VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

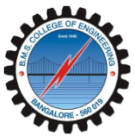
LAB REPORT
on

COMPUTER NETWORKS

Submitted by

ANSHA PRASHANTH (1BM20CS018)

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 "LAB COURSE **TITLE**" carried out by **ANSHA PRASHANTH (1BM20CS018),** who is a 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.

Dr. Latha N R
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

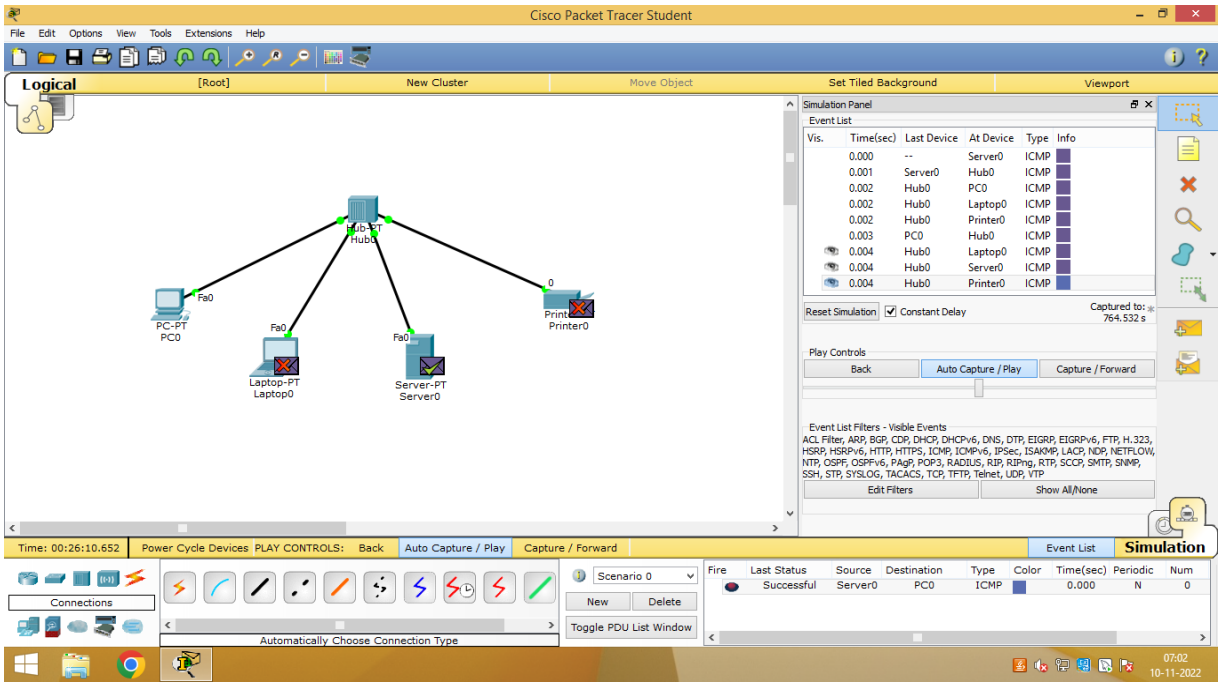
Index

Cycle-1: Experiment 1

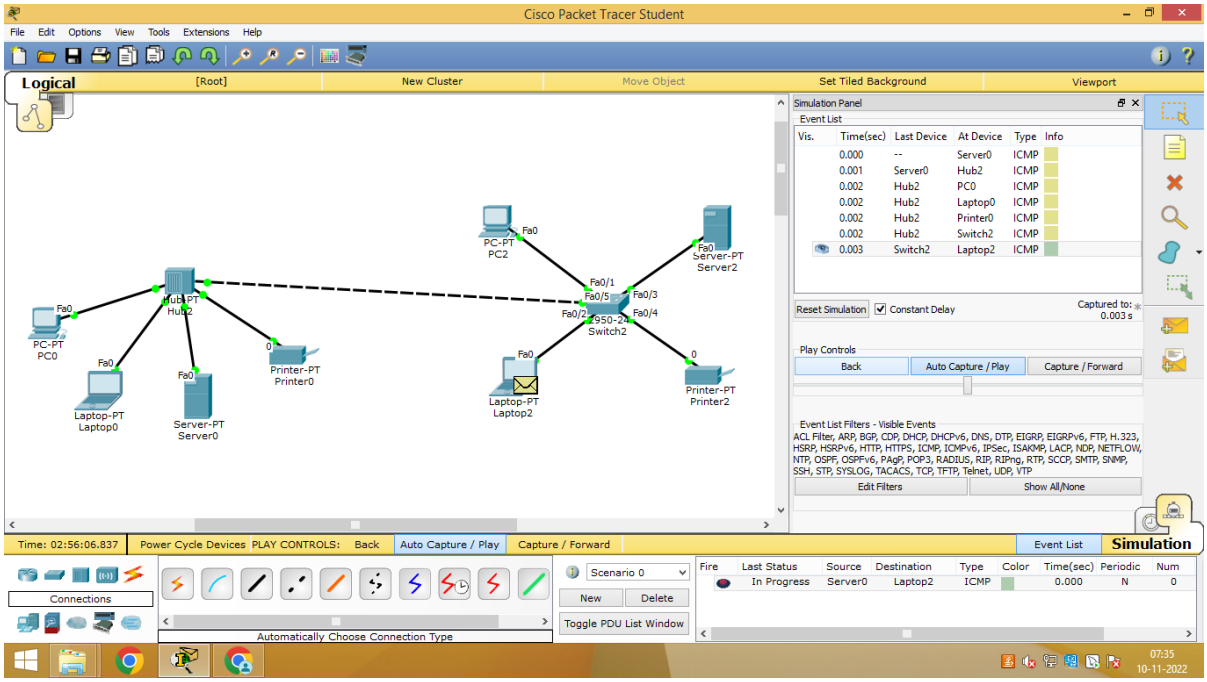
Aim of the experiment

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

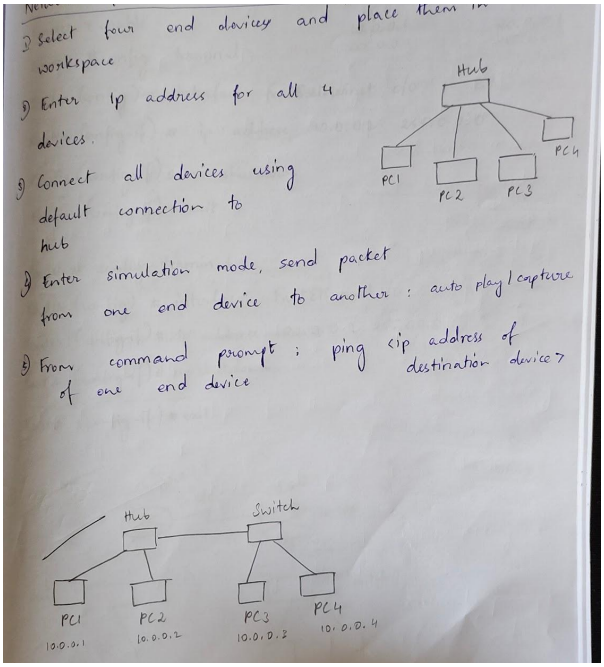
Hub Topology



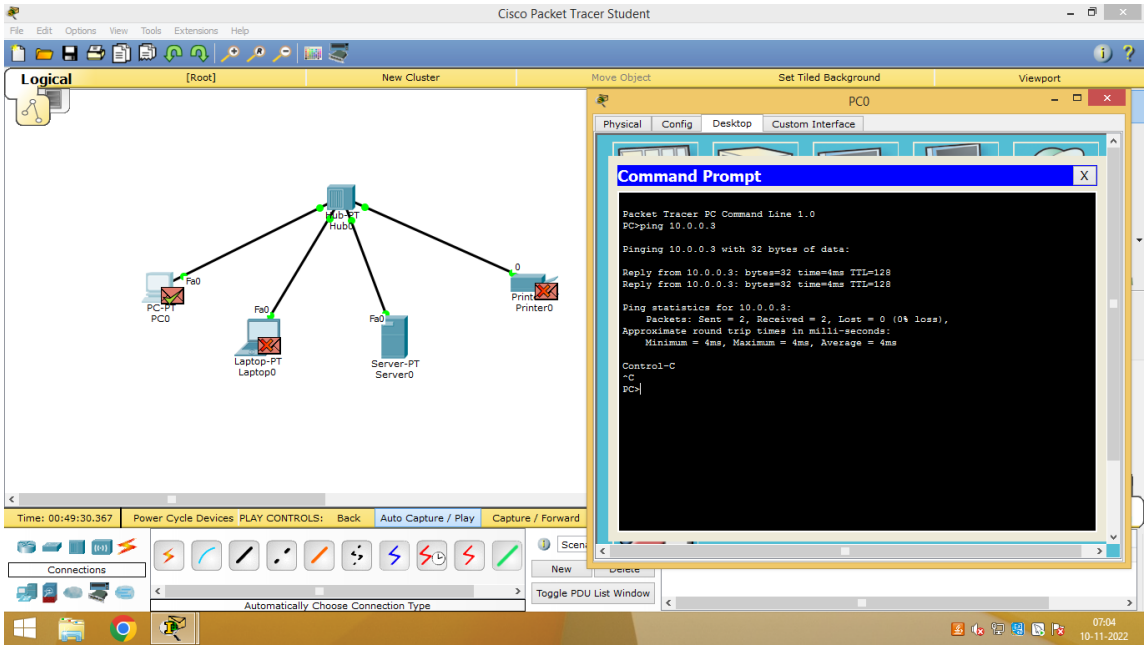
Switch Topology

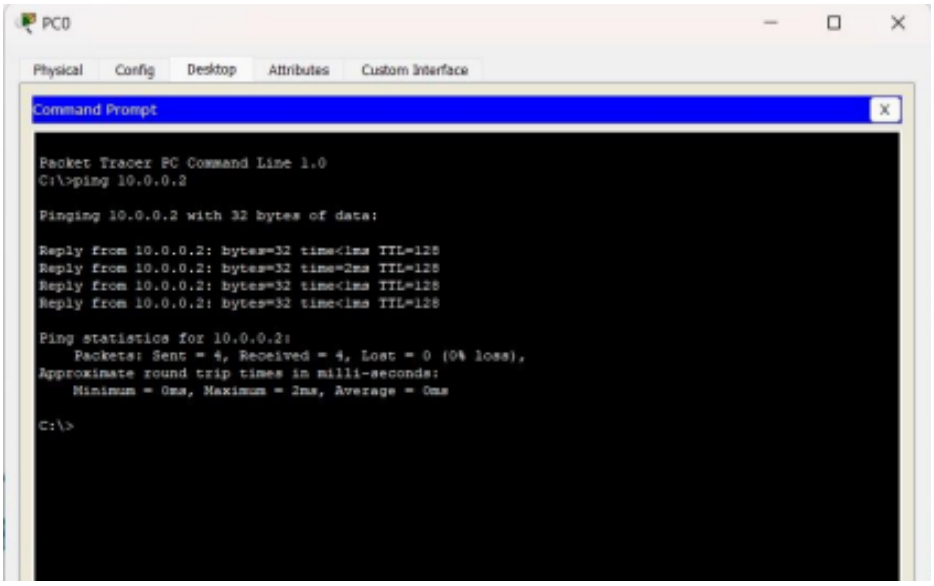

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.000 -- Server0 ICMP
0.001 Server0 Hub2 ICMP
0.002 Hub2 PC0 ICMP
0.002 Hub2 Laptop0 ICMP
0.002 Hub2 Printer0 ICMP
0.002 Hub2 Switch2 ICMP
0.003 Switch2 Laptop2 ICMP
Reset Simulation
Constant Delay
Captured to: 0.003 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
PC-PT PC2
Server-PT Server2
Hub-PT Hub2
2950-24 Switch2
PC-PT PC0
Printer-PT Printer0
Laptop-PT Laptop0
Server-PT Server0
Laptop-PT Laptop2
Printer-PT Printer2
Time: 02:56:06.837
Power Cycle Devices
PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
In Progress Server0 Laptop2 ICMP 0.000 N 0
07:35
10-11-2022

Procedure


1) Select four end devices and place them in workspace
2) Enter ip address for all 4 devices.
3) Connect all devices using default connection to hub
4) Enter simulation mode, send packet from one end device to another : auto play / capture
5) From command prompt ; ping <ip address of destination device> of one end device
Hub
PC1
PC2
PC3
PC4
Switch
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4

Output


Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 4ms, Average = 4ms
Control-C
^C
PC>
Hub-PT
Hub0
PC-PT
PC0
Laptop-PT
Laptop0
Server-PT
Server0
Printer0
Fa0
Time: 00:49:30.367
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Connections
Automatically Choose Connection Type
New
Toggle PDU List Window
07:04
10-11-2022

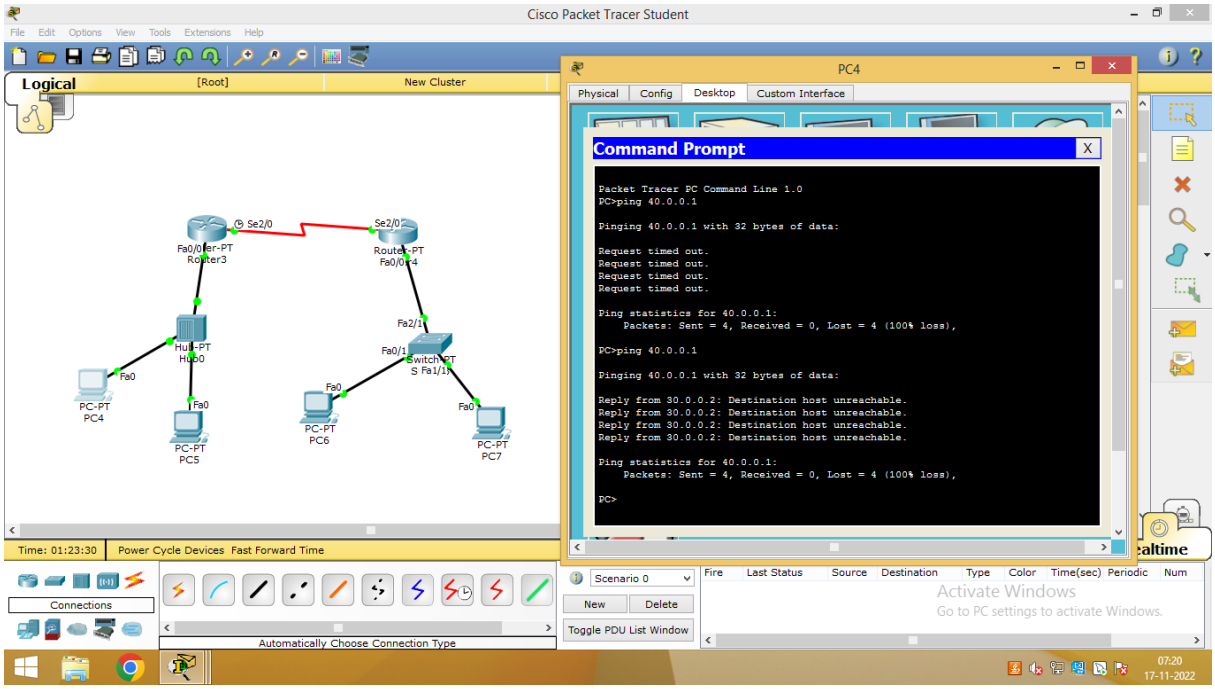

PC0
Physical
Config
Desktop
Attributes
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=2ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms
C:\>

Experiment 2

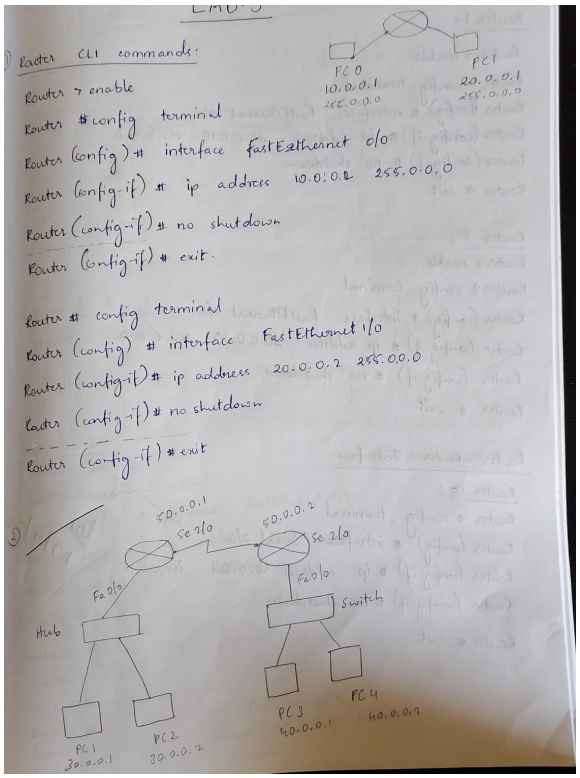
Aim of the experiment

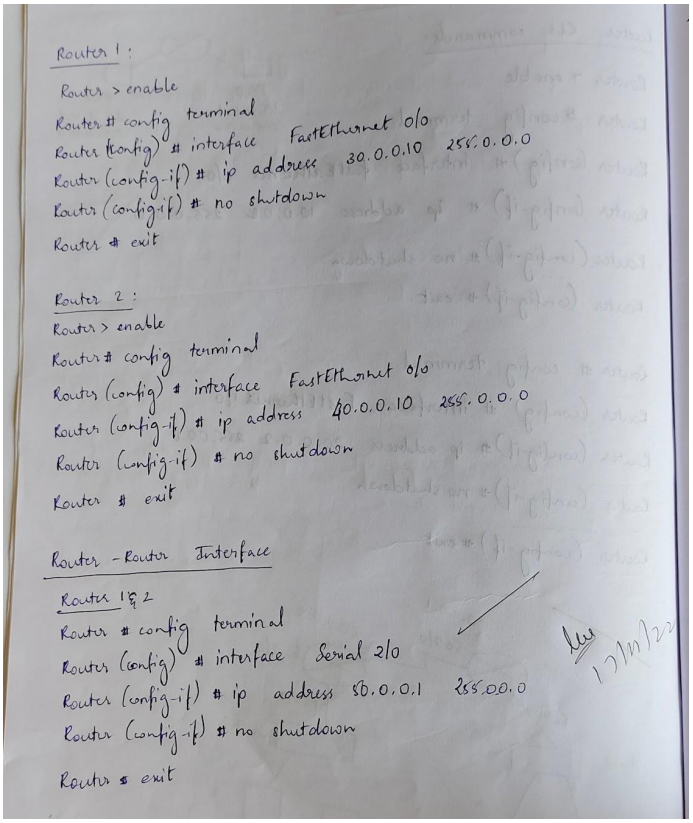
Configuring IP address to Routers in Packet Tracer. Exploring the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.

