

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT on COMPUTER NETWORKS

Submitted by

MEHUL TEJ (1BM20CS085)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING BENGALURU-560019 October-2022 to Feb-2023
(Autonomous Institution under VTU)

**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 **COMPUTER NETWORKS**” carried out by **MEHUL TEJ (1BM20CS085)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-2023. 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. Seema Patil

Designation
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak

Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

Sl. No.	Date	Experiment Title	
1	10-11-2022	Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.	
2	17-11-2022	Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply	
3	24-11-2022	Configuring default route to the Router	
4	01-12-2022	Configuring DHCP within a LAN in a packet Tracer	
5	08-12-2022	Configuring RIP Routing Protocol in Routers	
6	15-12-2022	Demonstration of WEB server and DNS using Packet Tracer	
7	29-12-2022	Write a program for error detecting code using CRC-CCITT (16-bits).	
8	12-01-2023	Write a program for distance vector algorithm to find suitable path for transmission.	
9	12-01-2023	Implement Dijkstra's algorithm to compute the shortest path for a given topology.	
10	05-01-2023	Write a program for congestion control using Leaky bucket algorithm.	
11	28-01-2023	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.	
12	29-01-2023	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.	

Interface Overview

File / Open / Save / Print

In this we have Menu Bar and in it we have menu bar tool, Common tools, Logical / physical workspace, Realtime / simulation bar, Device type selection bar, Device specific selection bar and User created packet window.

It has 2 workspaces and 2 modes. In Logical you can build your network and in simulation you can run controlled networking.

You can change setting according to your preferences. You can toggle b/w animation, sounds, show link lights etc. In simulation Buffer full action, prompt, clear event list etc.

: Admin bar (with profile)

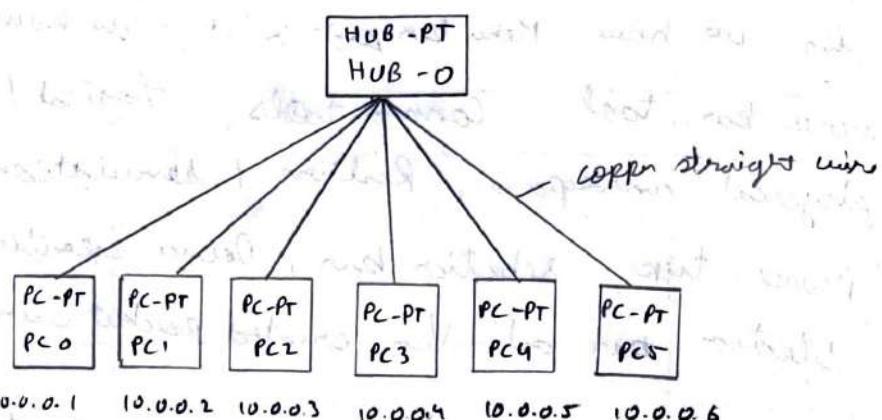
In admin panel you can disable access to a particular interface such as interface tabs, Interface locking etc.

Under Hide panel you can choose to show or hide phy, copy CCL, or UI & HTML, desktop. In fact you can change the font option. You can set user profiles from the menu bar.

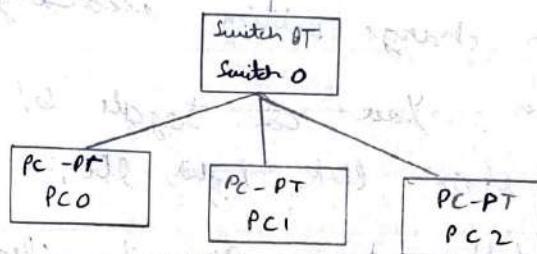
N
10/11/22 You have multiple Algorithms and you can save the project as PKZ file from the same as option.

Topology

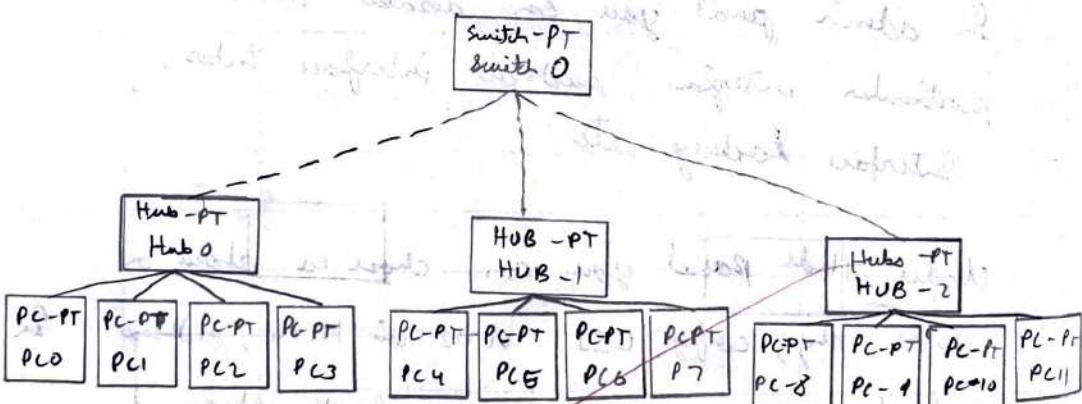
using HUB:



using Switch :



Using HUB and switch :



(HUB and switch)

Albatross with 29E has natural swimming & also (i)

Aim: creating a topology and simulate sending a simple PDU from one station to another via a switch for members of a working group

which has not thought the last words out
fully.

Step 1: A hub is connected to multiple PC's

With upper jaw raised at about 60 degrees (67)

Step 2: A simple AOU simulated from

within the *per1* and *PCBP1* genes, thereby adding to the complexity of the regulatory circuit.

wiehet ~~PC2~~ \rightarrow huisel \rightarrow ~~PC2~~ stop

Step 3: the two ad. PC's are connected to each other.

not needed external connection for IGBT soft turn-on (c)

Step 4: If the no. of parts is insufficient then add ^{Chamfer} extremes by clicking on the direction off the device and add the necessary parts ^{and last}

step 5: write the if's of all devices in a note
between the devices

Real time

Select a service PC and in the desktop tab select the command

~~Prompt option in command prompt type "ping 10.0.0.1"~~

Simulation Time

Wish will

Select a simple PC and select a source and destination computer. Clicking an auto option allows us to see few parts are transferred to and from device.

Wah

using switch

- i) add a generic switch and 3 PC to the ~~response~~ (obtained from Q&A)
 - ii) ~~Configure to IP address of each PC's in the configuration~~ to change that IP different for each device
 - iii) connect all PC's to the switch using copper cable
 - iv) If the port are insufficient then add extra ports by clicking + device. Turn off the device and necessary ports.
 - v) write the IP's of all devices in a table below the devices
- Real Time
- Select a source PC and both desktop tab set in command prompt option. In command prompt of type ping the destination by specifying its IP

Simulation Time

"Select it" a simple PC had select a source destination computer. Clicking on this option allows us to see how packets are transferred

Using hybrid

- i) Add a switch, 3 hubs and 12 PCs to workspace
- ii) connect the three hubs to the switch and 4 PCs to each of the hubs using copper cable

cross over ad copper straight through wires
respectively

- (v) configure the IP of each of the PC in config
and add a note below each PC containing IP
address

Real time mode:

Select the PC you want to send packet from. and
open its command prompt. specifying the
destination PC by specifying its IP address A
response is sent by the destination PC to
the source PC

simulation mode:

Add a single POU, by selecting the PC and
disk or notes option for right panel

Result

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 = 0ms

time = 0 ms

Reply from 10.0.0.3 : bytes = 32

time = 0 ms

Reply from 10.0.0.3 : bytes = 32

time = 0 ms

Reply from 10.0.0.3 : bytes = 32

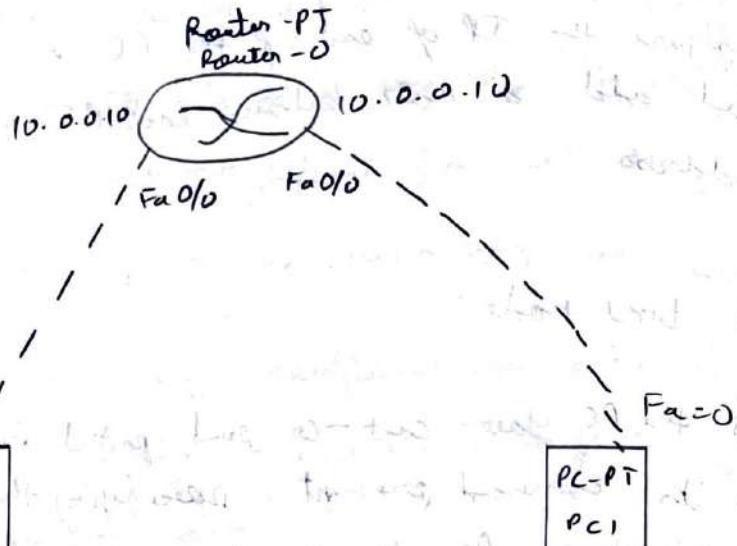
time = 0 ms

ping statistics POU 10.0.0.3

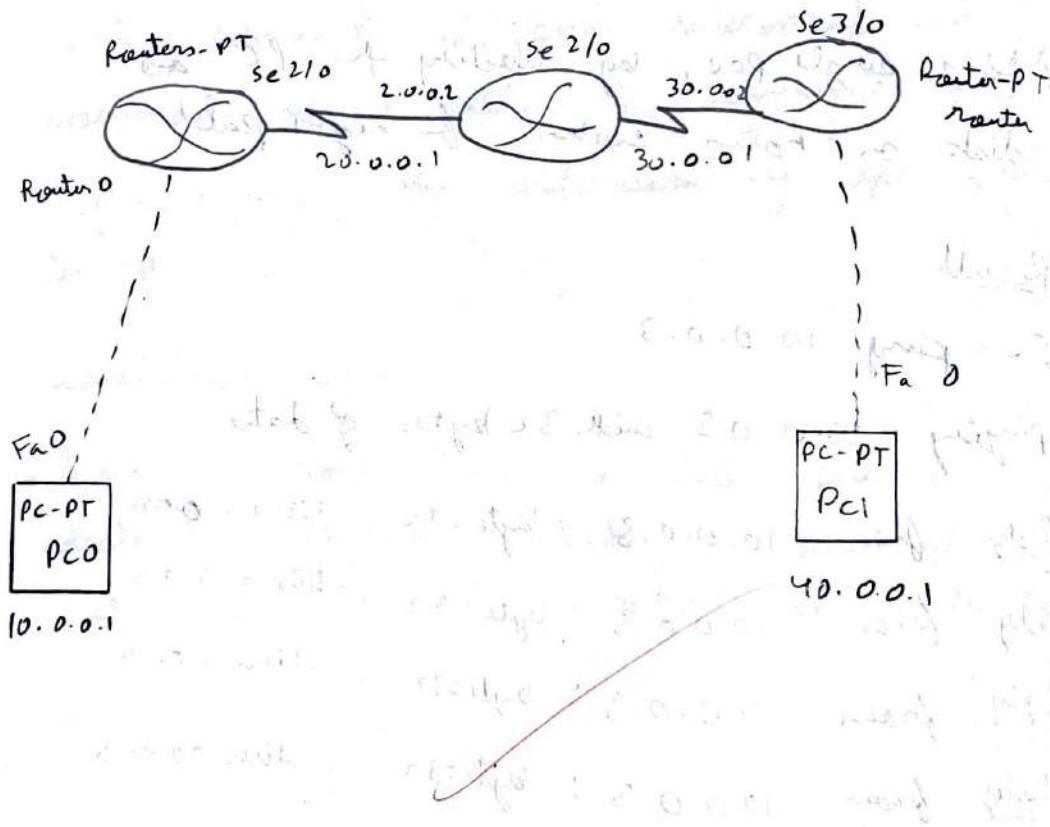
Packets sent = 0, received = 4, lost = 0

Topology

Single Raster



Final Topology



Lab 2

Routers

Aim: configuring IP address to router in packet tracer
explore the following messages ping responses
destination unreachable Request time out Reply

Procedure: single router

- add a monitor and two PC's to the workspace
- add a monitor and two PC's to the workspace
 - configure the IP address of each PC as 10.0.0.1 and 10.0.0.2 respectively and gateway of each PC to 20.0.0.2 respectively and gateway of each PC to 10.0.0.1 and 20.0.0.0 respectively connect the two PC's to the router using copper crossover cables
- click on the router gateway and type the following command to enable the interface

