

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

# COMPUTER NETWORKS

*Submitted by*

**KANKANALA APOORVA (1BM20CS067)**

*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 **COMPUTER NETWORKS**” carried out by **KANKANALA APOORVA (1BM20CS067)**, 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. Nandhini Vineeth  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

# Index

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

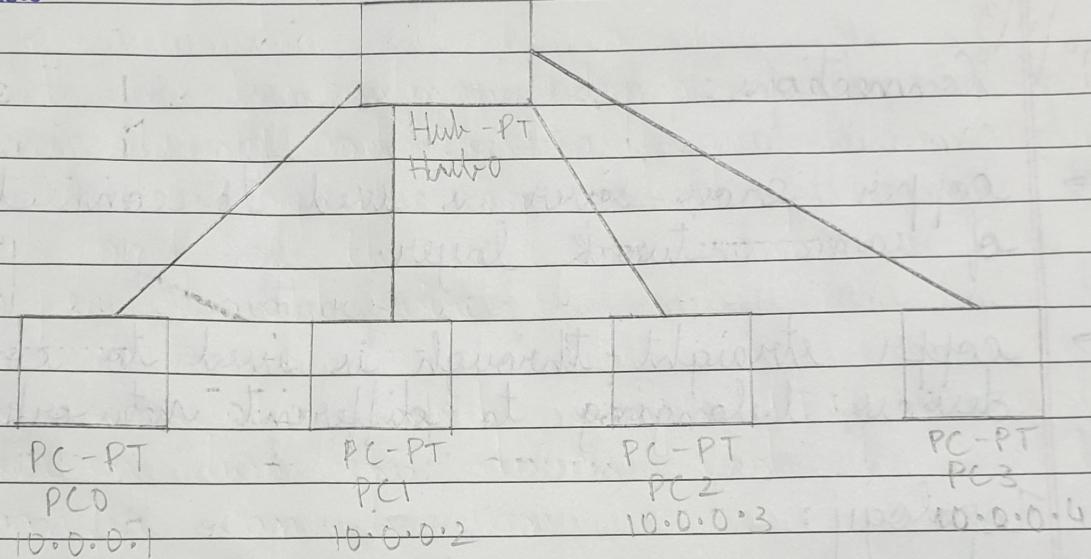
10/11/2022

DATE / /  
PAGE

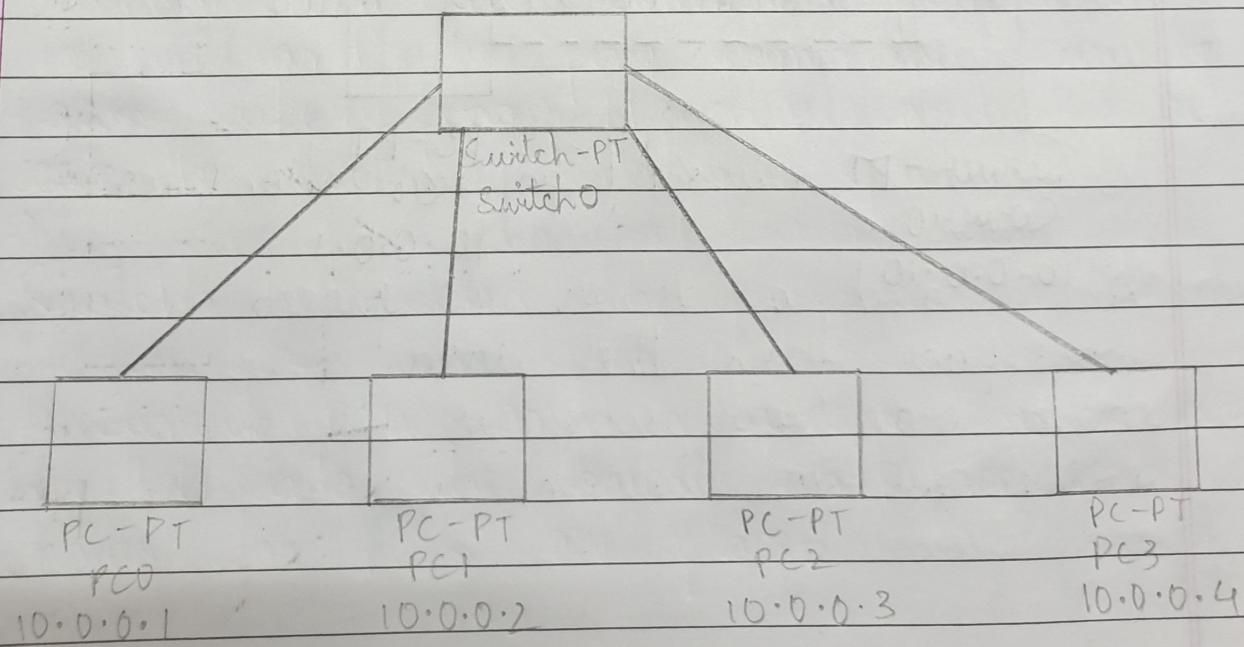
Aim: Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices

Topology:

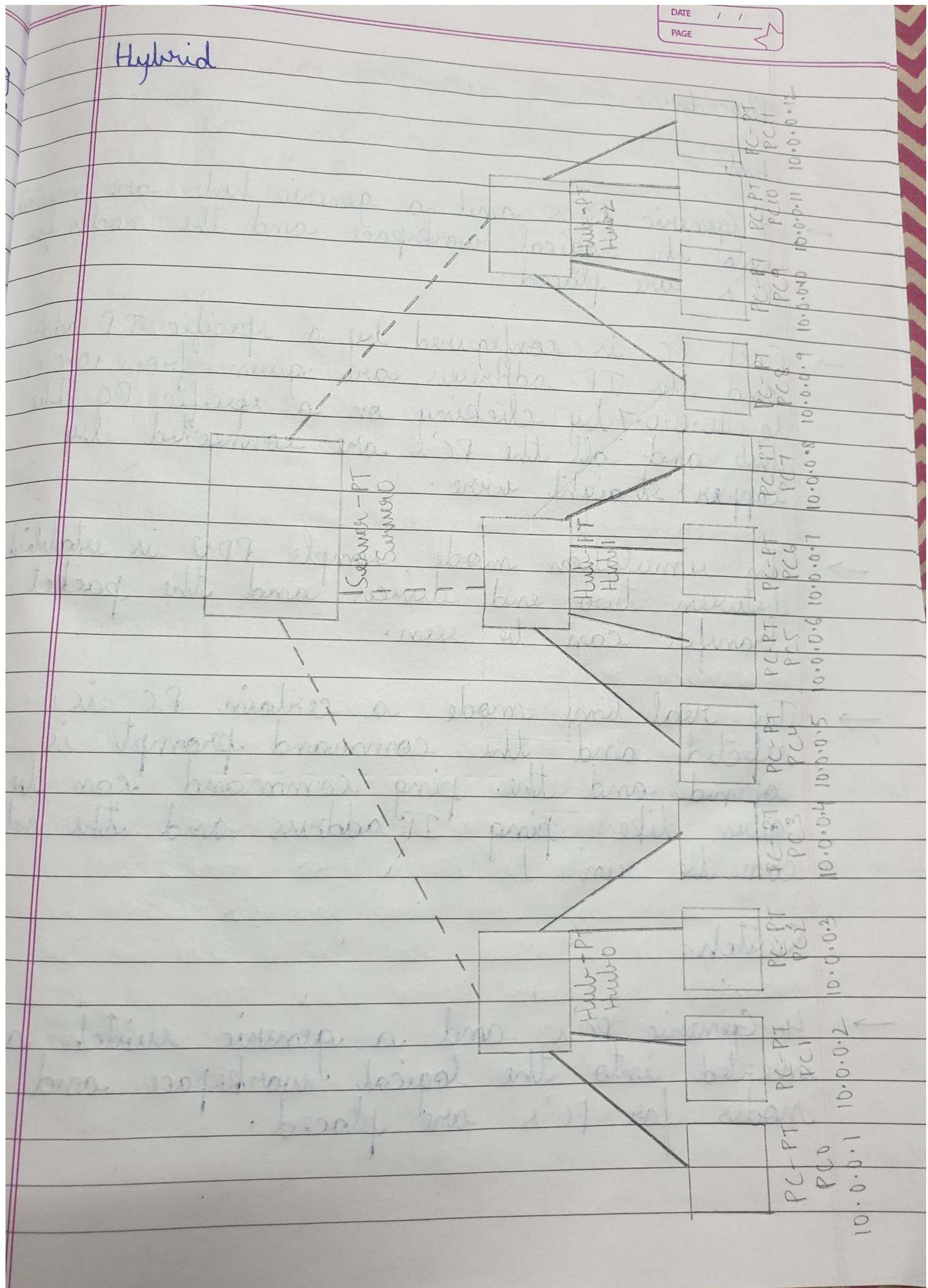
Hub



Switch



# Hybrid



## Procedure:

### Hub

- 7 Generic PC's and a generic hub are inserted into the logical workspace and the nodes for PC's are placed.
- Each PC is configured by a specific IP address and the IP addresses are given from 10.0.0.1 to 10.0.0.7 by clicking on a specific PC. The hub and all the PC's are connected by copper straight wire.
- In simulation mode, simple PDV is established between two end devices and the packet transfer can be seen.
- In real-time mode a certain PC is selected and the command prompt is opened and the ping command can be given like ping IP address and the output can be seen.

### Switch

- 4 Generic PC's and a generic switch are inserted into the logical workspace and the nodes for PC's are placed.

- Each PC is configured by a specific IP address and the IP addresses are given for the PC's from 10.0.0.1 to 10.0.0.4 by clicking on a specific PC. The switch and all the PC's are connected by using copper cross-over straight wire.
  - In simulation mode, simple PDU is established between two end devices and auto-capture is clicked. The packet transfer can be seen between the switch and the PC's.
  - In real-time mode, a PC is selected and command prompt is opened from the desktop option and the ping command can be given or sent to any other end-device and the output can be seen.
- Hybrid
- 12 generic PC's are inserted on the logical workspace where 4 PC's are connected to one hub where there are 3 hubs each with 4 PC's and the three hubs are connected to a single switch.
  - All the nodes for 12 PC's are placed and the PC's are connected to their respective hub by copper - straight wire and the three hubs are connected to their switch by copper cross-over wire. The IP addresses are specified

for all the 12 PC's from 10.0.0.1 to 10.0.0.12

- In simple PDU simulation mode, simple PDU is established between two end devices and packet transfer can be seen from the source to the destination.
- In real-time mode, a PC can be selected and the command prompt can be opened for pinging another PC from the present PC and the output can be seen on the command prompt screen.

