Experiment No: 01

Name of experiment: ICMP packet passing using hub and switch and trace test from cmd.

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 09.11.2023

Date of Submission: 14.05.2024

Dept of Computer Science & Engineering

Jahangirnagar University

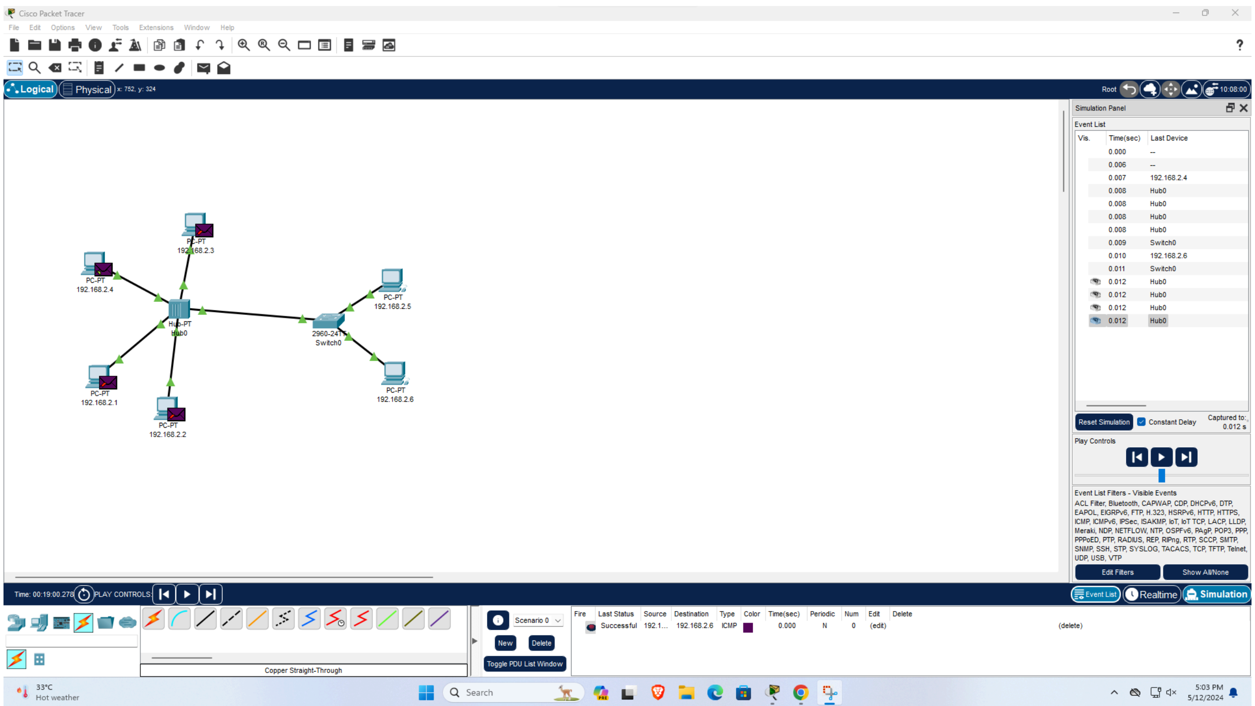
Savar, Dhaka-1342

Objective:

In this experiment we are going learn about ICMP packet passing using hub and switch. We are also going to learn about trace test from cmd.

Apparatus: Cisco Packet tracer, PC

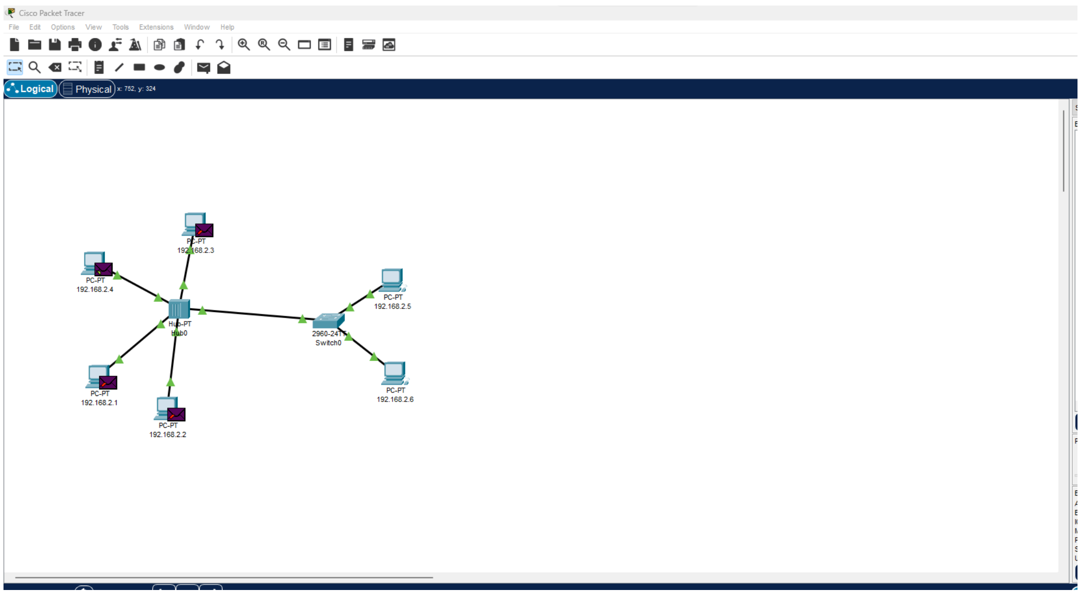
Network diagram:



Procedure and Result:

Here we are using Cisco Packet tracer to see the ICMP packet passing using hub and switch. We will first connect some PC using hub and see the message passing from one pc to another using hub. Here let's see packet passing from IP address 191.168.2.3 to IP address 191.168.2.1.

Next we are going to add some pcs and switch to the circuit. Then we will see packet passing using hubs and switch. Here let's see packet passing from IP address 191.168.2.1 to IP address 191.168.2.5.



Now we will see track test from cmd.

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

Command Prompt

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jucse>tracert www.google.com

Tracing route to www.google.com [142.250.194.36]
over a maximum of 30 hops:

  1     3 ms     1 ms     1 ms  192.168.0.1
  2     2 ms    11 ms     4 ms  10.250.0.1
  3     2 ms     2 ms     5 ms  172.20.20.1
  4    22 ms    17 ms     1 ms  172.17.124.65
  5     4 ms     5 ms     5 ms  10.162.229.53
  6     6 ms     5 ms     4 ms  10.162.228.6
  7     4 ms     9 ms     3 ms  10.13.13.37
  8    27 ms    78 ms     5 ms  be-r2-2-ag1-2.summitgw.net [103.26.244.10]
  9    34 ms    35 ms    36 ms  72.14.242.176
 10    36 ms    38 ms    35 ms  108.170.251.97
 11    35 ms    34 ms    36 ms  142.251.52.229
 12    35 ms    35 ms    35 ms  del12s02-in-f4.1e100.net [142.250.194.36]

Trace complete.
```

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.

```
C:\Users\jucse>tracert www.juniv.edu

Tracing route to juniv.edu [72.249.68.156]
over a maximum of 30 hops:

  1     1 ms     1 ms     9 ms  192.168.0.1
  2     5 ms    15 ms     2 ms  10.250.0.1
  3     2 ms     1 ms     5 ms  172.20.20.1
  4     2 ms    21 ms     2 ms  172.17.124.65
  5     4 ms     4 ms     3 ms  10.162.229.53
  6    11 ms     4 ms     6 ms  10.162.228.6
  7    20 ms    11 ms    13 ms  192.168.80.1
  8    12 ms    11 ms    11 ms  103-16-152-74-noc.bsccl.com [103.16.152.74]
  9     *        *        *     Request timed out.
 10   183 ms   156 ms   209 ms  be3148.ccr31.mrs02.atlas.cogentco.com [154.54.76.165]
 11   157 ms   164 ms   255 ms  be3077.ccr31.bio02.atlas.cogentco.com [154.54.39.225]
 12   269 ms   304 ms   509 ms  be2331.ccr41.dca01.atlas.cogentco.com [154.54.85.241]
 13   306 ms   330 ms   486 ms  be2891.ccr21.cle04.atlas.cogentco.com [154.54.82.249]
 14   281 ms   382 ms   250 ms  be2717.ccr41.ord01.atlas.cogentco.com [154.54.6.221]
 15   340 ms   306 ms   325 ms  be2831.ccr21.mci01.atlas.cogentco.com [154.54.42.165]
 16   282 ms   352 ms   712 ms  be2706.rcr21.tul01.atlas.cogentco.com [154.54.31.93]
 17   416 ms   409 ms   301 ms  be2704.rcr21.okc01.atlas.cogentco.com [154.54.7.233]
 18   305 ms   304 ms   304 ms  te0-0-0-12.nr61.b023974-0.okc01.atlas.cogentco.com [154.24.1.
 19   307 ms   310 ms   264 ms  38.122.24.34
 20   364 ms   295 ms   271 ms  te0021.corertr-01.okc.tierpoint.net [74.112.93.10]
 21   381 ms   391 ms   276 ms  ae1000.jbdr-02.dal.tierpoint.net [206.123.64.108]
 22   272 ms   271 ms   272 ms  xe-0035.jcore-03.dal.tierpoint.net [206.123.64.27]
 23   421 ms   375 ms   408 ms  207.210.228.50
```

Experiment No: 02

Name of experiment: Configuration of DSL modem and Router

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 09.11.2023

Date of Submission: 14.05.2024

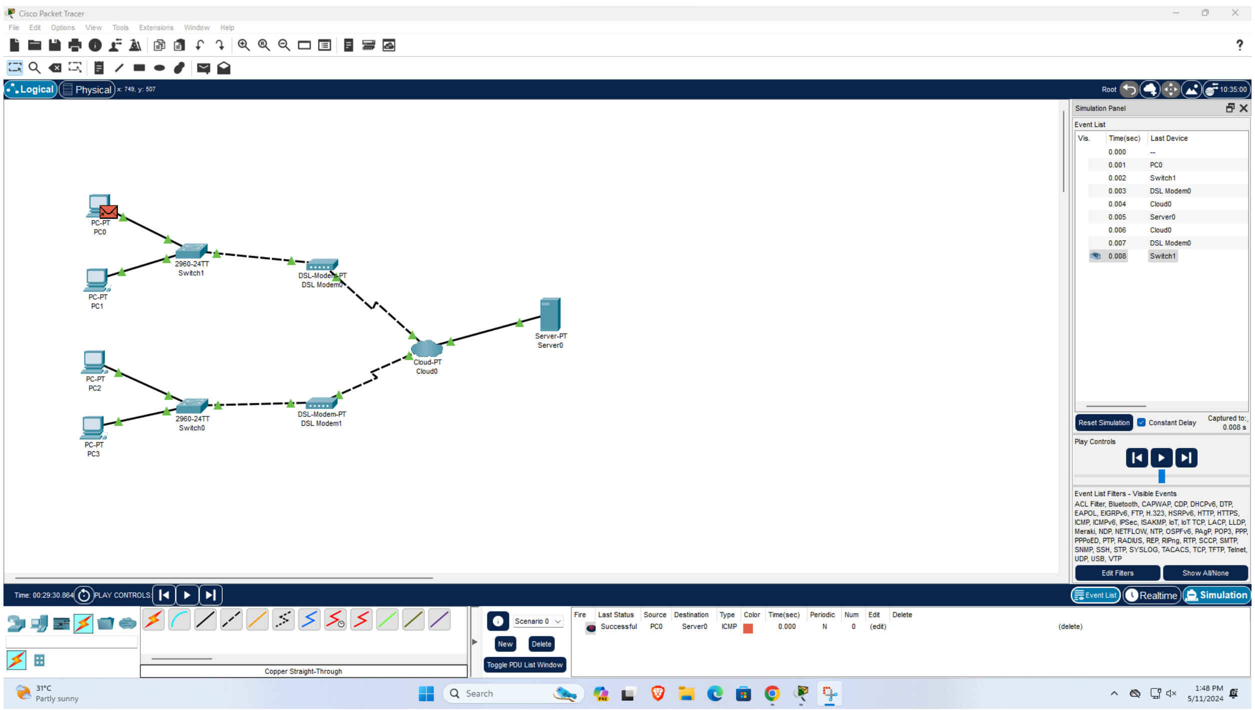
Dept of Computer Science & Engineering