Router > enable

Router # config

Router (config) # interface fast ethernet 0/0

Router (config-if) # ip address 10.0.0.10 25.0.0.1

Router (config-if) # no shutdown

Router (config-if) # exit

Router (config) # interface fast ethernet 1/0

Router (config-if) # ip address 20.0.0.10 25.0.0.10

Router (config-if) # no shutdown

Router (config-if) # exit

Router (config) # exit

Router >

After entering these commands the lights b/w PC's and router were turned green ping PC, then PC's turn desktop-s command prompt

Final topology:

Add 3 routers and 2 PC's to the workspace as shown;
connect the router and PC with a copper crossover
cable and the 2 routers using a serial S0/E cable

configure IP address and gateway for both PC's
as 10.0.0.1, 010.0.0.10 and 40.0.0.1
40.0.0.10

click on the router 0 go to CLI and type the command

Router > enable

Router # config .f

Router (config) # interface Fast Ethernet 0/0

Router (config-if) # ip address 10.0.0.10 255.0.0.0

Router (config-if) # no shut

Router (config-if) # exit

Router (config-if) # interface serial 2/0

Router (config-if) # no shut

Router (config-if) # exit

Router (config-if) # exit

Router # exit

Router >

Configure router 2 similarly as Router 0 with IP's
of Fa 0/0 as 10.0.0.10 and Se 3/0
as 30.0.0.2

Configure Router 1 in CLI with both interfaces
as Sc 2/0 as Se 3/0 with IP's 20.0.0.2
and 30.0.0.1

The next step all the routers used to be configured to compile green command

Result: ~~Router 10~~

Result: single router

ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data
request final out

Reply from 20.0.0.1: bytes=32

time <1ms TTL = 127

Reply from 20.0.0.1: bytes=32

time <1ms TTL = 127

Reply from 20.0.0.1: bytes=32

time <1ms TTL = 127

ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1: bytes=32 time <1ms TTL = 127

Reply from 20.0.0.1: bytes=32 time <1ms TTL = 127

Reply from 20.0.0.1: bytes=32

time <1ms TTL = 127

Ping statistics 20.0.0.1

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

Final configuration

Result: ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

Reply from 40.0.0.1: bytes=32 time <1ms TTL = 127

Reply from 40.0.0.1: bytes=32 time <1ms TTL = 127

Reply from 40.0.0.1: bytes=32 time <1ms TTL = 127

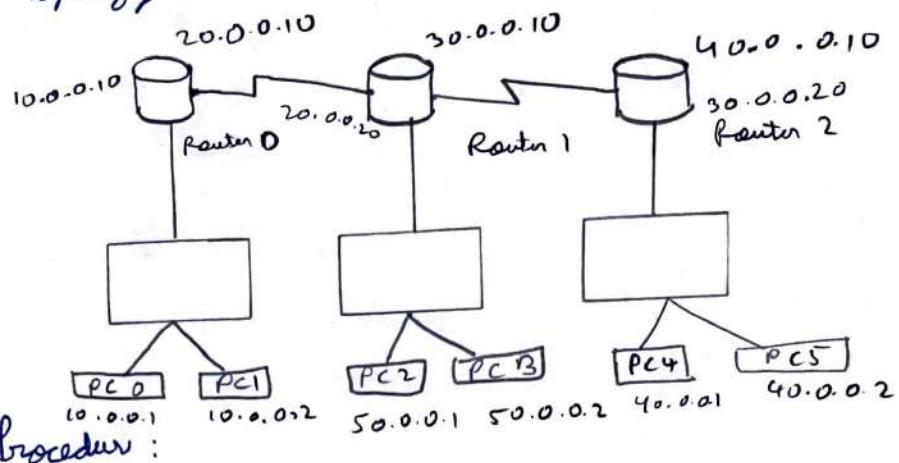
Reply from 40.0.0.1: bytes=32 time <1ms TTL = 127

Wdh11 Ping statistics for 40.0.0.1

Packets: sent 4 Received = 4 Lost = 0

Aim: Configuring default route to the router

Topology :



Procedure :

- ① Use 3 gen routers ad 6 gen PCs in the workspace along with 3 switches
- ② Place a net for each device and specify the IP address
- ③ Use copper straight wire to connect router ad switch
- ④ Use copper straight wires to connect switch and PC
- ⑤ click on a PC to set attributes for a PC and each PC has 3 attributes subnet mask, IP address ad gateway. This needs to be done for all 6 PCs
- ⑥ For Router 1 the config is done in the CLI.
The IP address ad subnet mask is set, Router 2 is a default router for Router 1 and this is done by command ip route 0.0.0.0 0.0.0.0 40.0.0.2
- ⑦ Router 2 IP address ad subnet masks are set for all 3 interfaces. It has static routing done by commands.

- ⑧ Router 3 is config'd with both interfaces with IP addresses and subnet mask. The default router for router 3 is router 2
- ⑨ Ping command is executed from 10.0.0.1 to 20.0.0.1 and from 10.0.0.1 to 30.0.0.2

Observation

- 1 router can have 2 default routers
- Default router for router 3 is also middle router
- Middle router get no default router because if a router is made default then there are chances of packets going to switch are sent to the router

Result

Ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data

Request timed out

Reply from 30.0.0.2 bytes = 32, time = 4 ms TTL = 125

Reply from 30.0.0.2 bytes = 32, time = 4 ms TTL = 125

Reply from 30.0.0.2 bytes = 32, time = 4 ms TTL = 125

~~Request timed out~~

~~Request timed out~~

0.0.0.222 5.0.0.1 test

WAB

time, source, etc. has to be carried in each

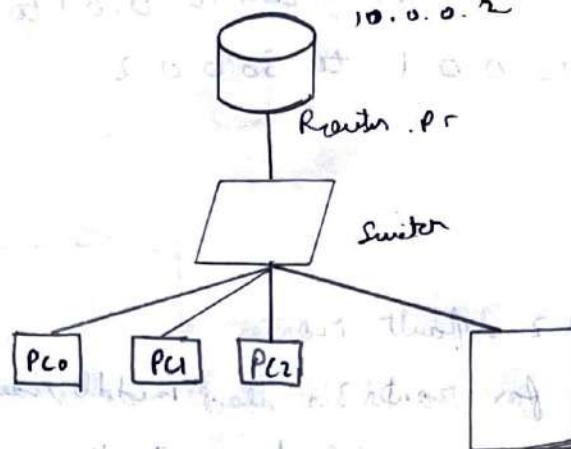
even housing number or something like that

down needs to be specified

Lab-4 about configuring DHCP within a LAN in a packet tracer

Aim: Configuring DHCP within a LAN in a packet tracer

Topology



Procedure

- ① Use one gen router one gen switch one gen server and 3 gen PCs in the workspace.
- ② Use copper straight cable to connect PCs and server.
- ③ Use fibre to connect switch and router.
- ④ Configure the server by adding IP address, subnet mask and gateway.
- ⑤ Configure the router by setting IP address, subnet masks and executing this command in CLI ip route 10.0.0.2 255.0.0.0
- ⑥ Click on server and go to services, select DHCP switch it on and add the gateway servers, IP address and subnet mask.

- ① click on the 1st pc and go to IP config. Select DHCP
Repeat this for all.
- ② ping command is executed from 10.0.0.3 to 10.0.0.5

Observation

pool of ip addresses exist from which IP address can be dynamically allocated.

This called Dynamic Host configuration

Protocol (DHCP)

Result

Ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data

Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

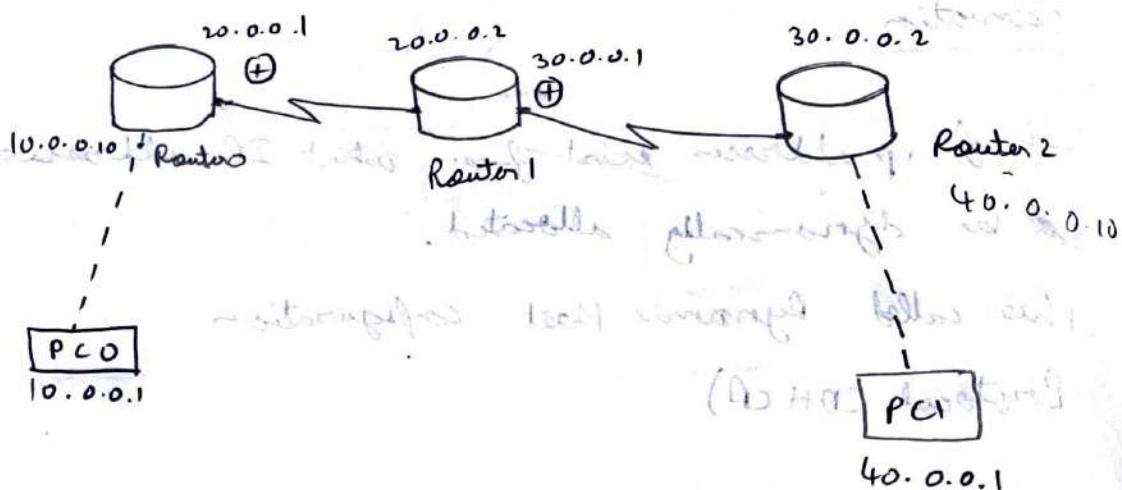
lping statistics for 10.0.0.5

packets sent = 4, received = 4, lost = 0

WAN

Aim: configuring RIP Routing Protocol in Routers

Topology:



Procedure

- Use 3 generic routers, 2 generic PC and place notes to indicate respective IP addresses
- Use serial OCE cables to connect routers and use copper cross cables to connect PC with router 1 and router 3
- Set IP address, gateway and subnet mask of
 - 10.0.0.1, 10.0.0.10, 255.0.0.0 for PC0
 - 40.0.0.1, 40.0.0.10, 255.0.0.0 for PC1
- Interface PC0 and Router 1
 - Interface fastethernet 0/0
 - IP address 20.0.0.10 255.0.0.0
 - no shut

- for interfacing serial 2/0 of router 1
 - interface serial 2/0
 - ip address 20.0.0.1 255.0.0.0
 - encapsulation PPP
 - clock rate 64000
 - no shutdown
- Use above commands for interfacing router which has clock symbol in cable. Then do it ad for other interfaces of routers using same above command except "clock rate 64000"
- Once all the lights are turned green follow the commands below each router
 - router rip
 - network 10.0.0.0
 - network 20.0.0.0
 - exit

- Repeat the same command for router 2 and router 3

~~Observation:~~

Use RIP routing becomes easy when large number of routers are present.

~~Result:~~

~~Pinging 10.0.0.1 with 32 bytes of data~~

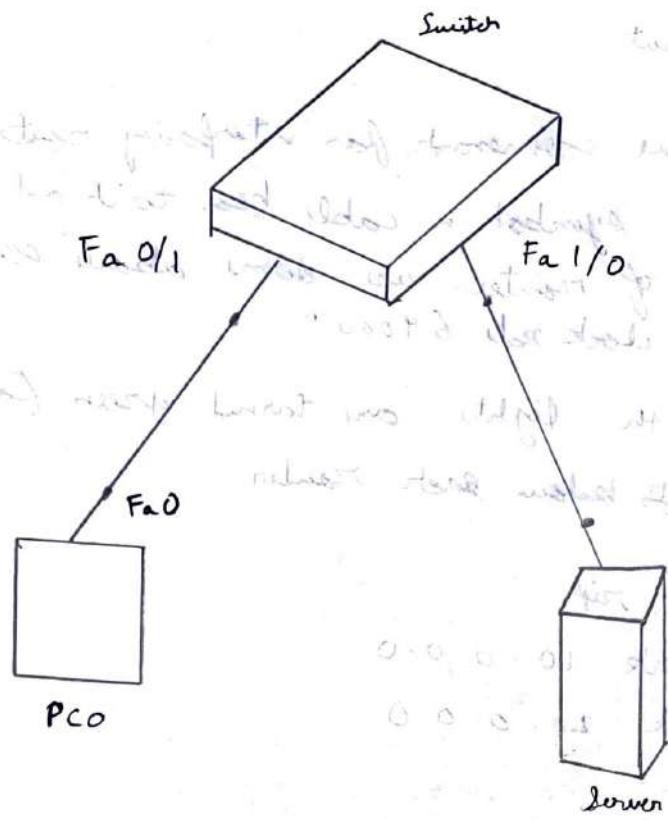
~~Reply from 10.0.0.1 byte = 32~~

~~Ping statistics for 10.0.0.1~~

~~Received: 4, sent = 4, lost = 0~~

Aim: Demonstration of Web server and DNS

Topology:



Procedure:

- Place a PC, Server and switch and then set the IP address of PC and server as 10.0.0.1 and 10.0.0.2.
- Open web browser from desktop tab of the PC and type "http://10.0.0.2"
- It will display a default page
- Open services and services enable HTTP and change or add the contents in the HTTP.
- After save refresh the browser of PC0 to the updated changes.