Observations :

Hub

learning outcome:

- 1) The hub broadcasts the message received by the source to all the other end devices but the message is read only by the specified destination and the destination responds back by sending a packet.
- 2) If the ports are not enough, the hub has to be clicked, the power has to be switched off and extra ports can be kept in the blank spaces and the power is switched on.

Result:

ping 10.0.0.7

pinging 10.0.0.7 with 82 bytes of data:  
 Reply from 10.0.0.7: bytes = 32 time = 0ms TTL=128  
 Reply from 10.0.0.7: bytes = 32 time = 0ms TTL=128  
 Reply from 10.0.0.7: bytes = 32 time = 0ms TTL=128  
 Reply from 10.0.0.7: bytes = 32 time = 0ms TTL=128

Ping statistics for 10.0.0.7:

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

Switch

Learning outcome:

- 1) The switch does not establish connection immediately, there is a certain time called learning time and message passing cannot be done until the green light connection is established.
- 2) The switch initially broadcasts for all the devices but afterwards the message is sent only to the destination specified.

Result:

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 TTL=128  
 Reply from 10.0.0.3: bytes = 32 time = 0ms TTL=128  
 Reply from 10.0.0.3: bytes = 32 time = 0ms TTL=128

Reply from 10.0.0.3 : bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3 :

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

Approximate round trip times in milli seconds  
Minimum=0ms, Maximum=0ms, Average=0ms

## Hybrid

Learning outcome:

- 1) The switch sends message to the particular hub which is connected to the destination PC but the hub which receives the message broadcasts to all the end devices of the particular hub while only the destination responds back.

Result:

Ping 10.0.0.11 with 32 bytes of data

Pinging 10.0.0.11 with 32 bytes of data  
Reply from 10.0.0.11 : bytes=32 time=0ms TTL=128

Reply from 10.0.0.11 : bytes=32 time=0ms TTL=128

Reply from 10.0.0.11 : bytes=32 time=1ms TTL=128

Reply from 10.0.0.11 : bytes=32 time=0ms TTL=128

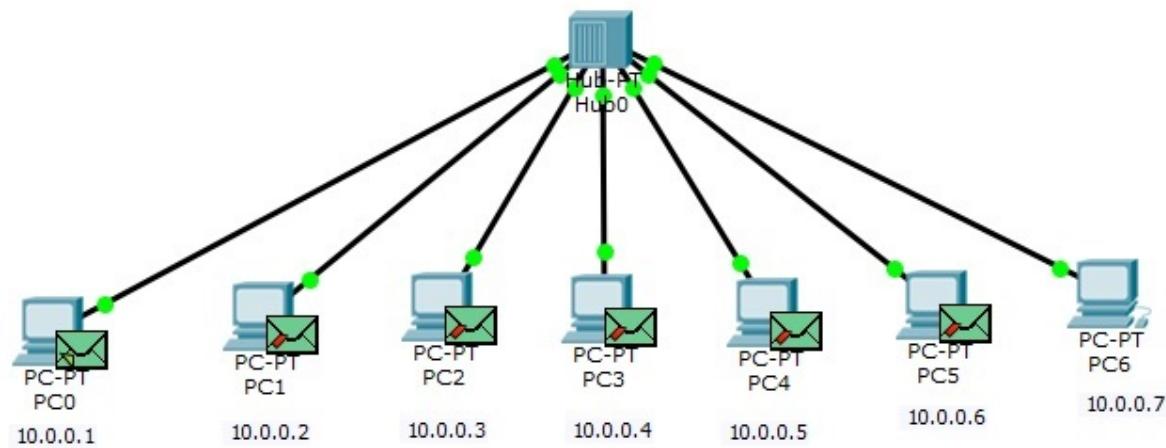
Ping statistics for 10.0.0.11 :

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

Approximate round trip times in milli -seconds

Minimum=0ms, Maximum=1ms, Average=0ms

1BM20CS067



## Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.7

Pinging 10.0.0.7 with 32 bytes of data:

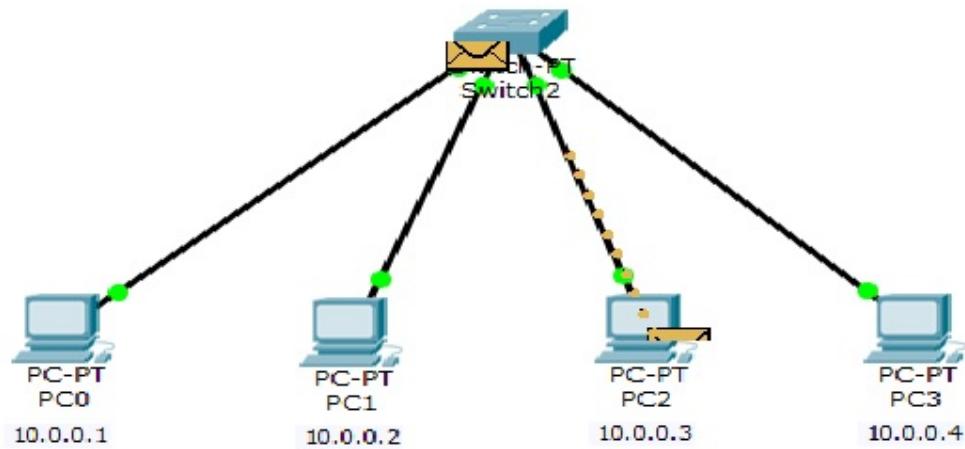
Reply from 10.0.0.7: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

1BM20CS067

1BM20CS067



## Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

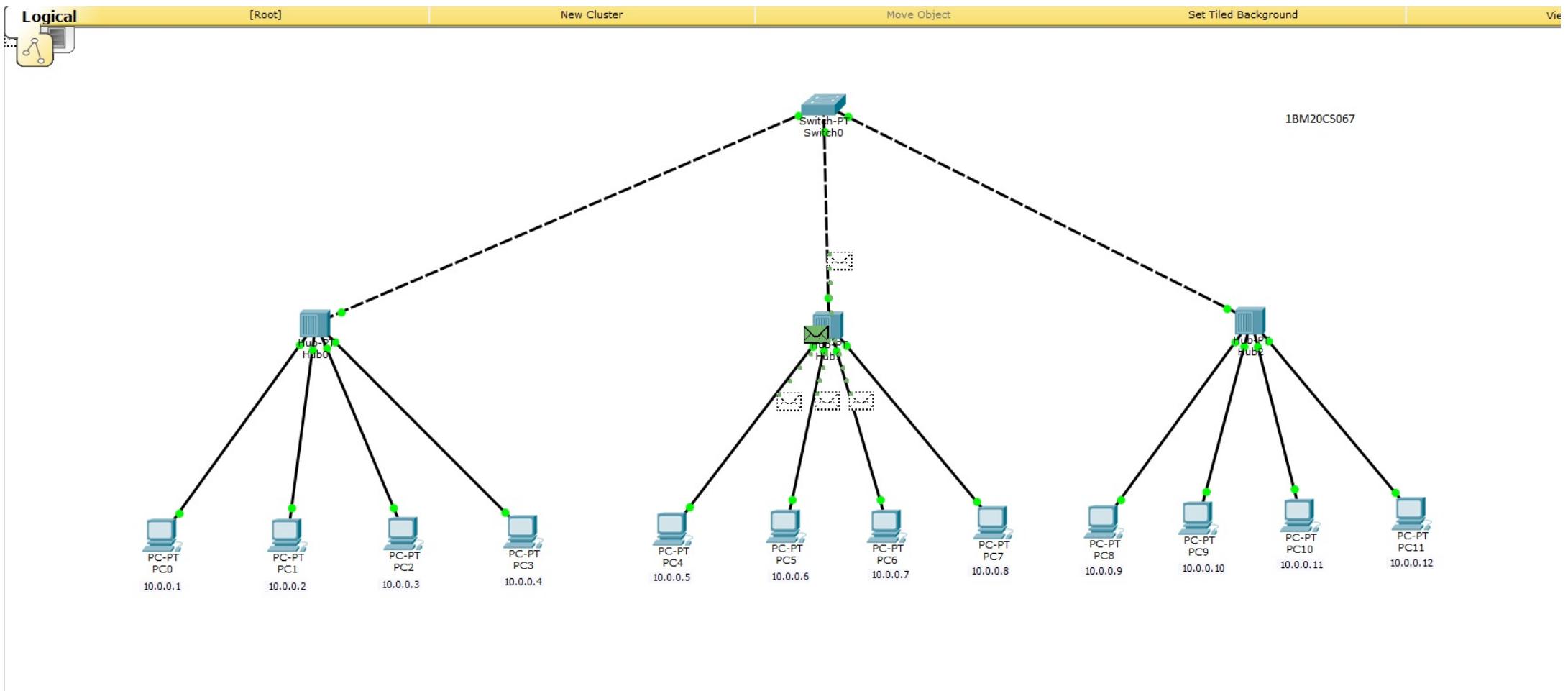
Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

1BM20CS067



## Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data:

Reply from 10.0.0.11: bytes=32 time=0ms TTL=128
Reply from 10.0.0.11: bytes=32 time=0ms TTL=128
Reply from 10.0.0.11: bytes=32 time=1ms TTL=128
Reply from 10.0.0.11: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