Topology

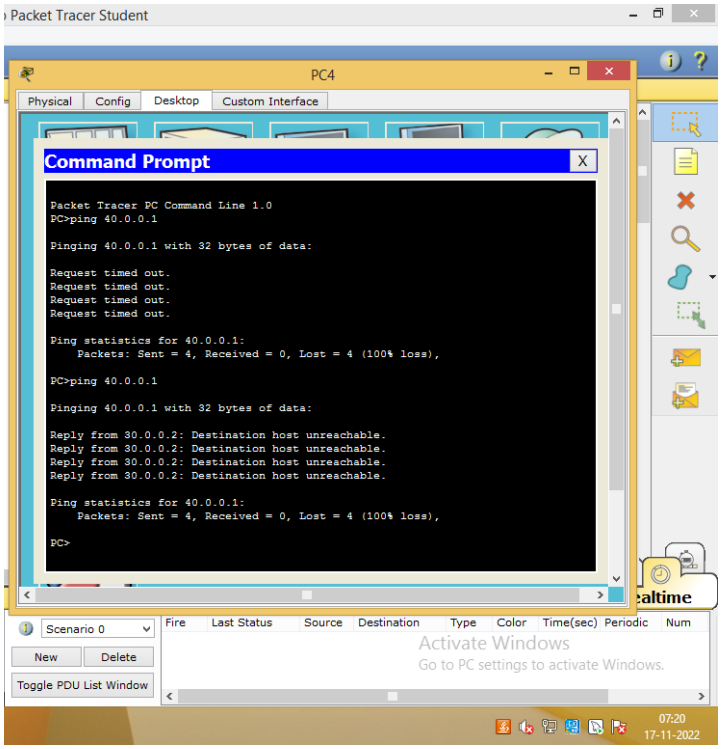


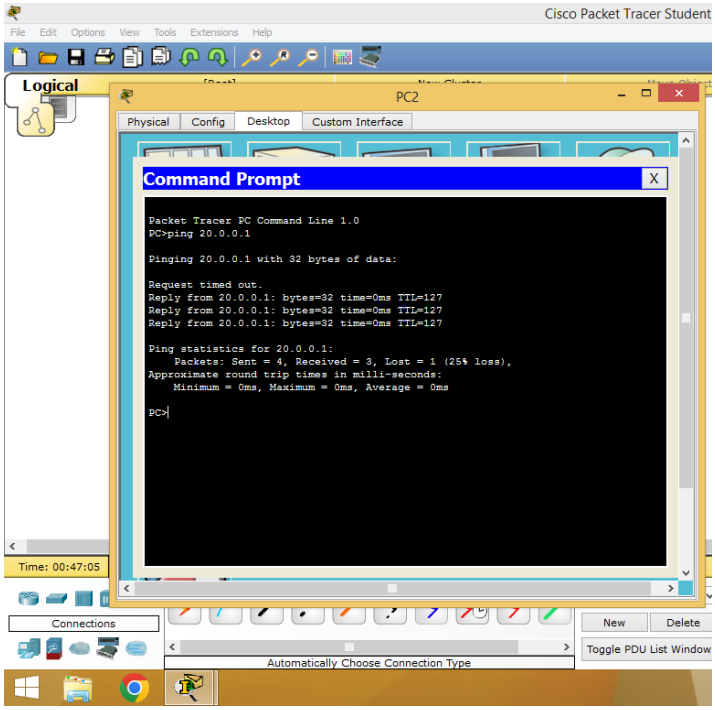
Procedure


LAB 3
1) Router CLI commands:
Router > enable
Router # config terminal
Router (config) # interface FastEthernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # exit
Router # config terminal
Router (config) # interface FastEthernet 1/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # exit
PC 0
10.0.0.1
255.0.0.0
PC 1
20.0.0.1
255.0.0.0
2)
50.0.0.1
Se 2/0
50.0.0.2
Se 2/0
Fa 0/0
Fa 0/0
Hub
Switch
PC 1
30.0.0.1
PC 2
30.0.0.2
PC 3
40.0.0.1
PC 4
40.0.0.2


Router 1:
Router > enable
Router # config terminal
Router (config) # interface FastEthernet 0/0
Router (config-if) # ip address 30.0.0.10 255.0.0.0
Router (config-if) # no shutdown
Router # exit
Router 2:
Router > enable
Router # config terminal
Router (config) # interface FastEthernet 0/0
Router (config-if) # ip address 40.0.0.10 255.0.0.0
Router (config-if) # no shutdown
Router # exit
Router - Router Interface
Router 1 & 2
Router # config terminal
Router (config) # interface Serial 2/0
Router (config-if) # ip address 50.0.0.1 255.0.0.0
Router (config-if) # no shutdown
Router # exit
17/11/22

Output


Packet Tracer Student
PC4
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 30.0.0.2: Destination host unreachable.
Reply from 30.0.0.2: Destination host unreachable.
Reply from 30.0.0.2: Destination host unreachable.
Reply from 30.0.0.2: Destination host unreachable.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Activate Windows
Go to PC settings to activate Windows.
07:20
17-11-2022

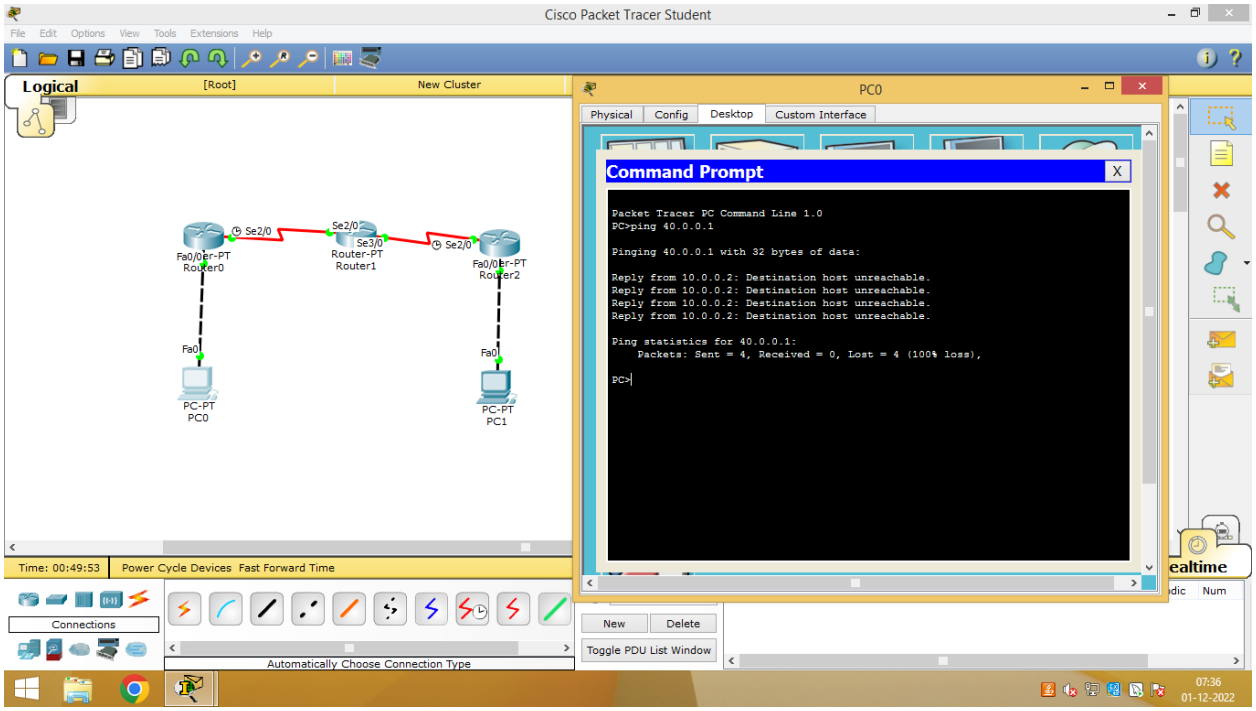

Cisco Packet Tracer Student
File
Edit
Options
View
Tools
Extensions
Help
Logical
PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>
Time: 00:47:05
Connections
Automatically Choose Connection Type
New
Delete
Toggle PDU List Window

Experiment 3

Aim of the experiment

Configuring default route to the Router

Topology

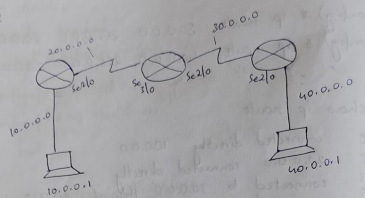


Procedure

11/11/22

LAB-5 : Default Routing

Topology:



Before Default routing:

From 10.0.0.1:

>ping 40.0.0.1

4 times { Reply from 10.0.0.2 : Destinat^n host unreachable

Router 0 - PC0:

Router >enable
Router # config terminal
Router (config) # interface FastEthernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router # exit

Router 2 - PC1:

Router >enable
Router # config terminal
Router (config) # interface FastEthernet 0/0
Router (config-if) # ip address 40.0.0.2 255.0.0.0
Router (config-if) # no shutdown

Router 0 - Router 1

- For Router 0 & Router 1:

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 shutdown
Router # exit

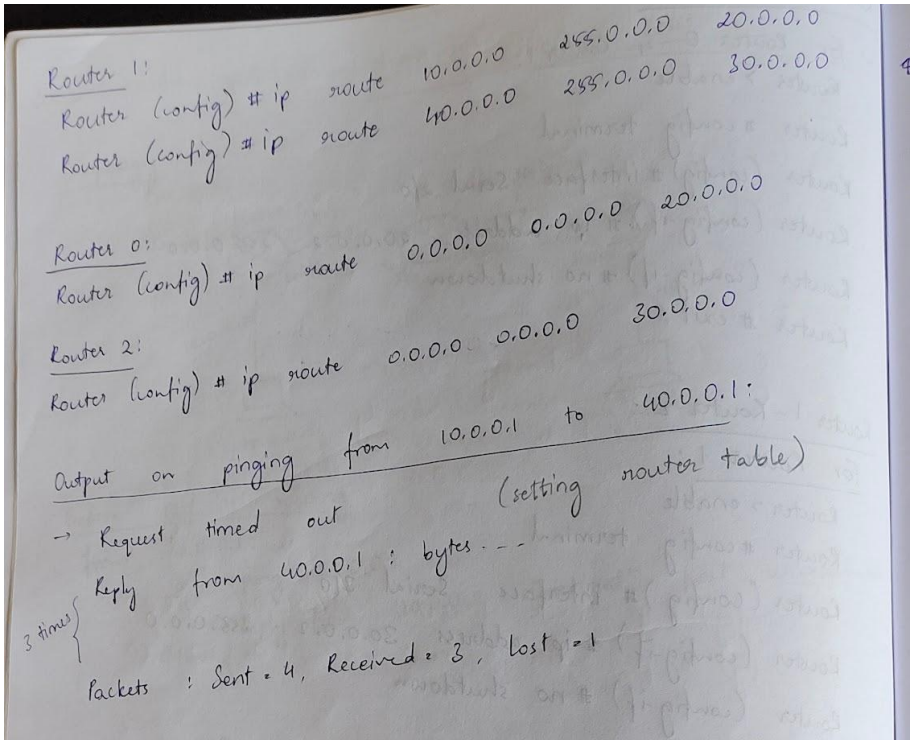
Router 1 - Router 2 :

For Router 1 :

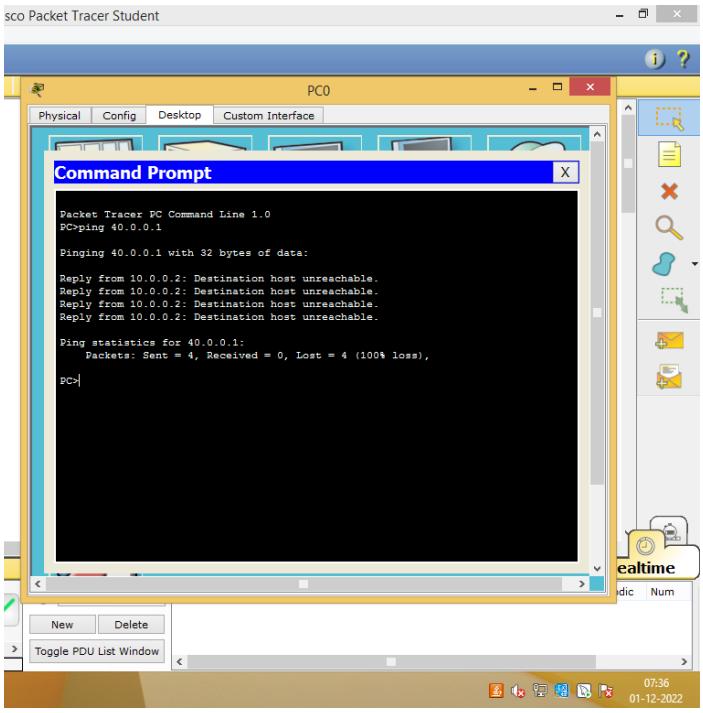
Router >enable
Router # config terminal
Router (config) # interface Serial 3/0
Router (config-if) # ip address 30.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router # exit

For Router 2:

Router # config terminal
Router (config) # interface Serial 2/0
Router (config-if) # ip address 30.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router # exit


Router 1:
Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.0
Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.0
Router 0:
Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.0
Router 2:
Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.0
Output on pinging from 10.0.0.1 to 40.0.0.1:
(setting router table)
→ Request timed out
3 times Reply from 40.0.0.1 : bytes - - -
Packets : Sent = 4, Received = 3, Lost = 1

Output

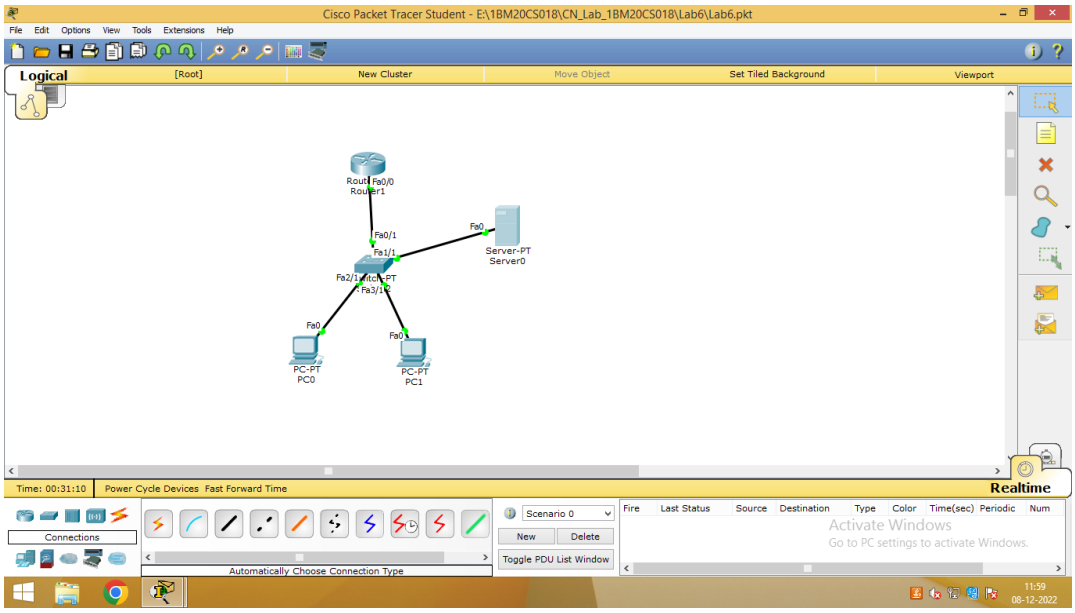

sco Packet Tracer Student
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>
ealtime
Num
New
Delete
Toggle PDU List Window
07:36
01-12-2022