Activate DNS:-

- enable DNS on the servers to activate it.
- enter the name and the IP address needs to be mapped.
- click on add to add the new mapping.
- now give the name in the web browser you saved to check if its working.

Custom page:-

- create a new page test.html and save it in http://server
- change hyperlink in index.html to link to created file
- check the output in the web browser of the PC by clicking on hyper link

Observations:-

- We can view the web page when we type 'caebmsce' in browser because 10.0.0.2 address is mapped to the name 'caebmsce'.
Mapping is required because its difficult for users to remember IP's. Hence it is mapped.

Result:-

Web Browser		Last visited
<URL>		http://cn/resume.html
Name	Method	*
age	20	
college	BMSCE	
experience	2 years	

Aim: Write a program for error detection for CRC with the receiver end.

```
#include <stdio.h>
#include <string.h>
#define N 8
#define gen-poly "1001100011000111"
char data[28];
char check-value[28];
char gen-poly[16];
int data-length, i, j;

void XOR() {
    for (j = 1; j < N; j++) {
        check-value[j] = ((check-value[j] == gen-poly[j]) ? '0' : '1');
    }
}

void receiver() {
    printf("Enter the received data: ");
    scanf("%s", data);
    printf("\n-----\n");
    printf("Data received: %s", data);
    CRC();
    for (i = 0; i < N - 1; i++) {
        if (check-value[i] != '1') {
            printf("\n Error Detected \n\n");
        } else {
            printf("\n No error detected \n\n");
        }
    }
}

int CRC() {
    int i, j;
    for (i = 0; i < data-length; i++) {
        if (data[i] == '1') {
            for (j = 0; j < N; j++) {
                if (check-value[j] == '1') {
                    check-value[j] = '0';
                } else {
                    check-value[j] = '1';
                }
            }
        }
    }
}
```

```

void crc () {
    for (i=0; i < N; i++)
        check_value [i] = data[i];
    do {
        if (check_value [0] == '1')
            Kroc ();
        for (j=0; j < N-1; j++)
            check_value [j] = check_value [j+1];
        check_value [j] = data [i+j];
    } while (i < data.length + N - 1);
}

int main () {
    printf ("Enter data to be transmitted:");
    scanf ("%s", data);
    printf ("Enter the gen polynomial:");
    scanf ("%s", gen_poly);
    data_length = strlen (data);
    for (i = data_length; i < data_length + N - 1; i++)
        data [i] = '0';
    printf ("\n-----");
    printf ("\n Data padded with n-1 zeros : %s", data);
    printf ("\n-----");
    crc ();
    printf ("\n CRC or check Value is : %s", check_value);
    for (i = data_length; i < data_length + N - 1; i++)
        data [i] = check_value [i - data_length];
    printf ("\n-----");
    printf ("\n Final data to be sent : %s", data);
    printf ("\n-----\n");
    receiver ();
    return 0;
}

```

output

enter the data: 10001000000100000

enter poly: 1011101

Data padded with n-1: 10001000000100001000000

CRC or check value: 0100110001

Final data set: 1000100000100001010011

enter the received data: 10001000000100001010011

Data received: 10001000000100001010011

No error detected

Wait

Lab - 8

```
# include <iostream> std::cout < iostream>
# include <iomanip.h>
using namespace std;
```

```
# define bufferSize 500
```

```
void bucketInput(int a, int b) = for testing
```

```
{ if (a > bufferSize) { cout << "Input too large"; }
```

```
cout << "Initiate Bucket overflow";
```

```
else {
```

```
sleep(5); // sleep for 5 sec
```

```
while(a > b) {
```

```
cout << "Initiate Bucket overflow";
```

```
cout << "Initiate Bucket overflow";
```

```
a -= b; // SPP = 100 bytes taken
```

```
sleep(5); // Initiates sleep for 5 sec
```

```
}
```

```
if (a > 0) {
```

```
cout << "Initiate Bucket overflow";
```

```
cout << "Initiate Bucket overflow";
```

```
}
```

```
int main()
```

```
{
```

```
int op, pktSize;
```

```
cout << "Enter output rate: ";
```

```
in >> op;
```

```
for (int i = 1; i <= s; i++)
```

```
{
```

```
sleep(rand() % 10);
```

```
pktSize = rand() % 700;
```

```
cout << "Initiate Bucket overflow";
```

```
size = << pktSize;
```

```
Bucket Input(pktSize, up?);
```

3
sent cc and;
return 0;
}

Output

Enter output rate: 100

Packet no 1

Packet size = 186

100 bytes outputted

Last 86 bytes sent

Bucket output successful

Packet no 3

Packet size = 535

Bucket overflow

Packet no 4

Packet size = 492

100 bytes outputted

100 bytes outputted

100 bytes outputted

100 bytes outputted

Packet no 5

Packet size = 521

Bucket overflow

```

#include < stdio.h>
#include < stdlib.h>

int Bellman_Ford (int g[20][20], int V, int E, int edge[20][2])
{
    int i, u, v, k, distance[20], parent[20], s, flag = 1;
    for (i = 0; i < V; i++)
        distance[i] = 1000, parent[i] = -1;
    printf ("Enter source: ");
    scanf ("%d", &s);
    distance[s - 1] = 0;
    for (i = 0; i < V - 1; i++)
    {
        for (k = 0; k < E; k++)
        {
            u = edge[k][0], v = edge[k][1];
            if (distance[u] + g[u][v] < distance[v])
                distance[v] = distance[u] + g[u][v], parent[v] = u;
        }
    }
    for (k = 0; k < E; k++)
    {
        u = edge[k][0], v = edge[k][1];
        if (distance[u] + g[u][v] < distance[v])
            flag = 0;
    }
    if (flag)
    {
        for (i = 0; i < V; i++)
            printf (" vertex %d / -> cod = %d \n", i + 1, distance[i]);
        getch();
    }
}

```

Output :

Buffer size = 4 out of buffer size $= 10$

Buffer size = 7 out of buffer size $= 10$

Buffer size $= 10$ out of buffer size $= 10$

Bucket less ≤ 4 means total number of elements ≤ 4

Buffer size = 4 out of buffer size $= 10$

Bucket less ≤ 7 means total number of elements ≤ 7

Buffer size = 7 out of buffer size $= 10$

Bucket less ≤ 10 means total number of elements ≤ 10

Bucket less ≤ 10 means total number of elements ≤ 10

Bucket less ≤ 10 means total number of elements ≤ 10

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

$\{0\} \cup \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$

Socket programming

client by

from socket import *

server Name = "Desktop-HMP001C"

server Port = 12530

client socket = socket (AF_INET, SOCK_STREAM)

client socket. connect ((server Name, server Port))

print = input ("Enter file name")

client socket. send (server. encode())

file contents = client socket. recv (1024). decode()

print ("From Server : ", file contents)

client socket. close()

from socket import *

server Name = "127.0.0.1"

server Port = 12000

client socket = socket (AF_INET, SOCK_DGRAM)

value = input ("Enter delivery")

client socket. sendto (bytes (value), ("127.0.0.1", 12000))

(server Name, server Port))

file contents, server Address = client socket. recvfrom (2048)

print ("From Server : ", file contents)

client socket. close()

server of .py

from socket import *

server Port = 12000

server socket = socket (AF_INET, SOCK_STREAM)

server socket.bind (("127.0.0.1", server Port))

print ("The Server is ready to receive")

while 1:

Server . client Address = Server Socket read from (2048)

file = open ("server", "r")

I = file.read(2048)

Server Socket . sendto (bytes('1', "WFO")) client
put ("sent back to client", 1)

file.close()

from socket import *

serverName = 'DESKTOP-1MPODEC'

serverPort = 12530

serverSocket = socket (AF_INET, SOCK_STREAM)

serverSocket.bind (serverName, serverPort)

serverSocket . listen (5)

print ("the sever is ready to receive")

client, addr =

serverSocket . accept ()

server = socket.socket (socket.AF_INET, socket.SOCK_STREAM)

file = open ("server", "r")

file.read(1024)

server . send (file . encode ())

file . close ()

server . close ()

HTTP request

"http://127.0.0.1:5000"

"GET / HTTP/1.1"

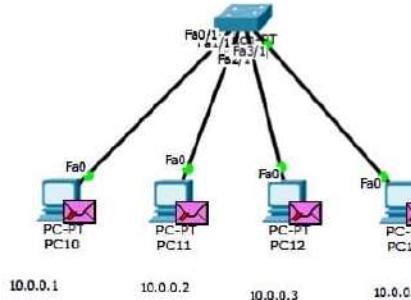
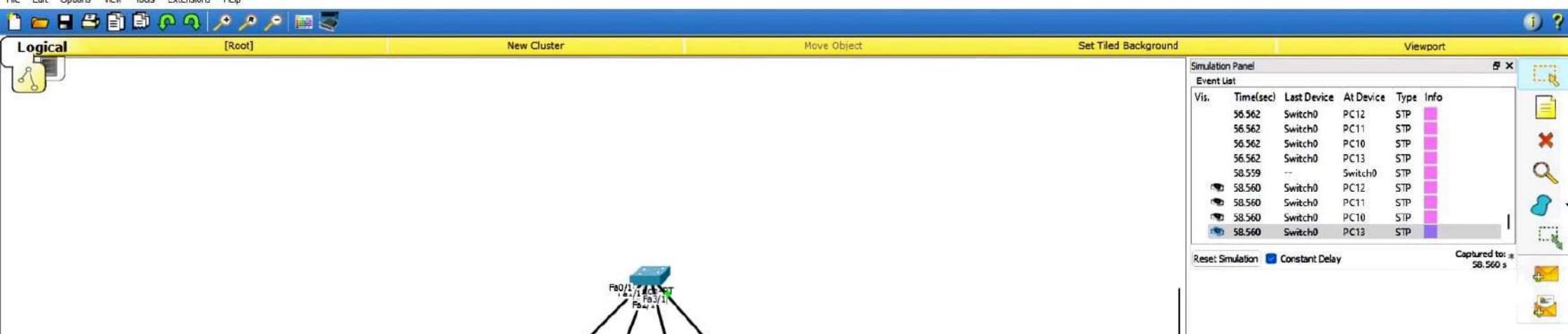
"Host: 127.0.0.1:5000"

"User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.113 Safari/537.36"

"Accept: */*

Output :

Server started
waiting for a client ...
connected
Server started
waiting for a client ...
client accepted



Simulation Panel					
Event List					
Vis.	Time(sec)	Last Device	At Device	Type	Info
	56.562	Switch0	PC12	STP	
	56.562	Switch0	PC11	STP	
	56.562	Switch0	PC10	STP	
	56.562	Switch0	PC13	STP	
	58.559	--	Switch0	STP	
⌚	58.560	Switch0	PC12	STP	
⌚	58.560	Switch0	PC11	STP	
⌚	58.560	Switch0	PC10	STP	
⌚	58.560	Switch0	PC13	STP	

Constant Delay

Captured to: ...
FF_FFD...

Play Controls

Back	Auto Capture / Play	Capture / Forward
------	---------------------	-------------------

Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDR, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, RADIUS, RIP, RIPng, RTP, SCOP, SMTP, SNMP, SSH, STP, SVI, TACACS+, TCR, TFTP, Tunnel, UDPM, VTR

[Edit Filters](#)

Show All/None

Time: 01:31:33.704 Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC9	PC7	ICMP	■■■■■	0.000	N	0	(edit)	(delete)
●	Successful	PC10	PC11	ICMP	■■■■■	0.000	N	1	(edit)	(delete)



Logical

[Root]

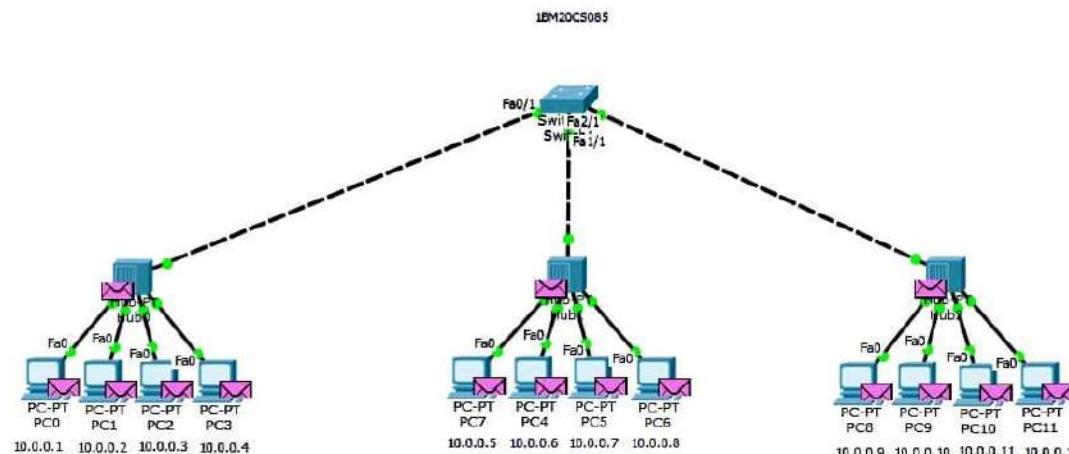
New Cluster

Move Object

Set Tiled Background

Viewport

?