Jahangirnagar University
Savar, Dhaka-1342

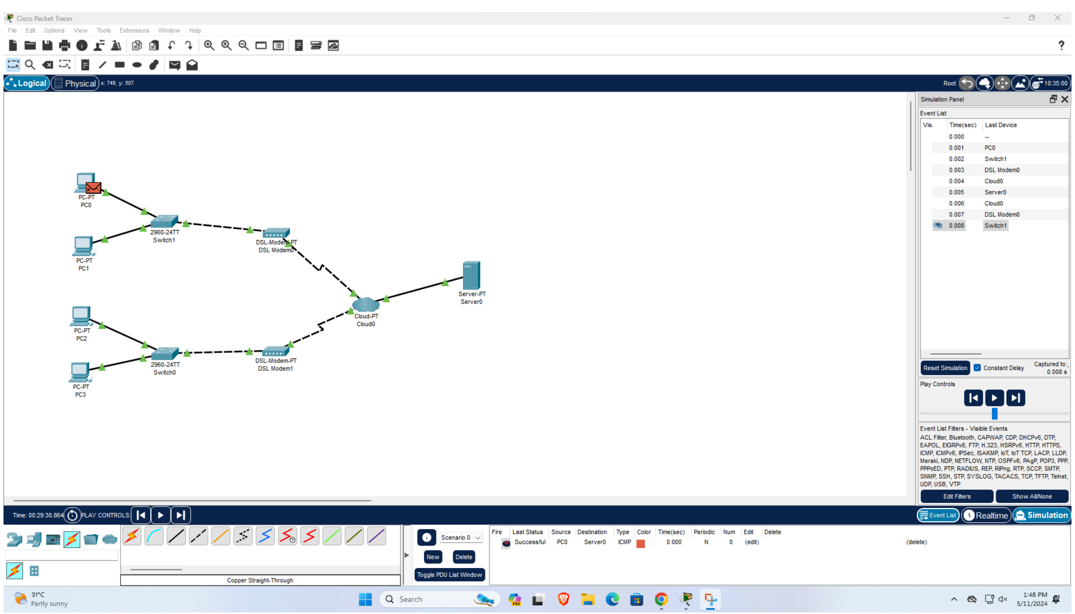
Objective:

The objective of this experiment is to implement DSL (Digital Subscriber Line) modem and routing in WAN. Then we will simulate it in packet tracer.

Apparatus: Cisco Packet tracer, PC

Network diagram:





Procedure and Result:

After performing this experiment, we will be able to connect an ISP to the phone line using DSL (Digital Subscriber Line). The computer is connected to a DSL modem that converts between digital packets and analog signals that can pass unhindered over the telephone line. At the other end, a device called a DSLAM (Digital Subscriber Line Access Multiplexer) converts between signals and packets.

Experiment No: 03

Name of experiment: Configuration of DSL modem and Router

Submitted by

Name: Shabrina Akter Shahana

Roll:2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 15.11.2023

Date of Submission: 14.05.2024

Dept of Computer Science & Engineering

Jahangirnagar University

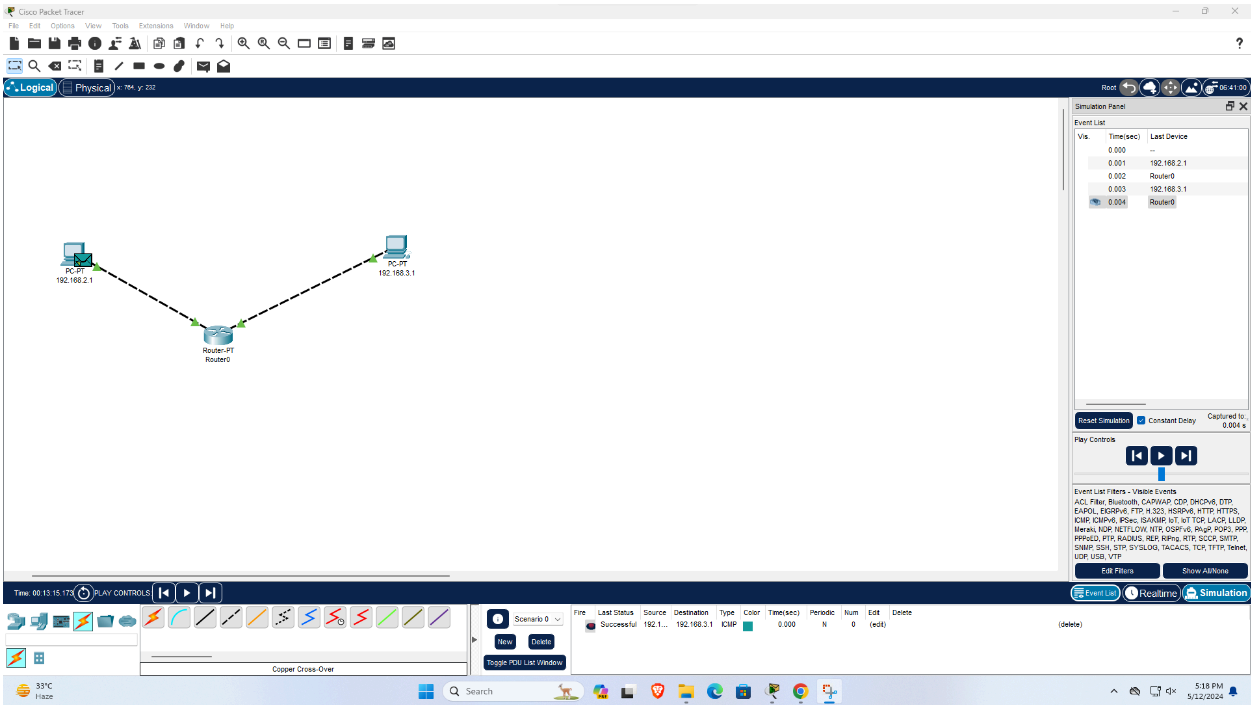
Savar, Dhaka-1342

Objective:

The objective of this experiment is to sent packets from one pc to another through routers

Apparatus: Cisco Packet tracer, PC

Network diagram:



Procedure and Result:

In Cisco Packet Tracer, connect two PCs to a router, assign unique IP addresses and subnet masks to each PC, then configure the router as the default gateway for both. Finally, ping from one PC to the other's IP to verify successful packet transmission through the router.

Experiment No: 04

Name of experiment: Router configuration using CLI

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 22.11.2023

Date of Submission: 14.05.2024

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

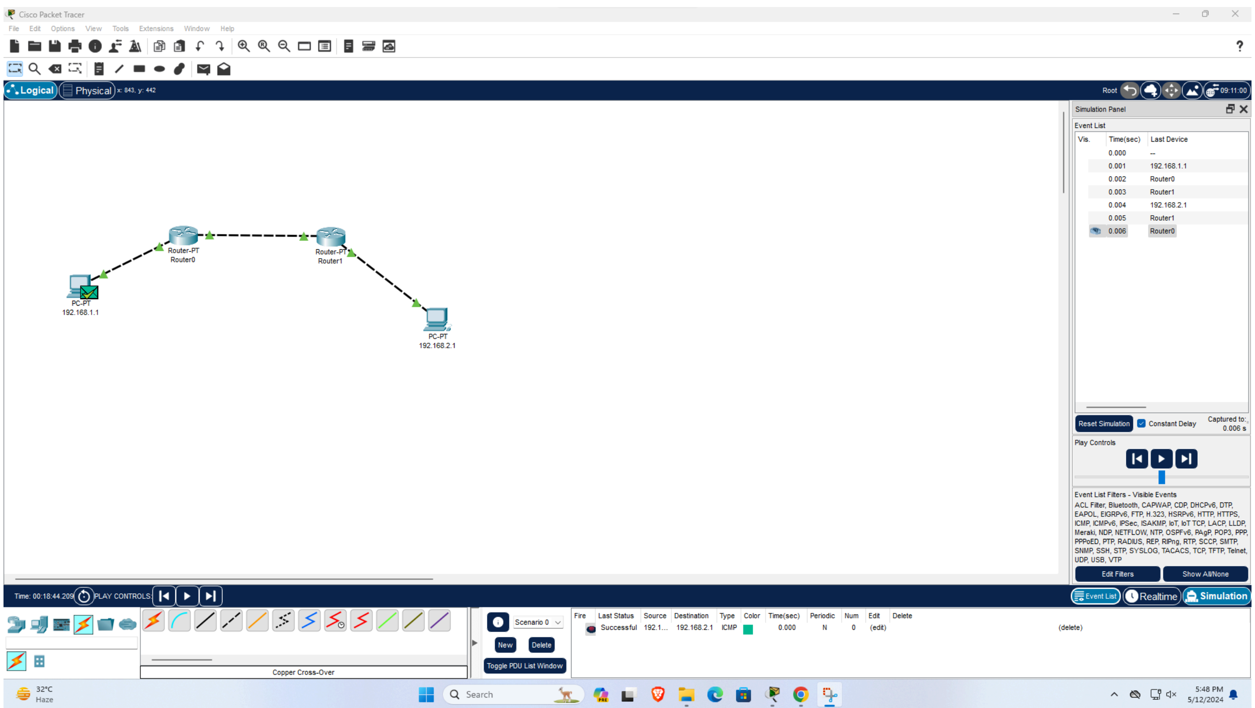