Experiment 4

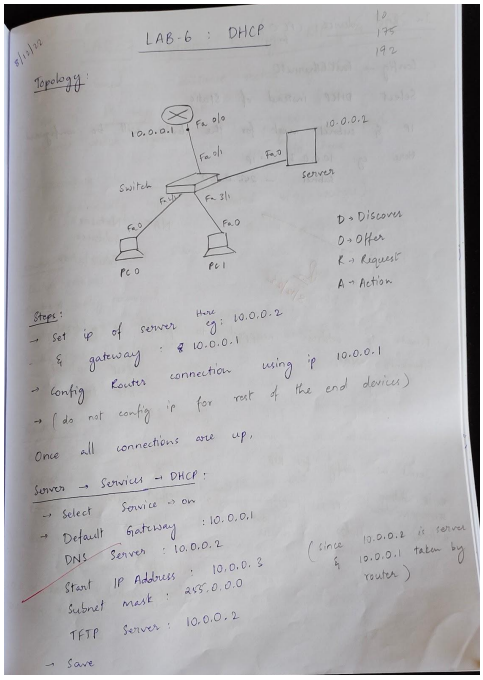
Aim of the experiment

Configuring DHCP within a LAN in a packet Tracer

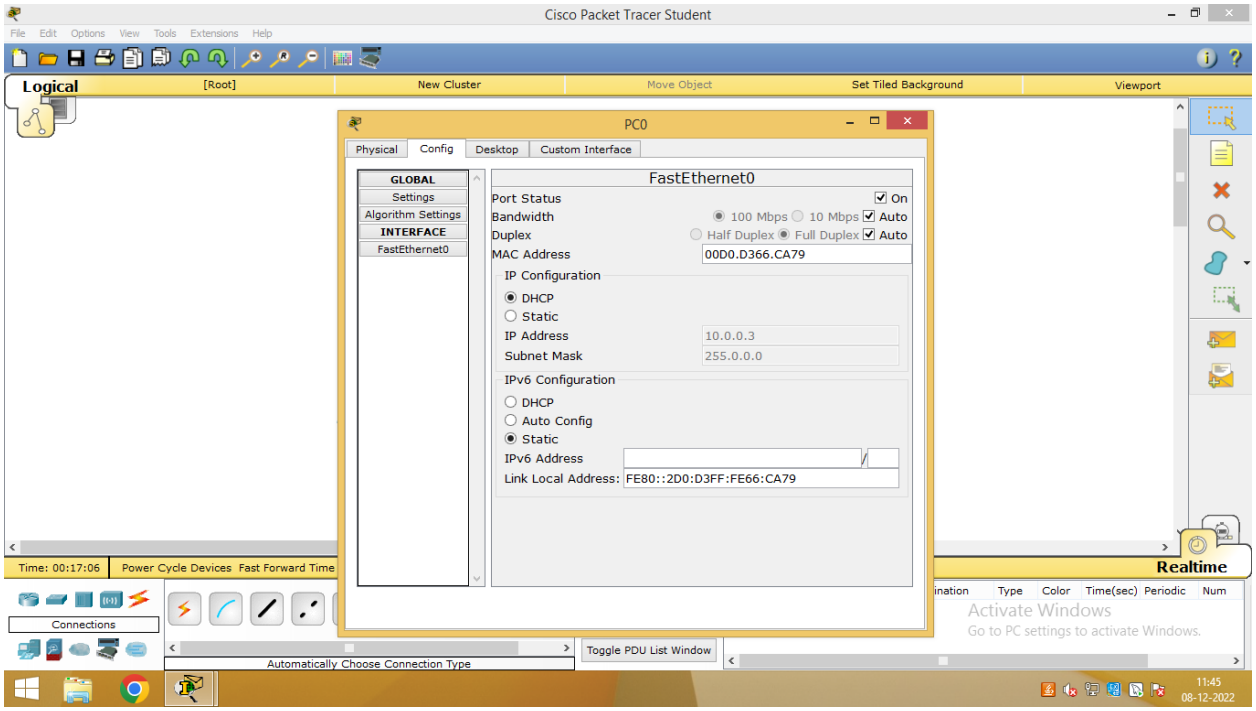
Topology

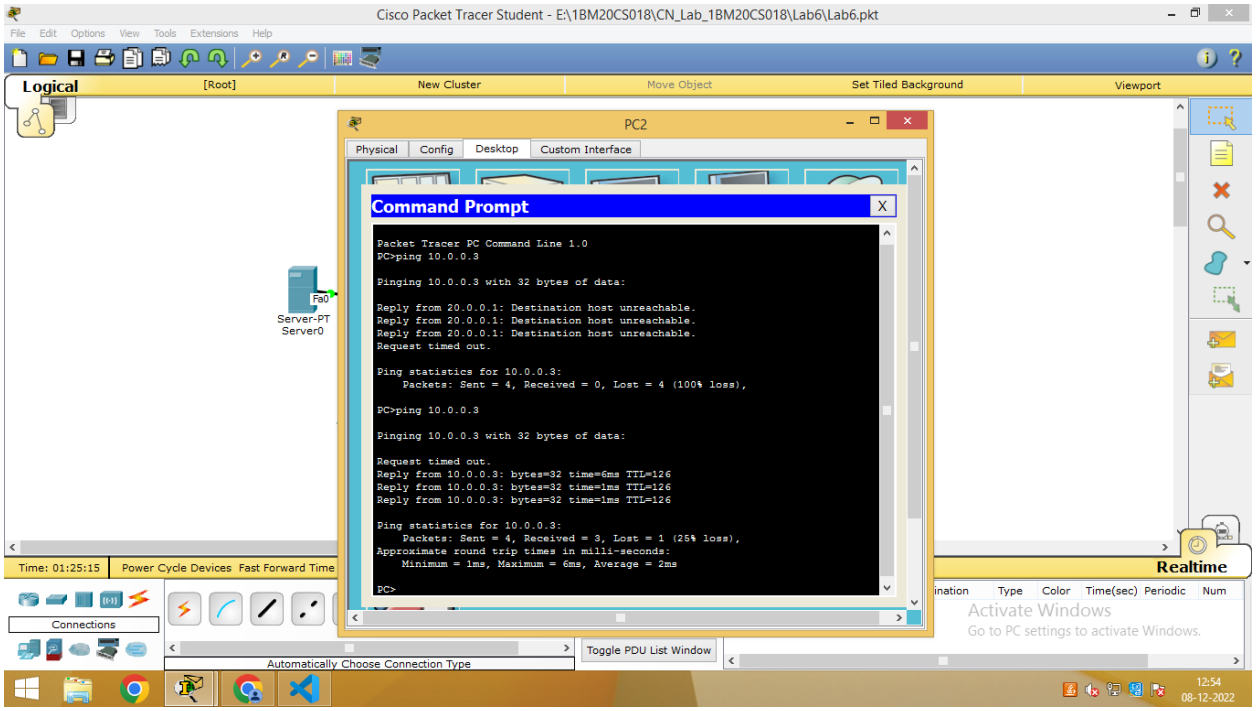


Procedure



Output


Cisco Packet Tracer Student
PC0
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00D0.D366.CA79
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address:
FE80::2D0:D3FF:FE66:CA79
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Time: 00:17:06
Power Cycle Devices
Fast Forward Time
Realtime
Connections
Automatically Choose Connection Type
Toggle PDU List Window
Activate Windows
Go to PC settings to activate Windows.
11:45
08-12-2022

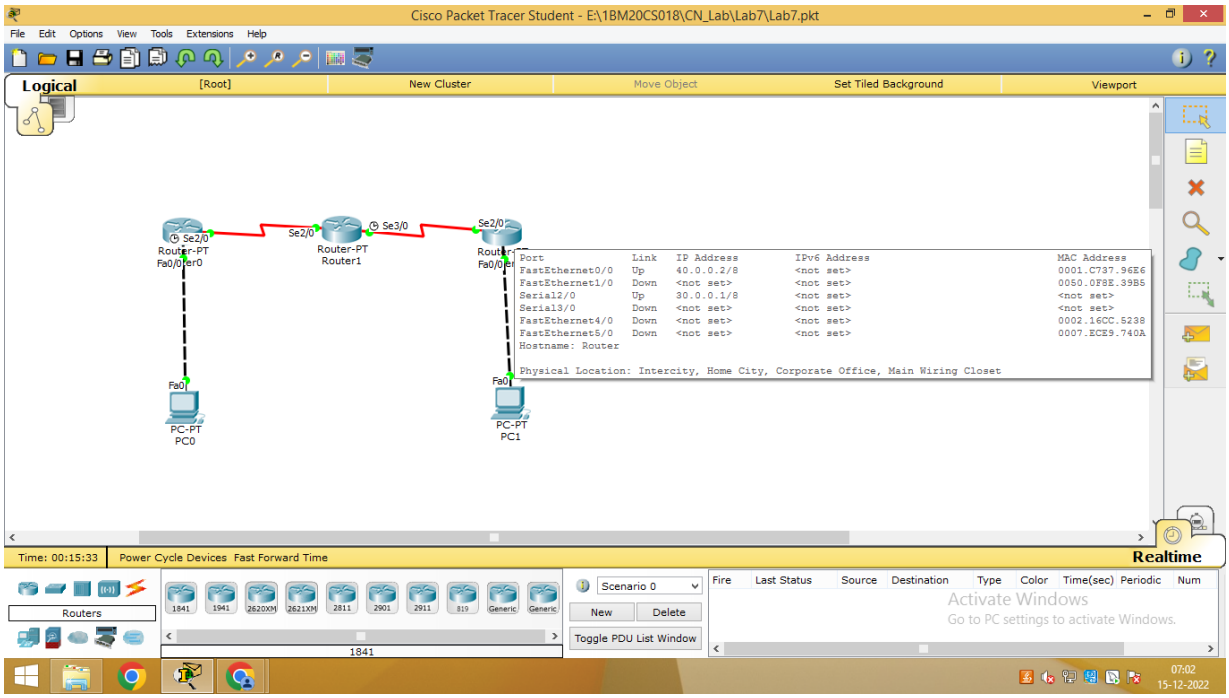

Cisco Packet Tracer Student - E:\1BM20CS018\CN_Lab_1BM20CS018\Lab6\Lab6.pkt
PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 20.0.0.1: Destination host unreachable.
Reply from 20.0.0.1: Destination host unreachable.
Reply from 20.0.0.1: Destination host unreachable.
Request timed out.
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 10.0.0.3: bytes=32 time=6ms TTL=126
Reply from 10.0.0.3: bytes=32 time=1ms TTL=126
Reply from 10.0.0.3: bytes=32 time=1ms TTL=126
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 6ms, Average = 2ms
PC>
Server-PT
Server0
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Time: 01:25:15
Power Cycle Devices
Fast Forward Time
Realtime
Connections
Automatically Choose Connection Type
Toggle PDU List Window
Activate Windows
Go to PC settings to activate Windows.
12:54
08-12-2022