1BM20CS067

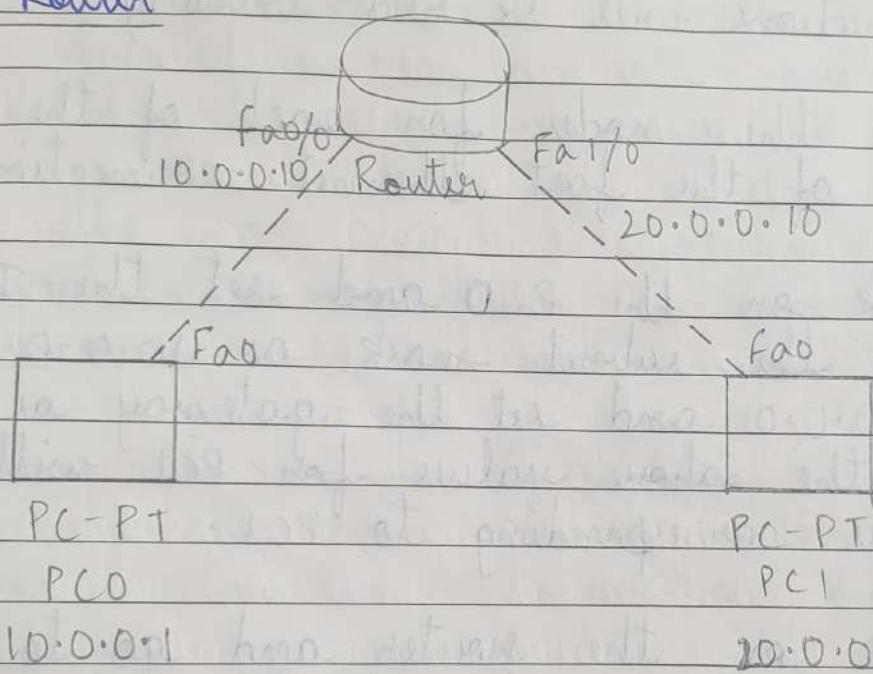
17/11/2022

DATE / /  
PAGE ↘

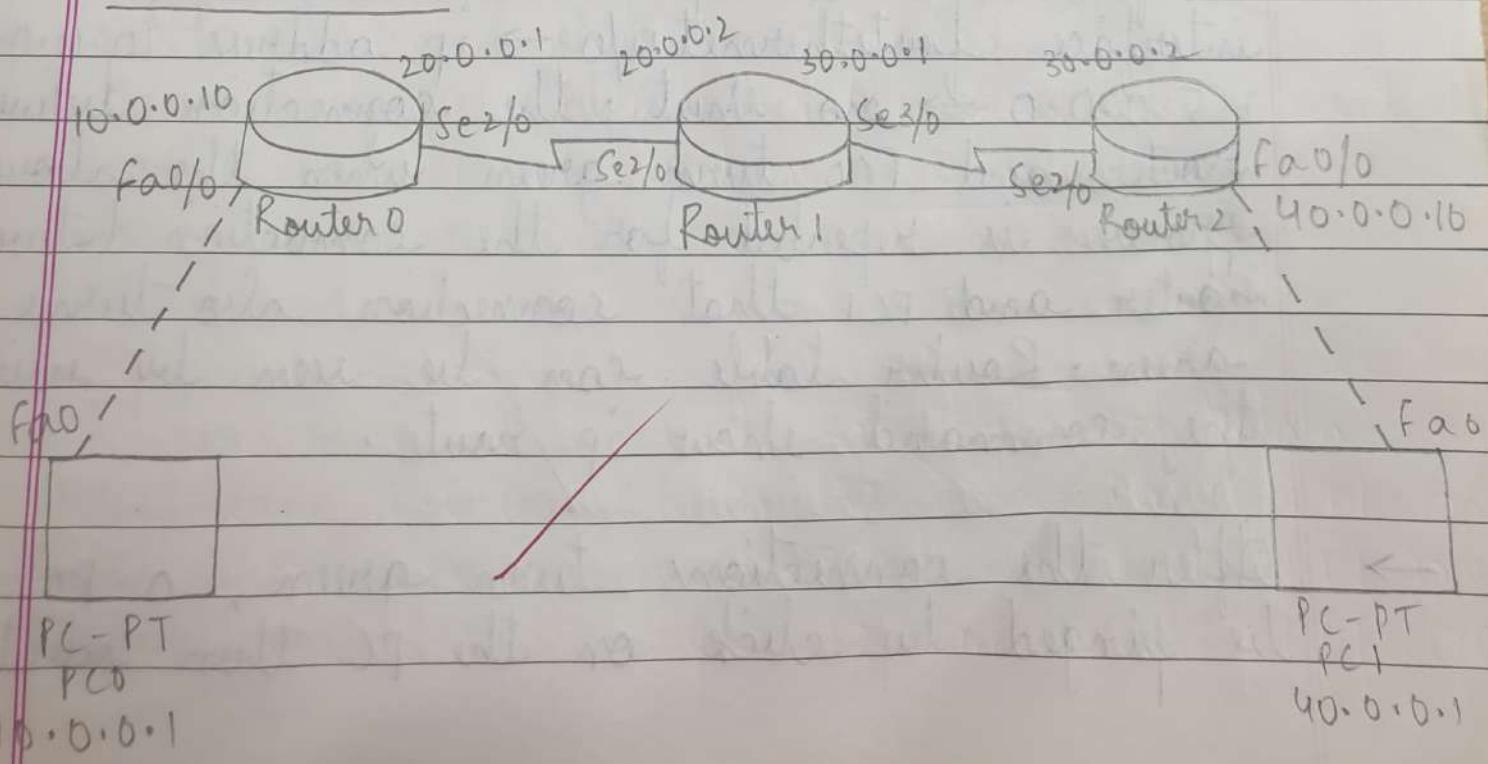
Aim: Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Response, Destination unreachable, Request timed out, Reply.

### Topology:

#### One-Router



#### Three routers



## Procedure :

### One-Router

- Place two generic PC's and a generic router and the router is connected to each of the PC's with a copper cross wire. The connections will be red initially.
- Place the modes for each of the PC's and each of the fast ethernet connections.
- Click on the PC0 and set the IP address and the subnet mask as 10.0.0.1 and 255.0.0.0 and set the gateway as 10.0.0.10. Set the above values for PC1 with the values corresponding to PC1.
- Click on the router and go to the command line interface (CLI) → don't continue with configuration dialog → enable → config t → interface fastethernet 0/0 → ip address 10.0.0.10 255.0.0.0 → no shut ; the connection between router and PC0 turn green when the above process is repeated for the connection between router and PC1 that connection also turns green. Router table can be seen by using the command show ip route.
- After the connections turn green, a PC can be pinged by click on the PC then selecting

desktop and then select command prompt.

### 3 - routers

- Place three generic routers and two generic PC's. First router is connected to the first PC and third router is connected to second PC by a copper cross-over wire and the three routers are connected among each other with the serial DCE cable. All the connections are red initially
- Place the nodes and the router and PC is connected through fast ethernet while the routers are connected through serial.
- Each of the PC is clicked and the IP address, subnet mask and gateway is set for each of the PC with the corresponding values.
- Router 1 is clicked > CLI > "no" > enable > config t > interface fastethernet 0/0 > ip address 10.0.0.10 255.0.0.0 > no shut → With these the first connection is established.  
config t > interface serial 2/0 > ip address 20.0.0.1 255.0.0.0 > no shut → Second connection is established.
- Router 2 is clicked > CLI > "no" > enable > config t > interface serial 2/0 > ip address 20.0.0.2 255.0.0.0 > no shut → With these first connection is established

→ Now, the correct reply is seen from when PCI is pinged by PCO

### Observations:

#### One - Router

When PCO pings PCI for the first time we get  
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 ms

Minimum = 0ms, Maximum = 4ms, Average = 1ms

But when PCO pings PCI again or if PCI reverse pings PCO we get the output when where all the 4 times replied reply is observed.

### 3 - Router

Before the routers are trained and PC1 is pinged by PC0 ~~will~~ get 8 routers we get  
ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.0.10: Destination host unreachable  
Reply from 10.0.0.10: Destination host unreachable  
Reply from 10.0.0.10: Destination host unreachable  
Reply from 10.0.0.10: Destination host unreachable

Pinging statistics 40.0.0.1

Packet: Sent = 4 Received = 0 Lost = 4 (100% loss)

ping 20.0.0.2

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Pinging statistics 20.0.0.2

Packet: Sent = 4 Received = 0 Lost = 4 (100% loss)

After the routers are trained

ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

Request timed out

Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

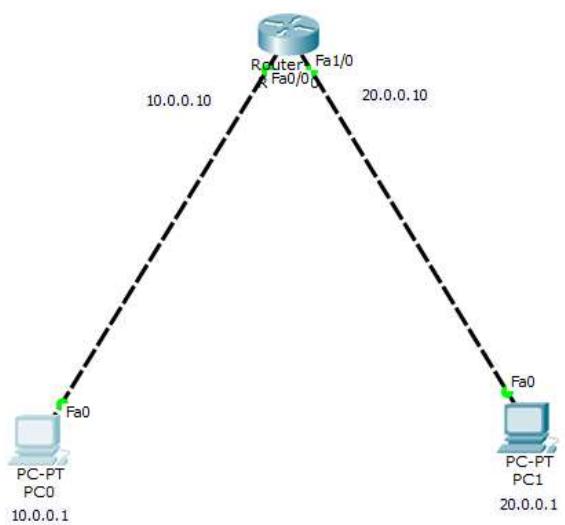
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Pinging statistics for 40.0.0.1

Plackets: Sent = 4 Received = 3 Loss = 1 (25% loss)

2/1/22

1BM20CS067



Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

1BM20CS067

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shut
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet1/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#no shut
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route
Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, FastEthernet1/0
Router#
```

```
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=4ms 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 = 4ms, Average = 1ms

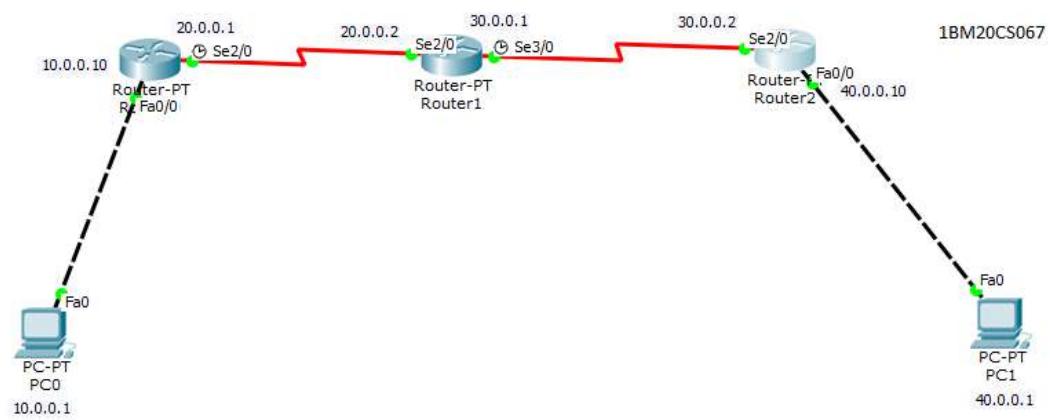
PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:
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=3ms 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 = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 0ms

PC>
```

1BM20CS067



```
Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
*LINK-5-CHANGED: Interface Serial2/0, changed state to up

Router(config-if)#interface serial2/0
*LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, cinterface serial2/0
Router(config-if)#exit
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
*LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

*LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)#exit
Router#
*SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

S   10.0.0.0/8 [1/0] via 30.0.0.1
S   20.0.0.0/8 [1/0] via 30.0.0.1
C   30.0.0.0/8 is directly connected, Serial2/0
C   40.0.0.0/8 is directly connected, FastEthernet0/0
Router#
```

1BM20CS067

```
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 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Request timed out.
Reply from 10.0.0.10: Destination host unreachable.