Objective:

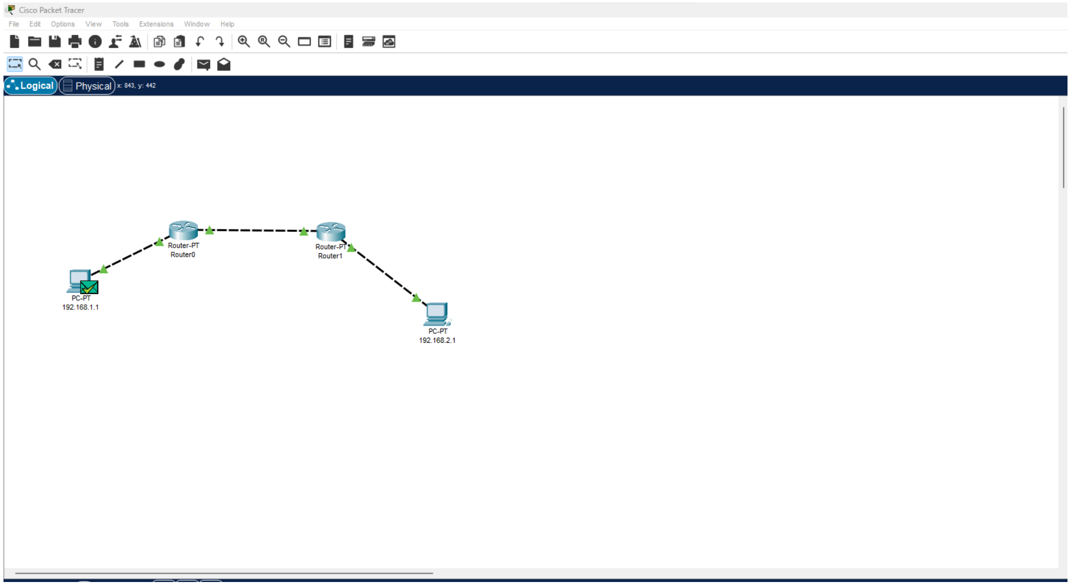
In this experiment we are going learn about router configuration using command line interface (CLI). We will connect routers with pc from command prompt and check their connection through cmd.

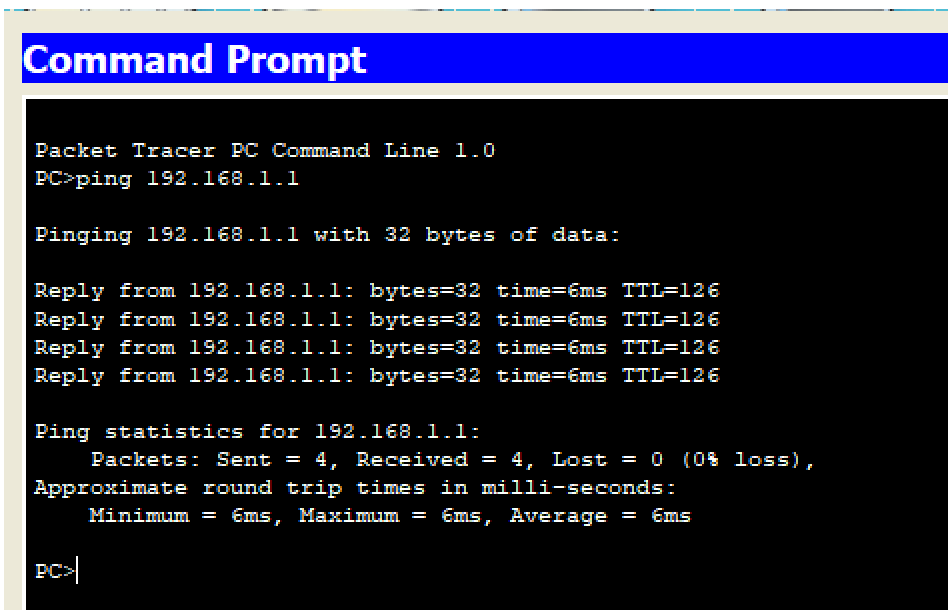
Apparatus: Cisco Packet tracer, PC

Network Diagram:

Procedure and Result:







After performing this lab, we will be able to route using CLI. In both packet tracer and command prompt packet is passed successfully. Since in router we use IP address instead of mac address so it is possible to send packet among different network id.

Experiment No: 05

Name of experiment: VLAN Configuration with Switch and Router

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 29.11.2023

Date of Submission: 14.05.2024

Dept of Computer Science & Engineering

Jahangirnagar University

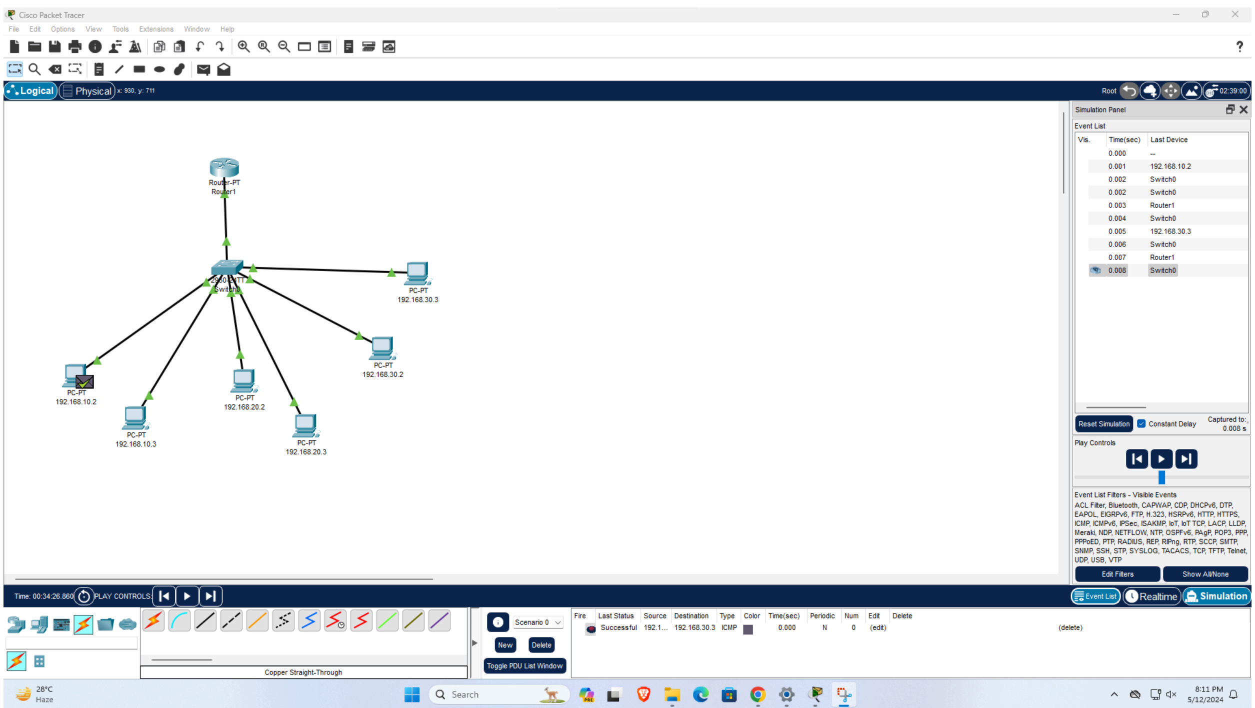
Savar, Dhaka-1342

Objective:

The objective of this experiment is to implement VLAN using switch and router. Then we will simulate it in packet tracer. For real life simulation we will taste and trace it in command prompt.

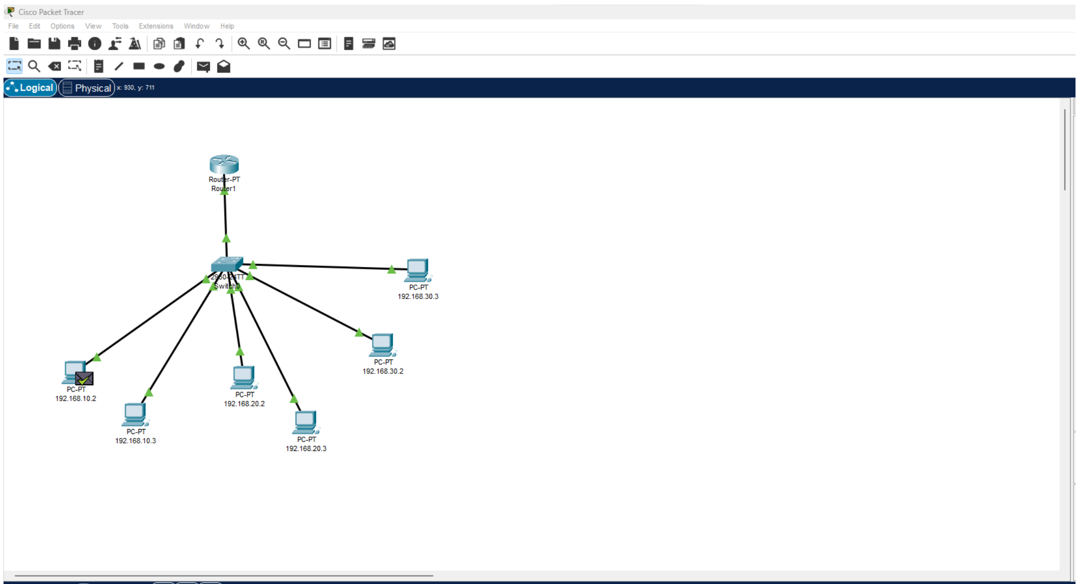
Apparatus: Cisco Packet tracer, PC

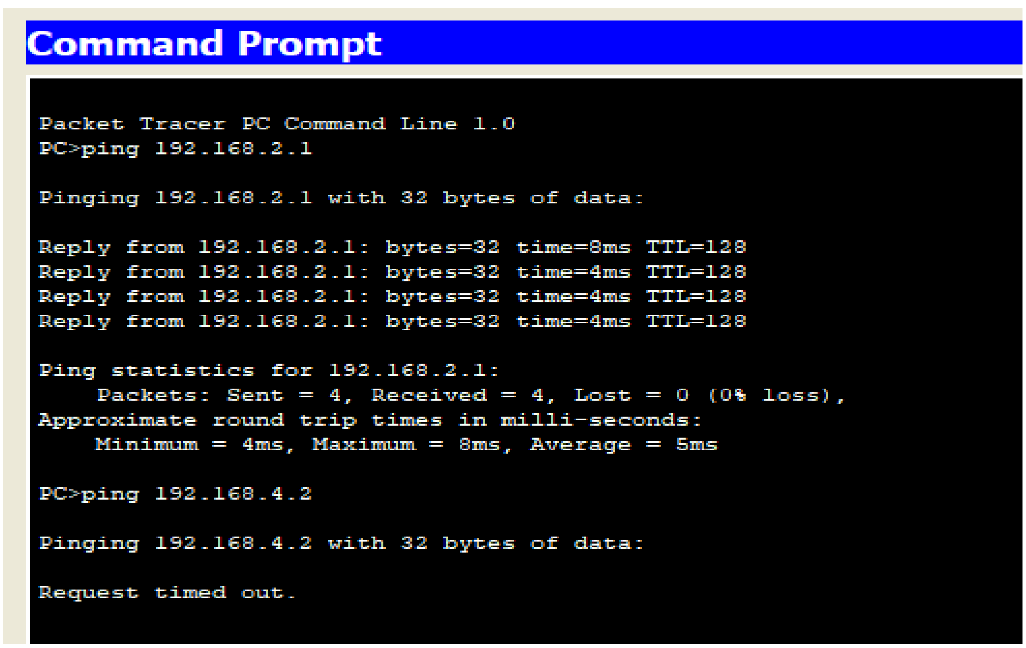
Network diagram:



Procedure and Result:

When we apply ping in the same VLAN the packet is sent successfully. But in case of different VLAN the packet sending is failed.





Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=8ms TTL=128
Reply from 192.168.2.1: bytes=32 time=4ms TTL=128
Reply from 192.168.2.1: bytes=32 time=4ms TTL=128
Reply from 192.168.2.1: bytes=32 time=4ms TTL=128

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 8ms, Average = 5ms

PC>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Request timed out.
```

