

# LAB-1

SURYA Gold

Date \_\_\_\_\_

Page \_\_\_\_\_

nslookup

used to get IP address

## Outcome Lab 2

We learnt about IP address, mac address and terminologies like Ethernet, Porters, ARP Table and many more.

In this tutorial we explored Wireshark packet tracer software it helped us to understand the concept of computer network in a better way. Through simulations, we have also established connection between client and server, and also added the IP address and DNS server for making necessary communication.

Abhishek  
Malik

## LAB-2

Aim: Implementation of hub and switches

Procedure: Open Cisco packet tracer, connect and show with hubs and switches. Then configure IP address for each IP device. Pass simple PDU across the end devices and note down result using simulation.

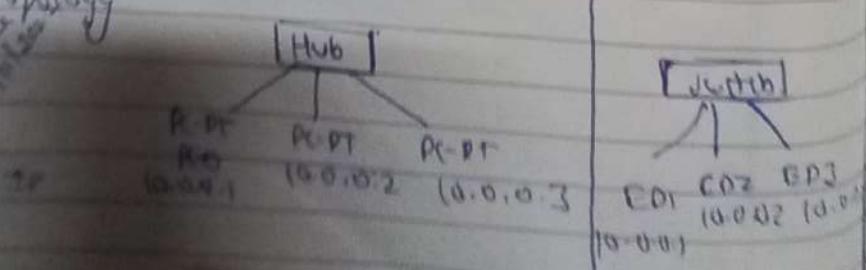
Topology: star topology

Result: Successful transfer of simple PDU from source to destination end devices.

Outcome / observation:

Hubs are devices which don't have storage and mapping (processing) features hence they just broadcast (transmit) to all the end devices of the same network. While switches performs as hub during initial state, but later transfer from source to destination and mapping feature using ARP table. transmission is said to be successful if PDU file of receiver host isn't connect to internet work, message loss by ping and hence represent response will be timed out.

→ Topology



## LAB - 3

Aim: (an)alyzing IP address to routers in path taken explore ping & response destination unreachable, reply, request, timer etc.

- Procedure :- Open Cisco packet tracer
- Connect two pc's to routers using copper (RJ45) over wire
  - Then configure a pc. and also configure router using CLI command's like enable, interface Fa0/0, ip address <ip-address>
  - If we ping from 1 pc to another pc then it will show timeout.
  - and ping to another device

Topology: Star topology

Result: Pinging 20.0.0.2 with 32 bytes of data:

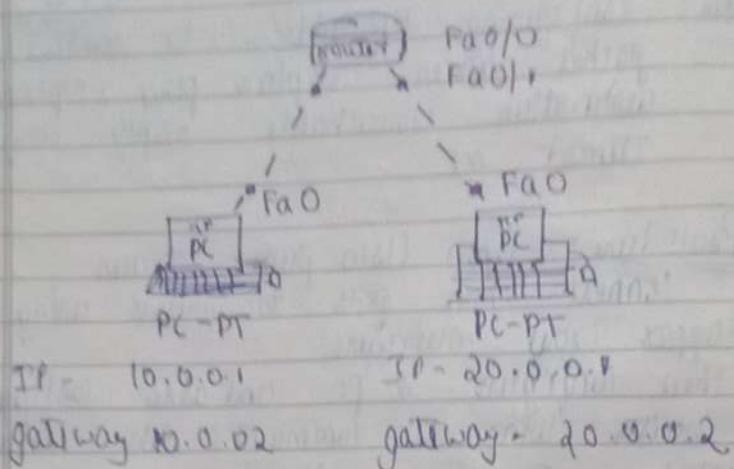
Reply from 20.0.0.2 : bytes=32 time=0ms TTL=255

Ping statistics for 20.0.0.2:

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

Approx round trip time in millisecond  
Minimum = 0ms, Maximum = 0ms

Question: To policy:



Answer:

(i) Succesfully: when we configure both the routers and PC (end devices) with appropriate ip addresses and by configuring subnet mask of interface of router as 255.0.0.0 and gateway of PC in xt as 10.0.0.2 and gateway of PC in xt as 20.0.0.2 which is of Fa0/0 interface followed by we successfully ping devices.

TimeOut: if IP address of end devices and/or gateway is not configured properly then packets are in Timeout.

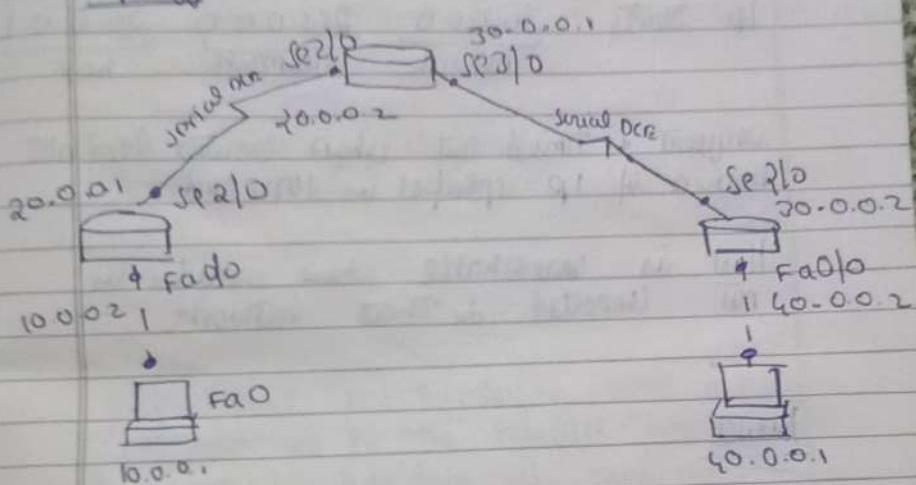
PC > ping 20.0.0.2

Request: direct and  
Request: failed and  
Request: timed out

Ping Statistic:  
Sent = 4 Received = 0

## LND-4

Aim: Configuring default route to router

TopologyProcedure:

Two routers are connected to and devices and routers are used to connect other two routers using serial DCE connection. Configure IP address for and devices and for interface of routers.

Set IP of PC2 to 40.0.0.1 and gateway to 40.0.0.2 (Configure routers of in CLI window using command)

- snakes
- config terminal
- interface fa0/0 on JE 2/0
- ip address
- no shutdown
- exit

such that all matches topology drawn later  
 Show ip route  
 for R1, 10.0.0.0 & 20.0.0.0

IP route 30.0.0.0 255.0.0.9 20.0.0.1  
 network      subnetmask      hop

Request is timed out when routes do not  
 known if IP specified is connected

Host is unreachable when routes by  
 not connected to that network

### Request

PC > ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data

Request timed out at host no route found

Request timed out

Request timed out

Ping statistics for 20.0.0.1

packets sent = 4, received = 0, lost = 4 (100% loss)

PC > ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data  
 destination host not reachable error

PC > ping 40.0.0.1

Ping to 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=8ms TTL=125

Ping statistics for 40.0.0.1

Packet: sent=4, received=4, Lost=0 (0%)

Observation

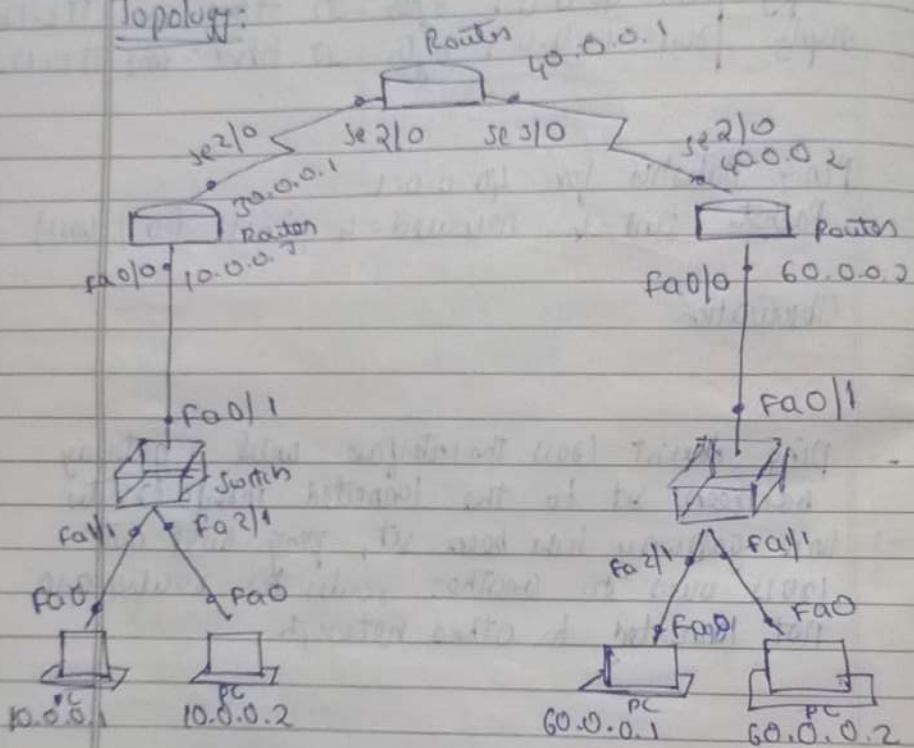
Notation  
Sat 1/20/2022

- > Ping doesn't (goes) to the interface until a gateway has been set to the connected interface/router
- > Once gateway has been set, ping will not (goes) over to another router as routers are not connected to other network.

## LAB-5

Aim: Configuring default route to router

Topology:



Procedure:

- ~~Two~~ <sup>3</sup> routers are connected to a switch which are then connected to routers
- Configure and delete and interfaces
- Configure IP address between routers on Virtual interface
- To configure IP address following commands are given

enable:

config terminal  
interface se 2/0  
ip address 10.0.0.2 255.0.0.0  
no shutdown

ip route 0.0.0.0 0.0.0.0 10.0.0.2

- Configure gateway address of and device with connected interface of router.
- A ping Inorder to establish default routes across routers, ip route is configured using command

ip route 0.0.0.0 0.0.0.0 10.0.0.2

Observation:

- ping doesn't cross interface until gateway has been sent to connected interface routers.
- once gateway has been set, ping will not cross over to another routers as the routers are not connected to other network and they won't know which routers network and they to take are where next hop of to take one where the next hop of to take signal is to be done
- routers are configured with ip-route. When network default route is configured between routers and subnet mask is not specified only via interface of connected router
- both ping connection between all the routers and and devices

## Result

PC  $\rightarrow$  ping 60.0.0.1

Pinging 60.0.0.1 with 32 bytes of data

Reply from 60.0.0.1 : bytes = 32 time = 3ms TTL =

Reply from 60.0.0.1 : bytes = 32 time 10ms TTL =

Ping statistics for 60.0.0.1:

Packets: sent = 4, received = 4, lost = 0

ping 60.0.0.1

Pinging timed out

Request timed out

Reply from 60.0.0.1 bytes = 32 time = 5ms TTL =

Request timed out

Ping statistics for 60.0.0.1

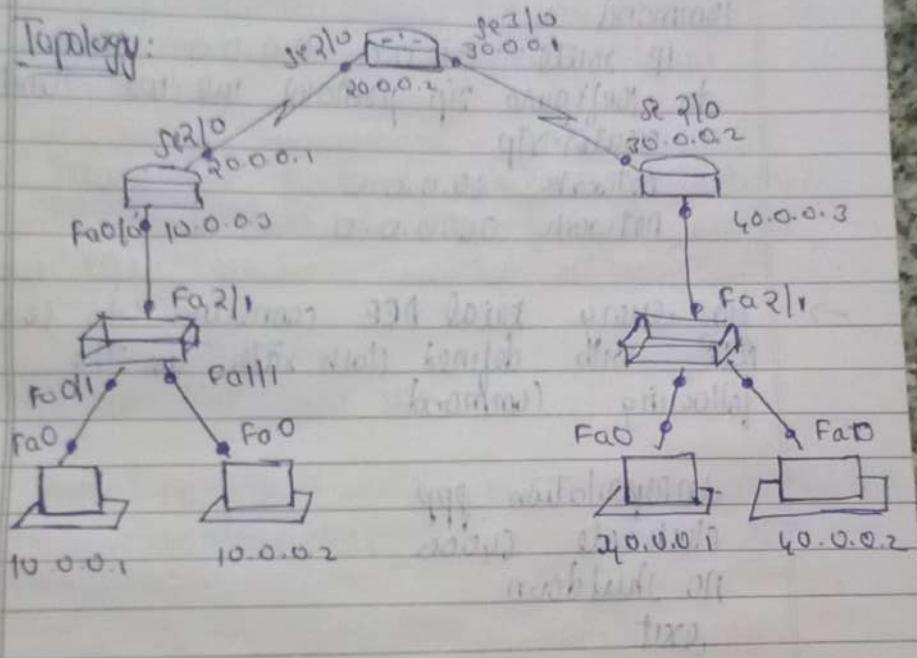
Packets: sent = 4, received = 2, lost = 2

*Request  
timed out*

## LAB-6

Aim : Unconfiguring RIP routing protocol in routers

Topology:



Procedure:

- 3 routers are connected to a switch which are then connected to switches and devices.
- Configure and disable and interface of routers
- Configuring IP address between routers on serial (DCE) interface
- To configure IP address following commands are given
  - enable
  - config terminal
  - interface S2/0
  - IP address 10.0.0.2 255.0.0.0
  - no shutdown
  - exit

- configure gateway address of and connect with connected interface of routers
- In order to establish default route on router ip route is configured using command

ip route 0.0.0.0 0.0.0.0 20.0.0.2  
 to configure rip protocol we use command  
 router rip  
 network 20.0.0.0  
 network 20.0.0.0

- for every serial DCE connector to config ppp with defined clock rate, use the following command

encapsulation ppp  
 clock rate 64000  
 no shutdown  
 exit

Observation of network

> ping 40.0.0.1

apply from 40.0.0.1 with 32 byte of dat

Reply from 40.0.0.1 bytes=32 time=2ms TTL=128  
 reply from 40.0.0.1 bytes=32 time=2ms TTL=128  
 reply from 40.0.0.1 bytes=32 time=2ms TTL=128  
 reply from 40.0.0.1 bytes=32 time=2ms TTL=128

Ping statistic for 40.0.0.1

Round-trip : Sent = 4, Received = 4; Lost = 0

Approximate round-trip in milliseconds  
Minimum = 2ms      Maximum = 40ms,  
Average = 10ms

Since RIP protocol has been established  
IP route does not have to be set for  
each router.