Ping statistics for 40.0.0.1:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.2:
  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:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=7ms TTL=128
Reply from 40.0.0.1: bytes=32 time=9ms TTL=128
Reply from 40.0.0.1: bytes=32 time=8ms TTL=128

Ping statistics for 40.0.0.1:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 7ms, Maximum = 9ms, Average = 8ms
PC>
```

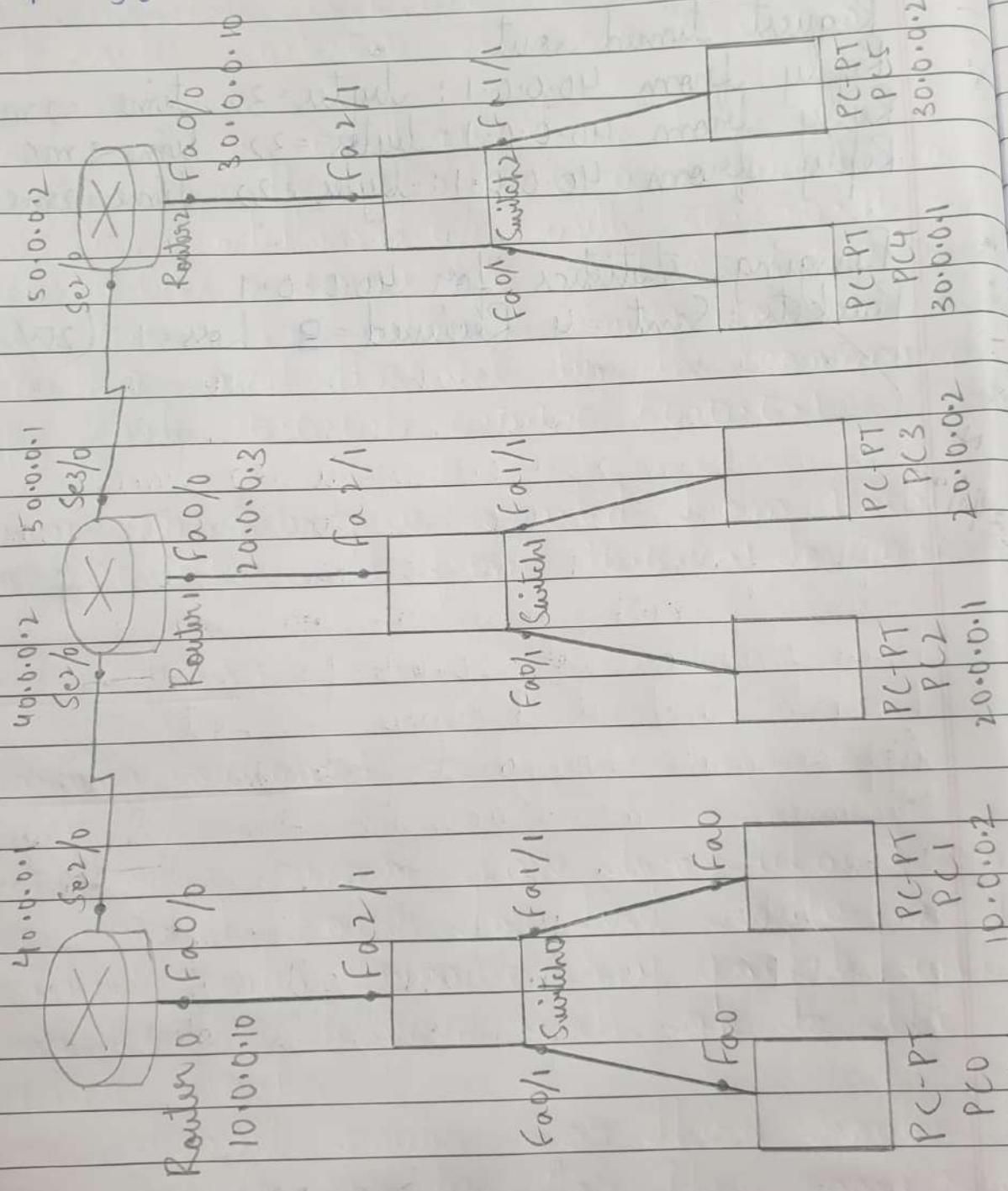
1BM20CS067

24/11/2022

DATE / /  
PAGE ↗

Ques: Configuring default route to the router

Topology:



## Procedure :

- Place 6 generic PC's, 3 switches and 3 routers and connect two PC's to each switch with copper straight through wire and each switch is connected to one router with a copper straight through wire and the three routers are connected among themselves by serial DCE cable and the nodes are placed for all the devices and networks.
- A PC is clicked to set the attributes for a PC and each PC has three attributes which are the IP address, subnet mask and the gateway and all the three are set according to the nodes placed. This process is done for all the 6 PC's.
- For Router 1, the configurations are done in the command line interface (CLI). The IP address and subnet mask are set for both the interfaces - fastethernet 0/0 as 10.0.0.10 & 255.0.0.0 and serial 2/0 as 40.0.0.1 and 255.0.0.0. Router 2 is the default router for Router 1 and this is done by the command ip route 0.0.0.0 0.0.0.0 40.0.0.2.
- For Router 2, the IP address and subnet mask are set for all three interfaces - fastethernet 0/0 as 20.0.0.3 & 255.0.0.0 and serial 2/0 as 40.0.0.2 & 255.0.0.0 and serial 3/0 as 50.0.0.1 & 255.0.0.0

Router 2 does not have any default router and the static routing is done for the network 10 & 40 by the following command

```
ip route 10.0.0.0 255.0.0.0 40.0.0.1  
ip route 30.0.0.0 255.0.0.0 50.0.0.2
```

- Router 3 is configured in both the interfaces with IP address and subnet mask for fastethernet 0/0 with 30.0.0.10 & 255.0.0.0 & serial 2/0 with 50.0.0.2 & 255.0.0.0. The default router for router 3 is router 2 and this set by the command ip route 0.0.0.0 0.0.0.0 50.0.0.1
- 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.

### Observations:

### Learning outcome:

- One router cannot have two default routers
- The default router for first router is the middle router because any packet which have to be delivered will go to the middle router.
- The default router for third router is the middle router for the same reason.
- The middle router does not have any default router because if one of the router is made default then there is a chance

that the packets which are to be sent to the switch are sent to the router.

Result :

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 = 1ms, TTL=126

Reply from 20.0.0.1: bytes = 32, time = 2ms, TTL=126

Reply from 20.0.0.1: bytes = 32, time = 6ms, TTL=126

ping 30.0.0.2

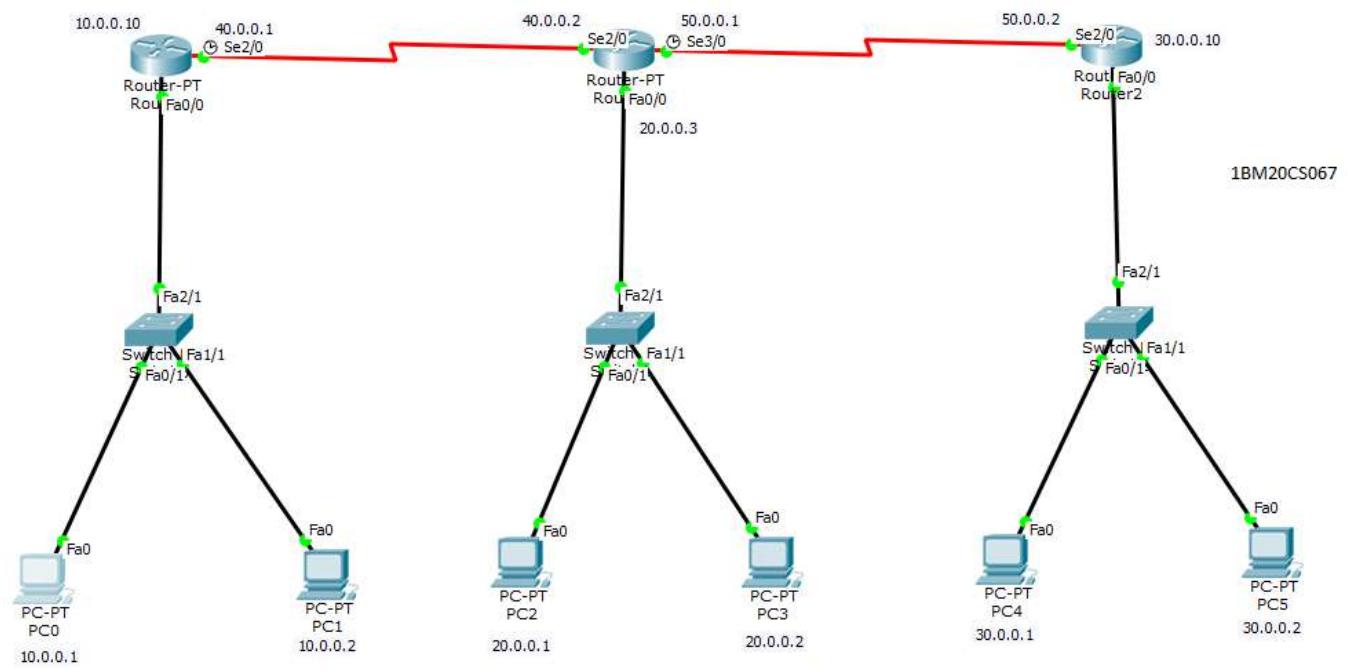
Pinging 30.0.0.2 with 32 bytes of data

Request timed out

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

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

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