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

Experiment No: 06

Name of experiment: Implementation of wireless LAN (wifi)

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 06.12.2023

Date of Submission: 14.05.2024

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

Objective:

The objective of this experiment is to implement wireless LAN in cisco packet tracer. For real life simulation we will taste and trace it in command prompt.

Apparatus: Cisco Packet tracer, PC

Network diagram:

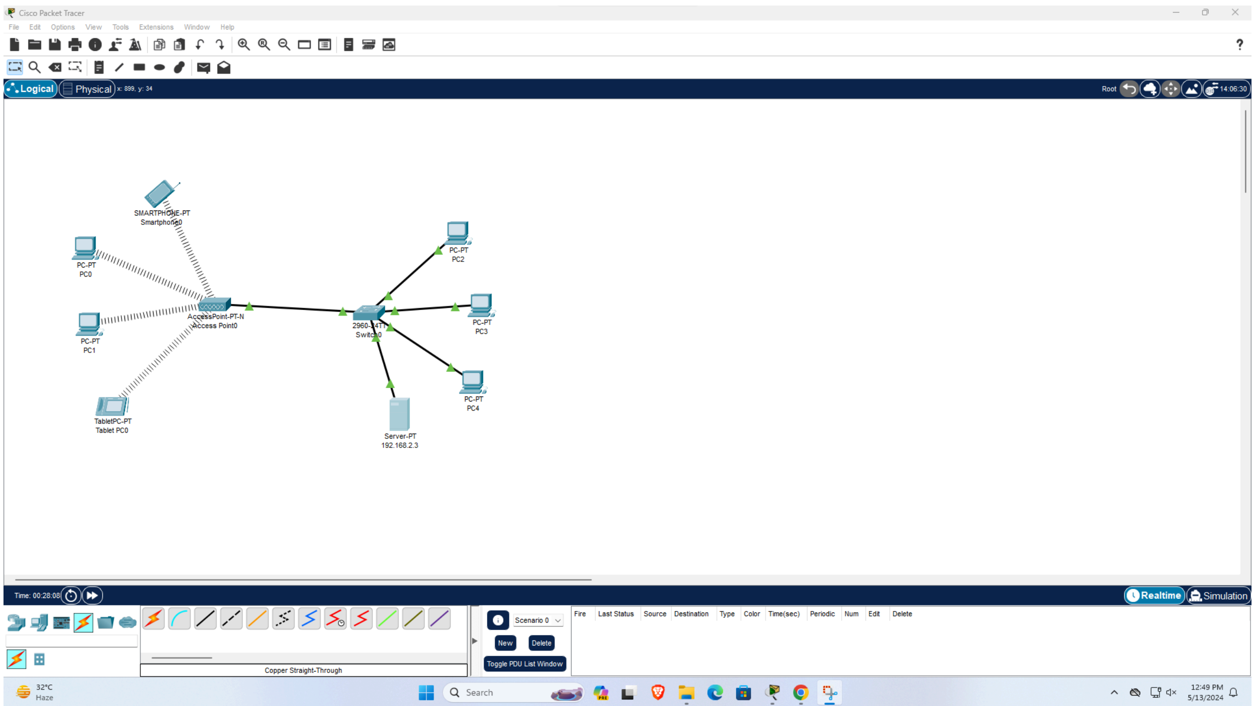
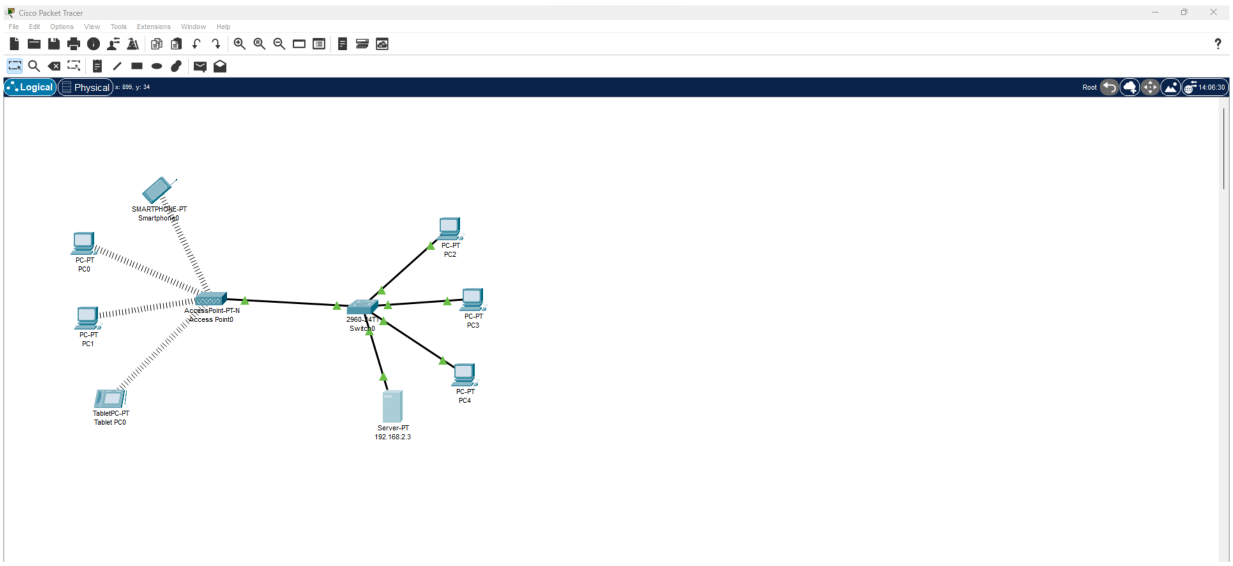


Fig: wireless LAN configuration

Procedure and Result:

We can connect wireless connection without authentication in access port.



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

Experiment No: 07

Name of experiment: Implementation of IP telephony

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 13.12.2023

Date of Submission: 14.05.2024

Dept of Computer Science & Engineering

Jahangirnagar University

Savar, Dhaka-1342

Objective:

The objective of this experiment is to implement a small network of IP telephony. Each telephone will be verified with its content of IP address and corresponding telephone number. Finally, the network will be tested by dialing to each other IP phone.