Before RIP was set

ping 10.0.0.1  $\rightarrow$  40.0.0.1 ; Destination host  
unreachable

Before RIP:

ping 10.x  $\rightarrow$  20.x  
Request Timed out

Only on correctly configuring gateways and  
protocols, in reply received properly.

Result:

(Routing Information Protocol) RIP is  
established in network correctly.

NOTE:

Even on proper connection and configuration  
The first packet of first inter network  
ping is timed out as switches have  
not learnt network yet

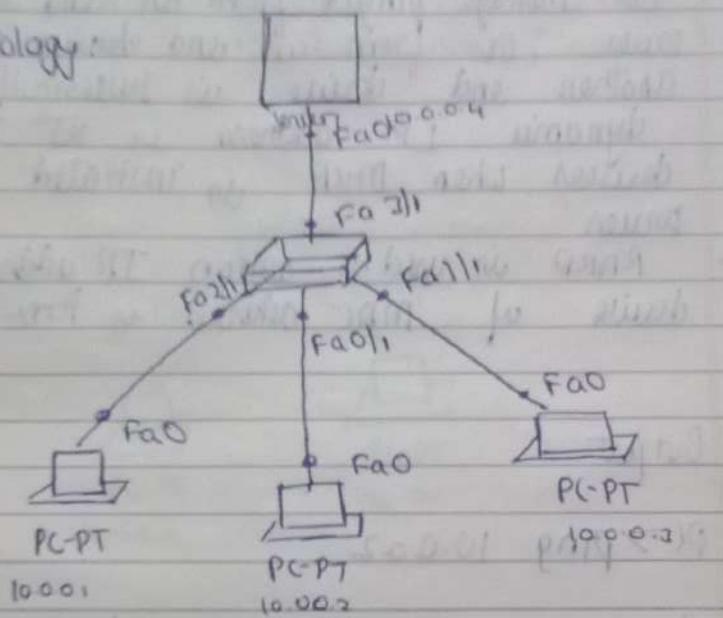
RIP - Routing Information Protocol -  
is dynamic routing protocol that  
use hop count as a metric to find  
best path between source and destination  
network

Meeting  
18/1/2022

## LAB-6

Aim: Configure DHCP service

Topology:



Procedure:

- 1. 3 end devices are connected to a switch which is then connected to a server.
- 2. Configure IP address of server to 10.0.0.4
- 3. Switch on the DHCP service
- 4. Find and save the IP addresses of all the end devices in switches to their appropriate IP and joins them to connect the end devices.
- 5. After saving the address of in server change the gateway/DNS in end devices to DHCP.

- Pinging the IP address of all the end devices

- Observation:
  - The message pinged from an end device to server, or from an end device to another end device, is successfully sent.
  - dynamic IP address is set to end devices when DHCP is initiated in server.
  - RARP will do assign IP address of device if MAC address is known.

Output

PC > ping 10.0.0.2

Pinging 10.0.0.2 with 52 bytes of data:

Reply from 10.0.0.2 byte=32 time=0ms TTL=128  
Reply from 10.0.0.2 byte=32 time=3ms TTL=128  
Reply from 10.0.0.2 byte=32 time=0ms TTL=128

Ping statistics for 10.0.0.1:  
Packets: Sent = 4, Received = 4, Lost = 0

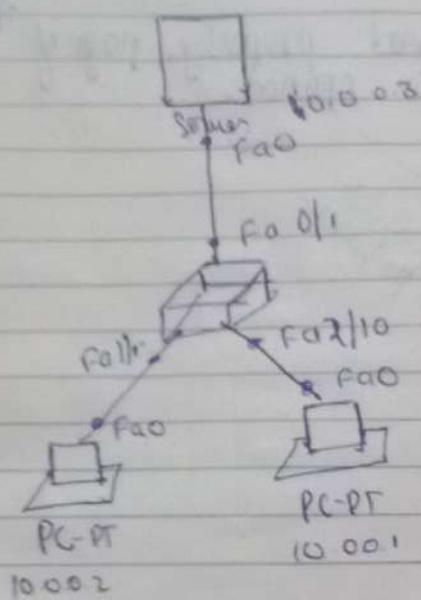
Approximate round trip in multi-hops:  
min = 0ms, max = 5ms, Avg = 3ms

## LAB-7

SURYA Gold

Aim: Configuring webserver and DNS Server

Topology:



Procedure:

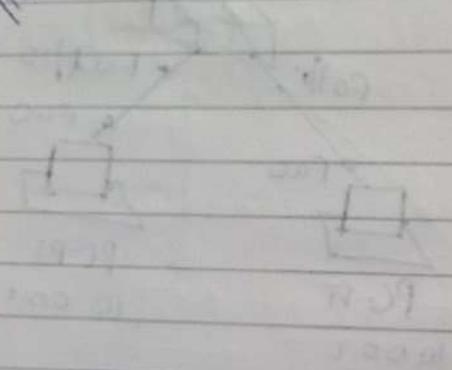
- Construct following topology. Where server is connected to end devices through switch.
- Configure IP address of server and end devices.
- Set HTTP and DNS servers to one state.
- Set server domain name & address of it as same as server, and save following.
- From one of end devices check if web server is accessible from end device by entering:

→ Simulation

Web browser module is opened, in and choose and set domain name ([www.bms.com](http://www.bms.com))

If system / series hasn't been configured properly i.e., if DNS servers and therefore gateway, the Host Unresolved is shown if configured properly, page of who has tracer is opened.

Neelima  
15/12/2022



much much work of configuration  
is required which has a lot of  
time for configuration of system

and system has to be set up  
and lot of time is required  
for configuration of system

Write a program for error detection using CRC (RC4)

```
#include <iostream>
using namespace std;

int main()
{
    int i=0, j=0, n;
    cout << "Enter no of data entry";
    cin >> n;
    int a[n+1];
    int g[] = {1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1};
    int rem[n];
    int m = n+1;

    for (i=0; i<n; i++)
    {
        if (i >= n)
        {
            a[i] = 0;
        }
        else
        {
            cin >> a[i];
        }
    }

    if (i < n)
    {
        rem[i] = a[i];
    }
}
```

```
for(i=0; i<n; i++) {  
    if(a[i] == 1) {  
        continue;  
    }
```

```
}  
for(j=0; j<m; j++) {  
    a[i+j] = a[i+j] ^ g[j];  
}
```

```
}
```

```
for(i=0; i<n; i++) {  
    a[i] = num[i];  
}
```

Output

3  
1  
1  
1

NL  
5/1/2023

111011000001100011  
00100000100000000

## Leaky - Bucket

SURYA Gold

```
#include <climits>
```

```
using namespace std;
```

```
int main() {
```

```
    int cap, out, in, n, bucket = 0;
```

```
    cout << "Enter CAPACITY of BUCKET \n";
```

```
    cin >> cap;
```

```
    cout << "Enter output rate \n";
```

```
    cin >> out;
```

```
    while (true) {
```

```
        cout << "ENTER 1 TO GIVE INPUT \n";
```

```
        cout << "ENTER 2 TO EXIT \n";
```

```
        cin >> n;
```

```
        switch (n) {
```

```
            case 1: cout << "ENTER INPUT RATE \n";
```

```
            cin >> in;
```

```
            if (in > cap)
```

```
                cout << "BUCKET OVERFLOW \n";
```

```
}
```

else?

bucket + in;

```
        if (bucket + in > cap)
```

```
            cout << "BUCKET OVERFLOW \n";
```

```
}
```

Ques.

butcher = 10  
butcher - basket - cat

if (basket == 15)  
    butcher = 2;

cout << "Capacity of basket " << basket  
    ?  
        break;

Ques 2 - cout << "Thank you ) 0";  
    exit;

Output 0:

3

Output:

3

Final capacity of basket  
300

Total output value

300

Ques.

Ques.

Ques.

## Distance-Vector (Bellman Ford)

SURYA Gold

```
#include <iostream>
using namespace std;
#define MAX 10
int n;
```

```
class Router {
    char adj_new[MAX];
    adj_old[MAX];
```

```
int table_new[MAX], table_old[MAX];
```

```
public:
```

```
{
```

```
    Router() {
```

```
        for (int i = 0; i < MAX; i++) {
            table_old[i] = table_new[i] = 99;
```

```
}
```

```
void copy() {
```

```
    for (int i = 0; i < n; i++) {
```

```
        adj_old[i] = adj_new[i];
```

```
        table_old[i] = table_new[i];
```

```
}
```

```
}
```

```
void display()
```

```
{
```

```
    cout << "Destination":
```

```
    for (int i = 0; i < n; i++)
```

```
        cout << "( " << i << " : " << table_new[i] << " )";
```

point is  $y = 0$ )  
vertical asymptote

1

odd half-plane

[vertical asymptote]

$\Rightarrow$  (Only one side falls outside the vertical  
asymptote)  $\Rightarrow$  if

3  
1

vertical asymptote

vertical asymptote

[vertical asymptote]

2nd quadrant

1  
last quadrant

2  
2nd quadrant

2  
2nd quadrant

Ans

entry of corresponding router in adjacent  
A B C D

Router table entry for A

Destination Router : A B C D E  
outgoing link : A B C D E  
Hop count 0 11 99 99

Destination Router : A B C D E  
outgoing link : A B C D E  
Hop count 0 11 99 99

Destination Router : A C D E F  
outgoing link : A C D E F  
Hop count 0 11 99 0 11

## Dijkstra

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

```
#include <limits.h>
```

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

```
using namespace std;
```

```
#define V 4
```

```
int minDistance(int dist[], bool sptset[])
```

```
{
```

```
    int min = INT_MAX, min_index;
```

```
    for (int v = 0; v < V; v++)
```

```
        if (!sptset[v]) min = dist[v] < min ? min : min;
```

```
        min_index = v;
```

```
    return min_index;
```

```
}
```

```
void printSolution(int dist[])
{
```

```
    cout << "Vertex ";
```

```
    for (int i = 0; i < V; i++)
        cout << " " << i << " ";
}
```

```
cout << endl;
```

```
int dijkstra(int graph[V][V], int src)
{
    int dist[V];
    bool sptset[V];
```

for (int i = 0; i < V; i++)  
    dist[i] = INT\_MAX;  
    sptSet[i] = false;

dist[0] = 0;

for (int count = 0; count < V - 1; count++)  
    int u = minDistance(dist, sptSet);  
    sptSet[u] = true;

for (int v = 0; v < V; v++)

    if (!sptSet[v] && graph[u][v] > 0)  
        dist[v] = dist[u] + graph[u][v];  
        if (dist[v] + graph[v][v] < dist[v])

            dist[v] = dist[v] + graph[u][v];

3

int main()

{  
    int graph[V][V];  
    cout << "Enter graph" << endl;

    for (int i = 0; i < V; i++)

        for (int j = 0; j < V; j++)  
            cin >> graph[i][j];

    Kruskal(graph, 0);

    return 0;

3

Output

into graph

0 9 25

9 0 6 3

2 6 0 0

5 3 0 0

Vertex

0

1

2

3

Distance from Junc(0)

0

8

2

5

# TCP-IP

SURYA Gold

Date

Page

from socket import \*

serverName = '127.0.0.1'

serverPort = 12000

2

clientSocket = socket(AF\_INET, SOCK\_STREAM)

clientSocket.connect((serverName, serverPort))

sentence = input("Enter file name")

clientSocket.send(sentence.encode())

fileContent = clientSocket.recv(1024).decode()

print("from server")

print(fileContent)

clientSocket.close()

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("Server is ready to receive")

connectionSocket, address = serverSocket.accept()

sentence = connectionSocket.recv(1024).decode()

file = open(sentence, 'x')

file.write(connectionSocket.recv(1024).decode())

```
connectionSocket.send(l.encode())
print("sentence")
file.close()
connectionSocket.close()
```

### Output

server is ready to receive  
sent content of server.py

## UDP

SURYA Gold

Date \_\_\_\_\_

Page \_\_\_\_\_

```
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))
fileContent, clientAddress = clientSocket.recvfrom(2048)
print(fileContent)
print(fileContent.decode("UTF-8"))
clientSocket.close()
```

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("UTF-8")
```

```
    file = open(sentence, "r")
    l = file.read(2048)
    serverSocket.sendto(l, clientAddress)
```