Simulation Panel					
Event List					
Vis.	Time(sec)	Last Device	At Device	Type	Info
█	15.835	Hub0	PC3	STP	
█	15.835	Hub1	PC7	STP	
█	15.835	Hub1	PC4	STP	
█	15.835	Hub1	PC5	STP	
█	15.835	Hub1	PC6	STP	
█	15.835	Hub2	PC8	STP	
█	15.835	Hub2	PC9	STP	
█	15.835	Hub2	PC10	STP	
█	15.835	Hub2	PC11	STP	

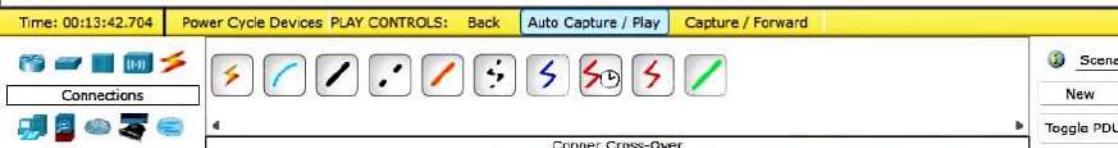
Reset Simulation Constant DelayCaptured to: *
15.835 s

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, SCOP, SMTP, SNMP, SSM, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Edit Filters

Show All/None

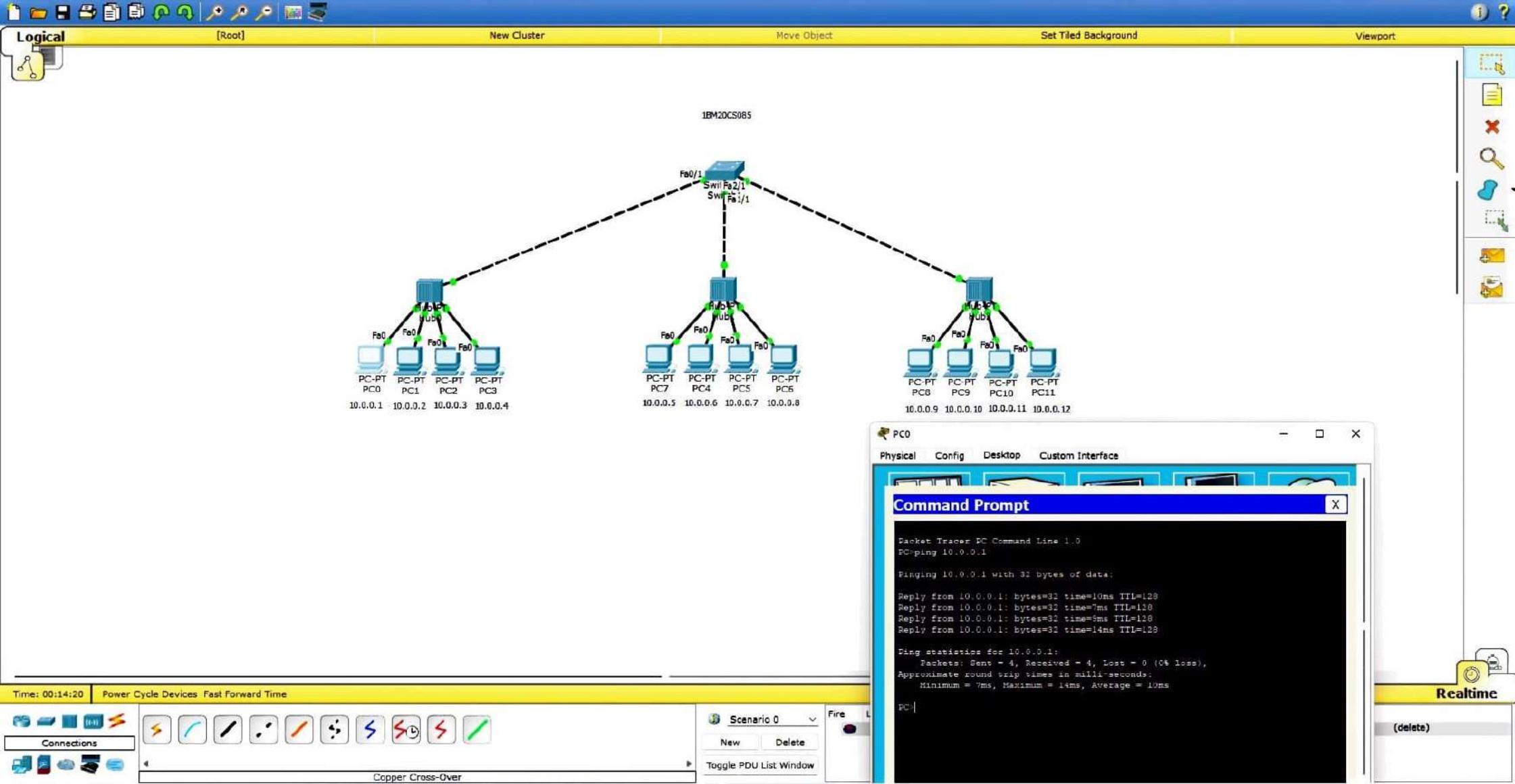


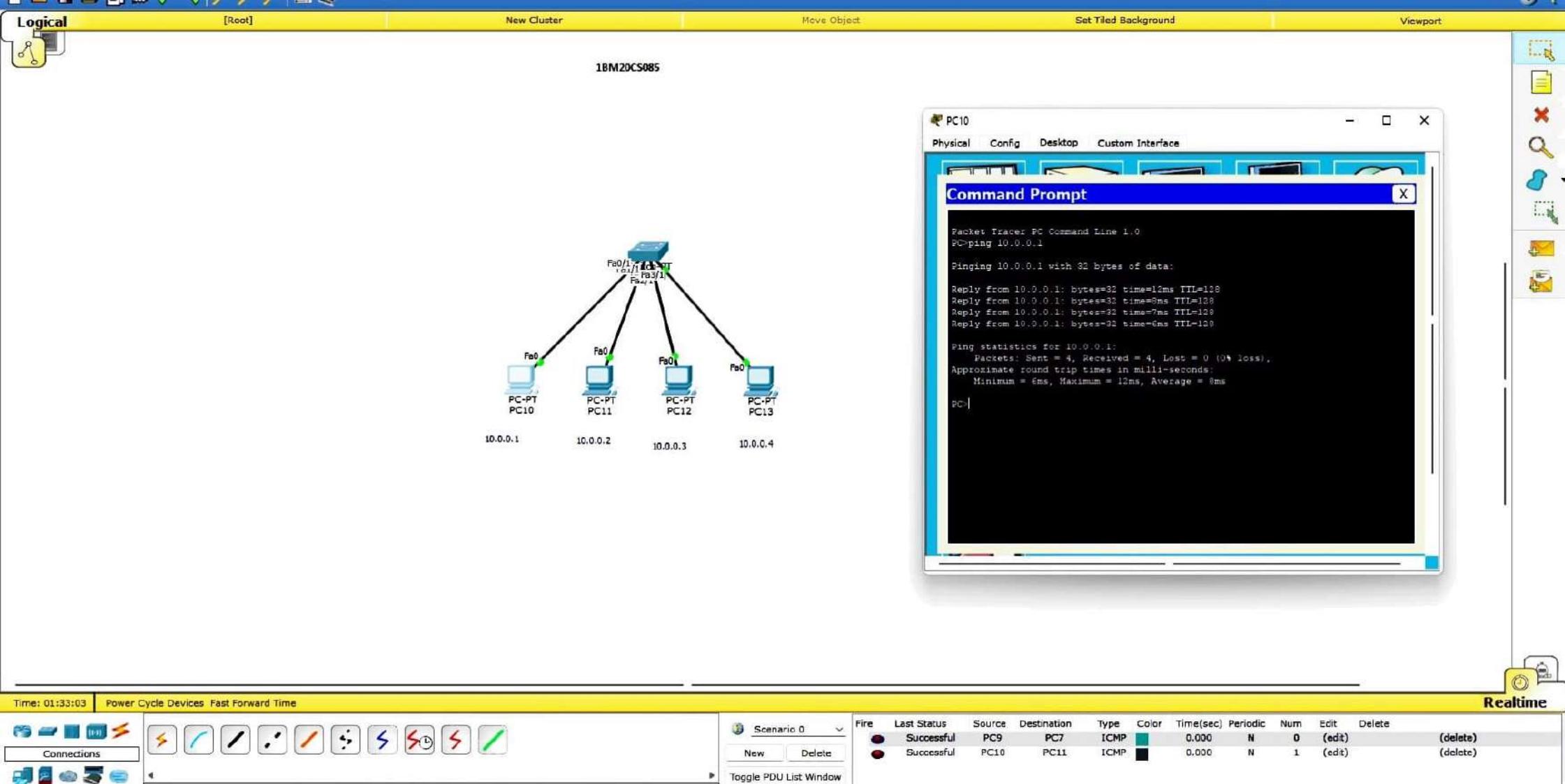
Scenario 0	Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	●	Successful	PC0	PC1	ICMP	█	0.000	N	0	(edit)	(delete)
<input type="button"/> Scenario 0											
<input type="button"/> New <input type="button"/> Delete											
<input type="button"/> Toggle PDU List Window											

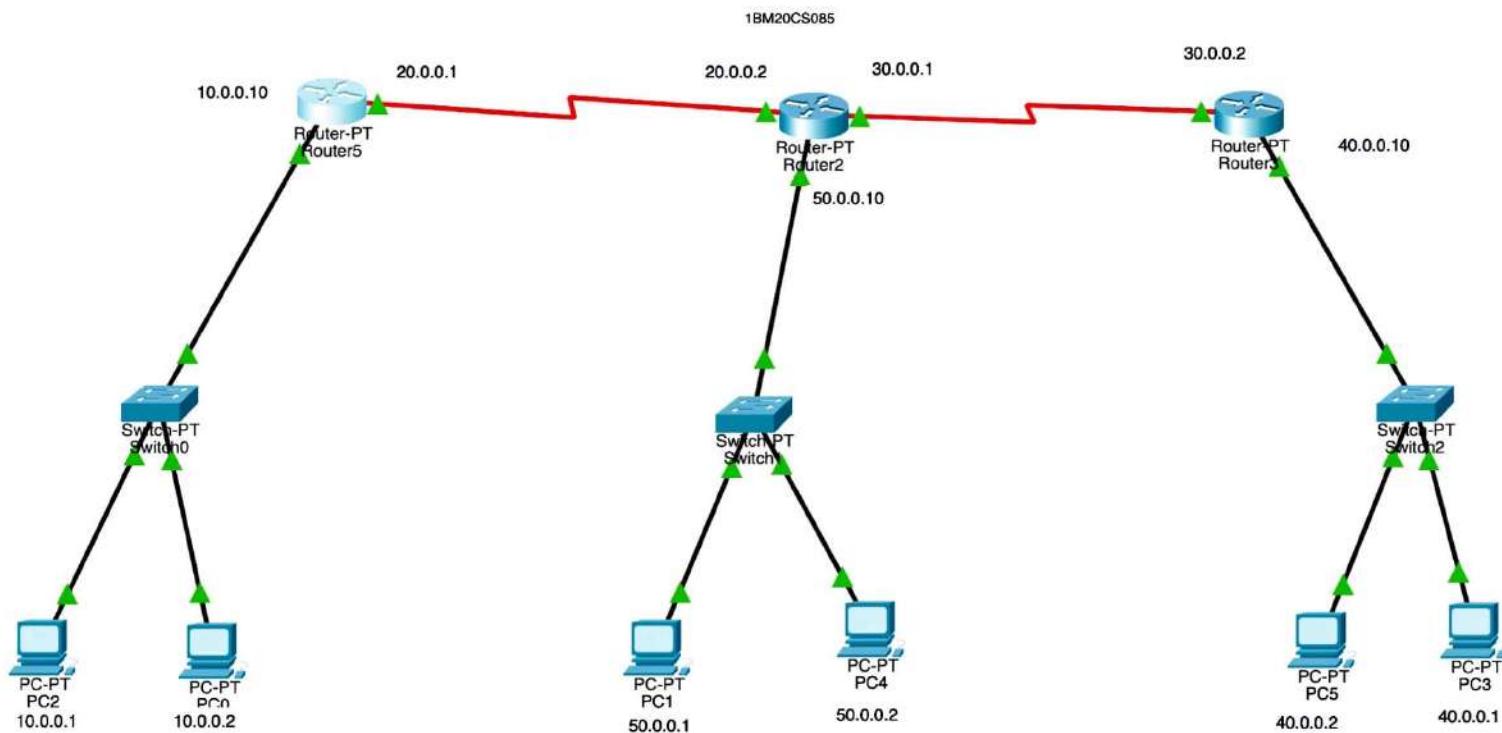


Event List

Simulation







Time: 00:01:39

Realtime Simulation

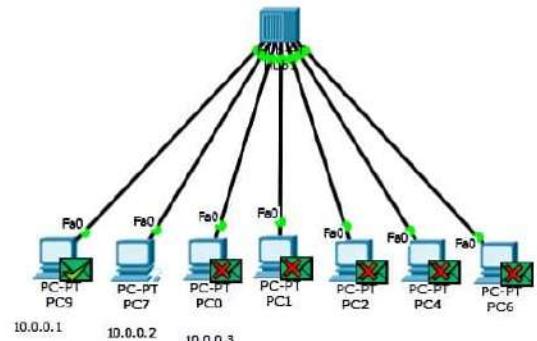


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Scenario 0	--	PC2	PC1	IC...	0.000	N	0	(...)	(delete)
		--	PC0	PC4	IC...	0.006	N	1	(...)	(delete)