```

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up           1BM20CS067

Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
interface
% Incomplete command.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 20.0.0.3 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface serial3/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

Router(config-if)#exit
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 30.0.0.0 255.0.0.0 50.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

S   10.0.0.0/8 [1/0] via 40.0.0.1
C   20.0.0.0/8 is directly connected, FastEthernet0/0
S   30.0.0.0/8 [1/0] via 50.0.0.2
C   40.0.0.0/8 is directly connected, Serial2/0
C   50.0.0.0/8 is directly connected, Serial3/0
Router#

```

```
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=1ms TTL=126
Reply from 20.0.0.1: bytes=32 time=2ms TTL=126
Reply from 20.0.0.1: bytes=32 time=6ms TTL=126
1BM20CS067

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 6ms, Average = 3ms

PC>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=4ms TTL=125
Reply from 30.0.0.2: bytes=32 time=4ms TTL=125
Reply from 30.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 3ms

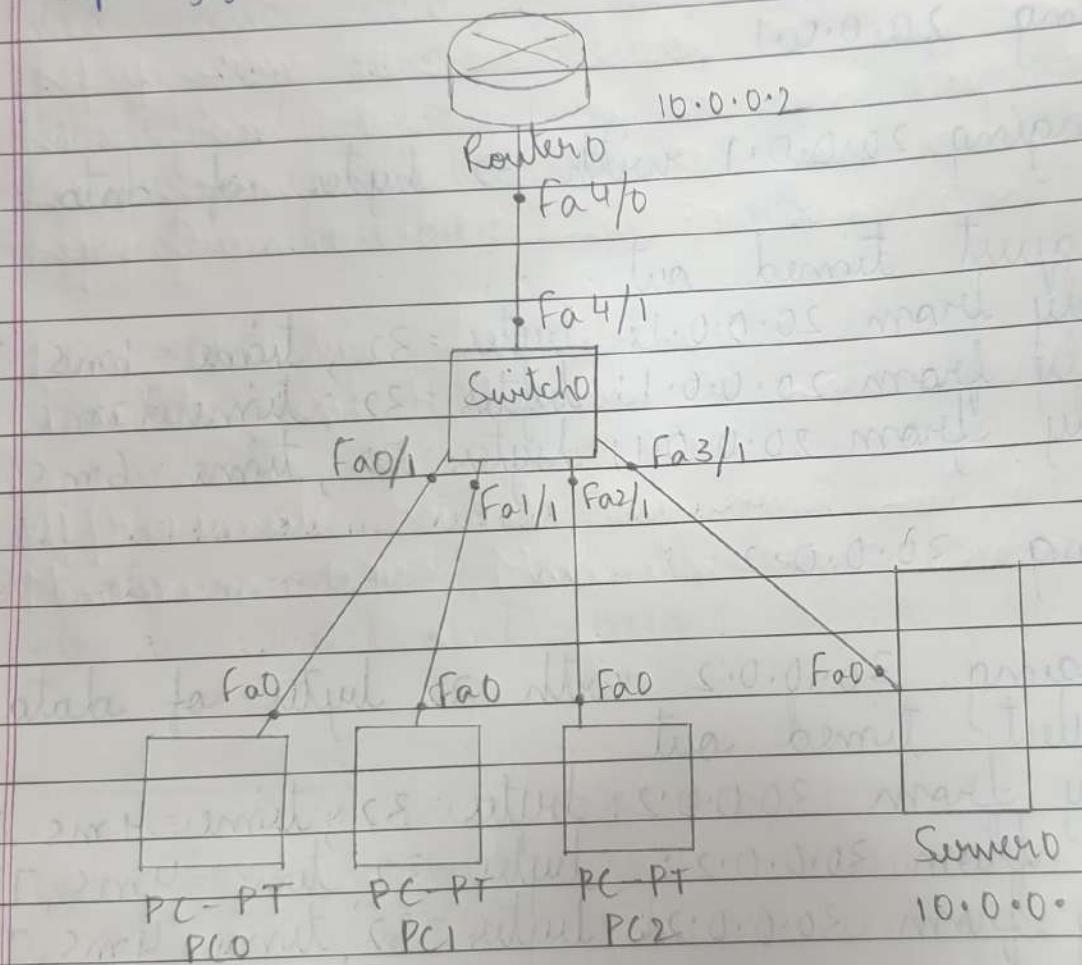
PC>
```

01/12/2022

PAGE

Aim: Configuring DHCP within a LAN in a packet tracer.

Topology:



Procedure:

- Place a generic router, a generic switch, a generic server and three generic PC's. Connect them with suitable connections among router - switch, switch - server and switch - PC.
- Place the nodes for server and router as 10.0.0.1 and 10.0.0.2.

- Click on the server → go to configuration tab and set IP address and subnet mask as 10.0.0.1 and 255.0.0.0. Set the gateway as 10.0.0.2.
- Click on the router → CLI tab → enable → config t → interface fastethernet 4/0 → Set IP address as ip address 10.0.0.2 255.0.0.0 → no shut. The ip address is set as 10.0.0.2
- Click on server → services tab → select DHCP and switch on the service → Set default gateway as 10.0.0.2, DNS server as 10.0.0.1 Start IP address as 10.0.0.3 and TFTP server as 10.0.0.1. After setting everything, the contents are saved.
- Each of the PC is clicked → desktop tab → IP configuration → Set as DHCP. The IP address, subnet mask, default gateway and DNS server are set automatically.
- Click on PC 10.0.0.3 and ping to 10.0.0.5.

### Observations :

Learning outcome : The server automatically sets the IP address, subnet mask and gateway to all the PC's and IP address is allocated serially in DHCP protocol.

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 = 0ms TTL = 128

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

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

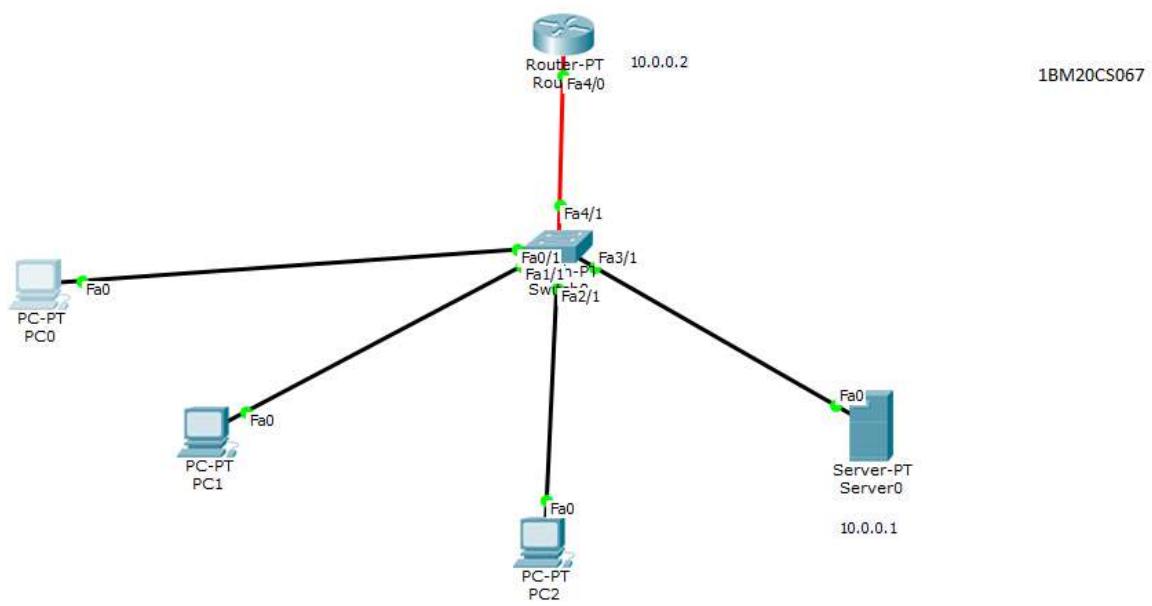
Reply from 10.0.0.5: bytes = 32, time = 0ms TTL = 128

Ping statistics for 10.0.0.5:

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

Approximate round trip times in ms:

Minimum = 0ms, Maximum = 4ms, Average = 1ms.

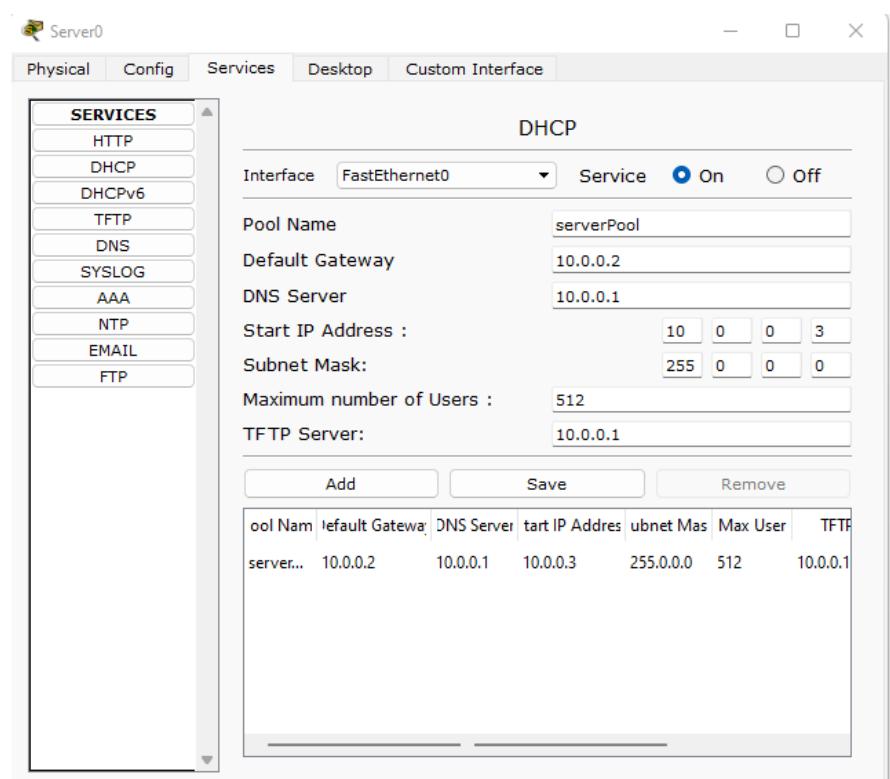