```
print("End contents of", end = ' ')
print(sentence)
file.close()
```

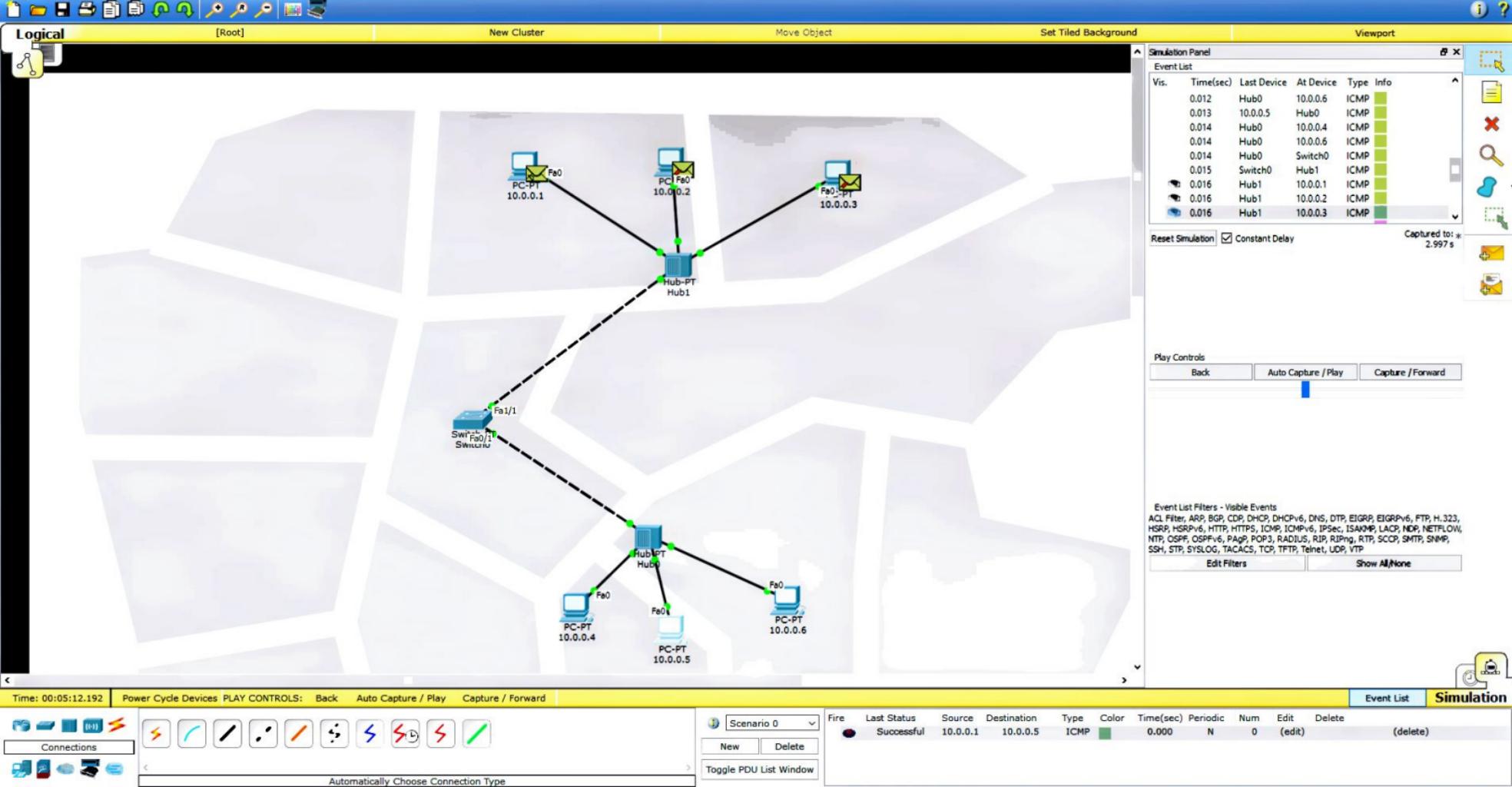
Output

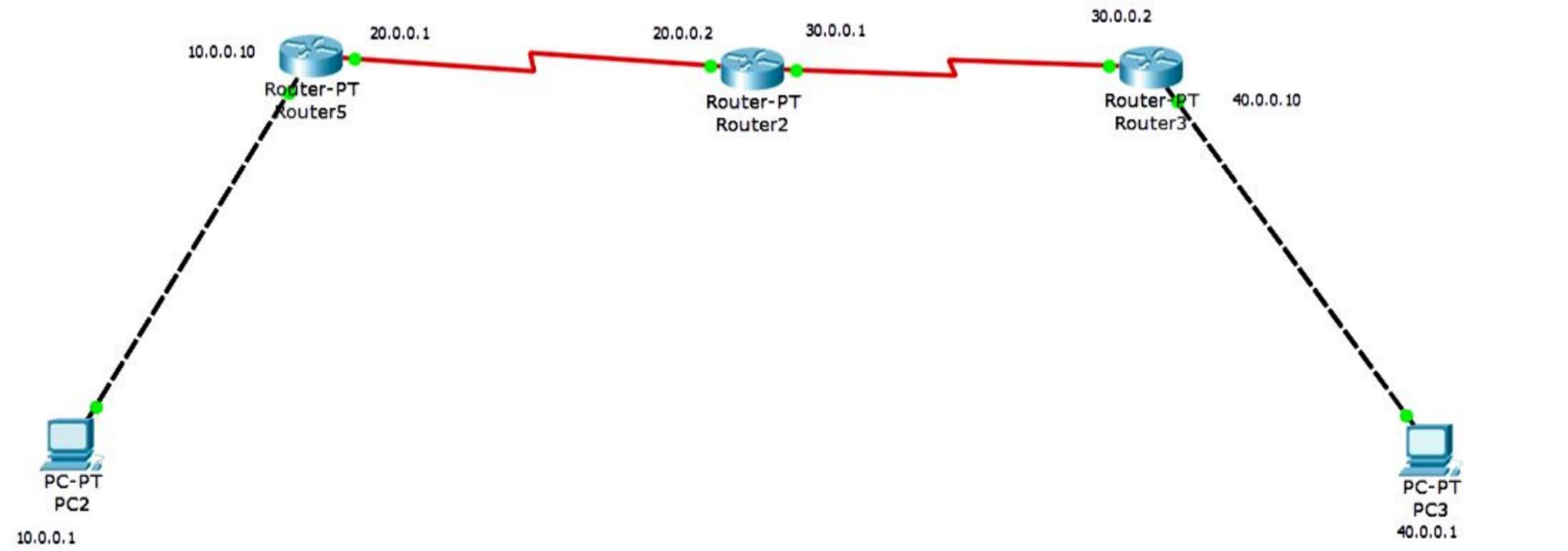
Sent content of servers UDP.py

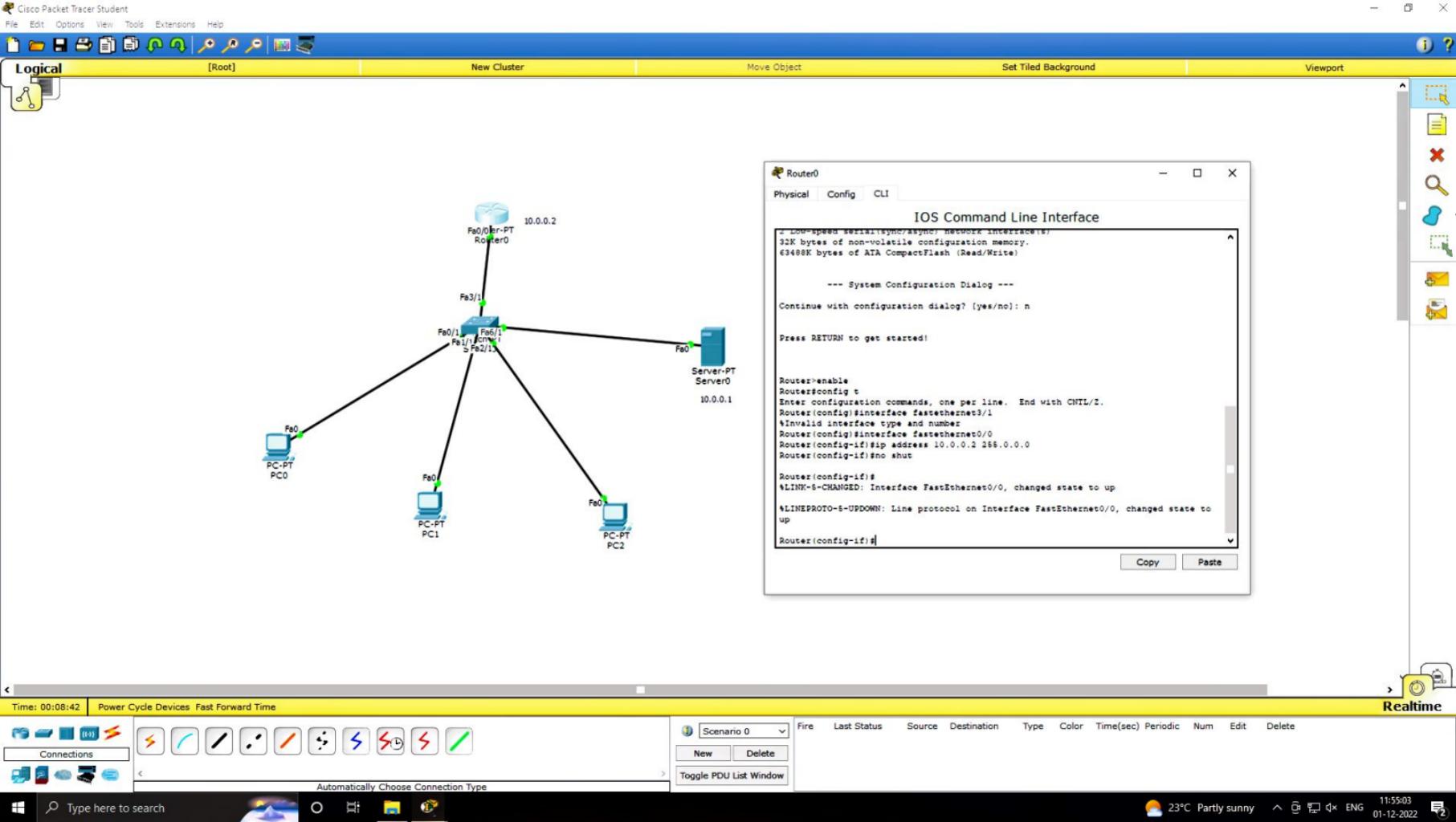
Reply from server:

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
```

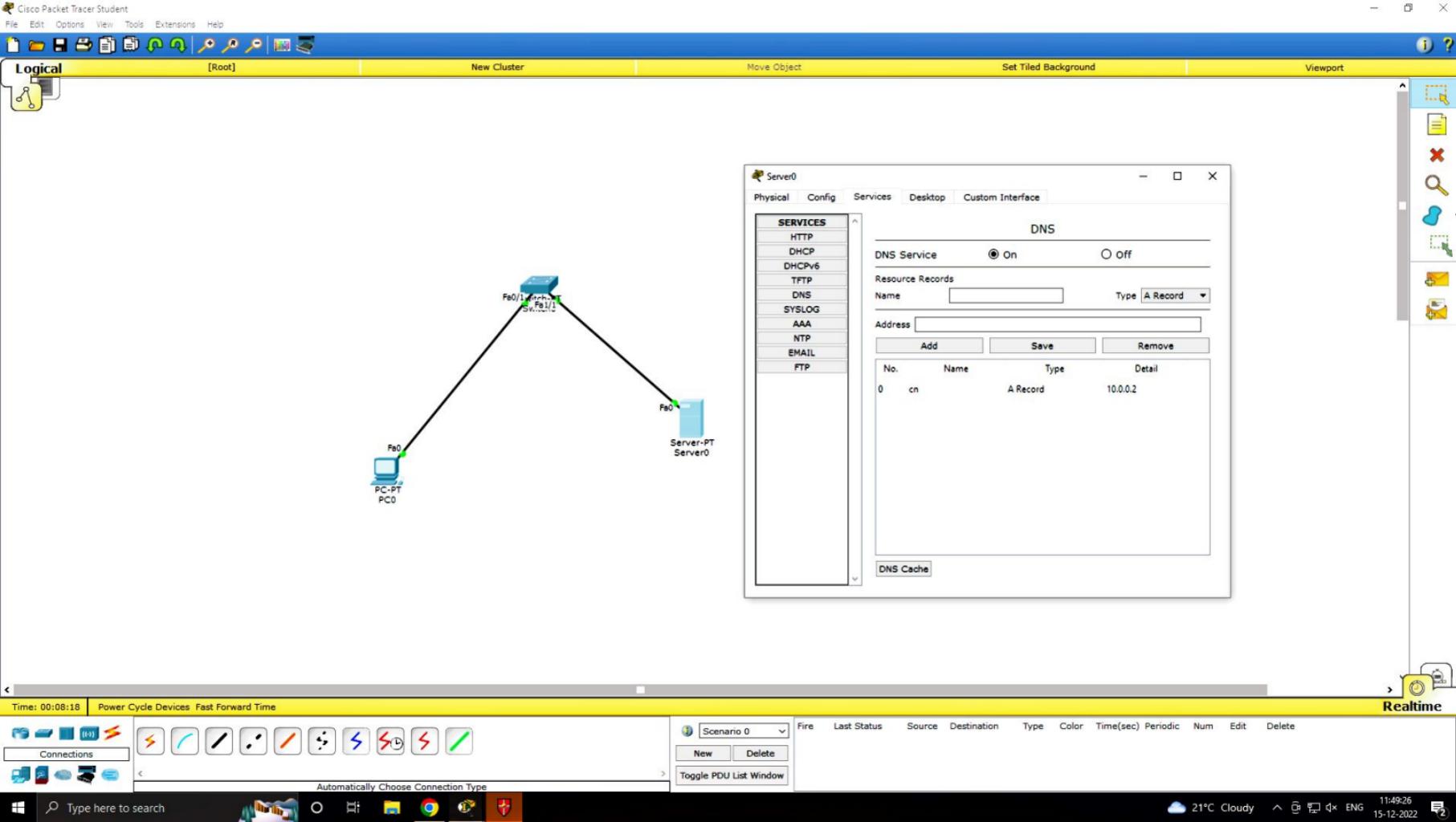
```
for i in sentence:
    print(str(i), end = '')
    file.write()
```

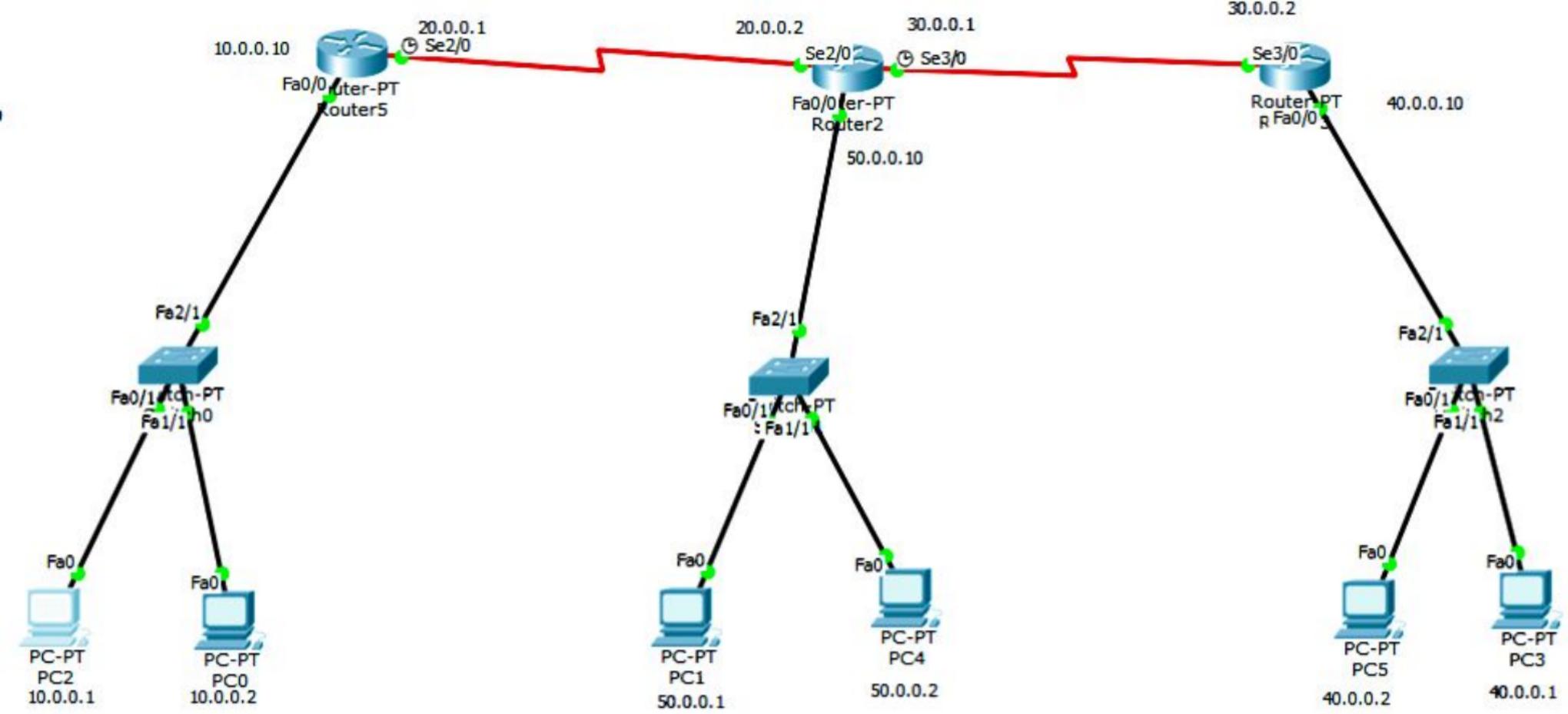












Enter the data:10000100000010001

Enter the key:10101

CRC=1101

Dataword=100001000000100011101

Process returned 0 (0x0) execution time : 37.143 s

Press any key to continue.