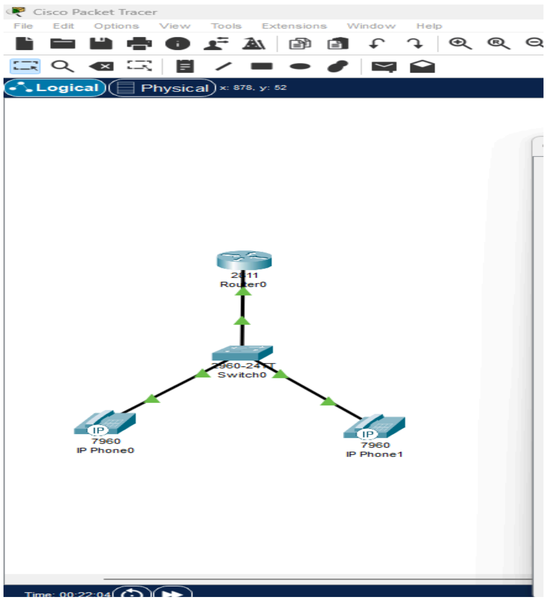
Apparatus: Cisco Packet tracer, PC

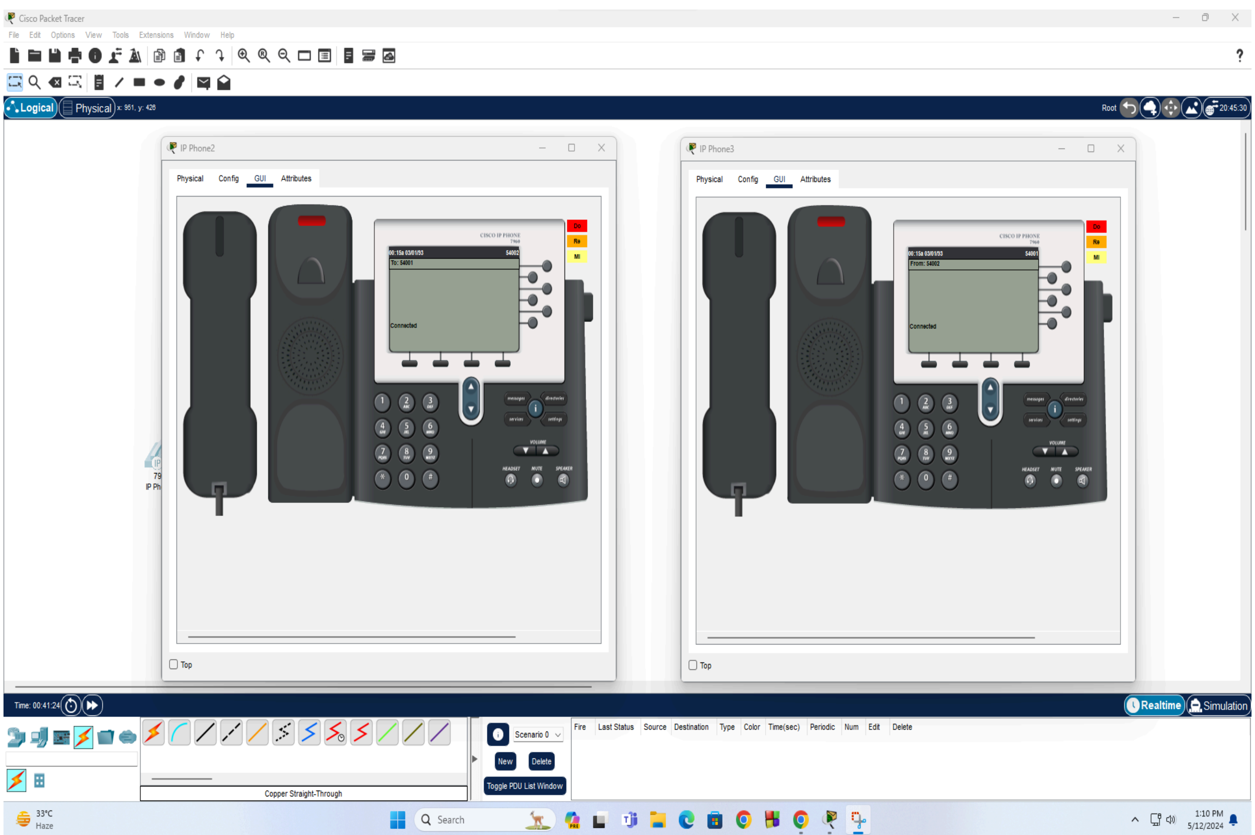
Network diagram:

Fig: Implementation of IP telephony

Procedure and Result:

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: 08

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

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 20.12.2023

Date of Submission: 14.05.2024

Dept of Computer Science & Engineering

Jahangirnagar University

Savar, Dhaka-1342

Objective:

The objective of this experiment is to establish connection between two nodes client and server of a network called socket programming. We will implement this experiment in python.

Apparatus: Visual studio, PC

Procedure :

VS Codes:

Server Side:

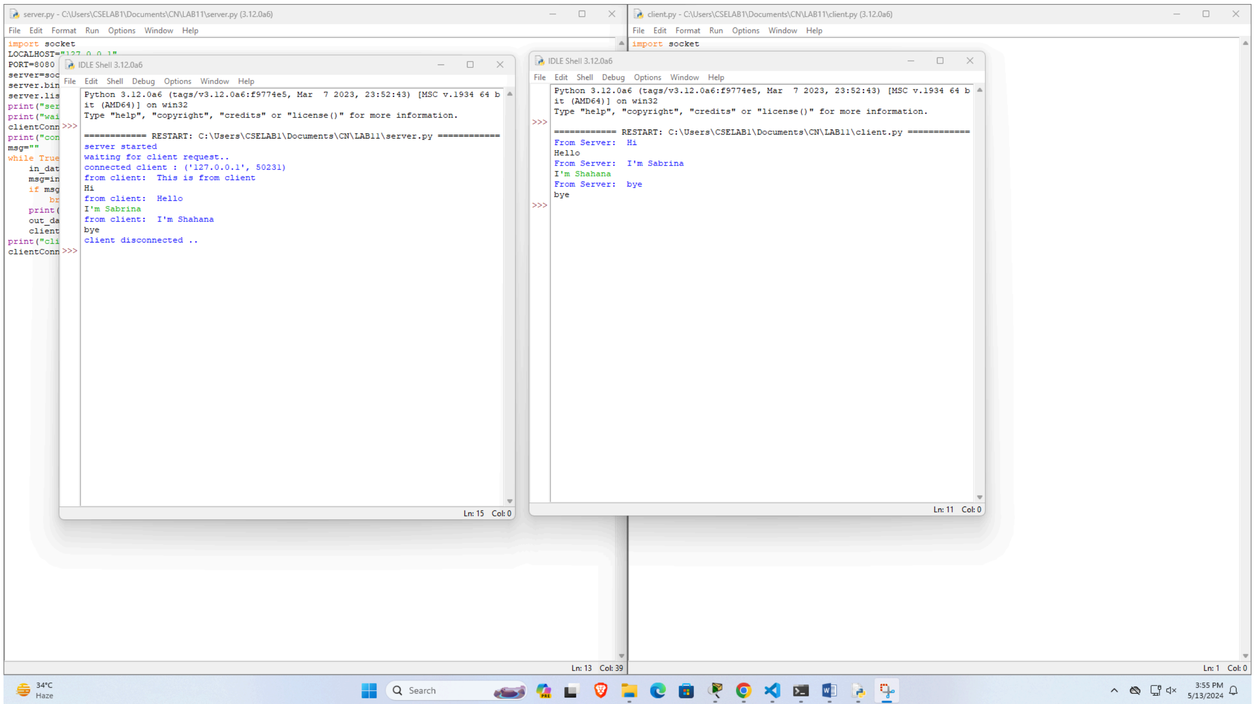
```
import socket
LOCALHOST="127.0.0.1"
PORT=8080
server=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
server.bind((LOCALHOST, PORT))
server.listen(1)
print("server started")
print("waiting for client request..")
clientConnection,clientAddress=server.accept()
print("connected client :" ,clientAddress)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

After performing this lab, we will be able to establish a connection between client and server. After a connection is established, the server prints out the client address and then waits for data. The client can SEND and RECEIVE data according to the protocols being used. When both client and server issue the CLOSE primitive, the connection will be torn down.


server.py - C:\Users\CSELAB1\Documents\CN\LAB11\server.py (3.12.0a6)
IDLE Shell 3.12.0a6
Python 3.12.0a6 (tags/v3.12.0a6:f9774e5, Mar 7 2023, 23:52:43) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
============ RESTART: C:\Users\CSELAB1\Documents\CN\LAB11\server.py ============
server started
waiting for client request..
connected client : ('127.0.0.1', 50231)
from client: This is from client
Hi
from client: Hello
I'm Sabrina
from client: I'm Shahana
bye
client disconnected ..
client.py - C:\Users\CSELAB1\Documents\CN\LAB11\client.py (3.12.0a6)
import socket
IDLE Shell 3.12.0a6
Python 3.12.0a6 (tags/v3.12.0a6:f9774e5, Mar 7 2023, 23:52:43) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
============ RESTART: C:\Users\CSELAB1\Documents\CN\LAB11\client.py ============
From Server: Hi
Hello
From Server: I'm Sabrina
I'm Shahana
From Server: bye
bye

Experiment No: 09

Name of experiment: DNS server configuration

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 10.01.2024

Date of Submission: 14.05.2024

জাহাঙ্গীরনগর বিশ্ববিদ্যালয়

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

Objective:

The objective of this experiment is to configure DNS server in packet tracer. Then we will simulate it in packet tracer and test from web browser and command prompt.

Apparatus: Cisco packet tracker, PC

Network diagram:

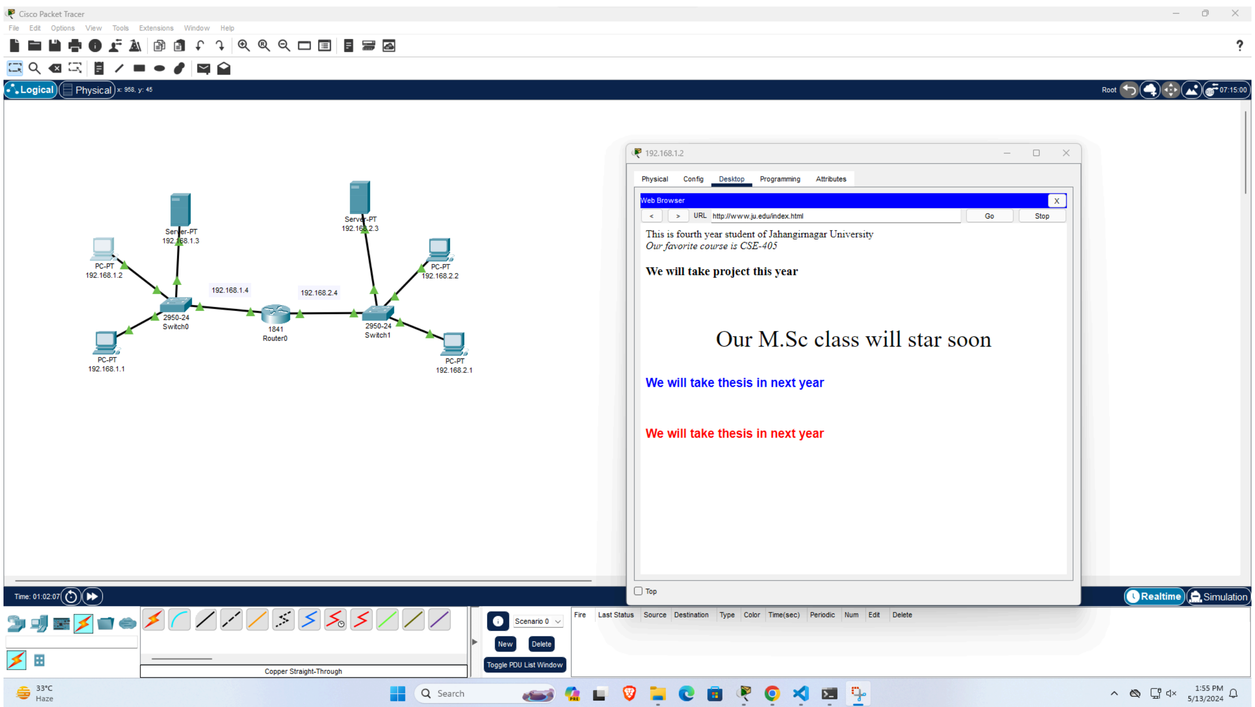
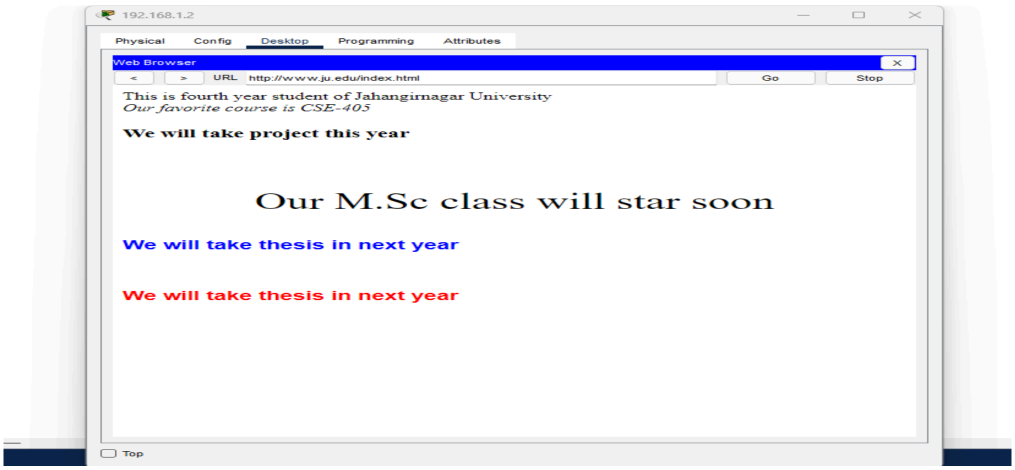
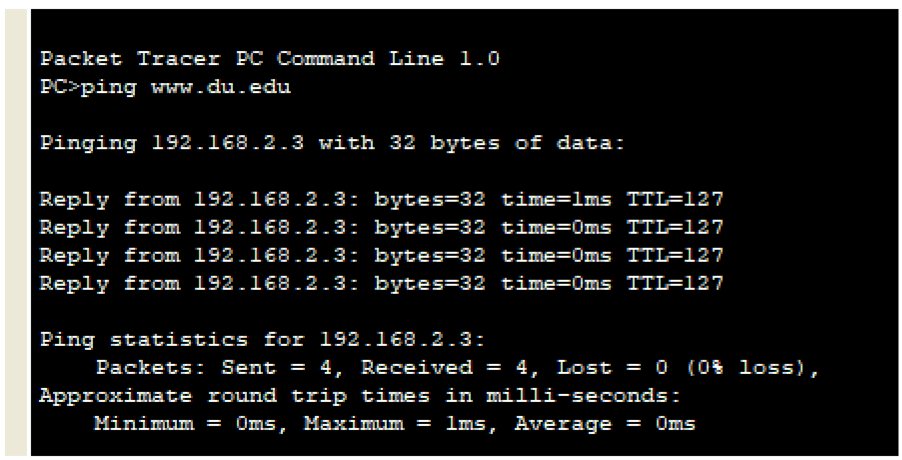


Fig: Implementation of DNS server

Procedure and Result:

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





Experiment No: 10

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

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 17.01.2024

Date of Submission: 14.05.2024

জাহাঙ্গীরনগর বিশ্ববিদ্যালয়

Dept of Computer Science & Engineering

Jahangirnagar University

Savar, Dhaka-1342

Objective:

The objective of this experiment is to implement RSA algorithm in text and image encryption/decryption. To encrypt a message, P, we compute C = Pe (mod n) and to decrypt C,we compute P = Cd (mod n). For image encryption & decryption we use RGB image.

Apparatus: Visual Studio, PC

Procedure:

Code:

```
import math

import random

from PIL import Image


# Function to generate prime numbers

def generate_primes(n):

    primes = []

    for num in range(2, n):

        prime = True

        for i in range(2, int(math.sqrt(num)) + 1):

            if (num % i) == 0:

                prime = False

                break

        if prime:

            primes.append(num)

    return primes
```

```
# Function to generate public and private keys
def generate_keys(p, q):
    n = p * q
    phi = (p - 1) * (q - 1)

    # Choose e such that e and phi(n) are coprime
    e = random.randrange(1, phi)
    g = math.gcd(e, phi)
    while g != 1:
        e = random.randrange(1, phi)
        g = math.gcd(e, phi)

    # Compute d, the modular inverse of e
    d = mod_inverse(e, phi)

    return ((e, n), (d, n))

# Function to compute modular inverse
def mod_inverse(a, m):
    m0, x0, x1 = m, 0, 1
    while a > 1:
        q = a // m
        m, a = a % m, m
        x0, x1 = x1 - q * x0, x0
    return x1 + m0 if x1 < 0 else x1

# Function to encrypt the image
def encrypt_image(image_path, public_key):
    image = Image.open(image_path)
```

```
    width, height = image.size
    pixels = list(image.getdata())

    e, n = public_key
    encrypted_pixels = []
    for pixel in pixels:
        encrypted_pixel = tuple(pow(component, e, n) for component in pixel)
        encrypted_pixels.append(encrypted_pixel)

    return encrypted_pixels, width, height

# Function to save the encrypted image
def save_encrypted_image(encrypted_pixels, width, height, output_path):
    encrypted_image = Image.new('RGB', (width, height))
    encrypted_image.putdata(encrypted_pixels)
    encrypted_image.save(output_path)

# Function to decrypt the image
def decrypt_image(encrypted_pixels, private_key):
    d, n = private_key
    decrypted_pixels = []
    for pixel in encrypted_pixels:
        decrypted_pixel = tuple(pow(component, d, n) for component in pixel)
        decrypted_pixels.append(decrypted_pixel)

    return decrypted_pixels

# Function to save the decrypted image
def save_decrypted_image(decrypted_pixels, width, height, output_path):
    decrypted_image = Image.new('RGB', (width, height))
```

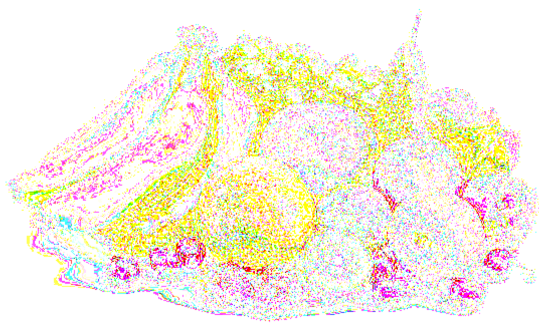
```
    decrypted_image.putdata(decrypted_pixels)
    decrypted_image.save(output_path)

# Main function
def main():
    image_path = 'image.jpg'  # Path to your image file
    output_path_encrypted = 'encrypted_image.png'  # Output path for encrypted image
    output_path_decrypted = 'decrypted.jpg'  # Output path for decrypted image