```
Continue with configuration dialog? [yes/no]: no
```

```
Press RETURN to get started!
```

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet4/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shut
```

1BM20CS067



```
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
Reply from 10.0.0.5: bytes=32 time=4ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
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 = 4ms, Average = 1ms

PC>
```

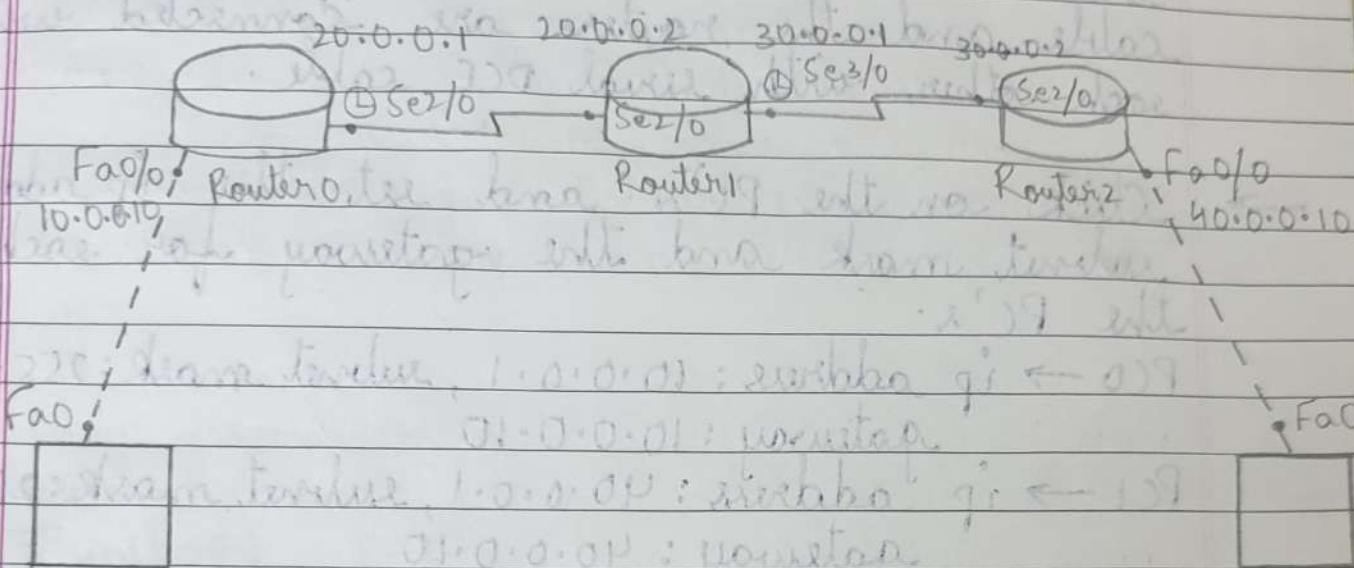
1BM20CS067

08/12/2022

DATE / /  
PAGE

Aim: Configuring RIP routing protocol in Routers.

Topology:



Serial DCE: Serial connections, often used for WAN links, must be connected between serial ports. We must enable clocking on the DCE side to bring up the line protocol. We can tell which end of the connection is the DCE side by the small "clock" icon next to the port.

Introduction: Routing Information protocol (RIP) is a protocol that routers can use to exchange network topology information. RIP uses a distance vector algorithm to decide which path to put a packet on to get to its destination.

Procedure:

- Place two PC's and three routers and connect the PC and router with copper cable over cable and the routers are connected with each other with serial DCE cable.
- Click on the PC's and set the ip address, subnet mask and the gateway for each of the PC's.
  - PC0 → ip address: 10.0.0.1, subnet mask: 255.0.0.0  
gateway: 10.0.0.10
  - PC1 → ip address: 40.0.0.1, subnet mask: 255.0.0.0  
gateway: 40.0.0.10
- Click on the first router → go to cli → type the commands >enable ; #config t ;
  - #interface fastethernet0/0 ; #ip address 10.0.0.10 255.0.0.0 ; #no shut ; #exit ;
  - #interface serial2/0 ; #ip address 20.0.0.1 255.0.0.0 ; #encapsulation ppp ; #clock rate 64000 ; #no shut
- Click on the second router → go to cli → type the commands >enable ; #config t ; #interface serial2/0 ; #ip address 20.0.0.2 255.0.0.0 ;
  - #encapsulation ppp ; #no shut ; #exit
  - #interface serial3/0 ; #ip address 30.0.0.1 255.0.0.0 ; #encapsulation ppp ; #clock rate 64000 ; #no shut

- Click on the third router → go to cli. → type the commands > enable ; config t ; #interface serial2/0 ; #ip address 30.0.0.2 255.0.0.0 ; #encapsulation PPP ; #no shut ; #exit  
#interface fastethernet0/0 ; #ip address 40.0.0.10  
255.0.0.0 ; #no shut
- Now all the basic configuration are set for all the PC's and routers. All the lights are turned green
- Now again click on first router and go to cli and type the following commands → #router rip  
#network 10.0.0.0  
#network 20.0.0.0 ; #exit
- Click on second router → go to cli and execute  
#router rip  
#network 20.0.0.0  
#network 30.0.0.0 ; #exit
- Click on third router → go to cli and execute  
#router rip  
#network 30.0.0.0  
#network 40.0.0.0
- Ping the PC 40.0.0.1 from 10.0.0.1

## Observations:

Learning outcome:

When RIP protocol is used we don't have to do static routing for all the routers i.e. we don't have to teach all the routers by providing with the next hop.

In dynamic routing (RIP protocol) we just have to specify the networks known by the router.

Result:

ping 40.0.0.1

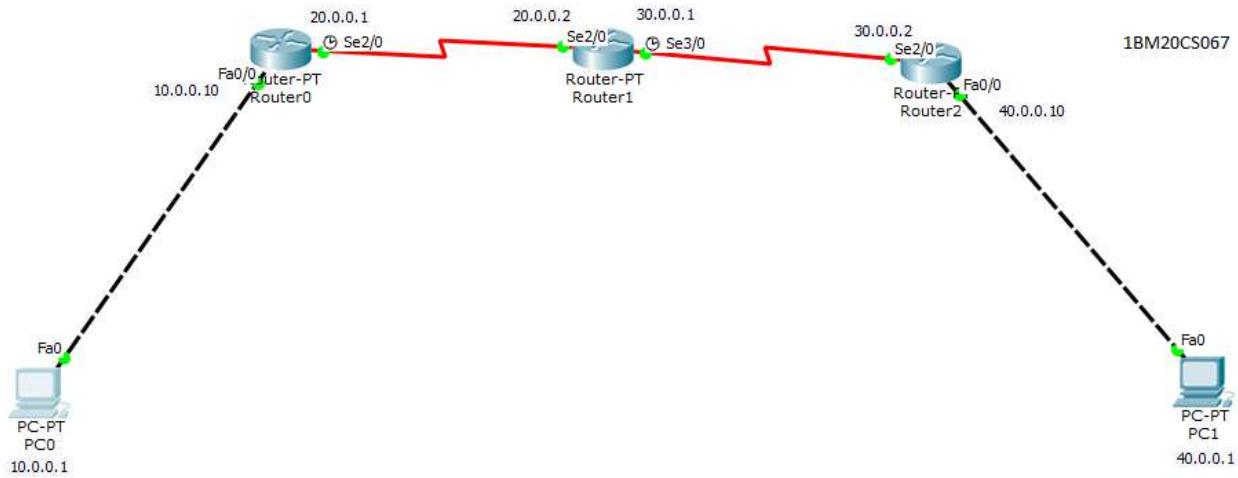
Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.

Reply from 40.0.0.1: bytes=32 time=4ms TTL=125

Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Reply from 40.0.0.1: bytes=32 time=9ms TTL=125



```
Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.          1BM20CS067
Router(config)#interface serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#exit
Router(config)#

```

```
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.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=6ms TTL=125
Reply from 40.0.0.1: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.1:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 2ms, Maximum = 7ms, Average = 5ms

PC>
```

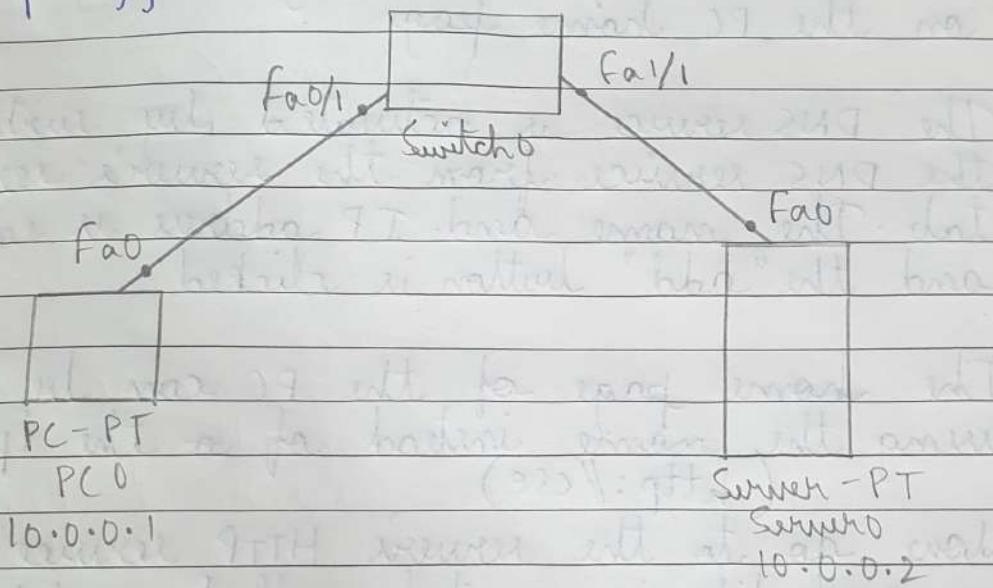
1BM20CS067

15/12/2022

DATE / /  
PAGE / /

Ques: Demonstration of web server and DNS using packet tracer.