(Select a Device to Drag and Drop to the Workspace)



1BM20CS085



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
0.002		Hub1	PC4	ICMP	
0.002		Hub1	PC6	ICMP	
0.003		PC7	Hub1	ICMP	
0.004		Hub1	PC9	ICMP	
0.004		Hub1	PC0	ICMP	
0.004		Hub1	PC1	ICMP	
0.004		Hub1	PC2	ICMP	
0.004		Hub1	PC4	ICMP	
0.004		Hub1	PC6	ICMP	

Reset Simulation

 Constant Delay

Capturing...

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, SCOP, SMTR, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Edit Filters

Show All/None

Time: 00:50:07.284

Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward

Event List

Simulation

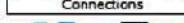


Scenario 0

New Delete

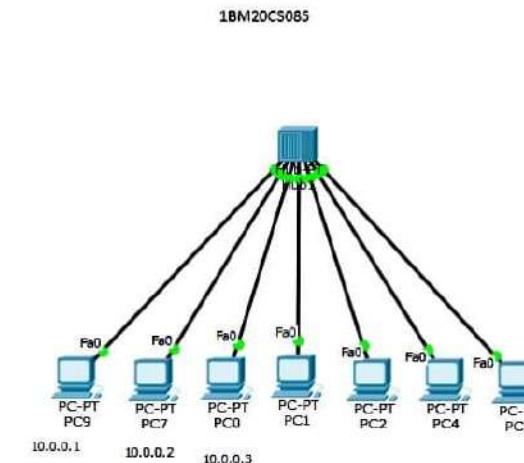
Toggle PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC9	PC7	ICMP	■	0.000	N	0	(edit)	(delete)



Copper Straight-Through

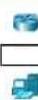




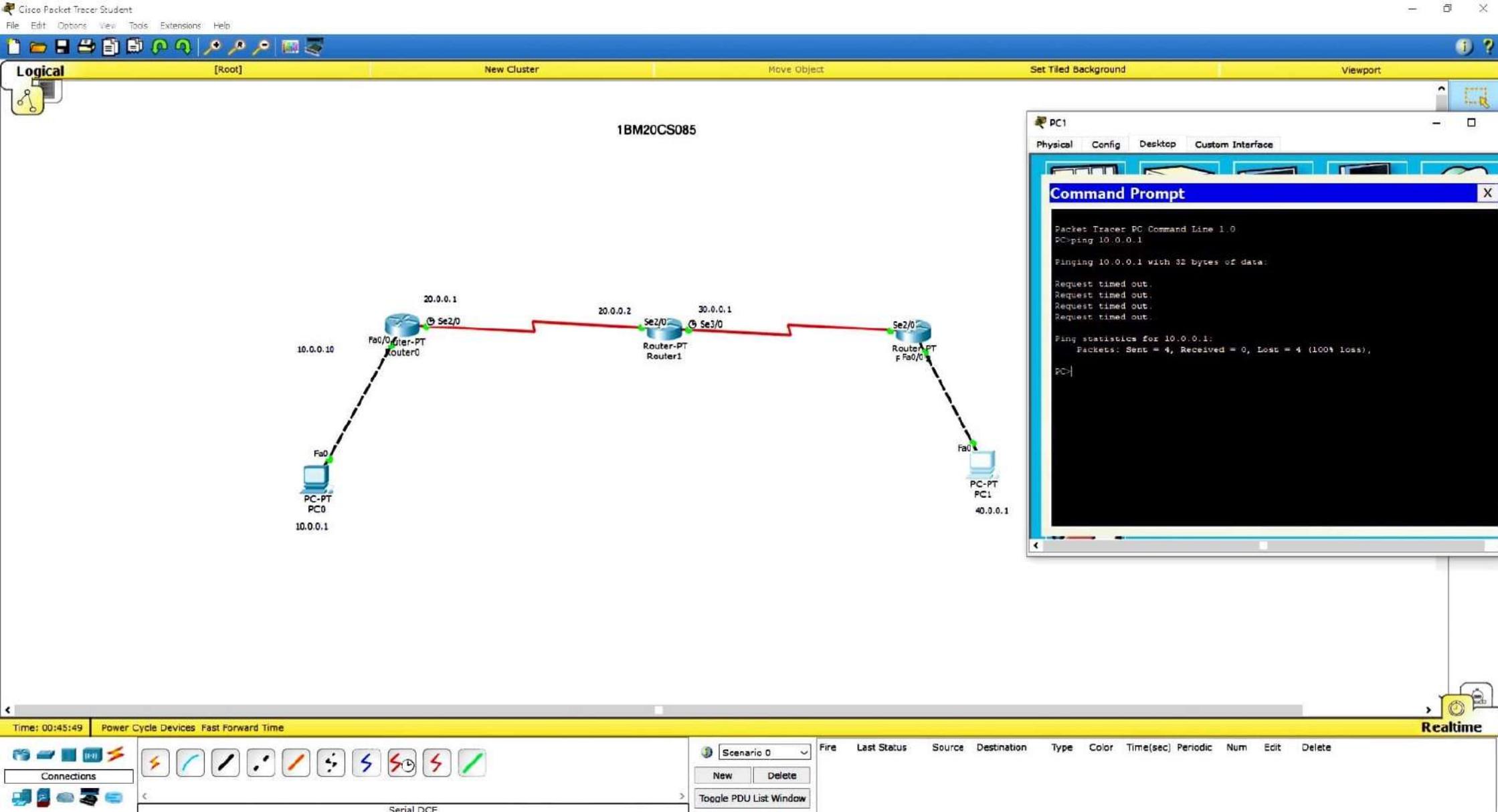
Time: 01:20:52

Power Cycle Devices Fast Forward Time

Realtime



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC9	PC7	ICMP	■■■■■	0.000	N	0	(edit)	(delete)





Logical

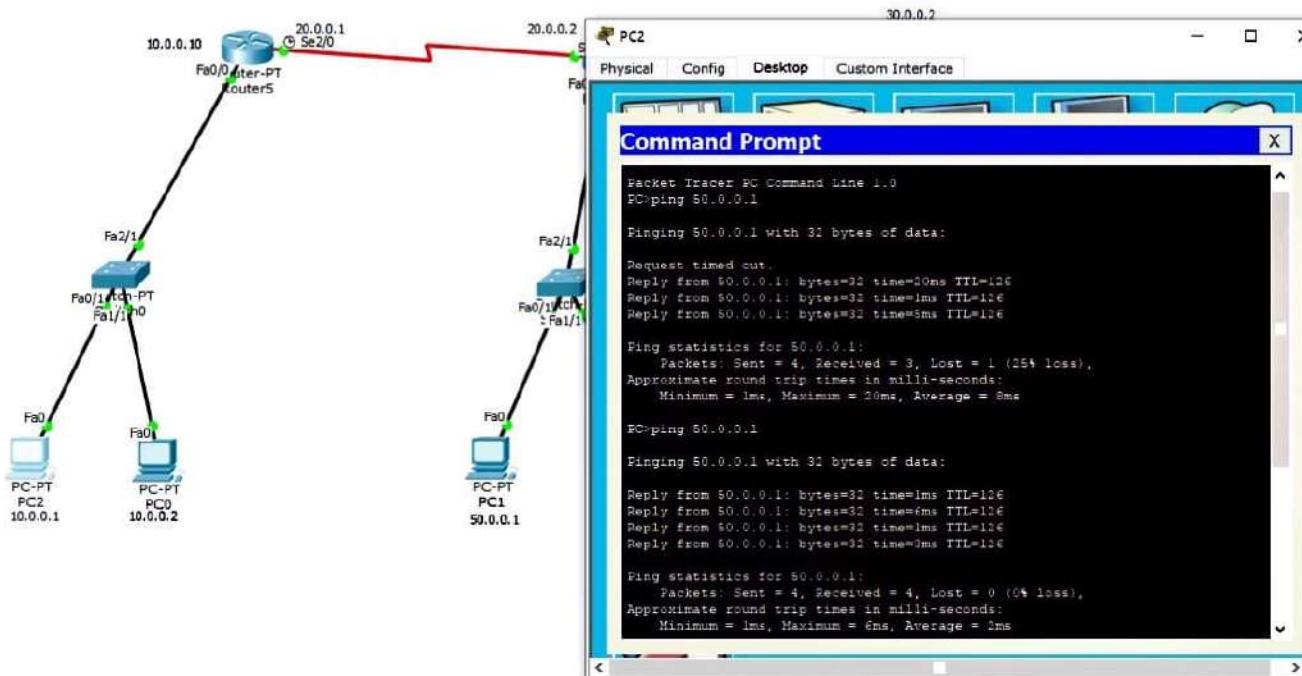
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Time: 00:47:33