Enter the bucket capacity: 50

Enter the outflow rate: 10

1.Insert

2.Exit

Enter choice: 1

Enter the packet size: 40

40

After outflow: 30

1.Insert

2.Exit

Enter choice: 1

Enter the packet size: 5

35

After outflow: 25

1.Insert

2.Exit

Enter choice: 1

Enter the packet size: 30

Bucket overflow1.Insert

2.Exit

Enter choice: 2

No more inputs. Program exited

Enter the number of vertices: 4

Enter the source vertex of the graph: 1

Enter no. of edges: 5

For edge 1=>

Enter source vertex :1

Enter destination vertex :2

Enter weight :4

For edge 2=>

Enter source vertex :1

Enter destination vertex :3

Enter weight :5

For edge 3=>

Enter source vertex :3

Enter destination vertex :2

Enter weight :7

For edge 4=>

Enter source vertex :2

Enter destination vertex :4

Enter weight :7

For edge 5=>

Enter source vertex :4

Enter destination vertex :3

Enter weight :-15

NEGATIVE CYCLE PRESENT..!!

Enter the no. of vertices: 5

Enter the adjacency matrix:

0 3 1 0 0

3 0 7 5 1

1 7 0 2 0

0 5 2 0 7

0 1 0 7 0

Enter the starting node: 0

Distance of 1 = 3

Path = 1 <-0

Distance of 2 = 1

Path = 2 <-0

Distance of 3 = 3

Path = 3 <-2 <-0

Distance of 4 = 4

Path = 4 <-1 <-0

C:\Users\Srikanth\PycharmProjects\pythonProject1\Scripts\python.exe C:\U

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()
```

Process finished with exit code 0

C:\Users\Srikanth\PycharmProjects\pythonProject1\Scripts\python.exe

Enter file name: serverudp.py

Reply from 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)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    l=file.read(2048)
    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print ('\nSent contents of ', end = ' ')
    print (sentence)
# for i in sentence:
#     print (str(i), end = '')
    file.close()
```

Process finished with exit code 0



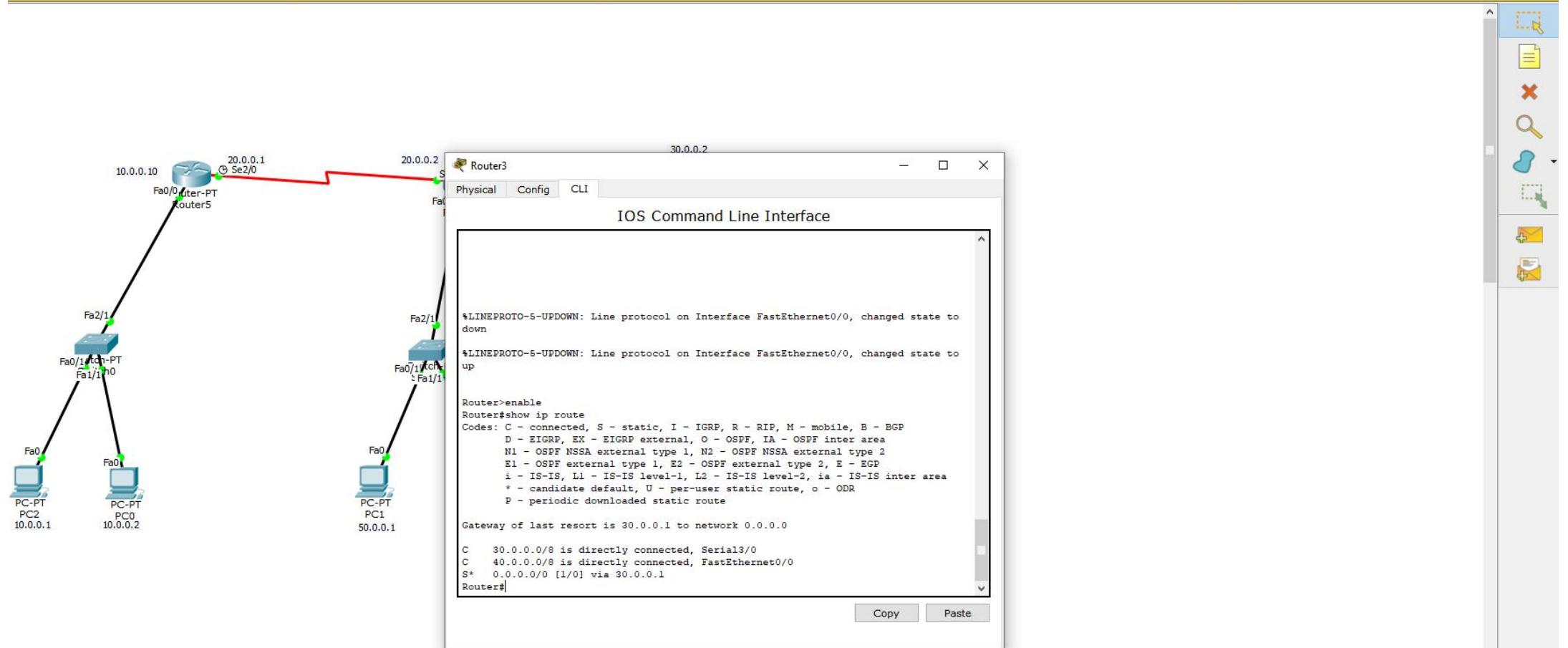
[Root]

New Cluster

Move Object

Set Tiled Background

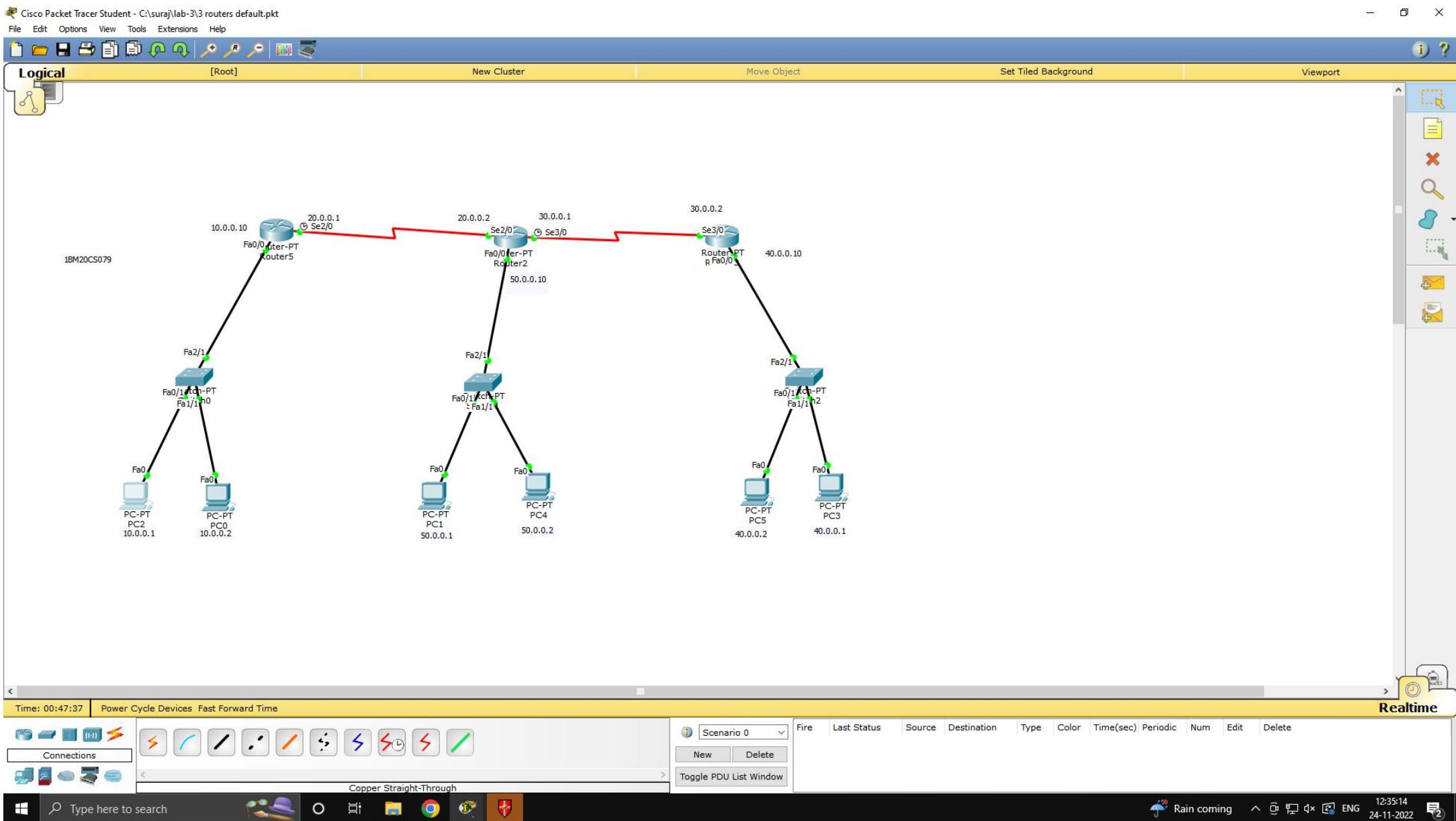
Viewport

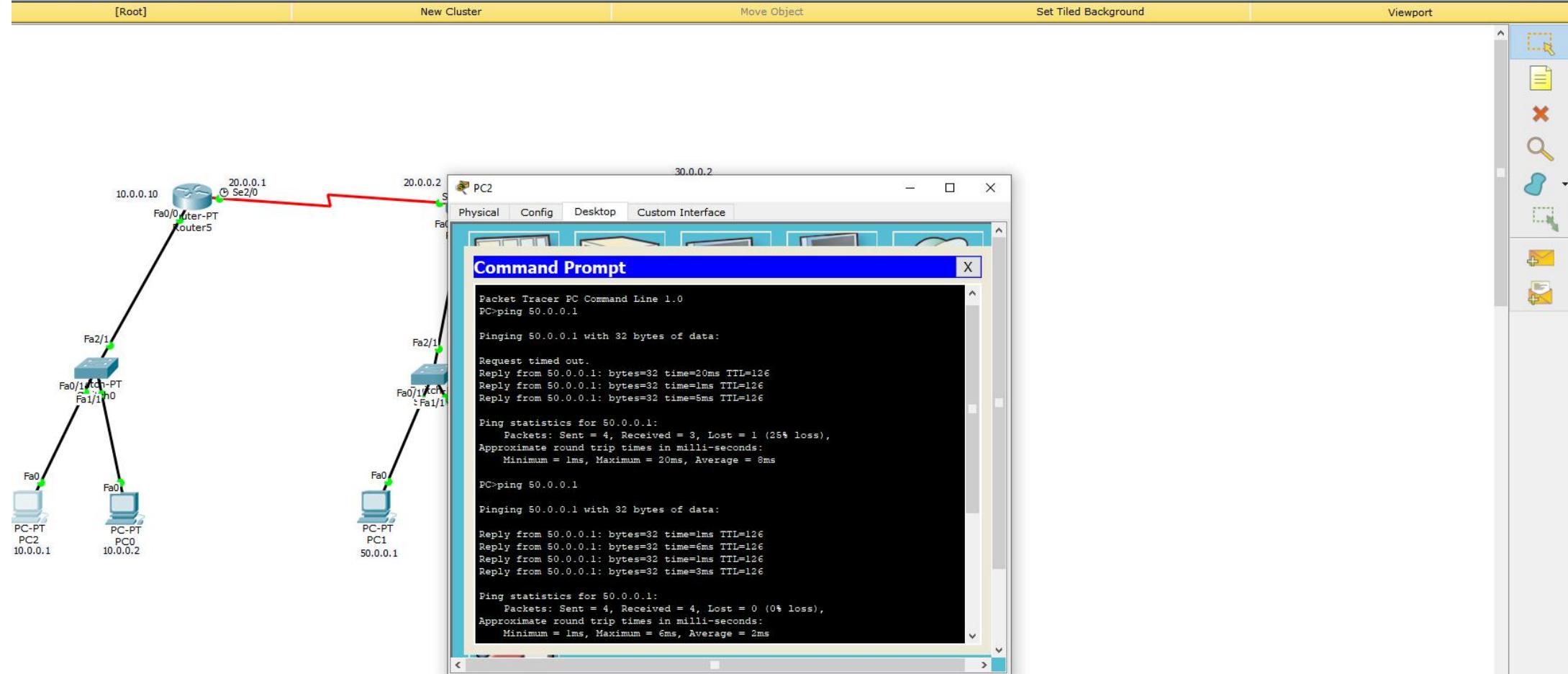


| Scenario 0 | Fire   | Last Status | Source | Destination | Type | Color | Time(sec) | Periodic | Num | Edit | Delete |
|------------|--------|-------------|--------|-------------|------|-------|-----------|----------|-----|------|--------|
| New        | Delete |             |        |             |      |       |           |          |     |      |        |