Experiment 5

Aim of the experiment

Configuring RIP Routing Protocol in Routers

Topology



Procedure

15/10/22

LAB-7 Dynamic Routing - RIP & EIGRP

- RIP is based on distance vector algorithm
- Max of 15 hops → RIP
- Distance vector algorithm ⇒ routing tables refresh time-to-time
- (Networking as rumour
  ⇒ (periodically info is updated b/w routers)

Router 0 - end device config

```
Router > enable
Router # config terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
                   # no shutdown
```

Router 2 - end device config

```
Router > enable
Router # config terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 40.0.0.2 255.0.0.0
                   # no shutdown
```

Router 0 : (src)

```
Router # config terminal
Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
                   # encapsulation PPP
                   # clock rate 64000
                   # no shutdown
```

Router 1 : (dest

```
(config) # interface serial 2/0
(config-if) # ip address 20.0.0.2 255.0.0.0
            # encapsulation PPP
            # no shutdown

(as src)
(config) # interface serial 3/0
         # ip address 30.0.0.2 255.0.0.0
         # encapsulation PPP
         # clock rate 64000
         # no shutdown
```

Router 2 :

```
(config) # interface serial 2/0
         # ip address 30.0.0.1 255.0.0.0
         # encapsulation PPP
         # no shutdown
```

Router 0 :

```
(config) # router rip
(config-router)# network 10.0.0.0
               # network 20.0.0.0
               # exit
```

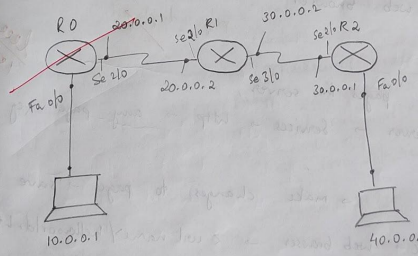
Router 1 :

```
(config) # router rip
         # network 20.0.0.0
         # network 30.0.0.0
         # exit
```

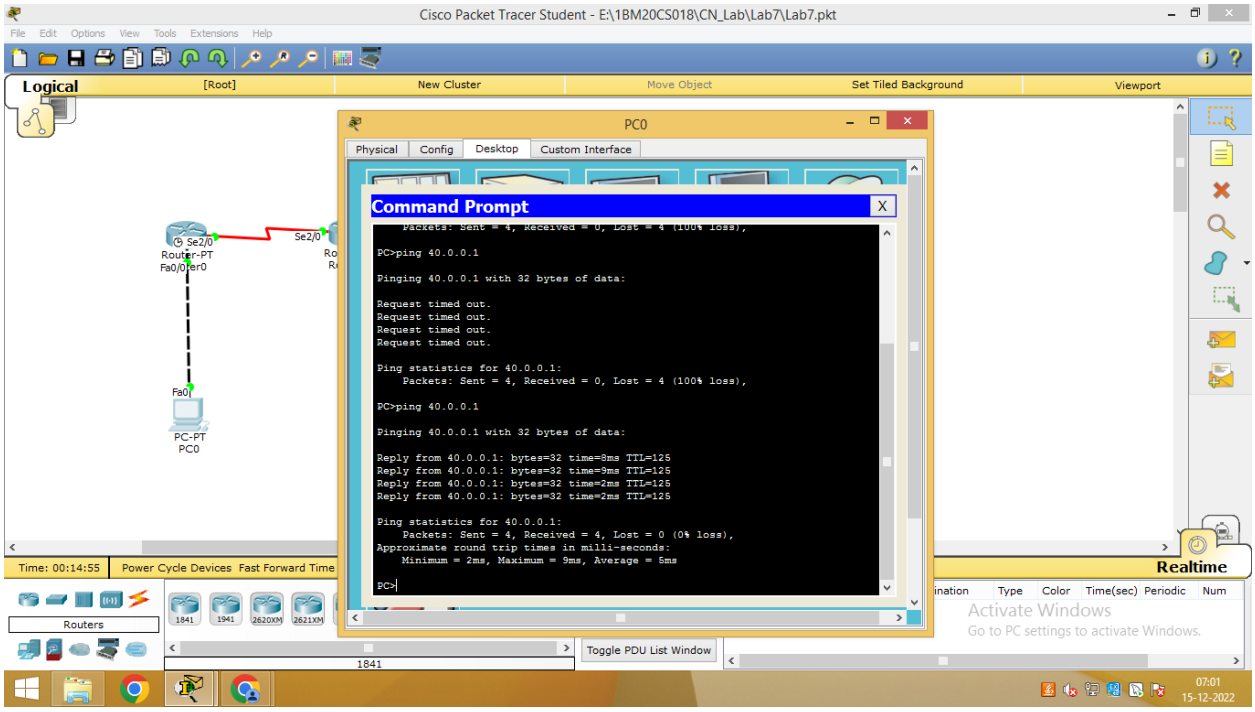
Router 2 :

```
(config) # router rip
         # network 30.0.0.0
         # network 40.0.0.0
         # exit
```

Topology:



Output

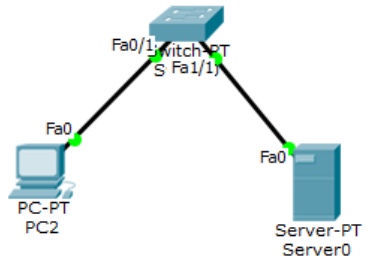

Cisco Packet Tracer Student - E:\1BM20CS018\CN_Lab\Lab7\Lab7.pkt
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
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=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 9ms, Average = 5ms
PC>
Router-PT
PC-PT PC0
Time: 00:14:55
Power Cycle Devices
Fast Forward Time
Realtime
Routers
Toggle PDU List Window
Activate Windows
Go to PC settings to activate Windows.
07:01
15-12-2022

Experiment 6

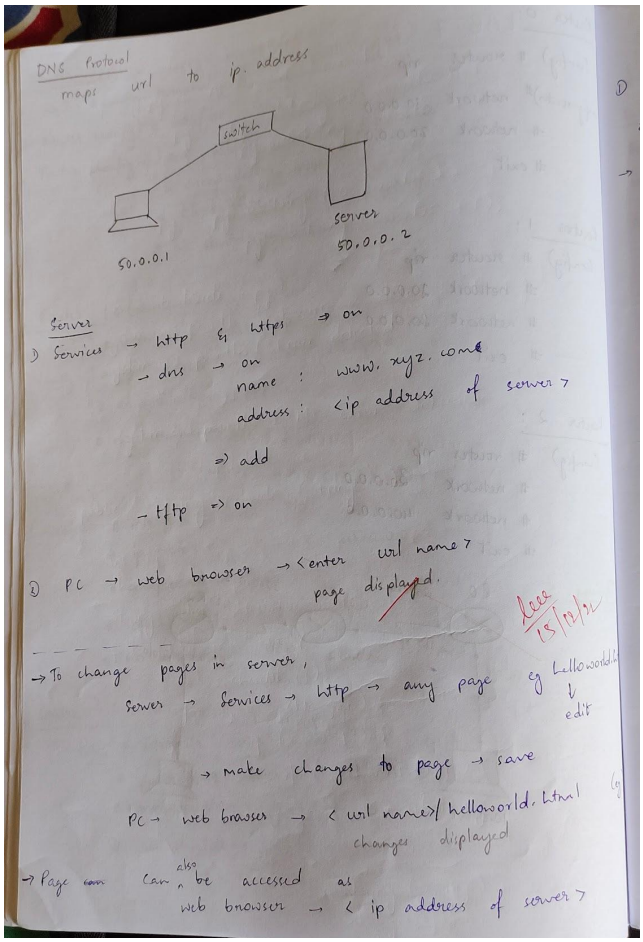
Aim of the experiment

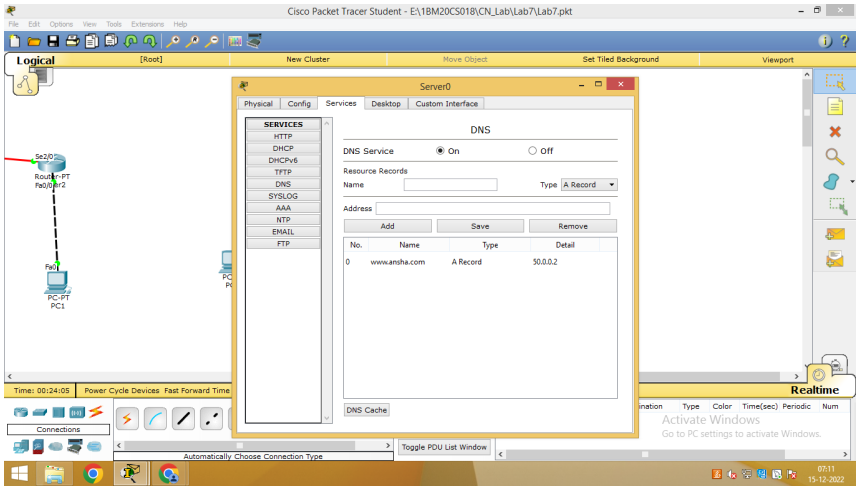
Demonstration of WEB server and DNS using Packet Tracer