Topology:



Procedure:

- Place a PC, a server and a switch and connect them with suitable connections and then place the mode.
- The IP address of and subnet mask are set for the PC as 10.0.0.1 255.0.0.0 and 10.0.0.2 255.0.0.0 for the server.
- The web browser in the desktop tab are invoked for the PC and the ip address of the PC is given as http://10.0.0.2. Then a home page is displayed.

- Now the services tab of the server is clicked and HTTP is selected. A window is seen with all the files and when the index.html file is changed it gets reflected on the PC home page
- The DNS service is activated by switching the DNS service from the service's services tab. The name and IP address is entered and the "add" button is clicked
- The home page of the PC can be opened using the name instead of the ip address (<http://cse>)
- Now go to the server's HTTP service and a new file is created called cv.html and the file is saved.
- Open web browser of the PC and when the command <http://cse/cv.html> is given the new file cv.html is appeared on the screen.

Observations :

Learning outcome: DNS protocol helps to map a name with an IP address. This protocol is very useful because the users are comfortable with the names and the computers are comfortable with the IP address.

Result :

The file typed in HTTP of server is  
cv.html

<html>

<head>

CV

</head>

<body>

Name : Apoorva <br><br>

College : BMSCE

</body>

</html>

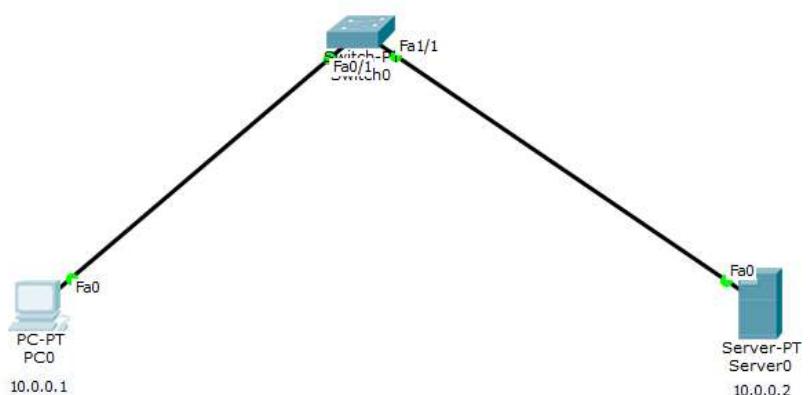
The output seen in the web browser of PC is

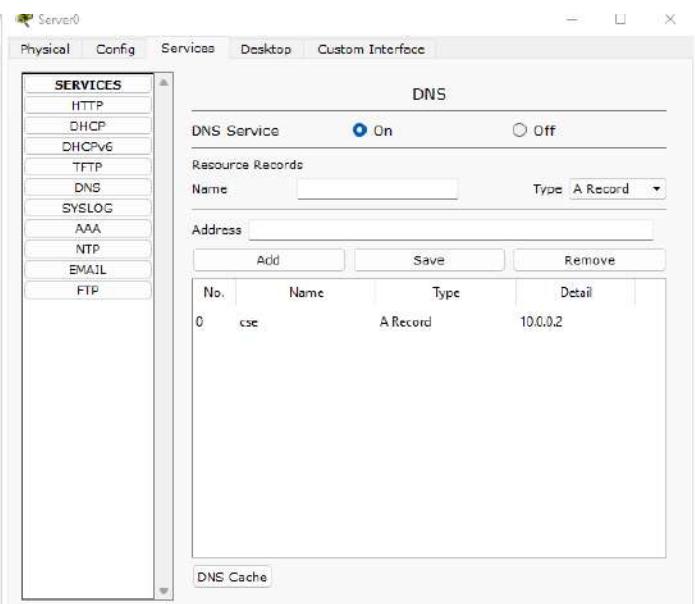
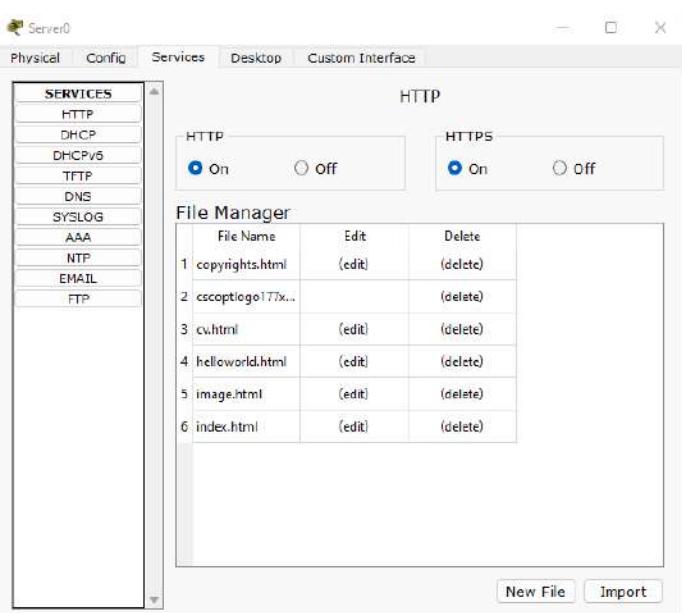
URL : http://cse/cv.html

CV Name : Apoorva

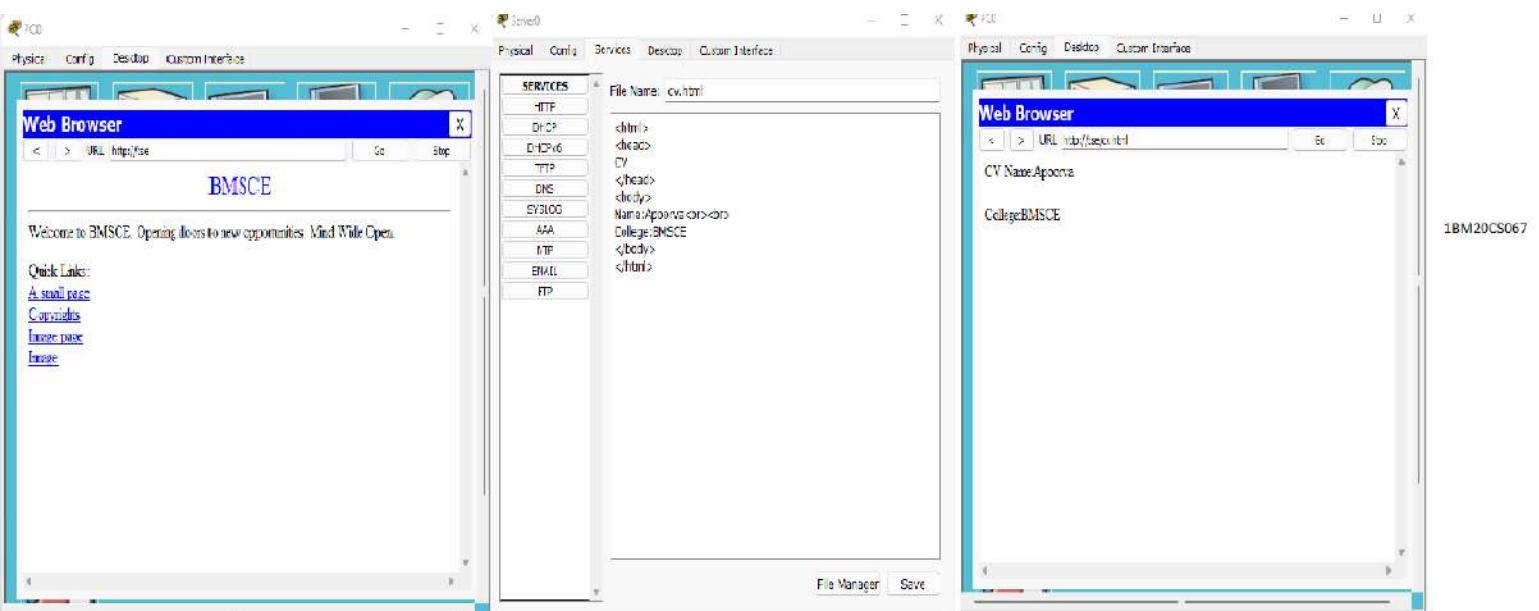
College : BMSCE

1BM20CS067





IBM20CS067



29/12/2022 Write a program to  
Error detection using CRC CITT (16)

```
#include <stdio.h>
#include <string.h>
#define N strlen(gen)
```

```
char data[28];
char check[28];
char gen[28];
int data_length, i, j;
```

void XOR()

{

```
for (j=1; j<N; j++)
```

{

```
if (check[j] == gen[j])
```

```
{ check[j] = '0';
```

    }

else

{

```
    check[j] = '1';
```

    }

}

    }

    }

    }

}

```
printf("Enter the received data : ");
```

```
scanf("%s", data);
```

```
printf("Data received : %s", data);
```

```
crc();
```

```
for (i=0; (i<N-1) && (check[i]!='1'); i++);
```

```
if (i<N-1)
```

```
printf (" \n Error detected ") ;
else
    printf (" \n No error detected ") ;
```

} void crc()

```
{ for (i=0; i<N; i++)
    check[i] = data[i] ;
```

```
do { if (check[0] == '1')
    XOR();
```

```
for (j=0; j< N-1; j++)
    check[j] = check[j+1] ;
```

```
    check[j] = data[i++];
```

```
} while (i <= data_length + N-1) ;
```