Toggle PDU List Window





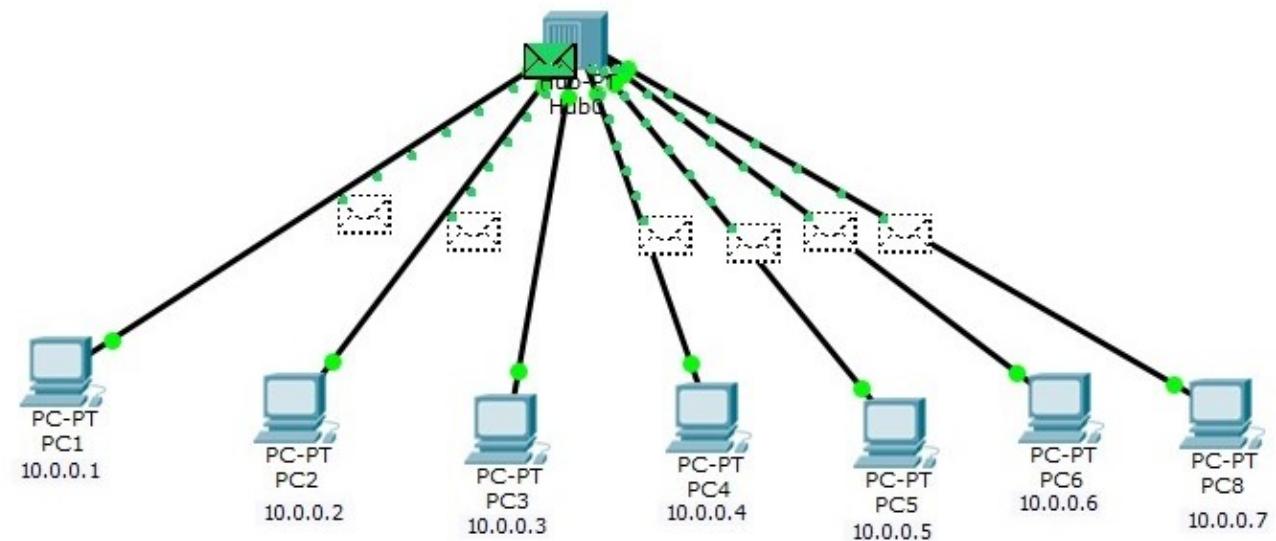


Scenario 0 | Fire | Last Status | Source | Destination | Type | Color | Time(sec) | Periodic | Num | Edit | Delete

New | Delete | Toggle PDU List Window

to search







Physical Config Desktop Custom Interface

## Command Prompt X

Packet Tracer PC Command Line 1.0

PC>PING 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.4:

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

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>

10.0.0.7

PC9

Physical Config Desktop Custom Interface

## Command Prompt

X

```
Packet Tracer PC Command Line 1.0  
PC>PING 10.0.0.10
```

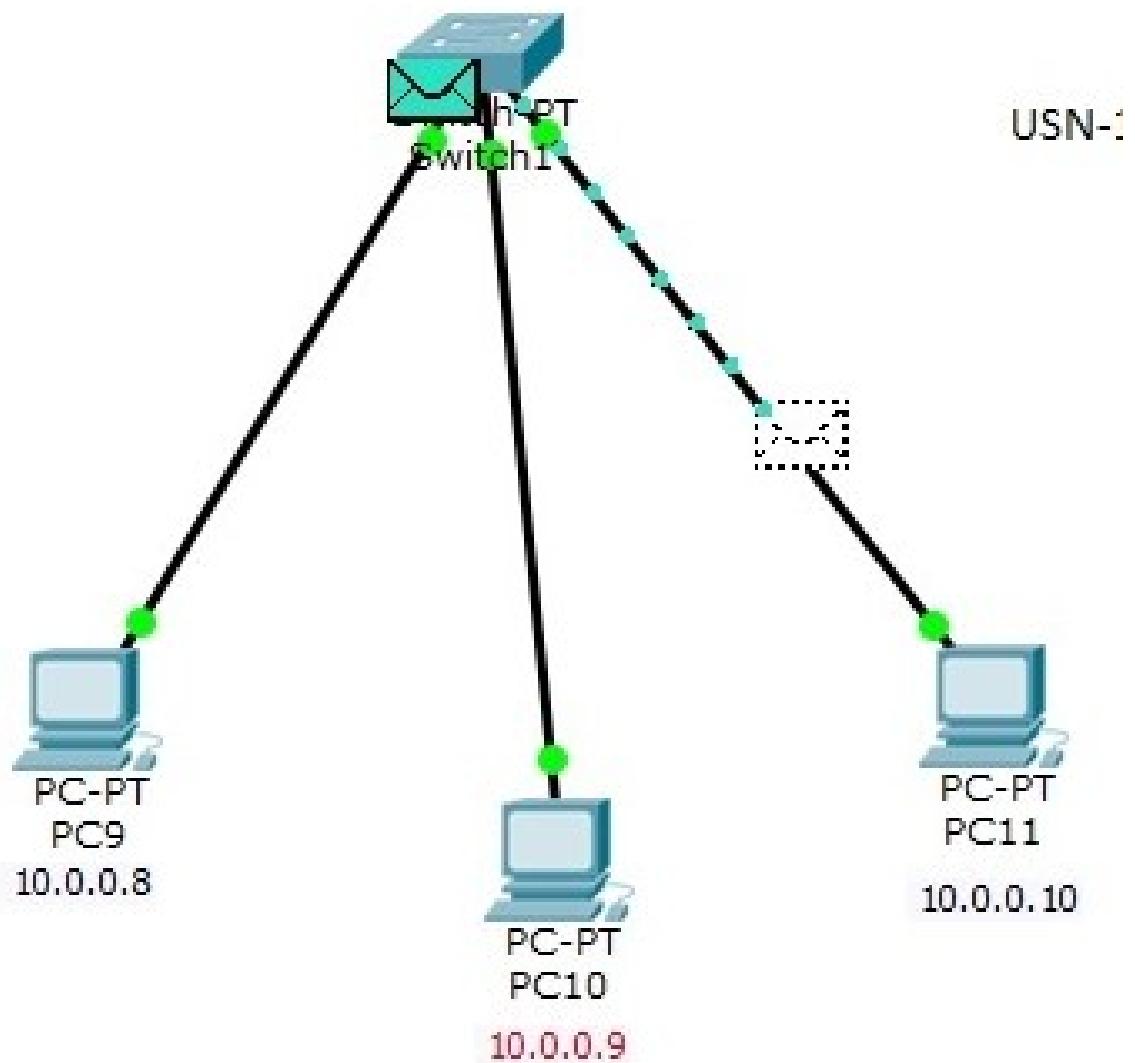
```
Pinging 10.0.0.10 with 32 bytes of data:
```

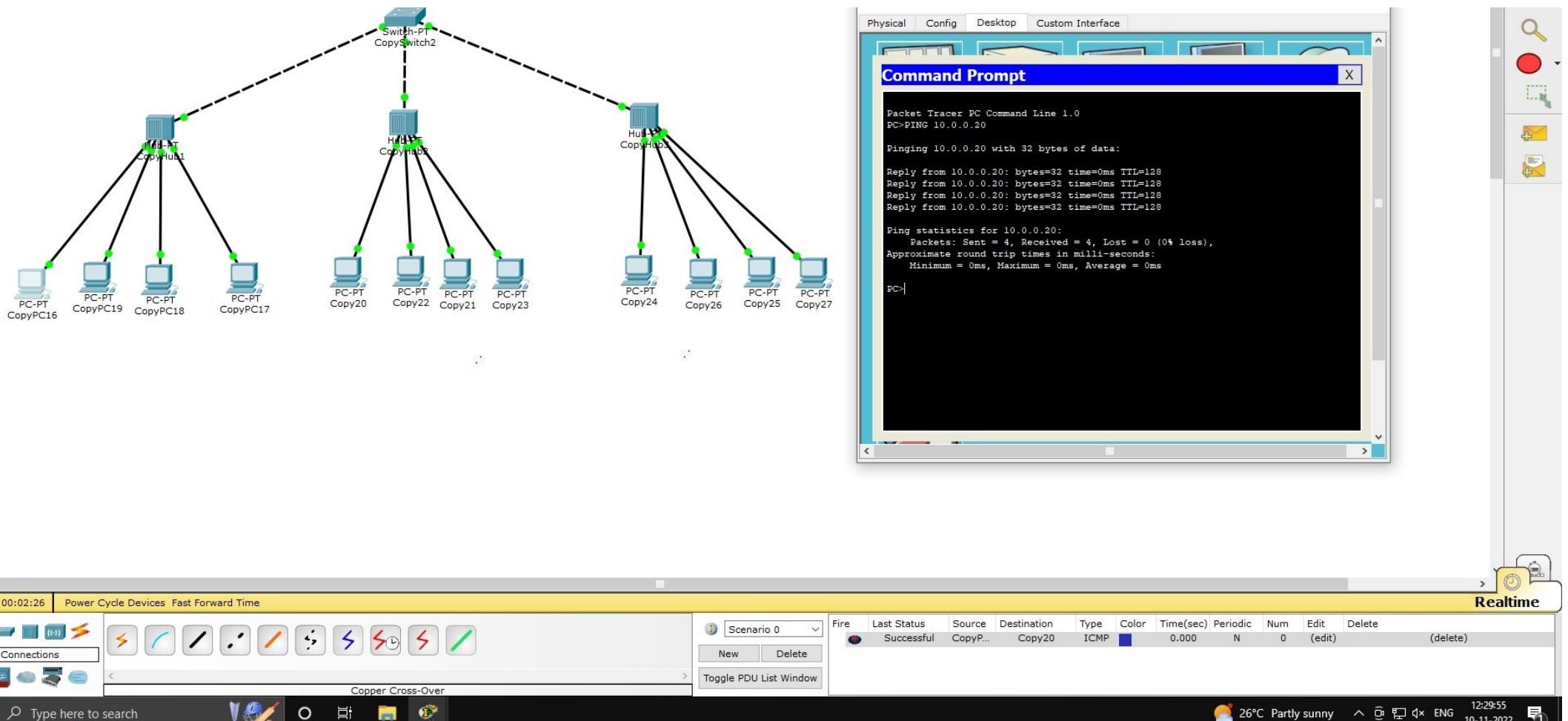
```
Reply from 10.0.0.10: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.10: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.10: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.10: bytes=32 time=4ms TTL=128
```