Topology

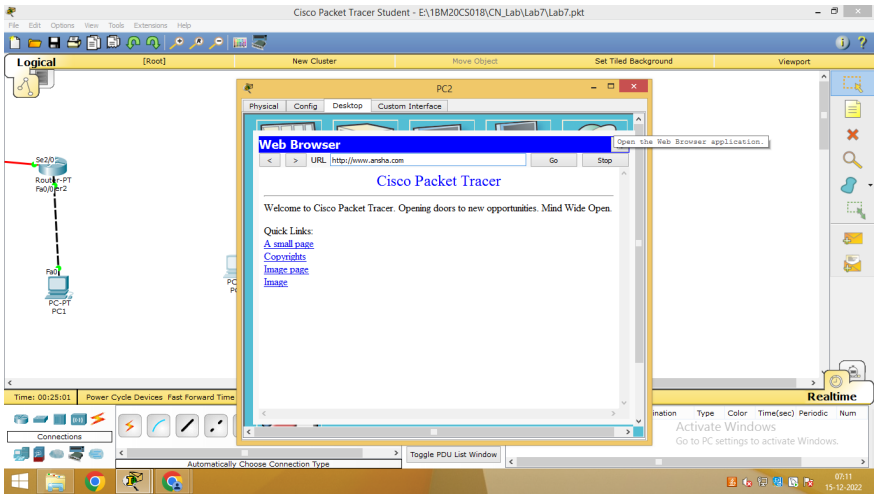


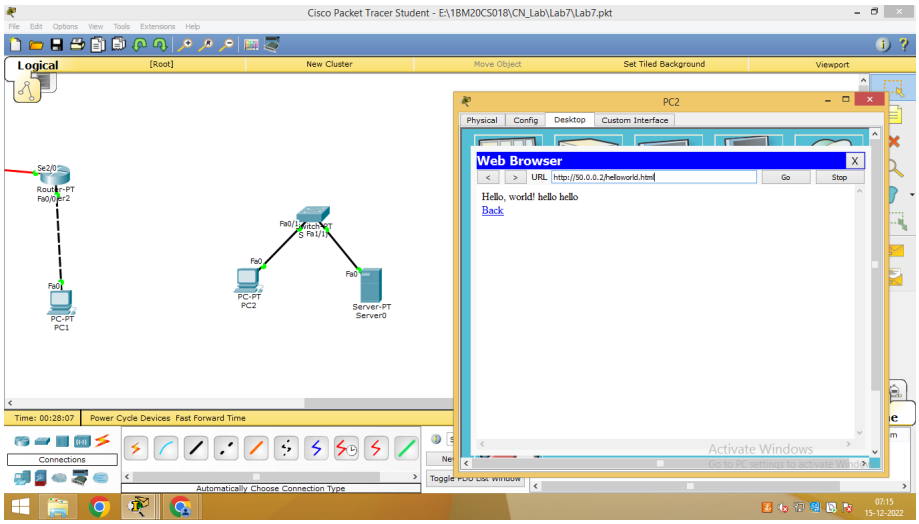
Procedure




Cisco Packet Tracer Student - E:\1BM20CS018\CN_Lab\Lab7\Lab7.pkt
Logical
Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DNS
DNS Service
On
Off
Resource Records
Name
Type
A Record
Address
Add
Save
Remove
No.
Name
Type
Detail
0
www.ansha.com
A Record
50.0.0.2
DNS Cache
Realtime
Activate Windows

Output


Cisco Packet Tracer Student - E:\1BM20CS018\CN_Lab\Lab7\Lab7.pkt
Logical
PC2
Physical
Config
Desktop
Custom Interface
Web Browser
URL
http://www.ansha.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
Realtime
Activate Windows


Cisco Packet Tracer Student - E:\1BM20CS018\CN_Lab\Lab7\Lab7.pkt
Logical
PC2
Physical
Config
Desktop
Custom Interface
Web Browser
URL
http://50.0.0.2/helloworld.html
Go
Stop
Hello, world! hello hello
Back
Server-PT
Server0
PC-PT
PC2
PC-PT
PC1
Activate Windows

Cycle-2 Experiment 1

Aim of the Experiment

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

Code

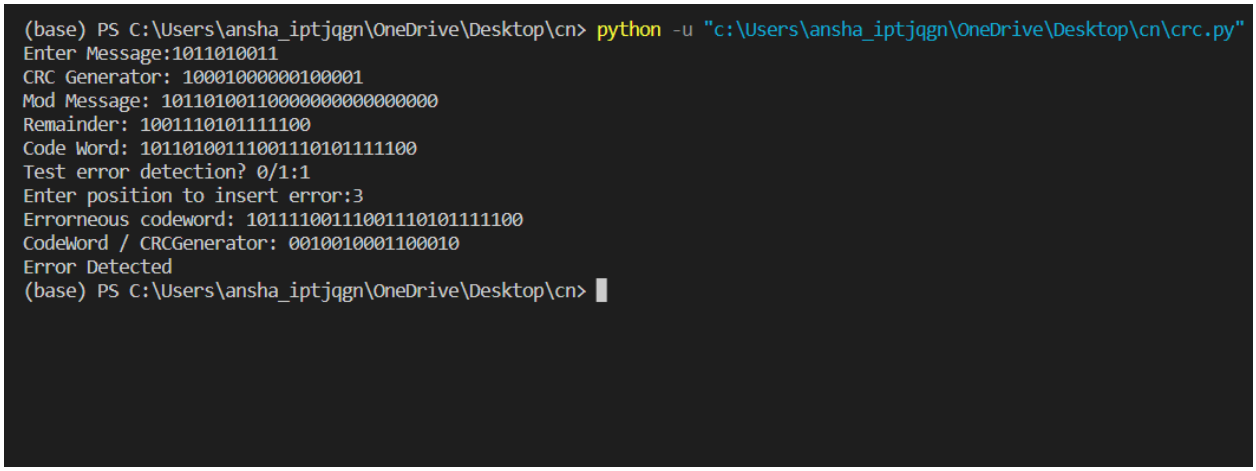
```
def xor(a, b):
    result = []
    for i in range(1, len(b)):
        if a[i] == b[i]:
            result.append('0')
        else:
            result.append('1')
    return ''.join(result)



def binaryDiv(genlen, msg, gen):
    pick = genlen
    tmp = msg[0:pick]
    while pick < len(msg):
        if tmp[0] == "1":
            tmp = xor(gen, tmp) + msg[pick]
```

```
        else:
            tmp = xor('0'*pick, tmp) + msg[pick]
        pick += 1
    if tmp[0] == '1':
        tmp = xor(gen, tmp)
    else:
        tmp = xor('0'*pick, tmp)
    return tmp


message = input("Enter Message:")
crcGenerator = "10001000000100001"
print("CRC Generator:", crcGenerator)
# get length of generator n
crcGenLength = len(crcGenerator)
# add trailing n-1 zeroes to the message
modMessage = str(int(message) * (10**(crcGenLength-1)))
print("Mod Message:", modMessage)
# rem = int(modMessage) / int(crcGenerator)
rem = binaryDiv(crcGenLength, modMessage, crcGenerator)
print("Remainder:", rem)
# generate codeword using remainder
```

```
codeword = str(int(modMessage) + int(rem))
print("Code Word:", codeword)

ch = int(input("Test error detection? 0/1:"))
if ch == 1:
    pos = int(input("Enter position to insert error:"))
    codeword = list(codeword)
    if codeword[pos+1] == '1':
        codeword[pos+1] = '0'
    else:
        codeword[pos+1] = '1'
    codeword = ''.join(codeword)
    print("Errorneous codeword:", codeword)

    # test = codeword / crcgenerator
    test = binaryDiv(crcGenLength, codeword, crcGenerator)
    print("CodeWord / CRCGenerator:", test)

    # if test = 0 => no error
    if int(test) == 0:
        print("No Error")
```

```
    else:
        print("Error Detected")
else:
    print("Skipping error insertion")
```

Output



Cycle 2: Experiment 2

Aim of the Experiment

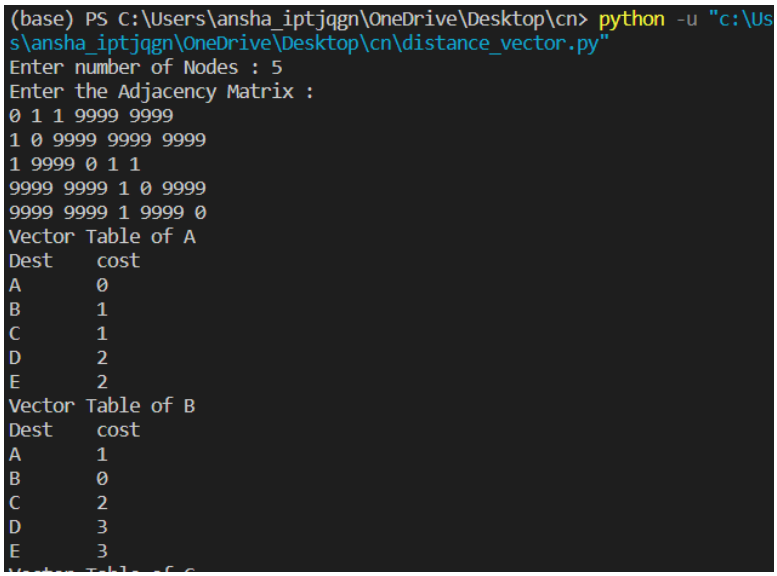
Write a program for distance vector algorithm to find suitable path for transmission.

