import socket
LOCALHOST="127.0.0.1"
PORT=8080
server=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
server.bind((LOCALHOST, PORT))
server.listen(1)
print("server started")
print("waiting for client request..")
clientConnection,clientAddress=server.accept()
print("connected client :" ,clientAddress)
msg=""
while True:
    in_data=clientConnection.recv(1024)
    msg=in_data.decode()
```

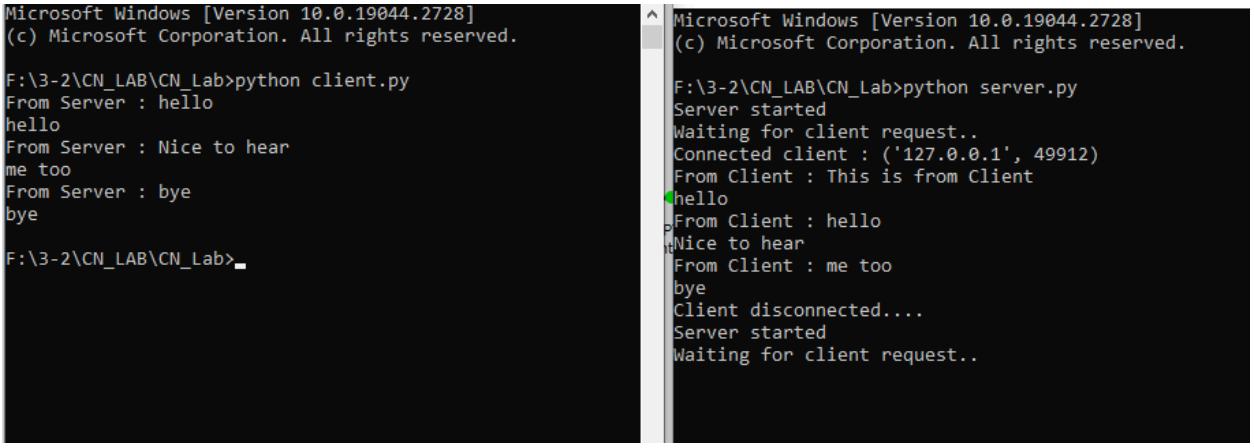
```
    if msg=='bye':
        break
    print("from client: " ,msg)
    out_date=input()
    clientConnection.send(bytes(out_date,'UTF-8'))
print("client disconnected ..")
clientConnection.close()
```

Client Side:

```
import socket
server="127.0.0.1"
port=8080
client=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((server,port))
client.sendall(bytes("This is from client",'UTF-8'))
while True:
    in_data=client.recv(1024)
    print("From Server: ",in_data.decode())
    out_data=input()
    client.sendall(bytes(out_data,'UTF-8'))
    if out_data=='bye':
        break
    client.close()
```

Result and discussion:

We will be able to connect to the server and client after completing this lab. The server prints the client address upon connection establishment and then waits for data. Data can be sent and received by the client in accordance with the protocols in use. The connection will be severed when the client and server both issue the CLOSE primitive.



Experiment No: 08

Name of experiment: DNS server configuration.

Objective:

This experiment aims to configure the packet tracer's DNS server. Next, using a packet tracer to replicate it, we will test it through a web browser and a command prompt.

Apparatus: Cisco packet tracer, PC

Circuit / Network diagram:

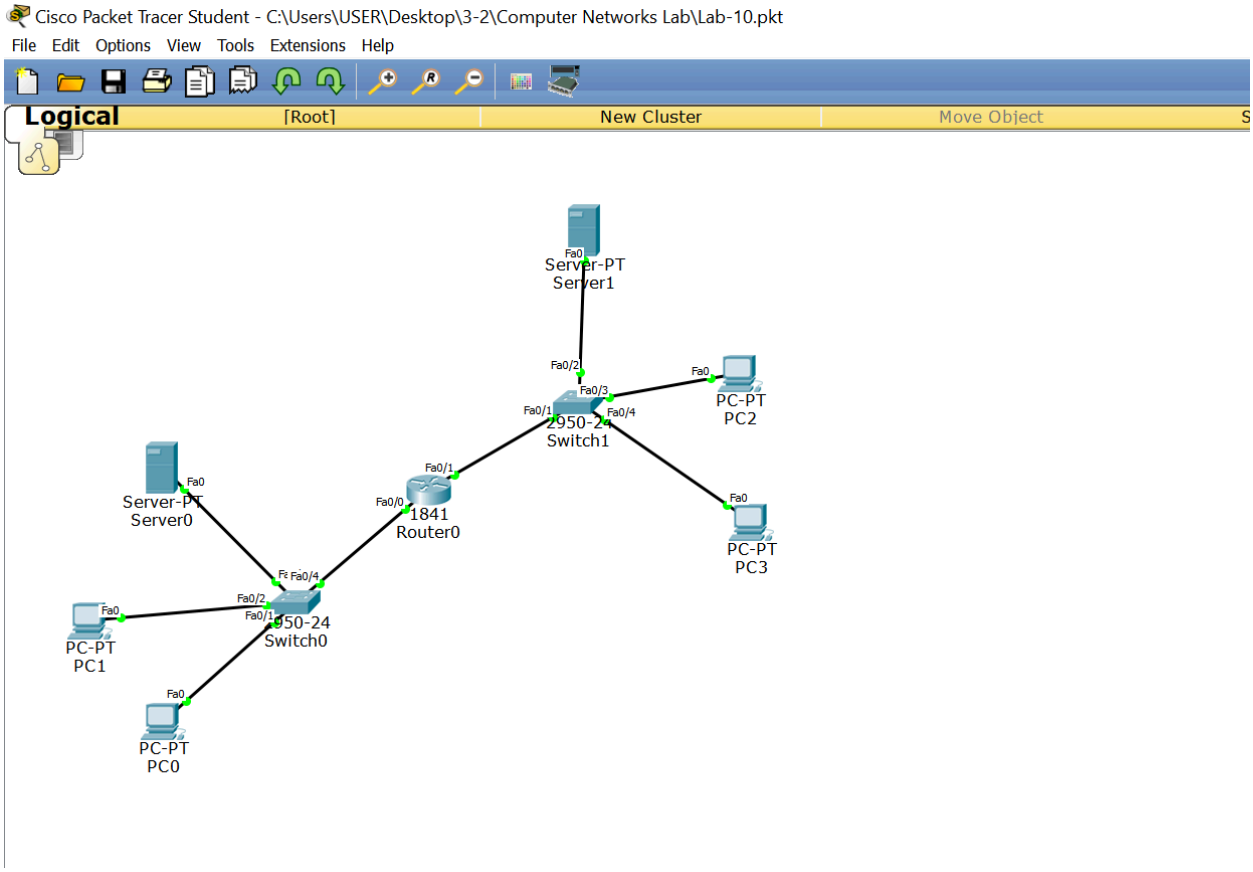
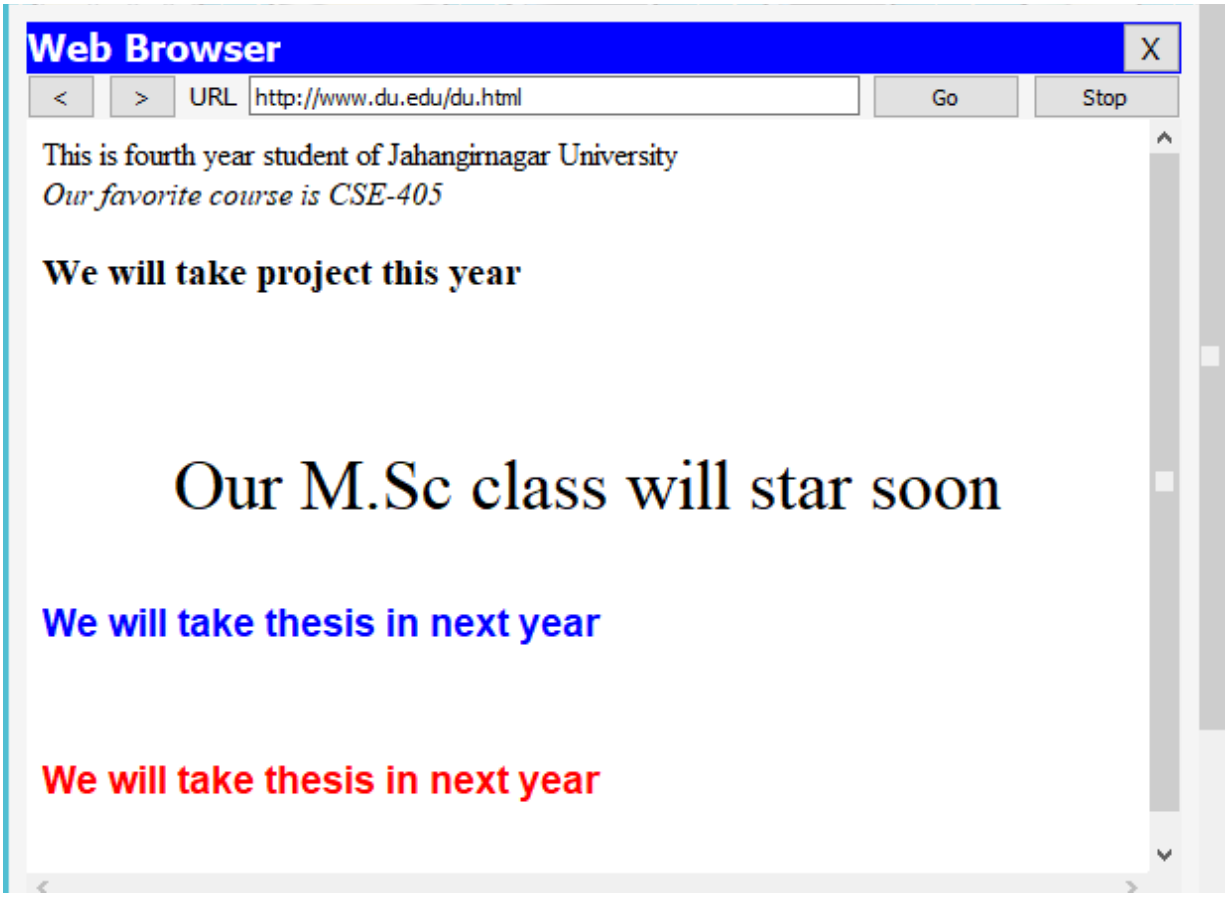


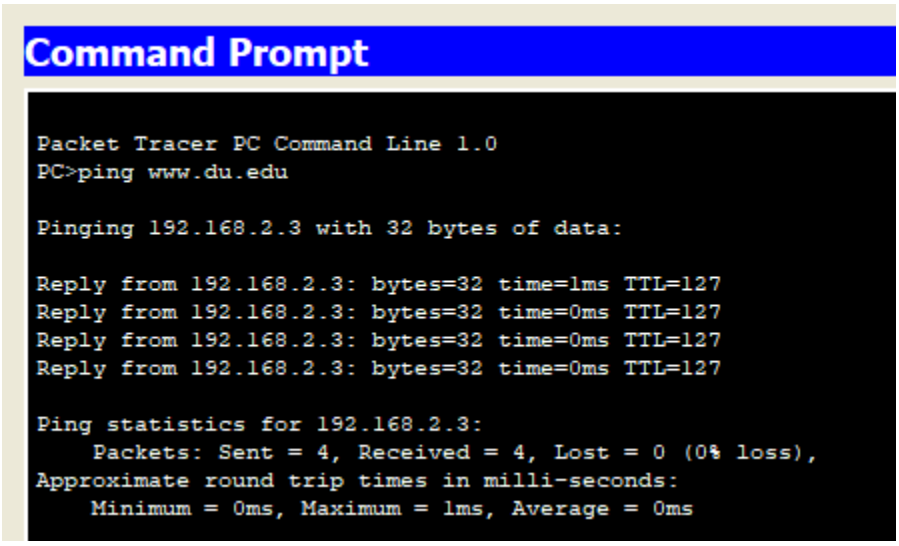
Fig: Implementation of DNS server

Result and discussion:

The browser result from PC with ip 192.168.1.2 to www.du.edu/du.html is-



Command prompt result:




```
PC>tracert www.du.edu

Tracing route to 192.168.2.3 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      192.168.1.4
  2    0 ms      0 ms      0 ms      192.168.2.3

Trace complete.

PC>ipconfig

FastEthernet0 Connection:(default port)

   Link-local IPv6 Address.....: FE80::290:2BFF:FE1C:E41D
   IP Address.....: 192.168.1.2
   Subnet Mask.....: 255.255.255.0
   Default Gateway.....: 192.168.1.4

PC>
```

Experiment No: 09

Name of experiment: Implementation of RSA algorithm in text and image encryption/decryption.

Objective:

This experiment aims to apply the RSA method to the encryption and decryption of text and images. We calculate C = Pe (mod n) to encrypt a message P, and P = Cd (mod n) to decrypt it. RGB images are used for image encryption and decryption.

Apparatus: MATLAB, PC

MATLAB CODE:

part-1

```
e = 3; n = 33; d = 7; %RSA parameters
 y='JAHANGIRNAGAR'; %input string
 z = double(y); %ASCII values
 S=z-60; %to reduce size of the integer
 for i=1:length(z)
 Encrypt(i)=mod(S(i)^e, n);
 end
 char(Encrypt)  % encrypted string
 Encrypt=double(Encrypt);
 for j=1:length(z)
 Decrypt(j)=mod(Encrypt(j)^d, n);
 end
 Recover=char(Decrypt+60)
clear all
```

close all

```
Command Window
>> e = 3; n = 33; d = 7; %RSA parameters
y='JAHANGIRNAGAR'; %input string
z = double(y); %ASCII values
S=z-60; %to reduce size of the integer
for i=1:length(z)
Encrypt(i)=mod(S(i)^e, n);
end
char(Encrypt) % encrypted string
Encrypt=double(Encrypt);
for j=1:length(z)
Decrypt(j)=mod(Encrypt(j)^d, n);
end
Recover=char(Decrypt+60)

ans =

    '□□↑□□□□□□□□□□'


Recover =

    'JAHANGIRNAGAR'

fx >>
```

Part-2

e = 3; n = 33; d = 7; N=256;

I=imread('peppers.png');

I=rgb2gray(I);

I=imresize(I,[N, N]);

subplot(2,2,1)

imshow(I)

title('Original Image')

I=double(I);