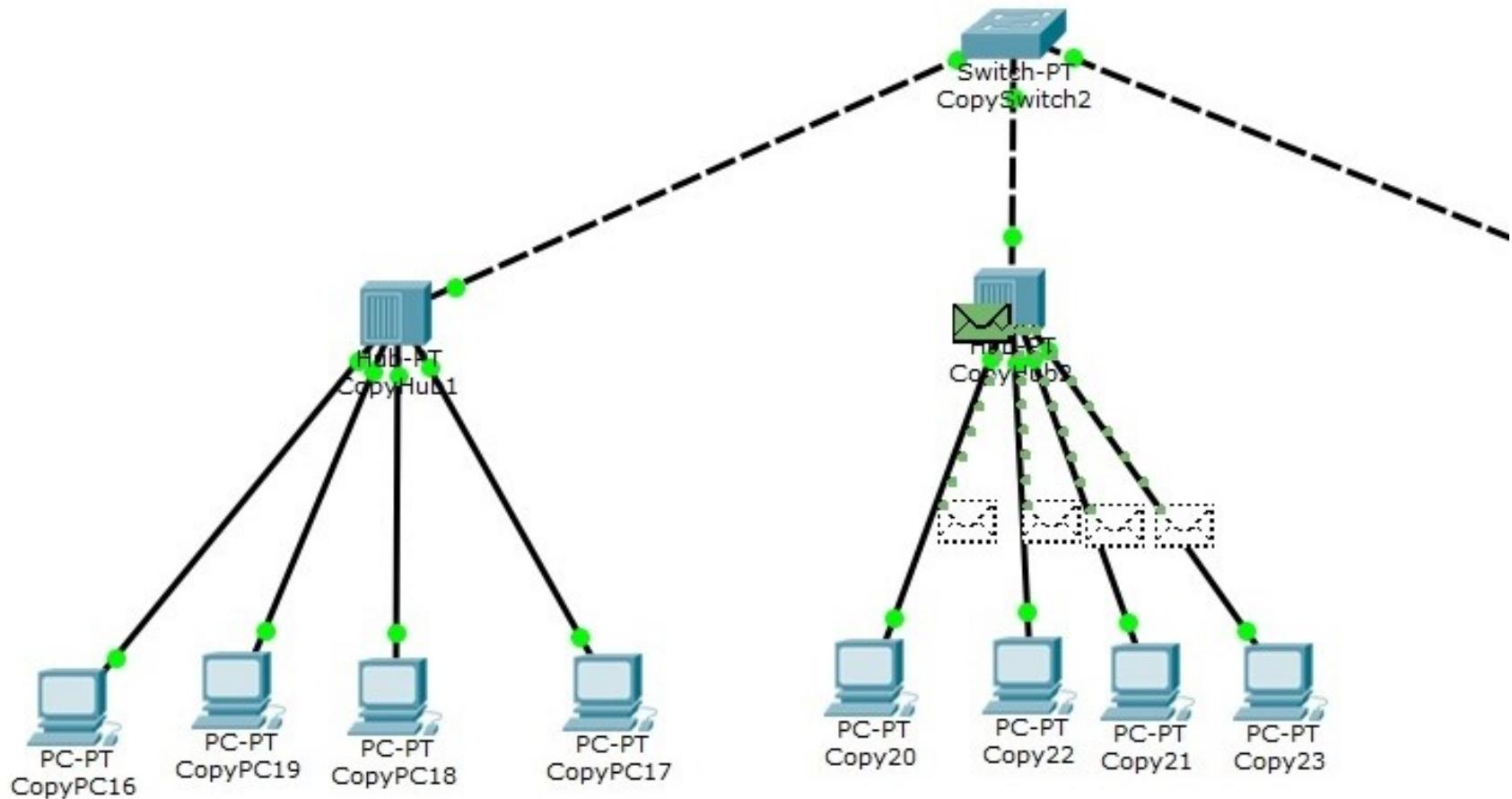
```
Ping statistics for 10.0.0.10:
```

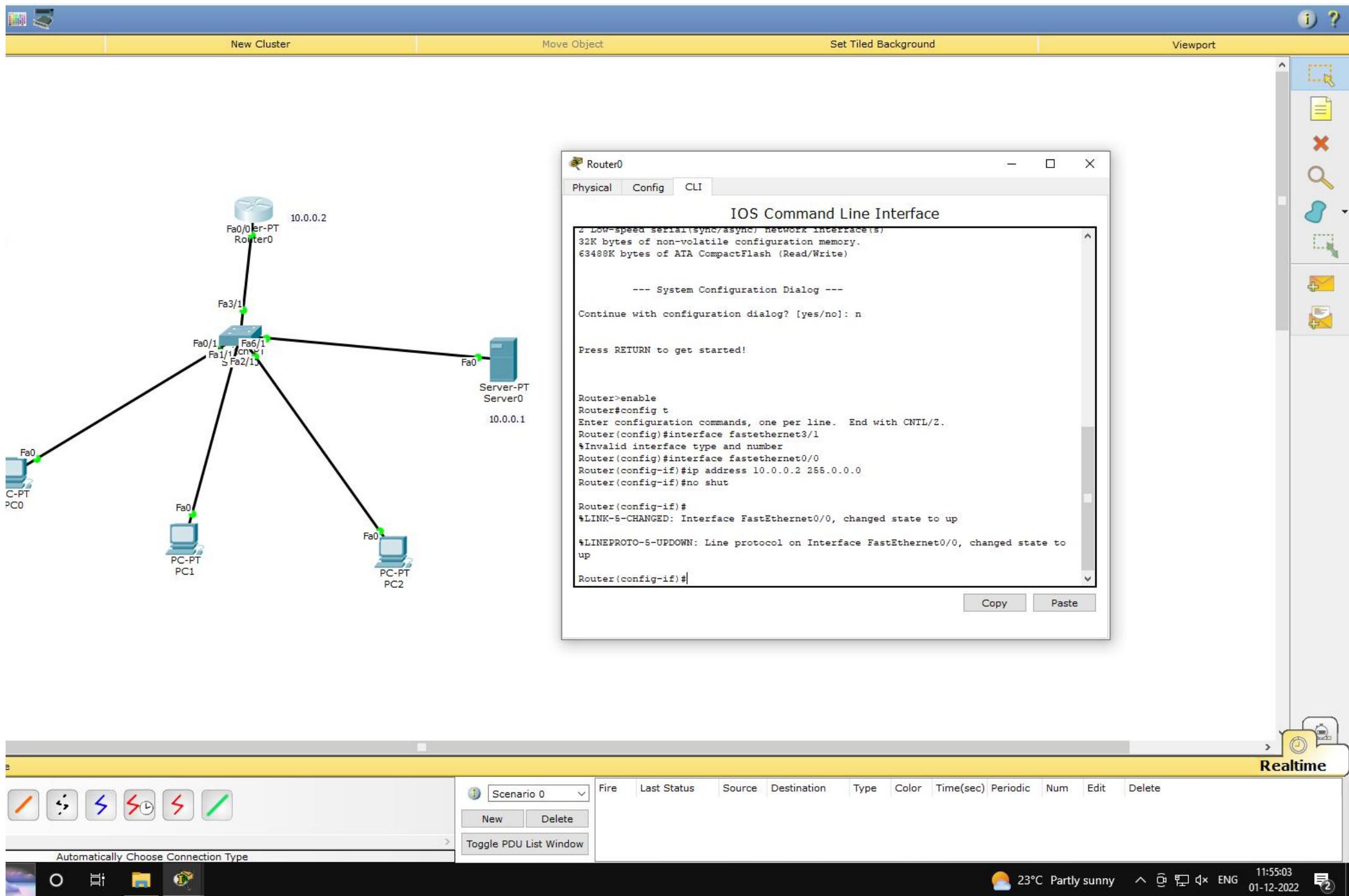
```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 4ms, Average = 1ms
```