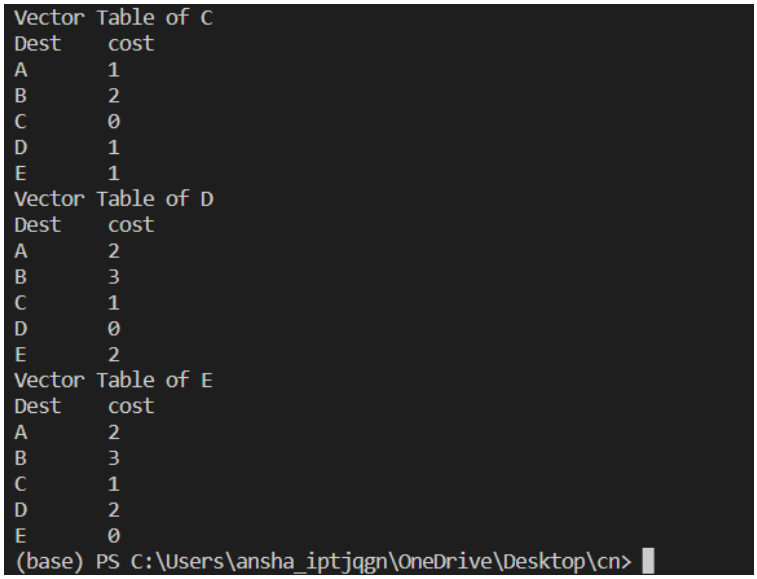
Code

```
class Topology:
    def __init__(self, n):
        self.matrix = []
        self.n = n

    def addlink(self, u, v, w):
        self.matrix.append((u, v, w))

    def table(self, dist, src):
        print("Vector Table of {}".format(chr(ord('A')+src)))
        print("Dest\tcost")

        for i in range(self.n):
            print("{0}\t{1}".format(chr(ord('A')+i), dist[i]))

    def bellmanford(self, src):
        dist = [9999] * self.n
        dist[src] = 0

        for _ in range(self.n - 1):
            for u, v, w in self.matrix:
                if dist[u] != 9999 and dist[u] + w < dist[v]:
                    dist[v] = dist[u] + w

        self.table(dist, src)

def main():
    matrix = []
    n = int(input("Enter number of Nodes : "))
    print("Enter the Adjacency Matrix :")
    for i in range(n):
        m = list(map(int, input().strip().split()))
        matrix.append(m)
    topo = Topology(n)
```

```
    for i in range(n):
        for j in range(n):
            if matrix[i][j] == 1:
                topo.addlink(i, j, 1)

    for a in range(n):
        topo.bellmanford(a)

main()
```

Output





Cycle-2 Experiment 3

Aim of the Experiment

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

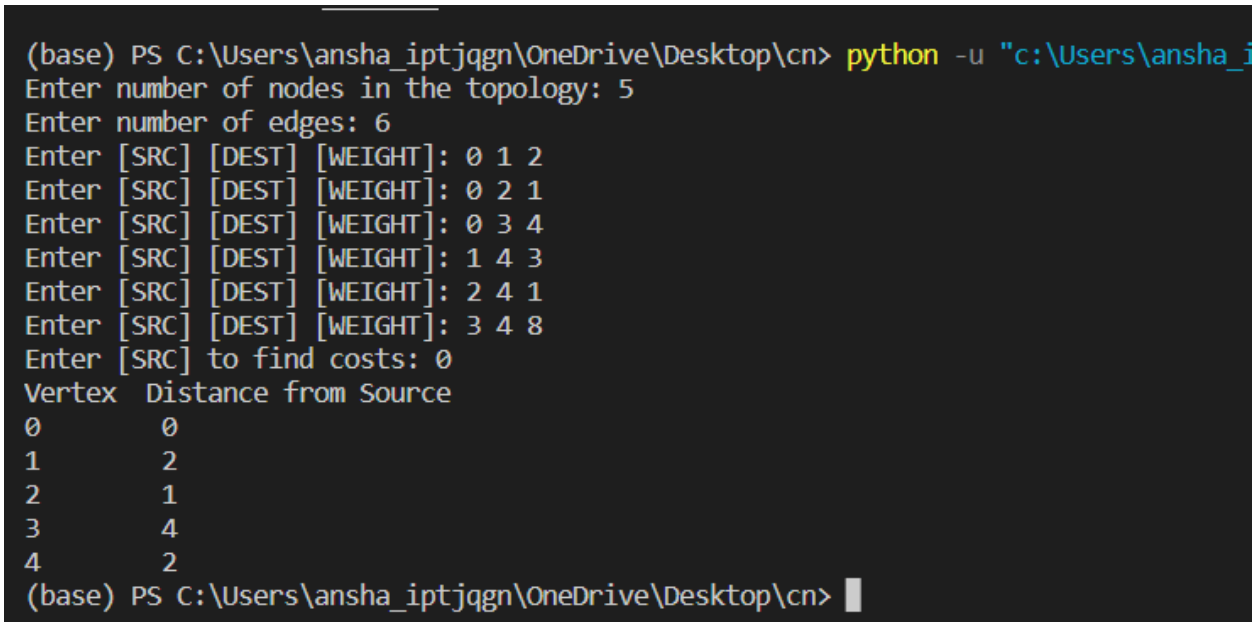
topology.

Code

```
class Graph():
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for column in range(vertices)]
                      for row in range(vertices)]

    def print_solution(self, dist):
        print("Vertex \tDistance from Source")
        for node in range(self.V):
            print(node, "\t", dist[node])


    def min_distance(self, dist, sptSet):
        min = 9999
        for v in range(self.V):
            if dist[v] < min and sptSet[v] == False:
                min = dist[v]
                min_index = v

        return min_index

    def add_edge(self, src, dest, weight):
        self.graph[src][dest] = self.graph[dest][src] = weight

    def dijkstra(self, src):

        dist = [9999] * self.V
        dist[src] = 0
        sptSet = [False] * self.V

        for cout in range(self.V):
            u = self.min_distance(dist, sptSet)
            sptSet[u] = True
            for v in range(self.V):
                if self.graph[u][v] > 0 and sptSet[v] == False and dist[v] > dist[u] + self.graph[u][v]:
                        dist[v] = dist[u] + self.graph[u][v]
```

```
        self.print_solution(dist)
g = Graph(int(input("Enter number of nodes in the topology: ")))
e = int(input("Enter number of edges: "))

for i in range(e):
    src, dest, cost = [int(_) for _ in input("Enter [SRC] [DEST] [WEIGHT]: ").split(' ')]
    g.add_edge(src, dest, cost)

src = int(input("Enter [SRC] to find costs: "))
g.dijkstra(src)
```

Output



Cycle-2 Experiment 4

Aim of the Experiment

Write a program for congestion control using leaky bucket algorithm.

Code

```
class LeakyBucket:
    def __init__(self, bucket_size, output_rate, packets):
        self.bucket_size = bucket_size
        self.output_rate = output_rate
        self.packets = packets

    def traffic_shaping(self):
        for i in range(len(self.packets)):
            packet_size = self.packets[i]
            print(f"Packet No: {i} Packet Size: {packet_size}")
            if packet_size > self.bucket_size:
                print("Bucket Overflow")
            else:
                while packet_size > output_rate:
                    print(f"{output_rate} bytes sent")
                    packet_size -= output_rate
                if packet_size:
                    print(f"Last {packet_size} bytes sent")

                print("Bucket output Successful")


bucket_size = int(input("Enter the Bucket Size: "))
output_rate = int(input("Enter the output rate: "))
packets = [int(x) for x in input("Enter the input packets: ").split()]
lb = LeakyBucket(bucket_size, output_rate, packets)
lb.traffic_shaping()
```

Output

```
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn> python -u "c:\Users\ansha_iptjqgn\OneDrive\Deskto
Enter the Bucket Size: 12
Enter the output rate: 2
Enter the input packets: 10
Packet No: 0 Packet Size: 10
2 bytes sent
2 bytes sent
2 bytes sent
2 bytes sent
Last 2 bytes sent
Bucket output Successful
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn>
```

Cycle-2 Experiment 5

Aim of the Experiment

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

Code

TCP Server:

```
from socket import *

serverName = "127.0.0.1"
serverPort = 12000
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()

    file = open(sentence, "r")
    l = file.read(1024)
    print("Recieved from client: ", l)

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

TCP Client:

```
from socket import *

serverName = "127.0.0.1"
serverPort = 12000
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

```
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn> cd sso
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn\sso> python -u "tcpserver.py"
The server is ready to receive
```

```
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn\sso> python -u "tcpclient.py"
Enter file name: a.txt
From Server:  inside a.txt file
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn\sso>
```

```
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn> cd sso
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn\sso> python -u "tcpserver.py"
The server is ready to receive
Recieved from client:  inside a.txt file
```

Cycle-2 Experiment 6

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

UDP 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()
```

UDP 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.decode())
clientSocket.close()
```

Output

```
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn\sso> python -u "udp_server.py"
The server is ready to receive
Sent back to client:  inside a.txt file
```

```
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn\sso> python -u "udp_client.py"
Enter file name: a.txt
From Server:  inside a.txt file
(base) PS C:\Users\ansha_iptjqgn\OneDrive\Desktop\cn\sso>
```