Power Cycle Devices Fast Forward Time



Realtime

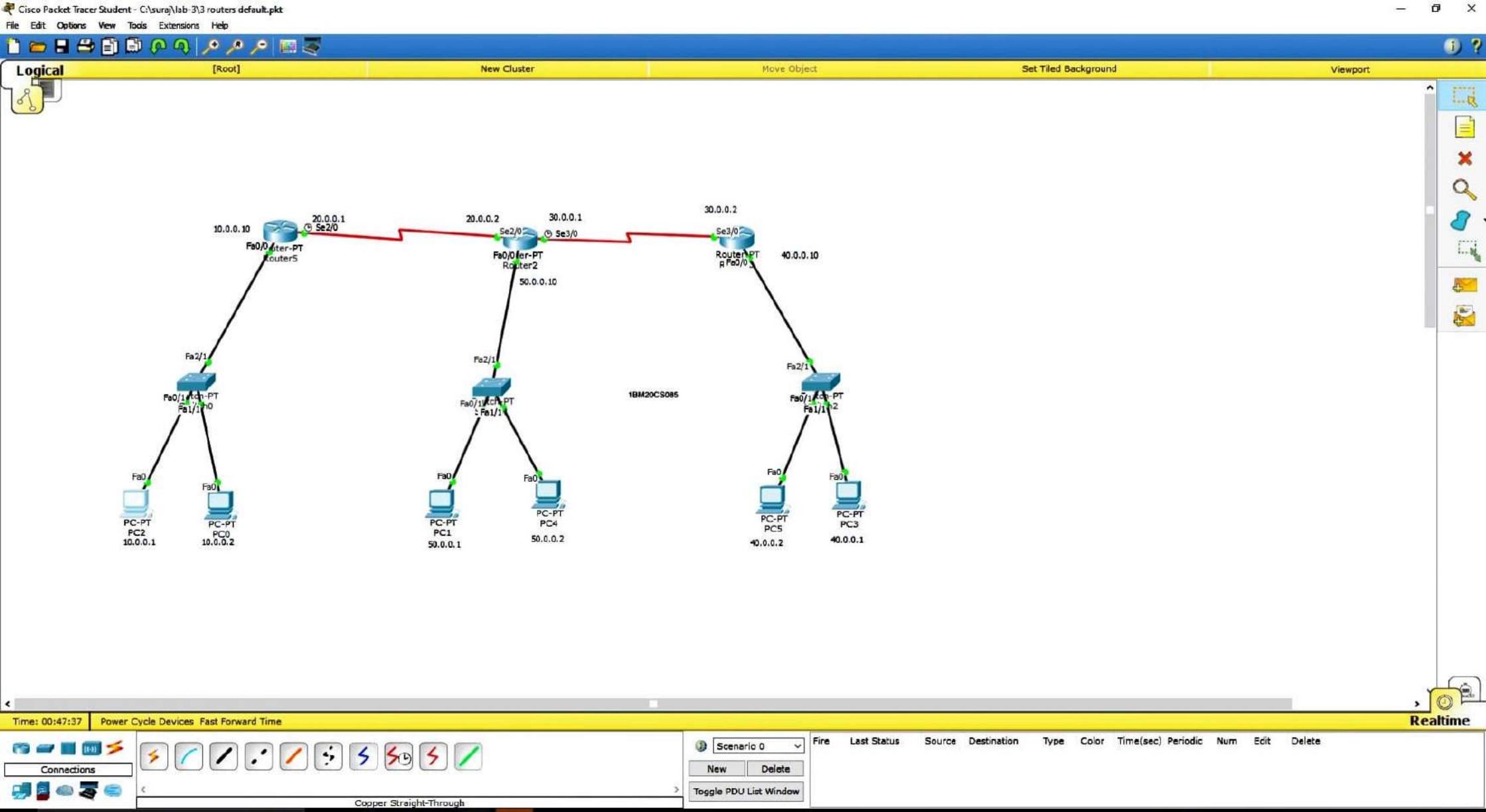


Scenario 0	Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
New	Delete										

Connections

Copper Straight-Through

Toggle PDU List Window





Logical

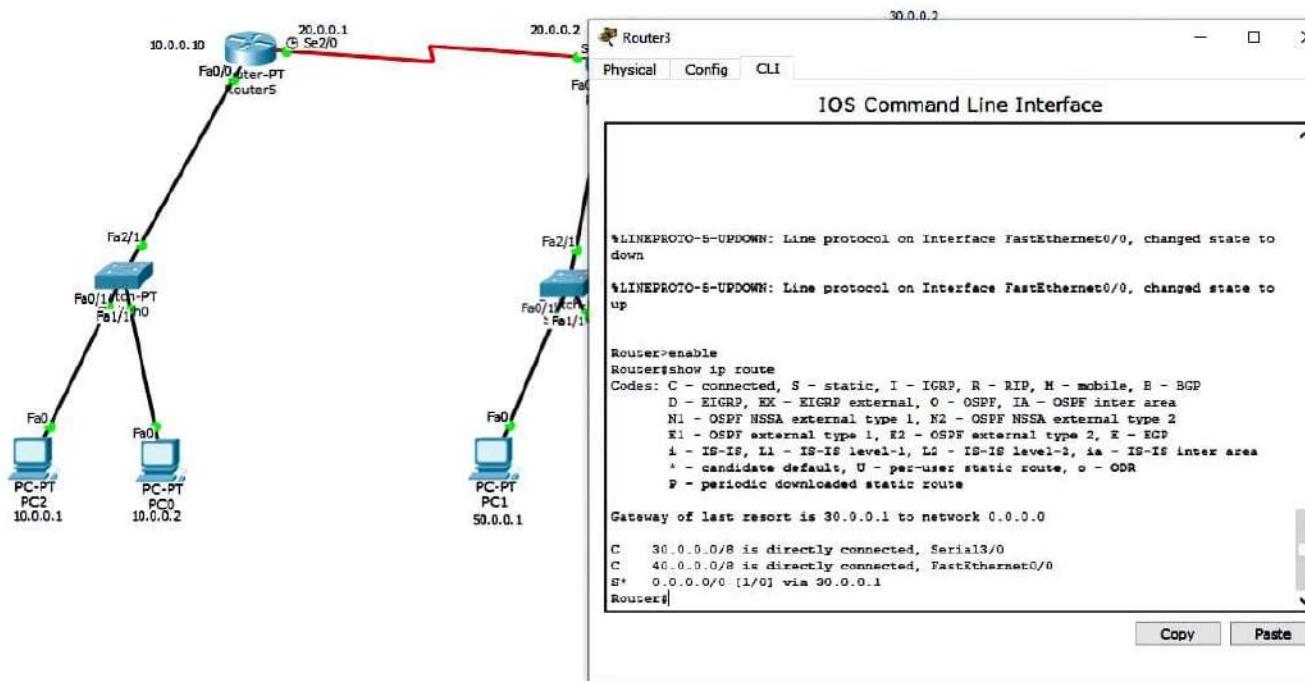
[Root]

New Cluster

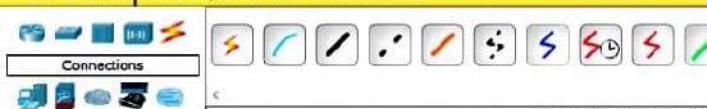
Move Object

Set Tiled Background

Viewport



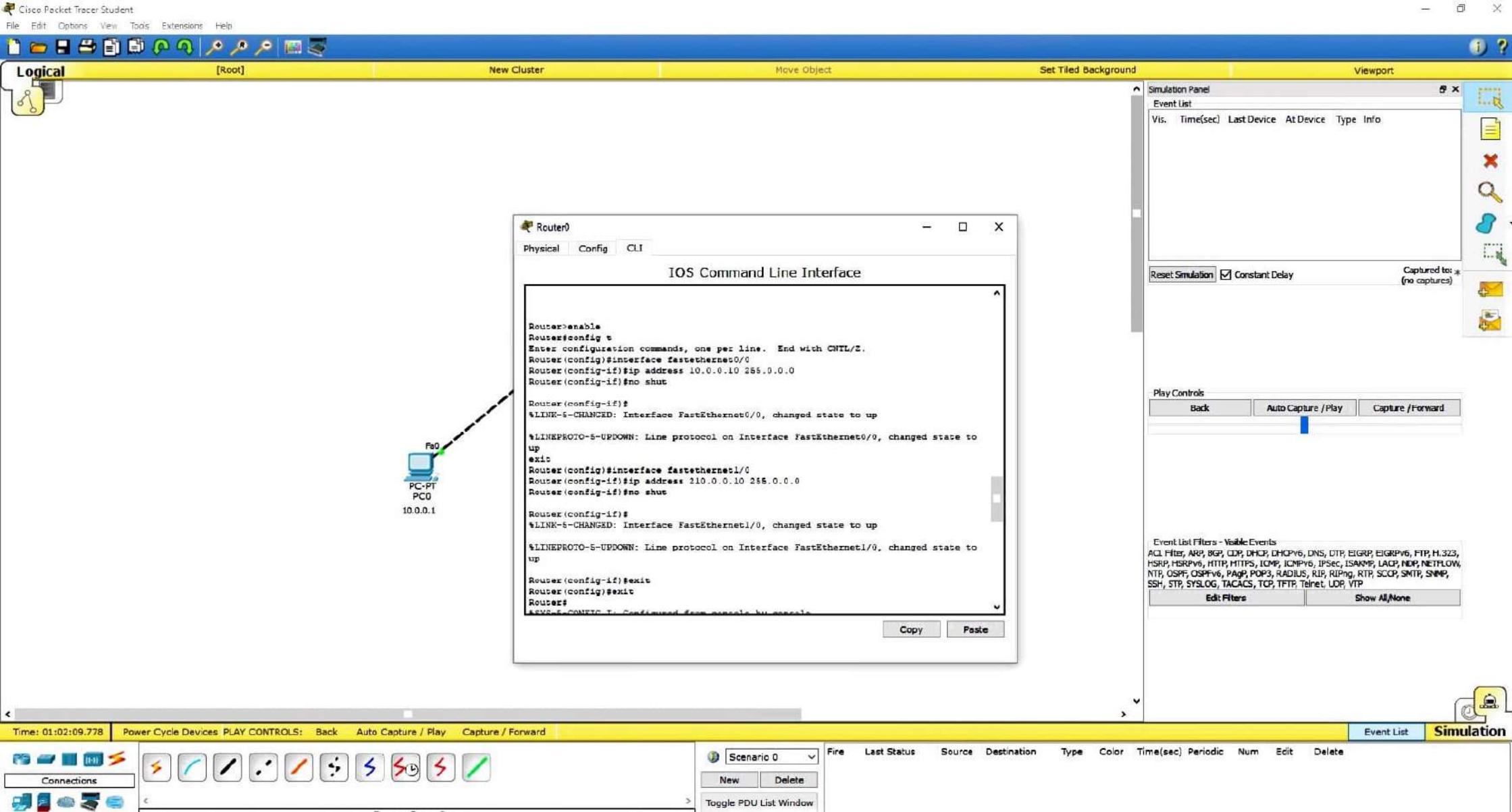
Time: 00:47:13 Power Cycle Devices Fast Forward Time



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

Toggle PDU List Window







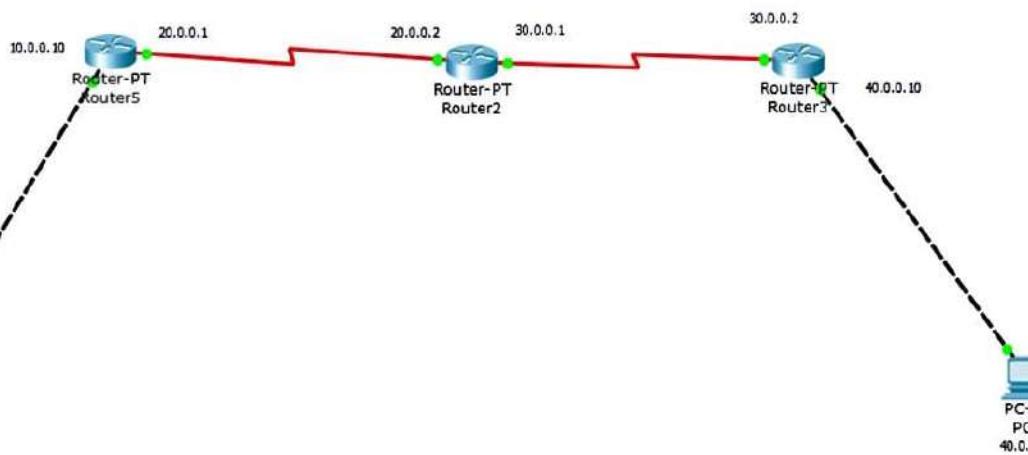
[Root]

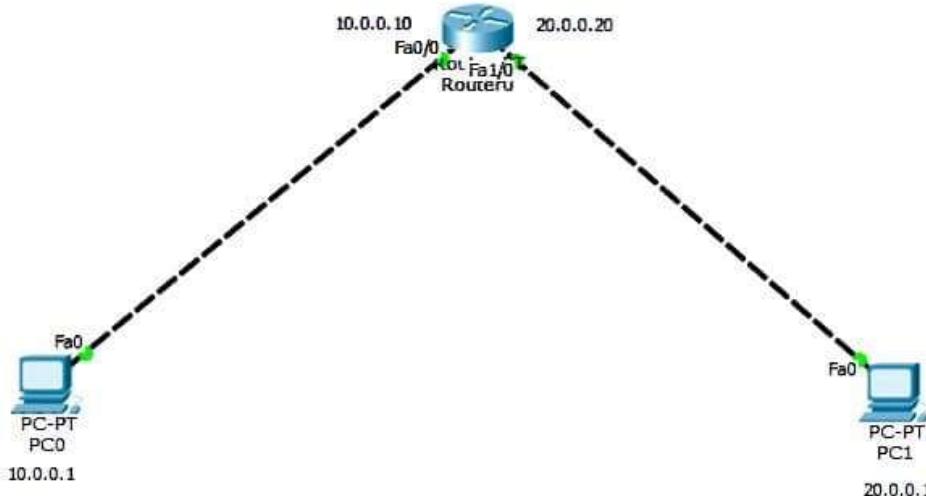
New Cluster

Move Object

Set Tiled Background

Viewport





PC1

Physical Config Desktop Custom Interface

Command Prompt

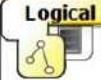
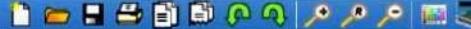
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=0ms TTL=127