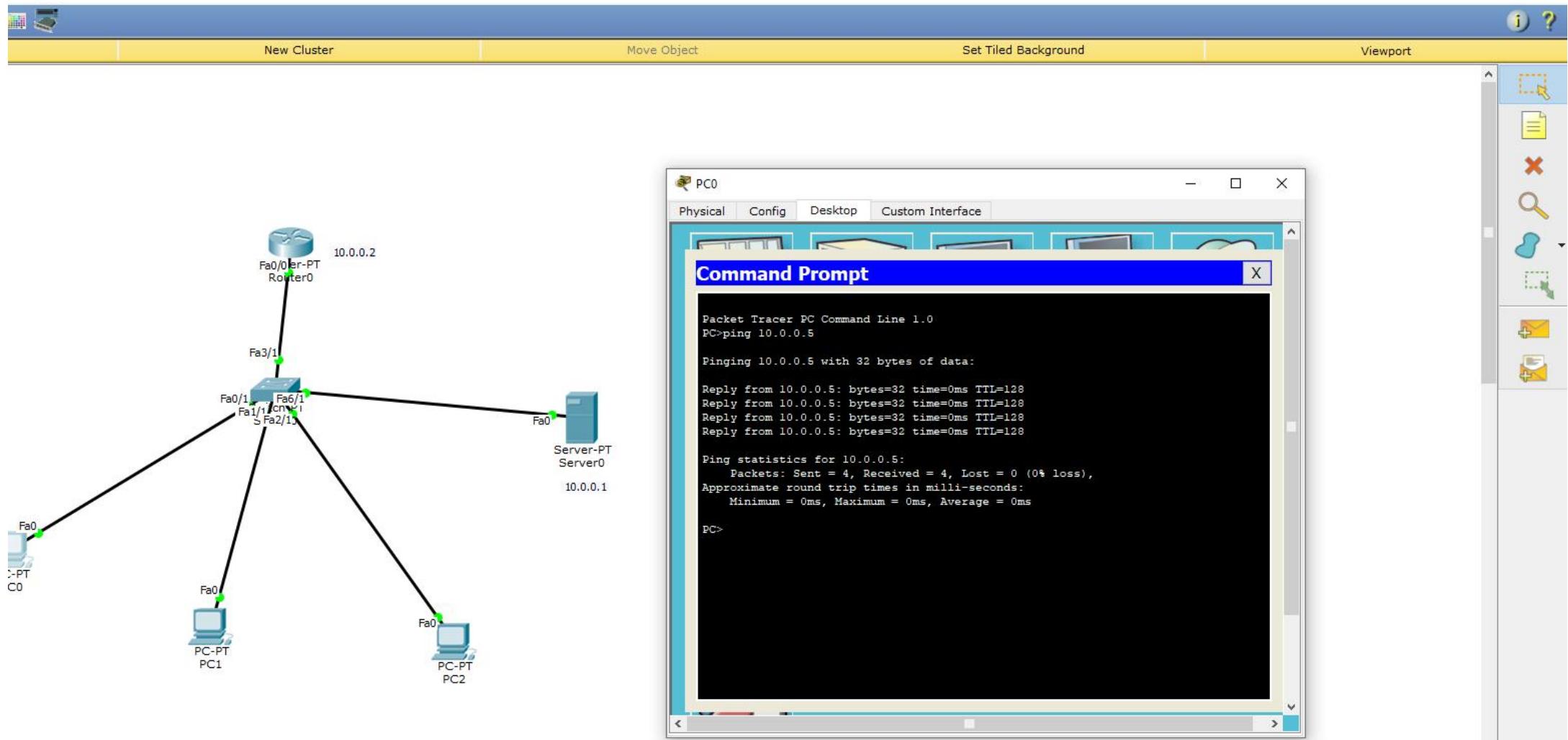
```
PC>
```











**PC0**

Physical Config Desktop Custom Interface

**Command Prompt**

```

Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.5

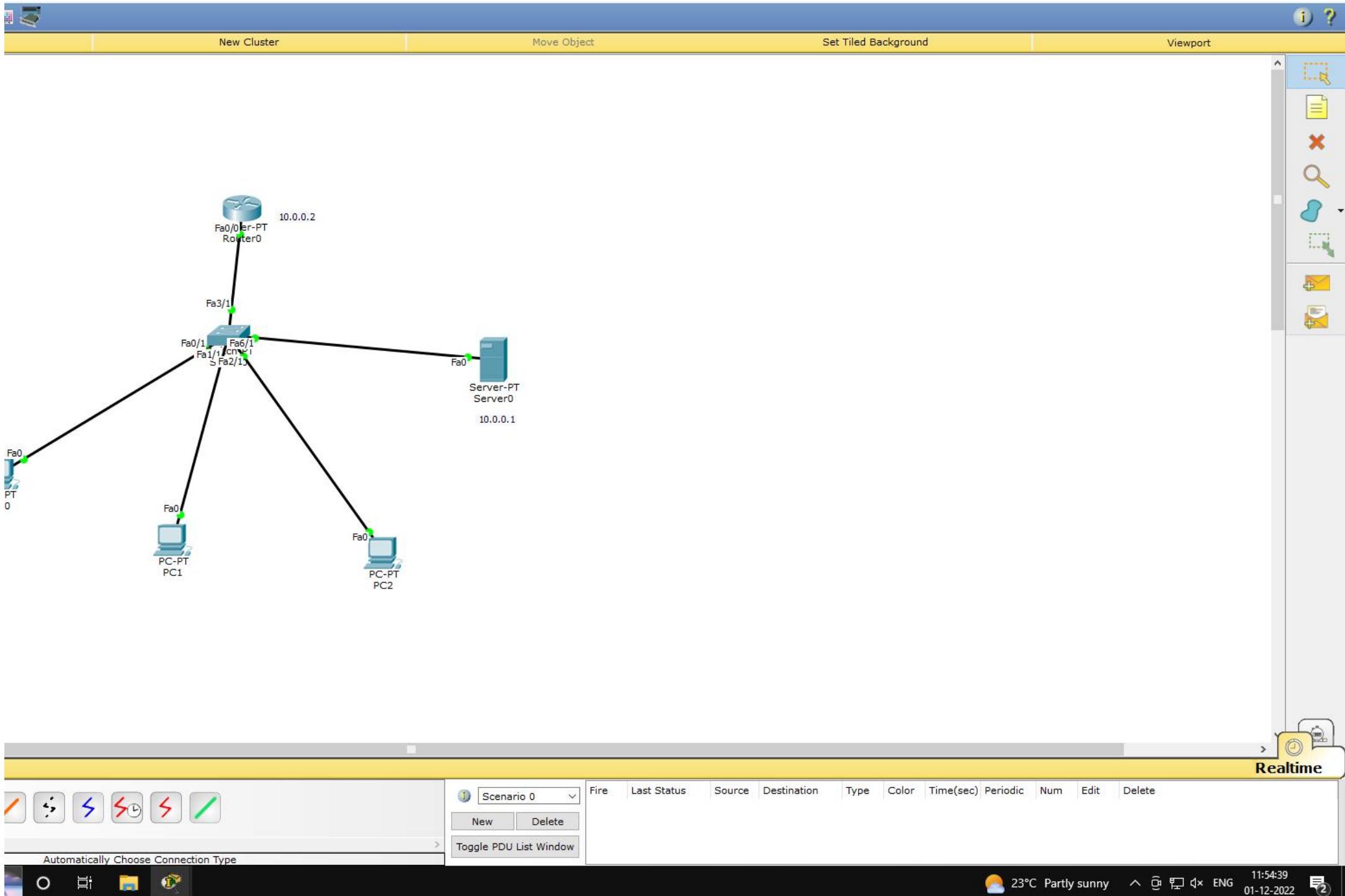
Pinging 10.0.0.5 with 32 bytes of data:

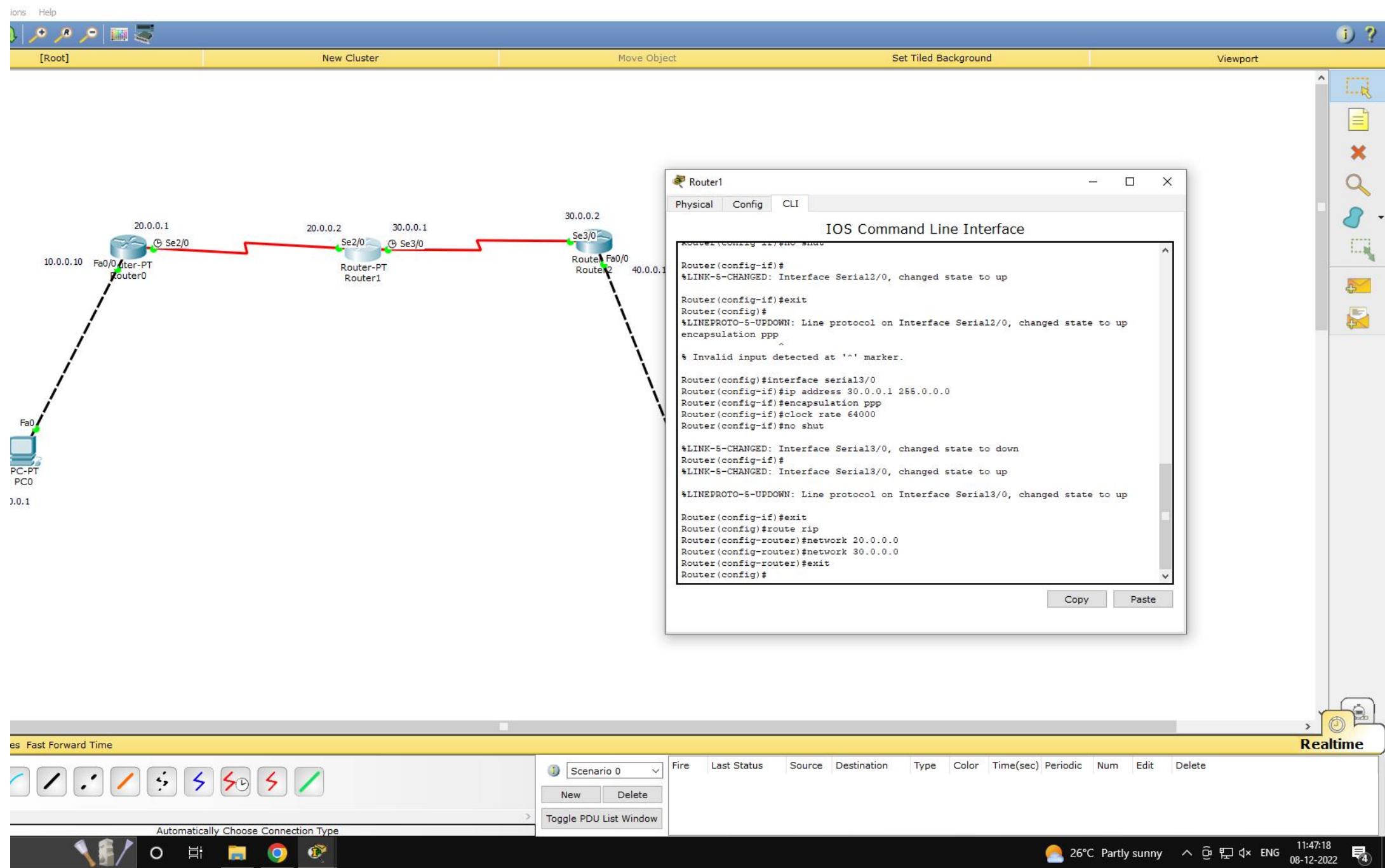
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128

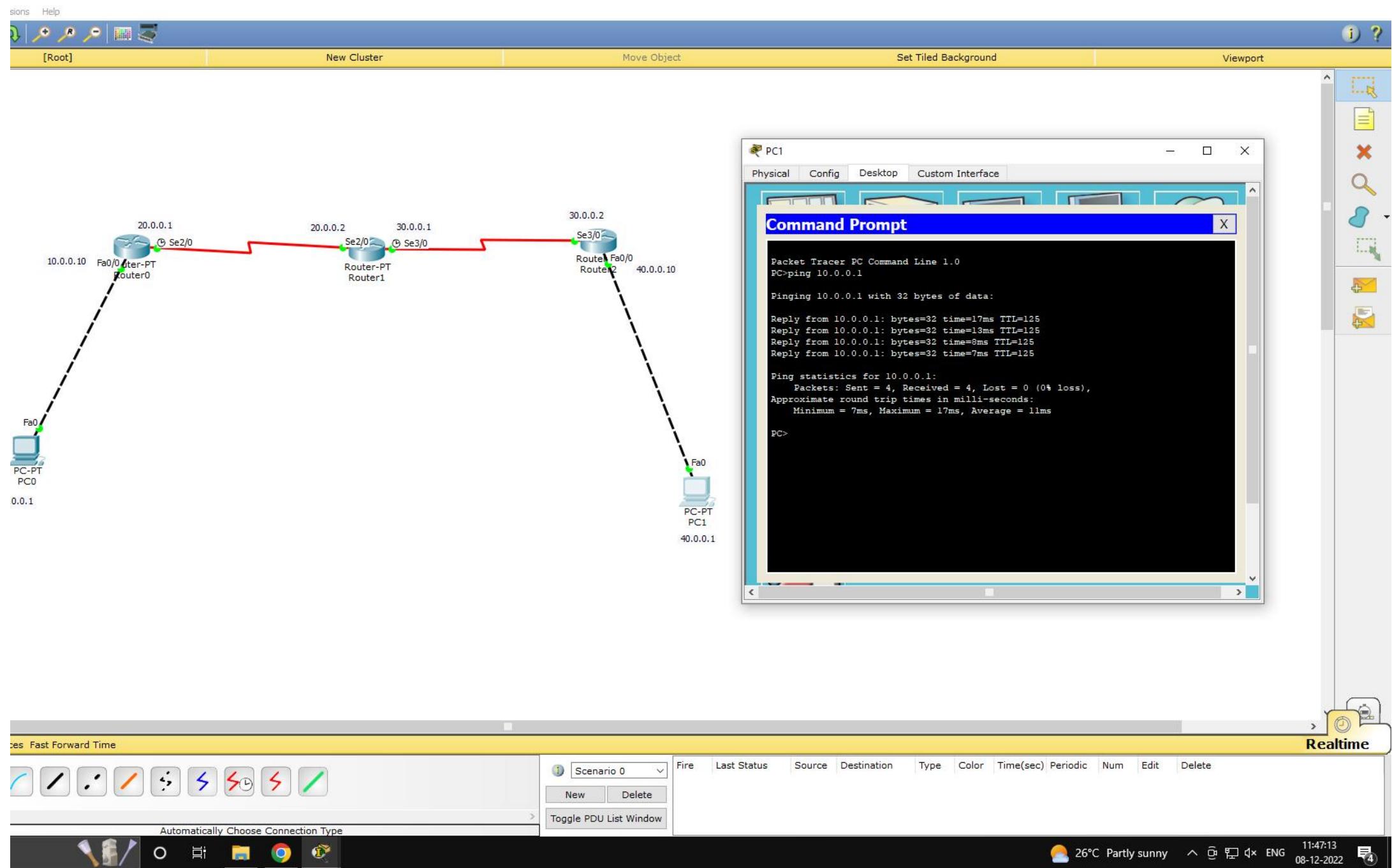
Ping statistics for 10.0.0.5:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

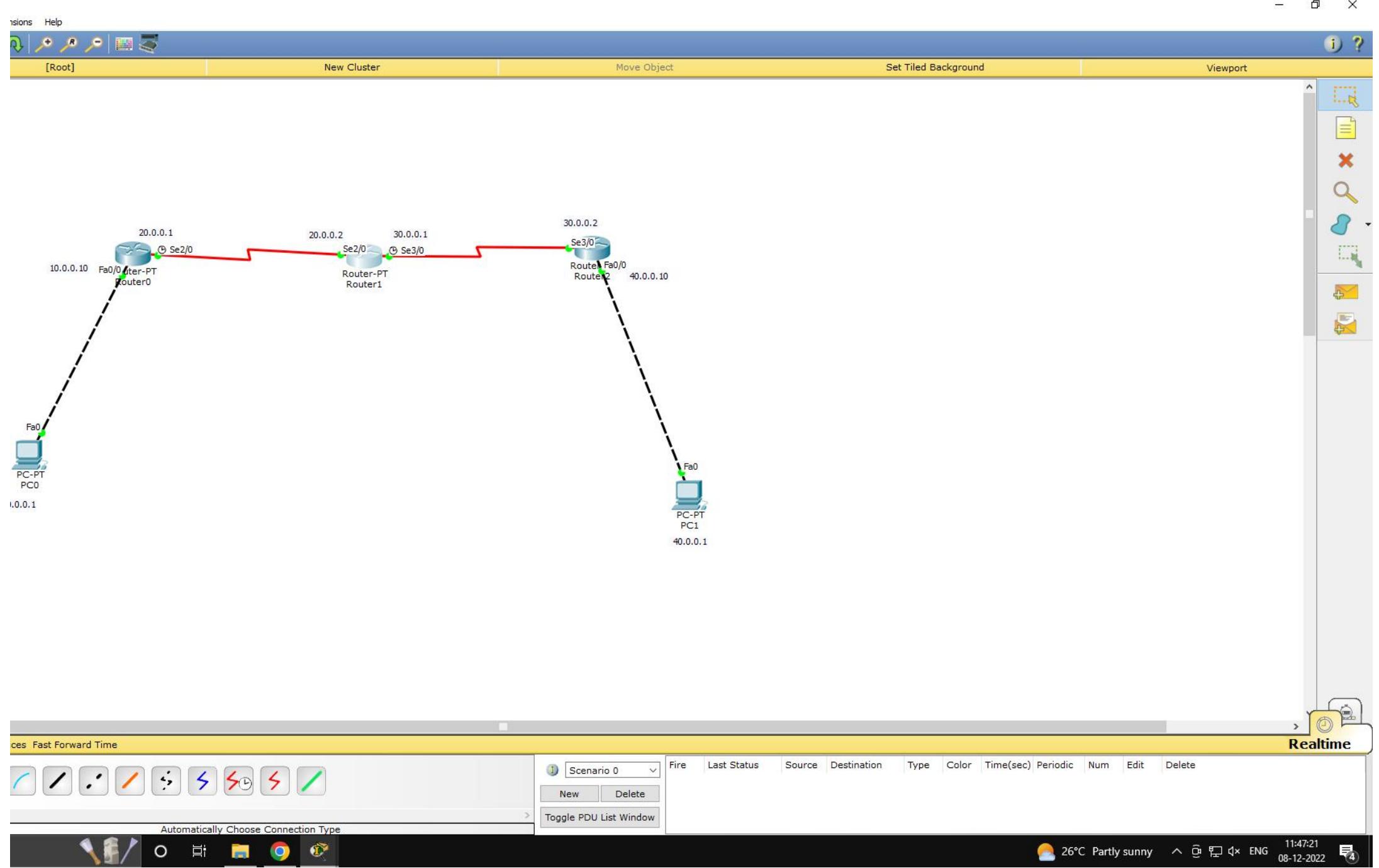
PC>
  
```











C:\Program Files\WindowsAp X +

- □ X

```
enter bucket capacity = 500
enter output rate = 5
enter packet size = 250
current bucket content = 245
enter packet size = 260
bucket is already full
240
enter packet size = 220
current bucket content = 455
enter packet size = 45
current bucket content = 495
enter packet size = |
```

C:\Program Files\WindowsAp X + v

```
enter bucket capacity = 500
enter output rate = 5
enter packet size = 600
packet size exceeded
current bucket content =  0
enter packet size = |
```

Enter the number the routers(<10): 5

Enter 1 if the corresponding router is adjacent to routerA else enter 99:

B C D E

Enter matrix:1 1 99 99

Enter 1 if the corresponding router is adjacent to routerB else enter 99:

A C D E

A B C E

Enter matrix:99 99 1 99

Enter 1 if the corresponding router is adjacent to routerE else enter 99:

A B C D

Enter matrix:99 99 1 99

Router Table entries for router A:-

Destination Router: A B C D E

Outgoing Line: A B C D E

Hop Count: 0 1 1 99 99

Router Table entries for router B:-

Destination Router: A B C D E

Outgoing Line: A B C D E

Hop Count: 1 0 99 99 99

Router Table entries for router C:-

Destination Router: A B C D E

Outgoing Line: A B C D E

Hop Count: 1 99 0 1 1

Router Table entries for router D:-

Destination Router: A B C D E

Outgoing Line: A B C D E

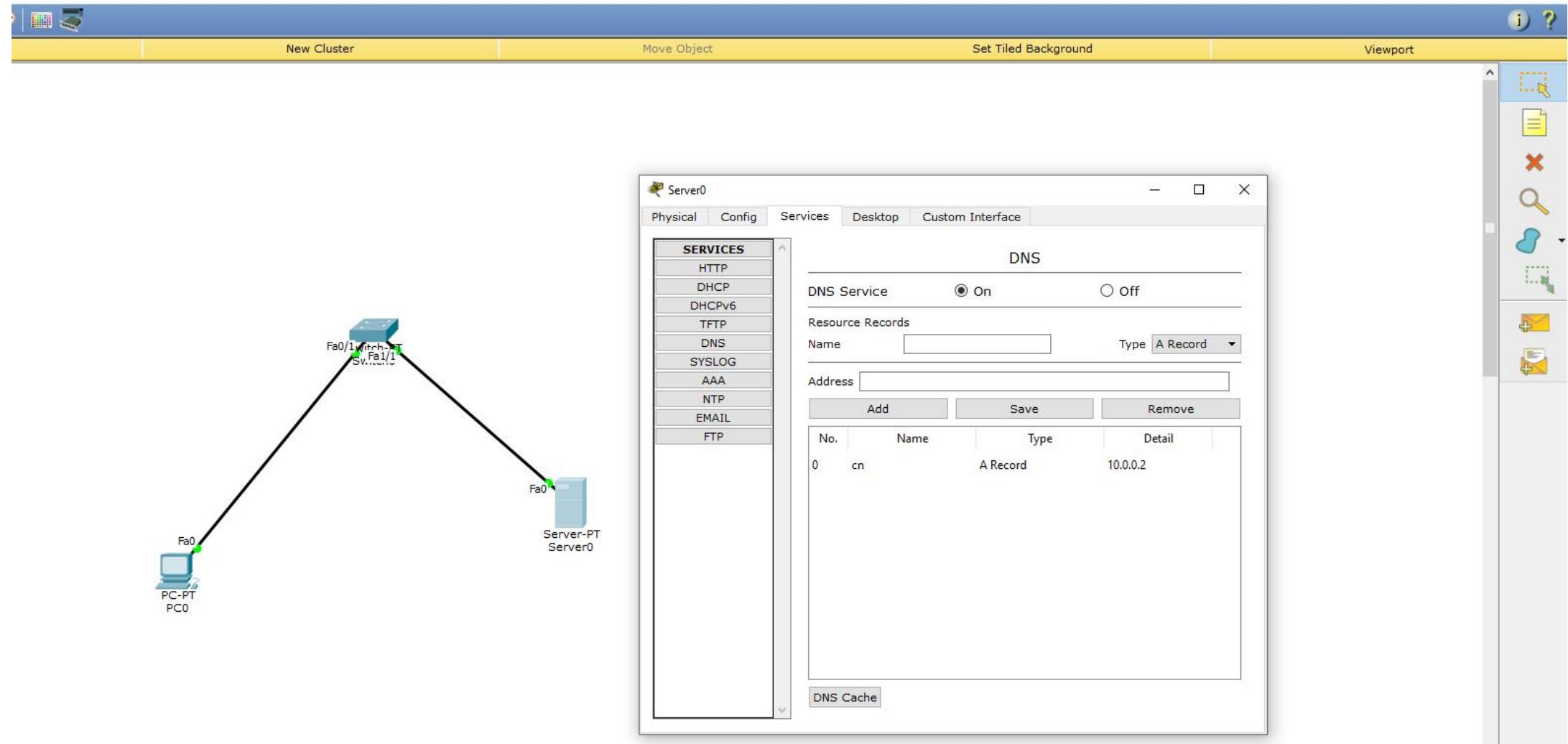
Hop Count: 99 99 1 0 99

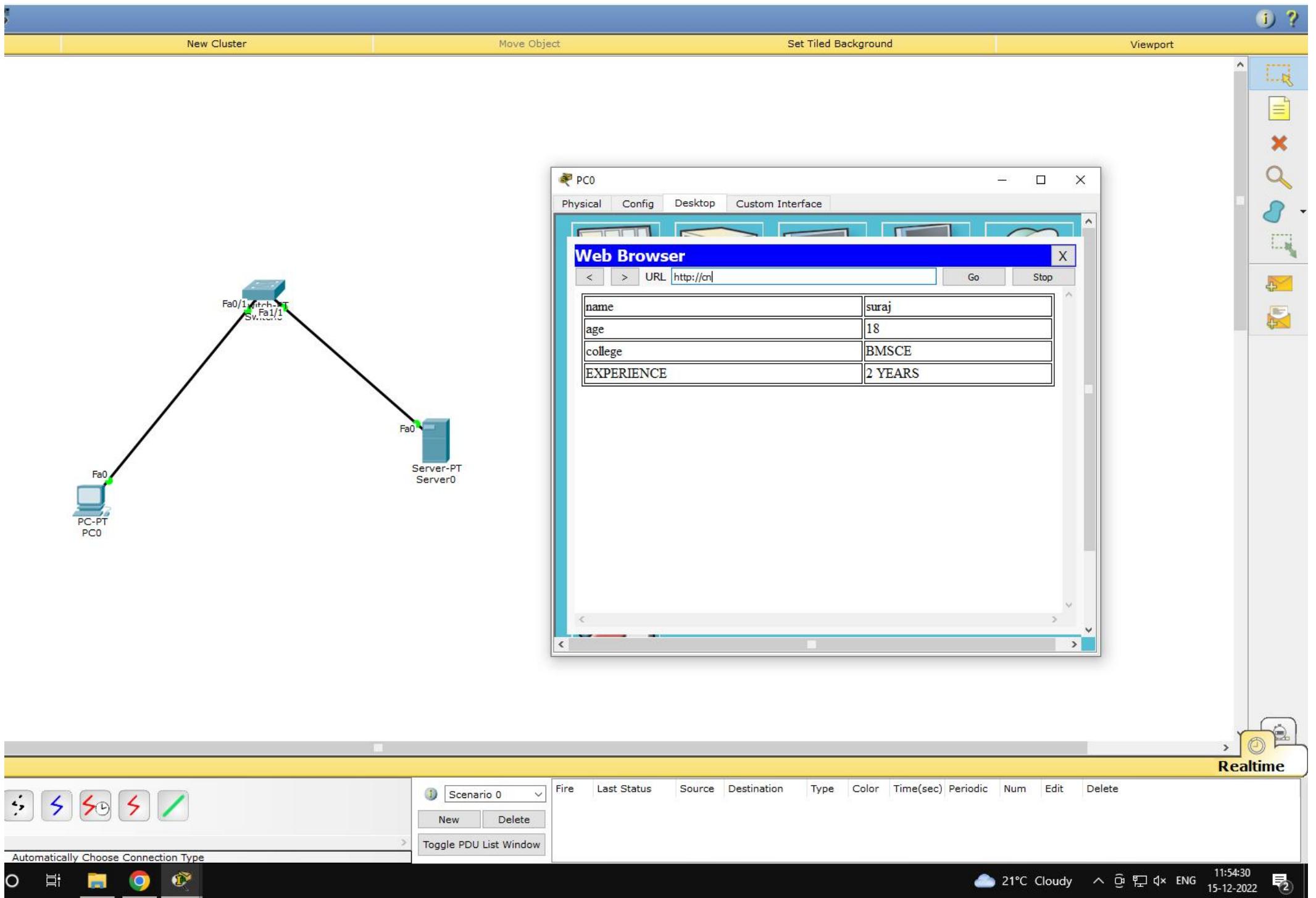
Router Table entries for router E:-

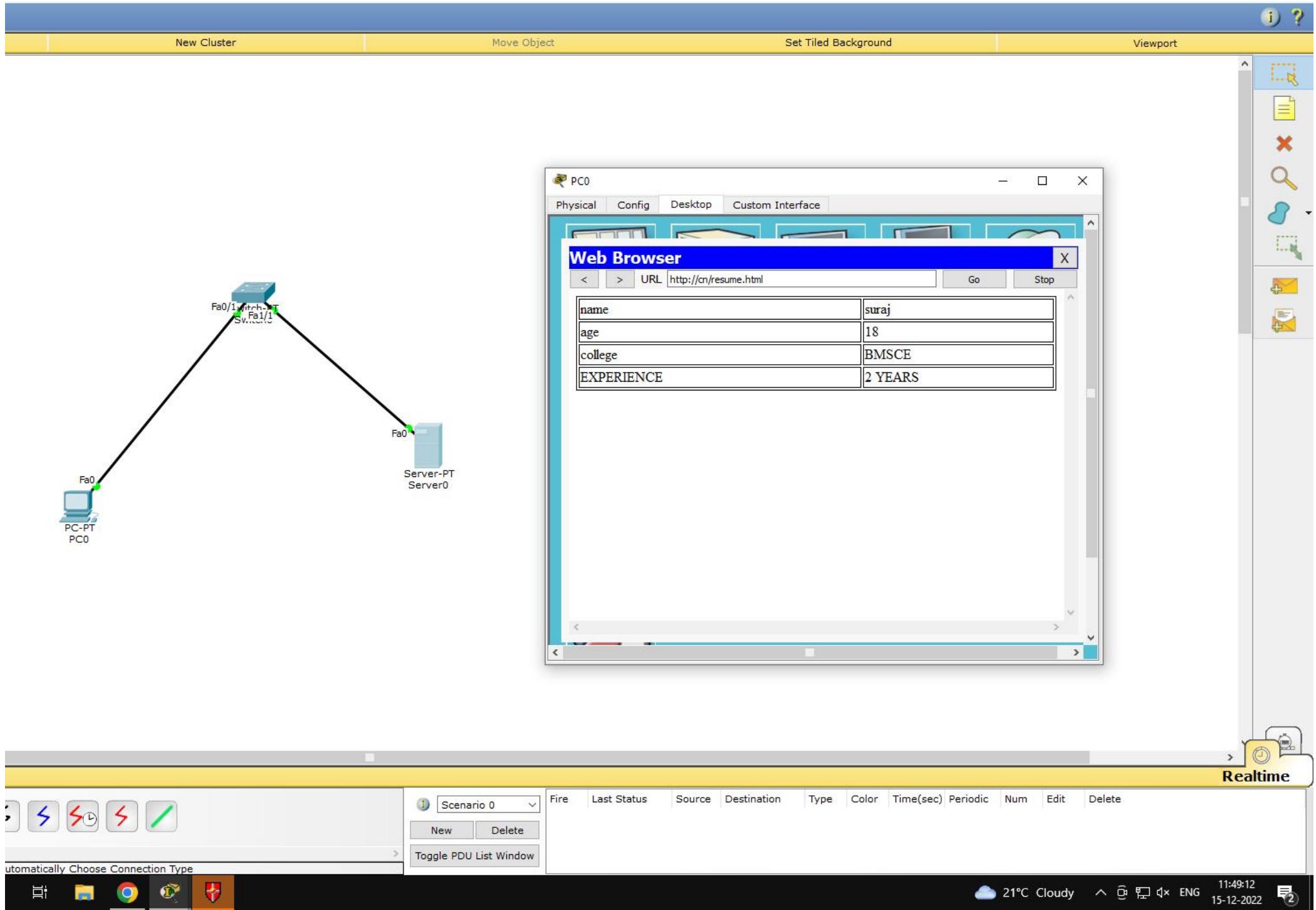
Destination Router: A B C D E

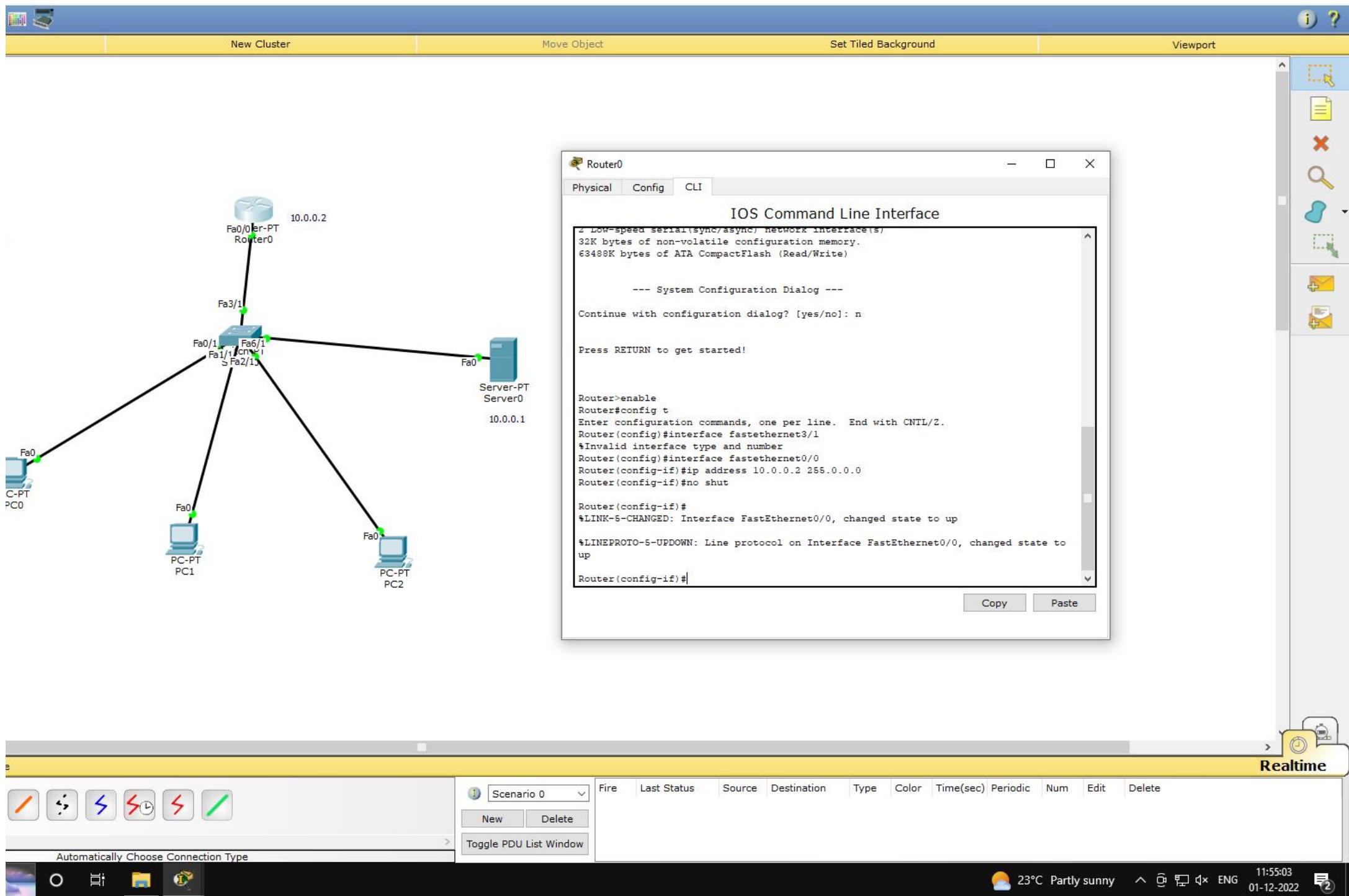
Outgoing Line: A B C D E

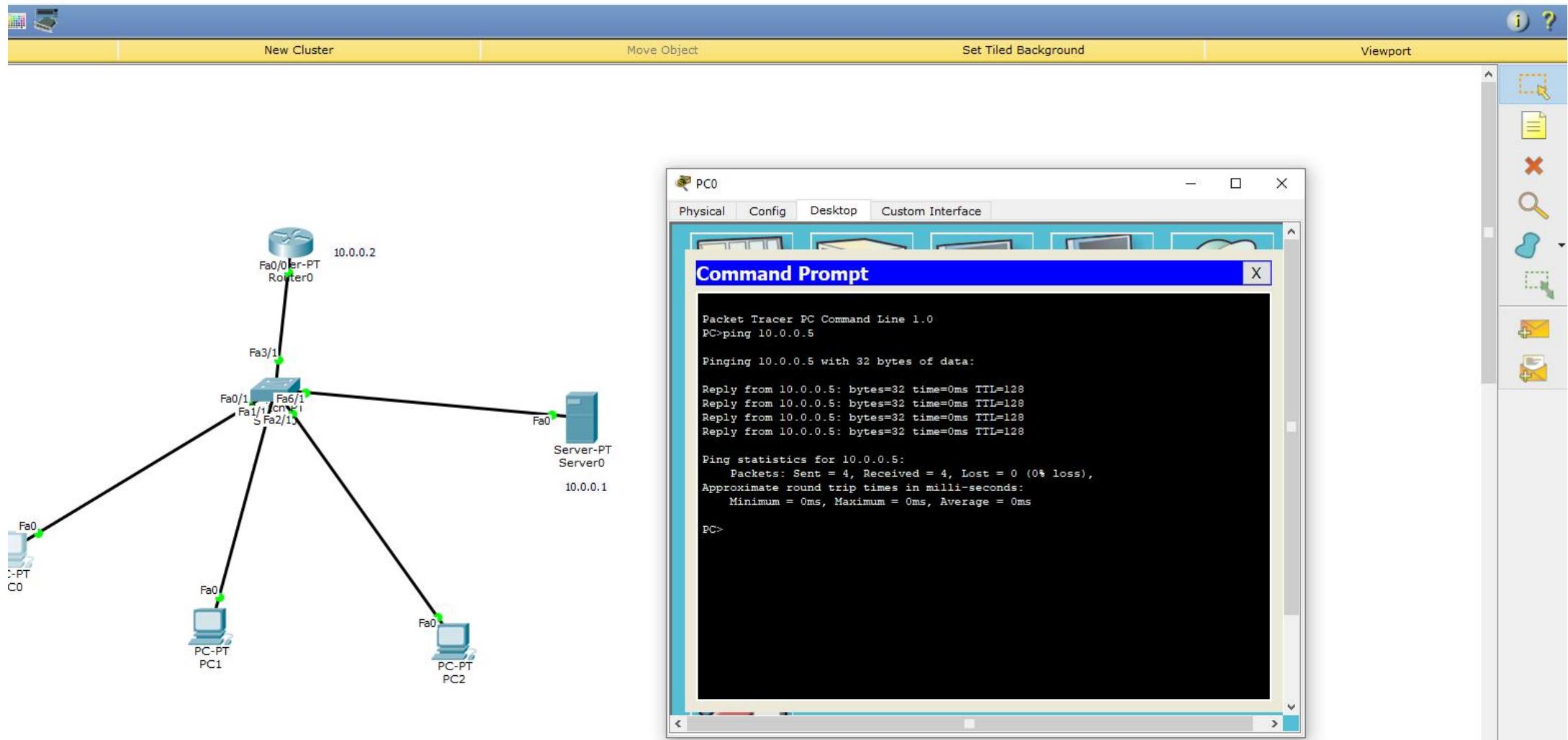
Hop Count: 99 99 1 99 0











**PC0**

Physical Config Desktop Custom Interface

**Command Prompt**

```

Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.5:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
  
```