} int main()

```
{ printf (" \nEnter data word :") ;
scanf ("%s", data) ;
```

```
printf (" \nEnter the generating polynomial :") ;
scanf ("%s", gen) ;
```

```
data_length = strlen (data) ;
```

```
for (i = data_length; i < data_length + N - 1; i++)
    data[i] = '0' ;
```

```

printf("In Data padded with zeros : %s", data);
crc();
printf("In CRC or check value is %s", check);
for(i = data_length; i < data_length + N - 1; i++)
    data[i] = check[i - data_length];
printf("Final data to be sent %s", data);
receiver();
return 0;
}

```

O/P-1

Enter data word : 1011010101

Enter generating polynomial : 1010

Data padded with zeros : 1011010101000

CRC is : 000

Final data to be sent : 1011010101000

Enter the received data : 1011010101000

Data received : 1011010101

No Error detected

O/P-2

Enter data word : 1011010101

Enter generating polynomial : 1010

Data padded with zeros : 1011010101000

CRC is : 000

Final data to be sent : 1011010101000

Enter the received data : 1011010101001

Data received : 1011010101001

Error detected

N  
9/11/22

```
Enter data to be transmitted: 10110101010
Enter the Generating polynomial: 1010
Data padded with n-1 zeros : 1011010101000
CRC or Check value is : 000
Final data to be sent : 1011010101000Enter the received data: 1011010101000
Data received: 1011010101000
No error detected
```

12/01/2023

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

DATE  
PAGE

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

int Bellman_Ford (int G[20][20], int V, int
                   int edge[20][20])
{
    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 \rightarrow cost = %.d parent = %.d
                \n", i+1, distance[i], parent[i]+1);

return flag;
}

int main()
{
    int V, edge[20][2], G[20][20], i, j, k=0;
    printf("Enter no. of vertices");
    scanf("%d", &V);
    printf("Enter graph in matrix form :\n");
    for(i=0; i<V; i++)
        for(j=0; j<V; j++)
    {
        scanf("%d", &G[i][j]);
        if(G[i][j] != 0)
            edge[k][0] = i, edge[k][1] = j;
    }

    if(Bellman_Ford(G, V, k, edge))
        printf("\n No negative weight cycle\n");
    else
        printf("\n Negative weight cycle exists\n");
    return 0;
}

```

D/P

Enter the no. of vertices: 4

Enter graph in matrix form:

0	5	17	3
2	0	3	5
8	5	0	2
1	3	2	0

Enter source: 1

Vertex 1 → cost = 0 parent = 0

Vertex 2 → cost = 5 parent = 1

Vertex 3 → cost = 5 parent = 4

Vertex 4 → cost = 3 parent = 1

Enter no. of vertices: 4  
Enter graph in matrix form:

0	5	17	3
2	0	3	5
8	5	0	2
1	3	2	0

Enter source: 1

Vertex 1 → cost = 0 parent = 0

Vertex 2 → cost = 5 parent = 1

Vertex 3 → cost = 5 parent = 4

Vertex 4 → cost = 3 parent = 1

N  
BFI/23

12/01/2020

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

DATE / /  
PAGE / /

```
#include <stdio.h>
#include <conio.h>
#define INFINITY 9999
#define MAX 10
```

```
void dijkstra (int G[MAX][MAX], int n, int startnode)
```

```
int main()
```

```
{
```

```
int G[MAX][MAX], i, j, n, u;
```

```
printf ("Enter no. of vertices");
```

```
scanf ("%d", &n);
```

```
printf ("Enter the adjacency matrix: \n");
```

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

```
    for (j=0; j<n; j++)
```

```
        scanf ("%d", &G[i][j]);
```

```
printf ("Enter the starting node");
```

```
scanf ("%d", &n);
```

```
dijkstra (G, n, u);
```

```
return 0;
```

```
}
```

```
void dijkstra (int G[MAX][MAX], int n, int startnode)
```

```
{
```

```
int cost[MAX][MAX], distance[MAX], pred[MAX];
```

```
int visited[MAX], count, mindistance, nextnode, i, j;
```

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

```
    for (j=0; j<n; j++)
```

```
        if (G[i][j] == 0)
```

$\text{cost}[i][j] = \text{INFINITY};$

else

$\text{cost}[i][j] = G[i][j];$

{  
for ( $i=0; i < n; i++$ )

{  
 $\text{distance}[i] = \text{cost}[\text{startnode}][i];$

$\text{pred}[i] = \text{startnode};$

$\text{visited}[i] = 0;$

}

$\text{distance}[\text{startnode}] = 0;$

$\text{visited}[\text{startnode}] = 1;$

$\text{count} = 1;$

while ( $\text{count} < n - 1$ )

{  
 $\text{mindistance} = \text{INFINITY};$

for ( $i=0; i < n; i++$ )

if ( $\text{distance}[i] < \text{mindistance} \& \& \text{!visited}[i]$ )

{  
 $\text{mindistance} = \text{distance}[i];$

$\text{nextnode} = i;$

}

$\text{visited}[\text{nextnode}] = 1;$

for ( $i=0; i < n; i++$ )

if ( $\text{!visited}[i]$ )

{  
if ( $\text{mindistance}[\text{nextnode}][i] < \text{distance}[i]$ )

$\text{distance}[i] = \text{mindistance} + \text{cost}[\text{nextnode}][i];$   
 $\text{pred}[i] = \text{nextnode};$

Y  
 $\text{count}++;$

Y  
 $\text{for}(i=0; i < n; i++)$

if( $i \neq \text{startnode}$ )

{ printf("Distance of node %d = %.d", i,  
 $\text{distance}[i]);$

printf("Path = %.d", i);

j = i;

do

{

j = pred[j];

printf(" - %.d", j);

} while( $j \neq \text{startnode}$ );

Y

O/P

Enter no. of vertices: 4  
 Enter the adjacency matrix:

0	5	9999	9999
2	0	4	9999
9999	9999	0	6
4	7	5	0

Enter the starting node: 0

Distance of node1 = 5

Path = 1 ← 0

Distance of node2 = 9

Path = 2 ← 1 ← 0

Distance of node3 = 15

Path = 3 ← 2 ← 1 ← 0

✓  
 N  
 of 1/2<sup>3</sup>

Enter no. of vertices: 4

Enter the adjacency matrix:

0	5	9999	9999
2	0	4	9999
9999	9999	0	6
4	7	5	0

Enter the starting node: 0

Distance of node1 = 5

Path = 1 ← 0

Distance of node2 = 9

Path = 2 ← 1 ← 0

Distance of node3 = 15

Path = 3 ← 2 ← 1 ← 0

05/01/2023 Write a program for congestion control using leaky bucket algorithm

DATE / /  
PAGE

```
#include <iostream>
using namespace std;
int main()
{
    int no, storage, output;
    int input, bucketSize, size_left;
    printf("Enter bucket size");
    scanf("%d", &bucketSize);
    printf("Enter output");
    scanf("%d", &output);
    printf("Enter number of packets");
    scanf("%d", &no);
    storage = 0;
    for(int i=0; i<no; i++)
    {
        printf("Enter input packet size");
        scanf("%d", &input);
        size_left = bucketSize - storage;
        if(input <= size_left)
        {
            storage += input;
        }
        else
        {
            printf("Packet cannot be taken as it is
greater than bucket size\n");
        }
        storage -= output;
    }
    printf("The size left is %d", size_left);
    return 0;
}
```

O/P

Enter bucket size 500

Enter output 50

Enter number of packets 5

Enter input packet size 200

Enter input packet size 100

Enter input packet size 600

Packet cannot be taken as it is greater than  
bucket size

Enter input packet size 100

Enter input packet size 200

The size left is 300

```
Enter bucket size500
Enter output50
Enter number of packets4
Enter input packet size100
Enter input packet size100
Enter input packet size600
Packet cannot be taken as it is greater than bucket size
Enter input packet size100
The size left is 450
```

28-01-2023

DATE / /  
PAGE

## Socket (TCP/IP)

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.

### Client TCP.py

```
from socket import *  
serverName = '127.0.0.1'  
serverPort = 12000
```

```
clientSocket = socket(AF_INET, SOCK_STREAM)  
clientSocket.connect((serverName, serverPort))  
sentence = input("\n Enter file name: ")
```

```
clientSocket.send(sentence.encode())  
filecontents = clientSocket.recv(1024).decode()  
print("\n From Server:\n")  
print(filecontents)  
clientSocket.close()
```

### Server TCP.py

```
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 ("I send contents of " + sentence)
```

```
    file.close()
```

```
connectionSocket.close()
```

D/P

Server TCP

The server is ready to receive

Sent contents of Server TCP.py

The server is ready to receive

Client TCP

Enter file name : Server TCP.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()
```

while 1 :

```

print ("Enter 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()

```

>>>

```

Enter file name: ServerTCP.py
From Servers:
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()

```

>>> |

29-01-2023

DATE / /  
PAGE

## Sockets (UDP)

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.

### ClientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name: ")
clientSocket.sendto(sentence.encode("utf-8"), (serverName, serverPort))
```

filecontents, serverAddress = clientSocket.recvfrom(2048)

```
print("\nReply from Server:\n")
print(filecontents.decode("utf-8"))
for i in filecontents:
    print(str(i), end = " ")
clientSocket.close()
```

### ServerUDP.py

```
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('Send contents of ', end='')
```

```
print(sentence)
```

```
# for i in sentence
```

```
# print(str(i), end='')
```

```
file.close()
```

O/P

ServerUDP

The server is ready to receive

Send contents of ServerUDP.py

The server is ready to receive

Client UDP

Enter the filename: ServerUDP.py

Reply from Server:

```
from socket import *
```

```
serverPort = 12000
```

```
serverSocket = socket(AF_INET, SOCK_DGRAM)
```

serverSocket.bind(("127.0.0.1", serverPort))  
while True:

```
print("The server is ready to receive")
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("\nSend contents of", end=" ")
print(sentence)
file.close()
```

>>>

```
The server is ready to receive
Sent contents of ServerUDP.py
The server is ready to receive
```

```
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))

while True:
    print("The server is ready to receive")
    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("\nSend contents of", end=" ")
    print(sentence)
    # for i in sentence:
    #     print(str(i), end=" ")
    file.close()

>>>
```