    # Generate prime numbers
    primes = generate_primes(100)
    p, q = random.choice(primes), random.choice(primes)

    # Generate keys
    public_key, private_key = generate_keys(p, q)

    # Encrypt image
    encrypted_pixels, width, height = encrypt_image(image_path, public_key)

    # Save encrypted image
    save_encrypted_image(encrypted_pixels, width, height, output_path_encrypted)

    # Decrypt image
    decrypted_pixels = decrypt_image(encrypted_pixels, private_key)

    # Save decrypted image
    save_decrypted_image(decrypted_pixels, width, height, output_path_decrypted)

if __name__ == "__main__":
    main()
```

your code goes here

Result and discussion:

Original Image:

Encrypted Image:



Decrypted Image:

Experiment No: 11

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

Submitted by

Name: Shabrina Akter Shahana

Roll: 2147

Exam Roll: 191323

Submitted to

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

Date of Performance: 23.01.2023

Date of Submission: 14.05.2024

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

Objective:

The objective of this experiment is to implement OSPF (Open Shortest Path First) Algorithm. Then we will simulate it in packet tracer. For real life simulation we will taste and trace it in command prompt.

Apparatus: Cisco Packet tracer, PC

Network diagram:

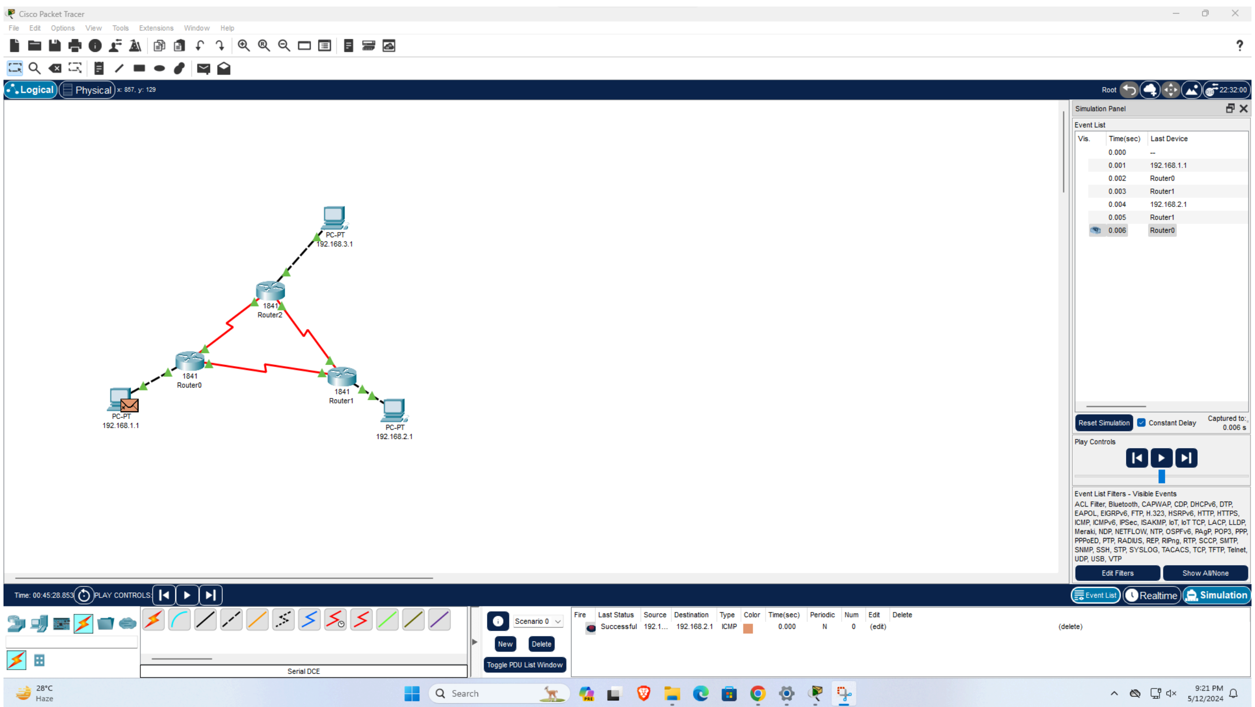
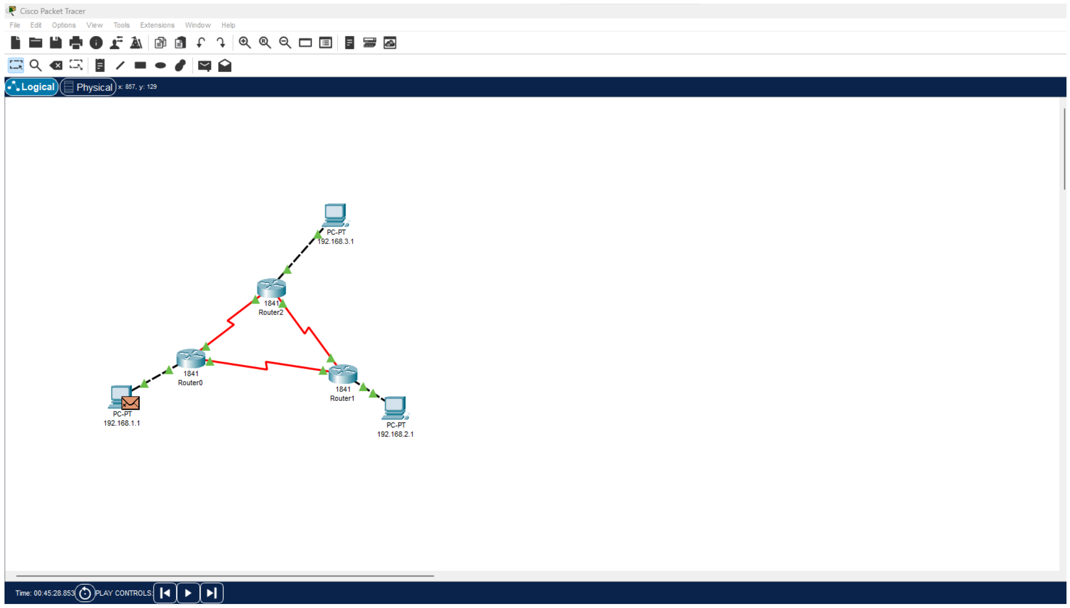


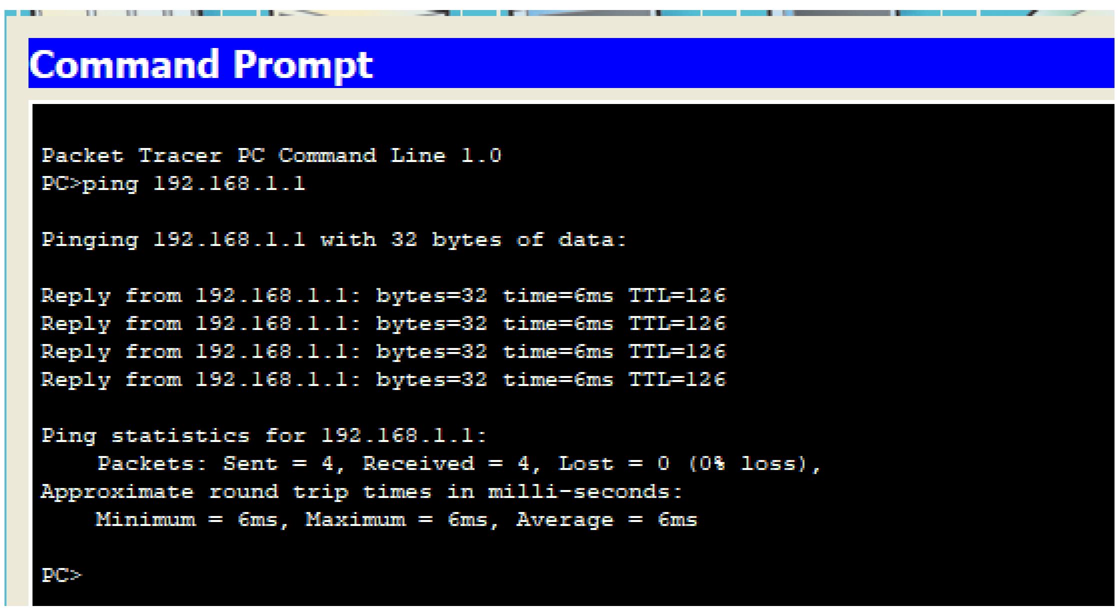
Fig: Implementation of ospf algorithm

Procedure:

Verification of the network in simulation mode



Verification of the network using ping




```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=6ms TTL=126
Reply from 192.168.1.1: bytes=32 time=6ms TTL=126
Reply from 192.168.1.1: bytes=32 time=6ms TTL=126
Reply from 192.168.1.1: bytes=32 time=6ms TTL=126

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 6ms, Average = 6ms

PC>
```

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