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

PC>
```



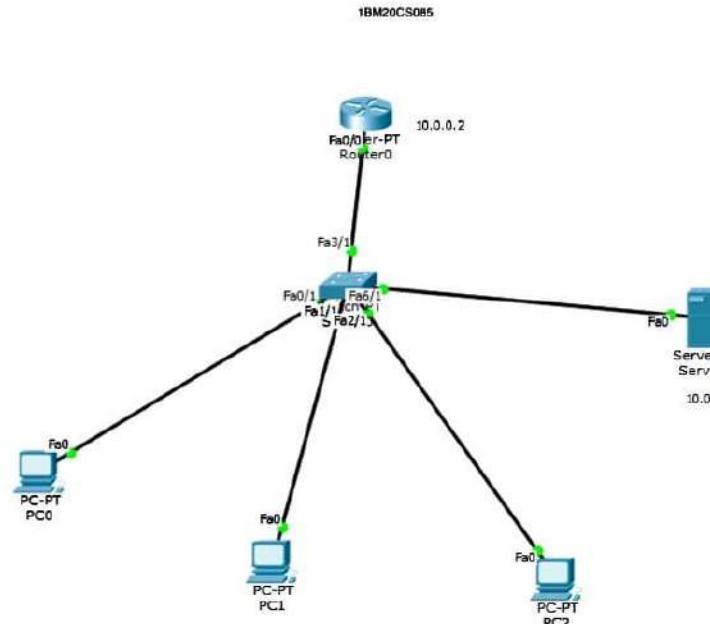
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Time: 00:17:38

Power Cycle Devices Fast Forward Time

Realtime



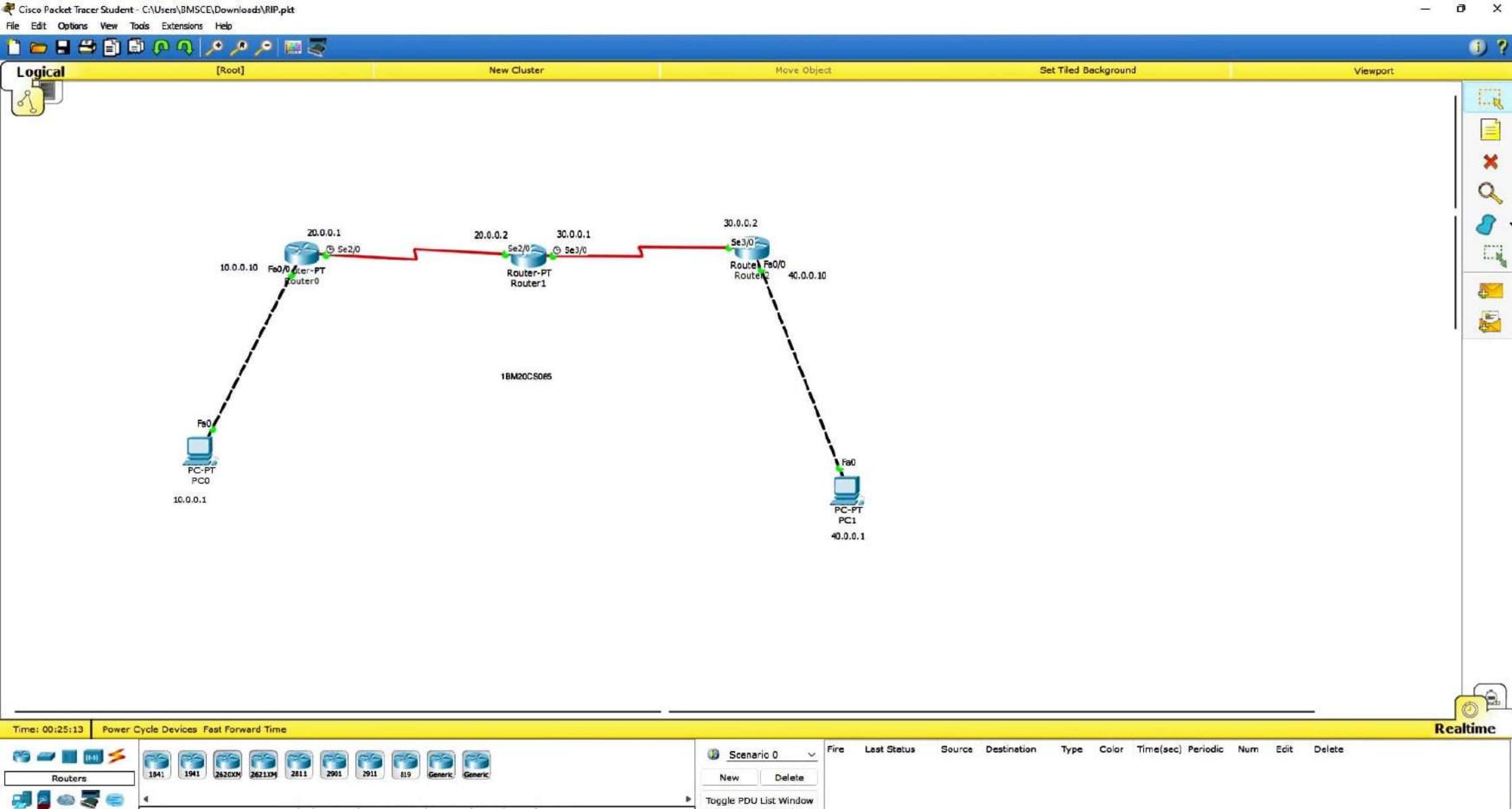
Scenario 0

New Delete

Toggle PDU List Window

4

(Select a Device to Drag and Drop to the Workspace)





Logical

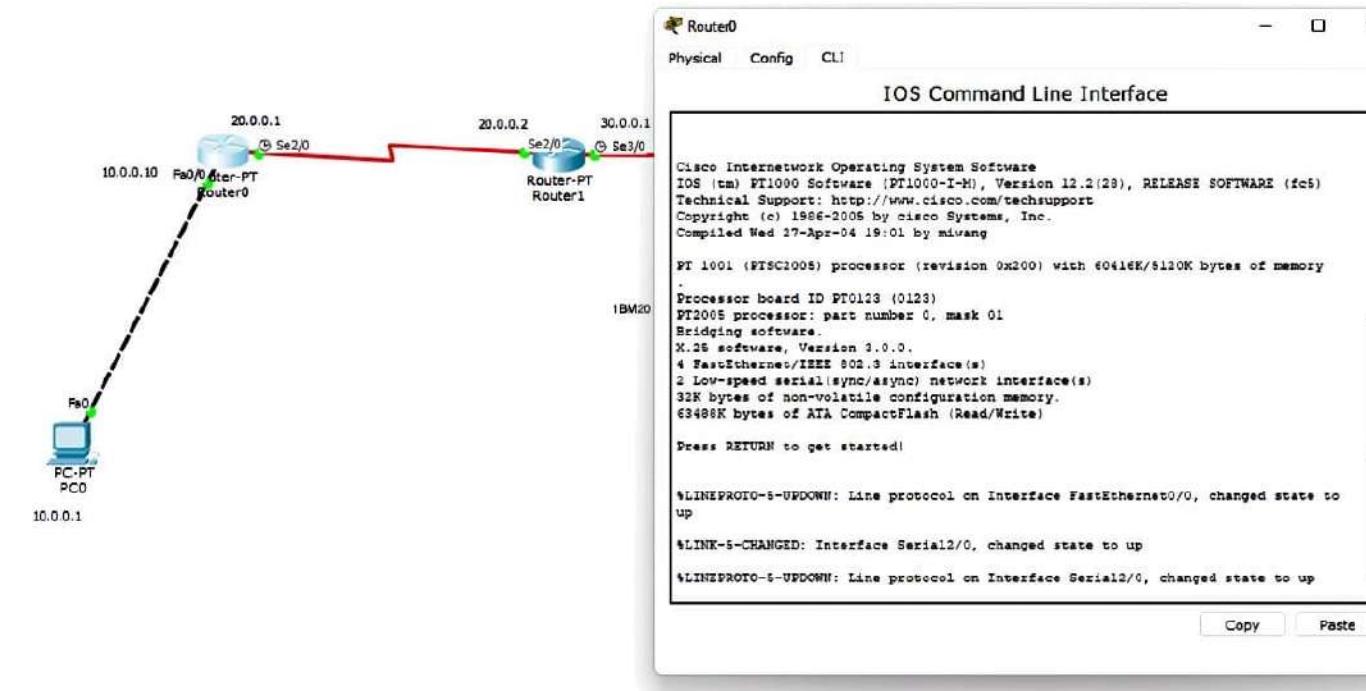
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Time: 00:26:38 Power Cycle Devices Fast Forward Time

Realtime



(Select a Device to Drag and Drop to the Workspace)



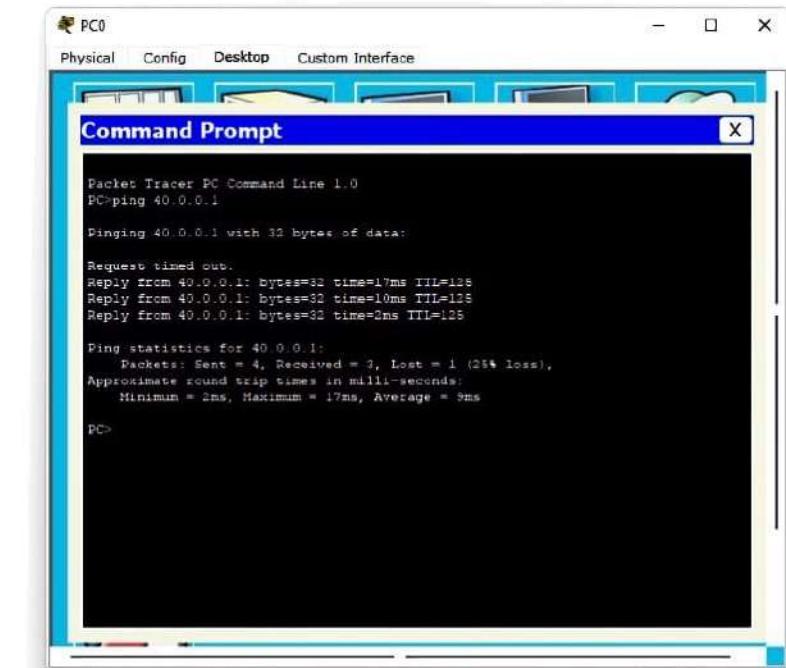
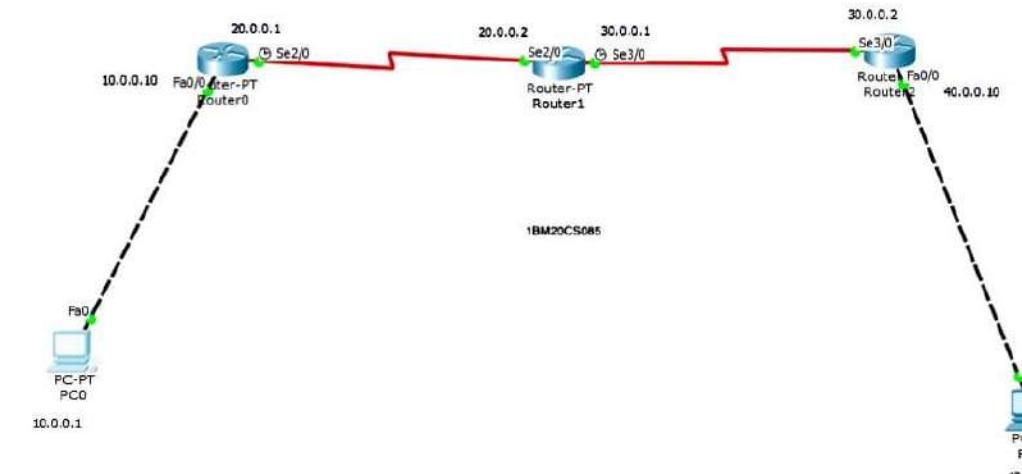
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



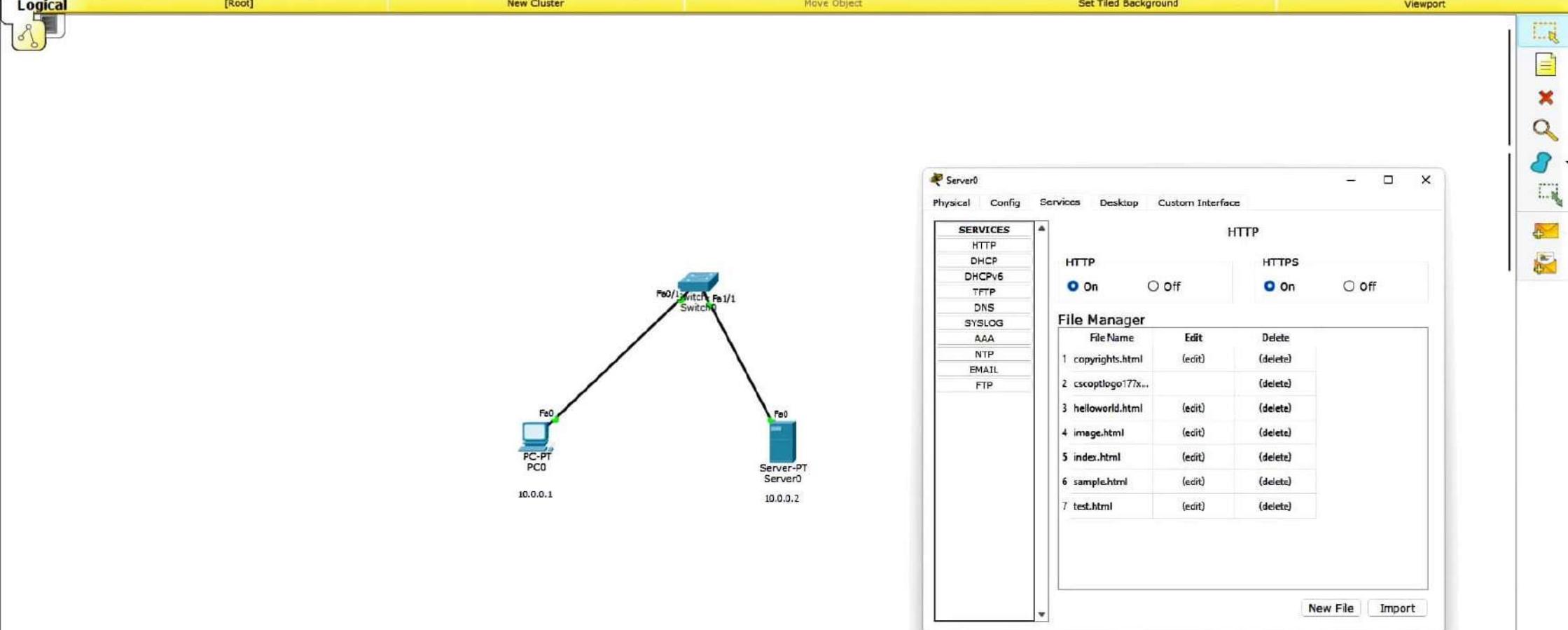
Time: 00:00:30

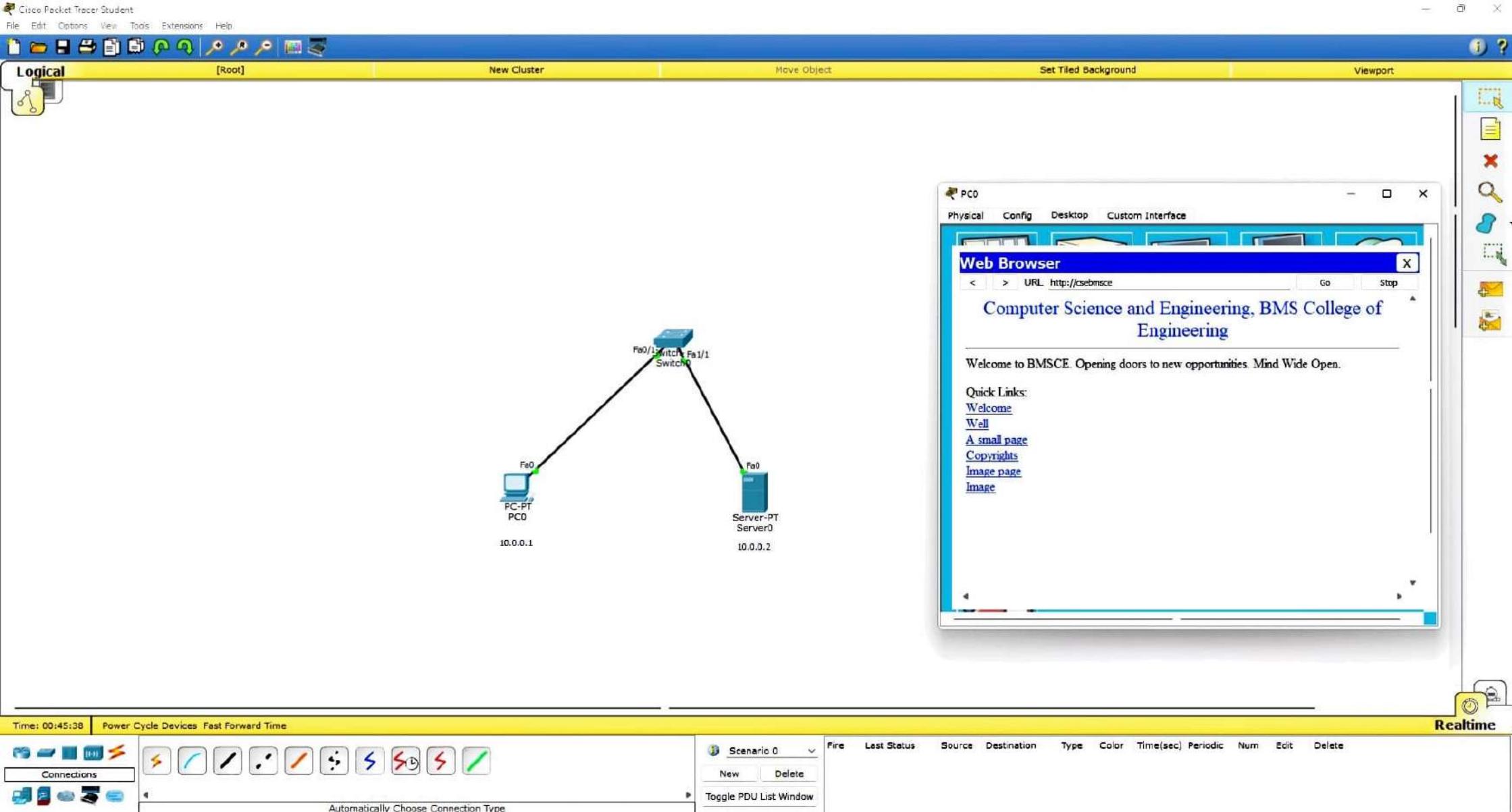
Power Cycle Devices Fast Forward Time

Realtime



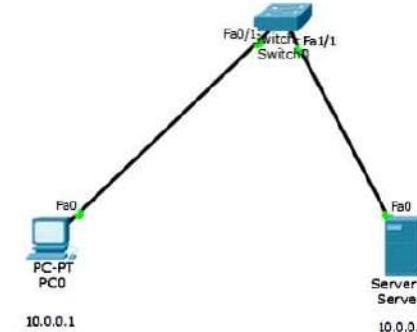
(Select a Device to Drag and Drop to the Workspace)





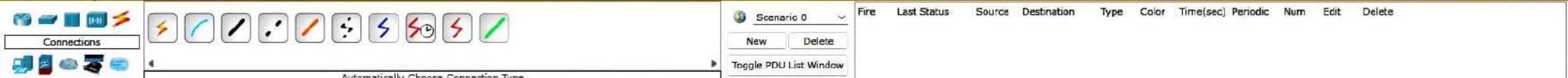


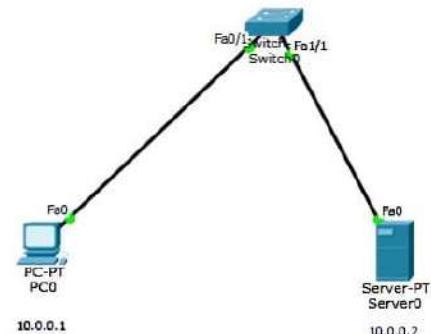
Logical [Root] New Cluster Move Object Set Tiled Background Viewport



Time: 00:44:11 Power Cycle Devices Fast Forward Time

Realtime



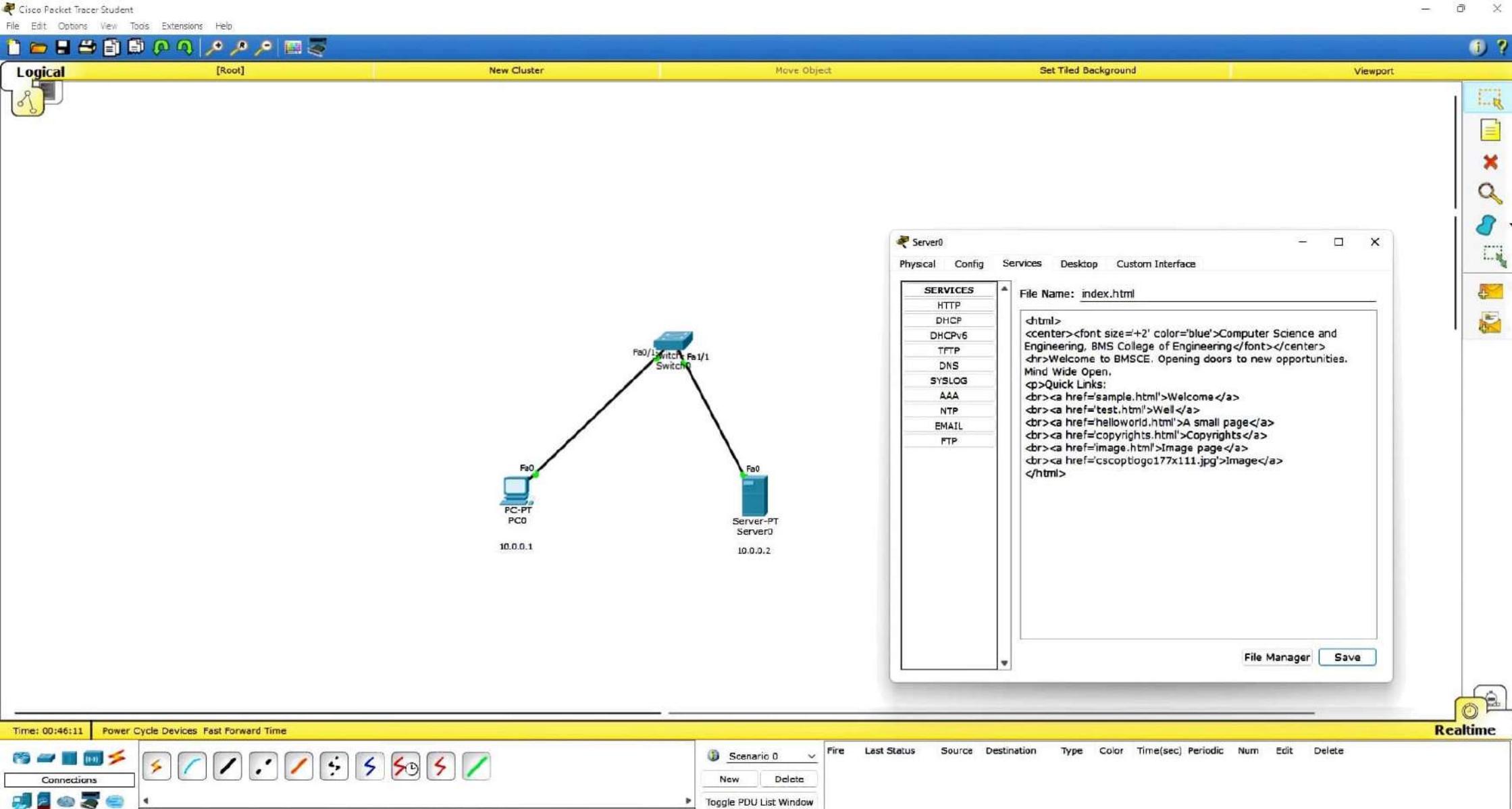


Time: 00:44:11

Power Cycle Devices Fast Forward Time



Realtime



Enter data to be transmitted: 10001000000100001

Enter the Generating polynomial: 1011101

Data padded with n-1 zeros : 10001000000100001000000

CRC or Check value is : 010011

Final data to be sent : 10001000000100001010011

Enter the received data: 10001000000100001010011

Data received: 10001000000100001010011

No error detected