```
R=mod(I,16);
%Remainder of the image
for i=1:N
for j=1:N
Q(i,j)=uint8((I(i,j)/16)-0.5);
% Quiescent of the image
end
end
Q=double(Q);
for i =1:N
for j = 1:N
Qe(i, j)=mod(Q(i,j)^e, n);
Re(i, j)=mod(R(i,j)^e, n);
%Decryption of image
Qd(i, j)=mod((Qe(i,j))^d, n);
Rd(i, j)=mod((Re(i,j))^d, n);
end
end
Rec=Qd*16+Rd;
subplot(2,2,2)
imshow(uint8(Qe))
title('Encrypted Quiescent Image')
subplot(2,2,3)
imshow(uint8(Re))
title('Encrypted Remainder Image')
```

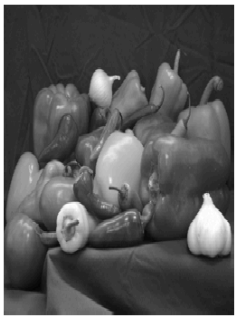
subplot(2,2,4)

imshow(uint8(Rec))

title('Decrypted Image')

Result and discussion:

Original Image



Encrypted Quiescent Image

Encrypted Reminder Image

Encrypted Quiescent Image

Experiment No: 10

Name of experiment: Implementation of OSPF(Open Shortest Path First). Algorithm

Objective:

This experiment aims to apply the Open Shortest Path First (OSPF) algorithm. After that, we'll use packet tracing to replicate it. We will taste it and trace it in the command prompt to simulate real life.

Apparatus: Cisco Packet tracer, PC

Circuit / Network diagram:

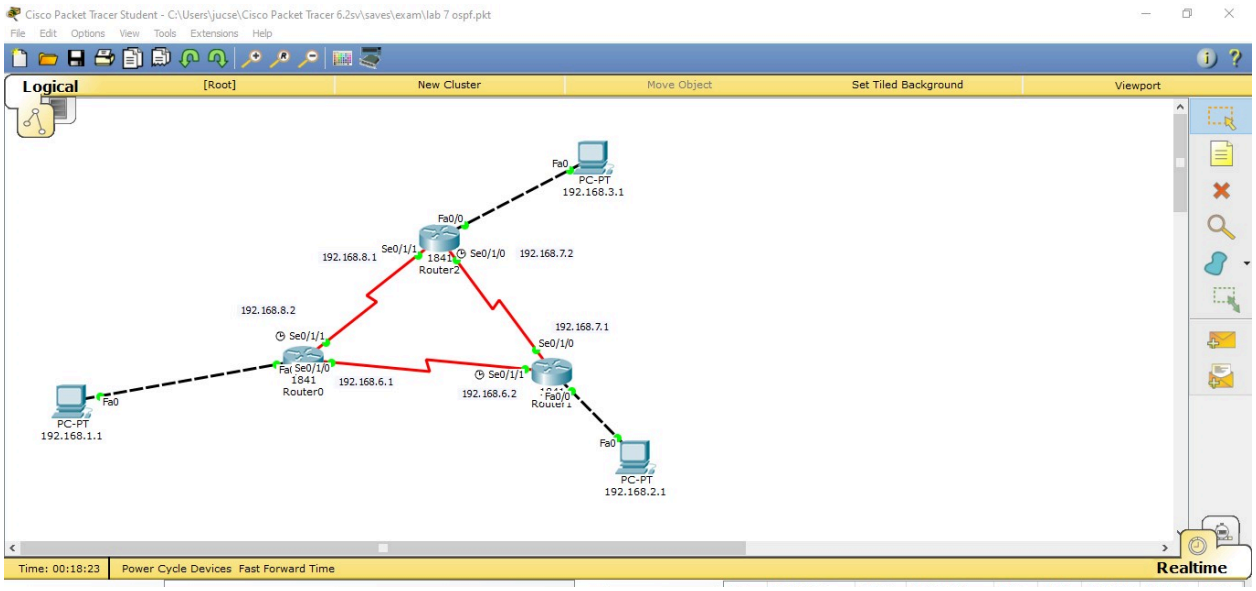
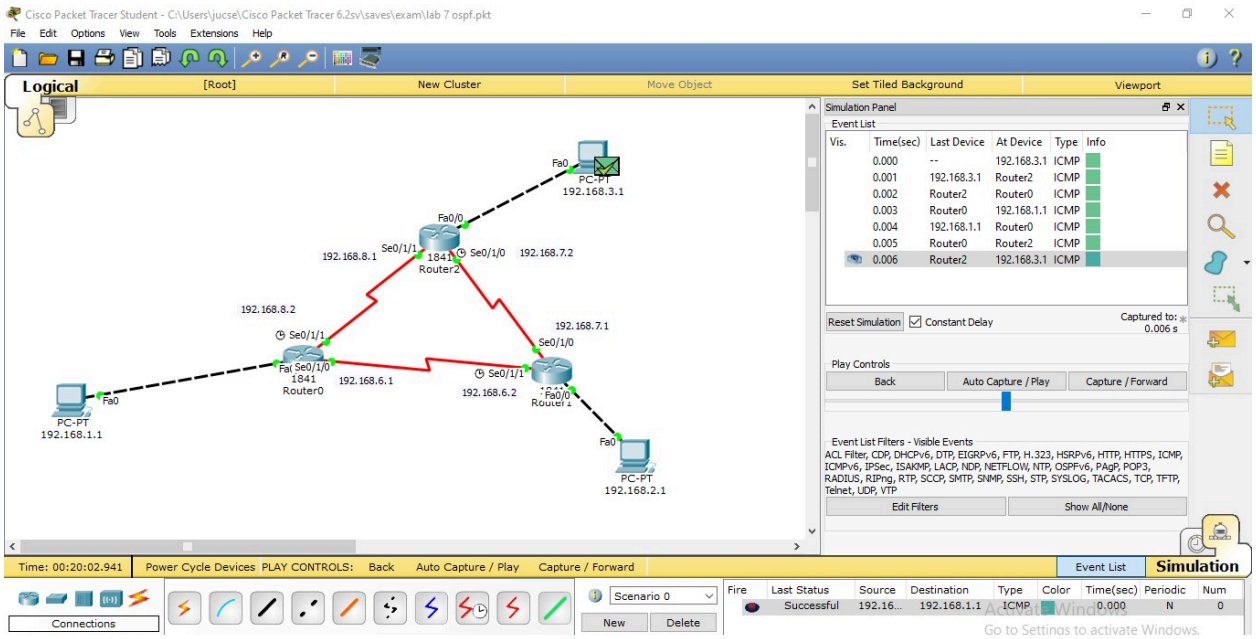


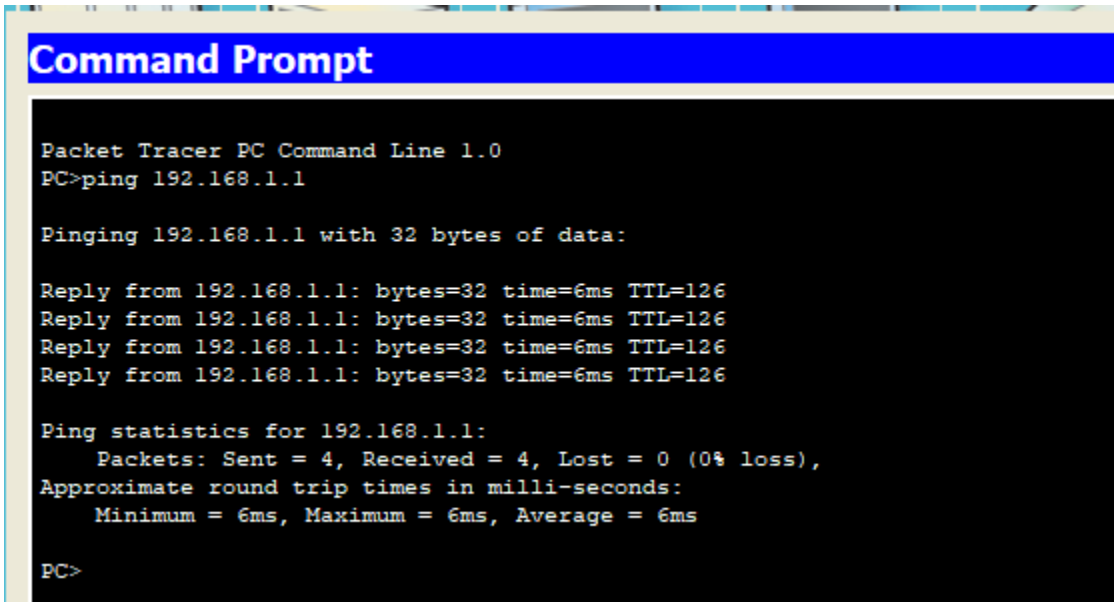
Fig: Implementation of ospf algorithm

Result and discussion:

Verification of the network in simulation mode



Verification of the network using ping



So after performing this lab, we can implement the OSPF algorithm. In both packet tracer and command prompt packet is passed successfully.