```
Enter bucket size
500
Enter output rate
100
Enter packet size
700
Packets too big for bucket
Amount of bucket filled 0
Do you want to enter another packet(1 for yes, 2 for no)
1
Enter packet size
200
Amount of bucket filled 100
Do you want to enter another packet(1 for yes, 2 for no)
1
Enter packet size
300
Amount of bucket filled 300
Do you want to enter another packet(1 for yes, 2 for no)
1
Enter packet size
100
Amount of bucket filled 300
Do you want to enter another packet(1 for yes, 2 for no)
2
```

...Program finished with exit code 0
Press ENTER to exit console. █

BELLMAN FORD

Enter no. of vertices: 4

Enter graph in matrix form:

0 5 4 999

5 0 6 3

999 3 1 6

2 0 1 4

Enter source: 1

Vertex 1 -> cost = 0 parent = 0

Vertex 2 -> cost = 5 parent = 1

Vertex 3 -> cost = 4 parent = 1

Vertex 4 -> cost = 8 parent = 2

No negative weight cycle

...Program finished with exit code 0

Press ENTER to exit console.

Enter no. of vertices:4

Enter the adjacency matrix:

0 5 4 999

5 0 6 3

999 3 1 6

2 0 1 4

Enter the starting node:1

Distance of node0=5

Path=0<-1

Distance of node2=4

Path=2<-3<-1

Distance of node3=3

Path=3<-1

...Program finished with exit code 0

Press ENTER to exit console.