VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COURSE TITLE

Submitted by

KOTTURU AMARNATH (1BM20CS074)

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 **KOTTURU AMARNATH (1BM20CS074),** 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
Assistant Professor
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.	

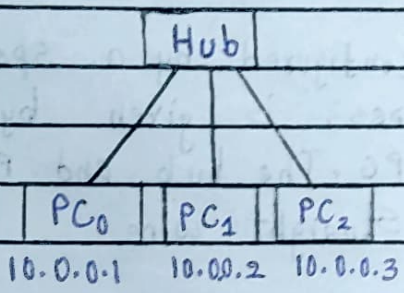
Program - 1

Date: 10/11/22
Page No.:

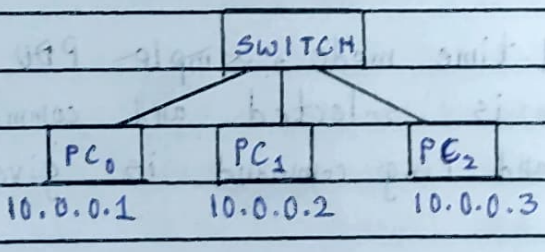
Aim: Creating a topology and simulating a simple PDU from source to destination using hub and switch as connecting devices

Topology

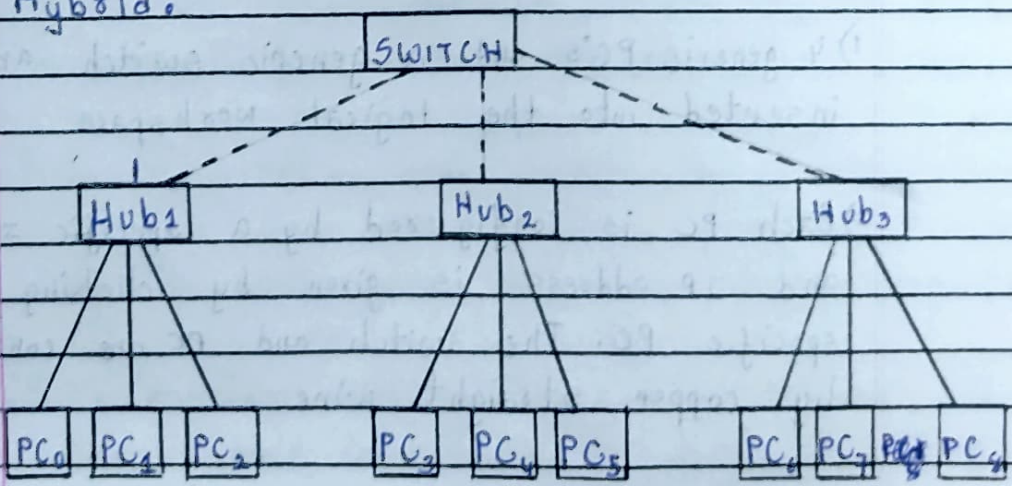
HUB:



SWITCH:



Hybrid:



Date :
Page No.:

Procedure:

Hub

1) 7 generic PC's and a generic hub are inserted into the logical workspace

2) Each PC is configured by a specific IP address and IP address is given by clicking on a specific PC. The hub and PC's are connected by copper Straight wire

3) In simulation mode, simple PDU is established b/w to send devices and Packet transfer can be seen

4) In real-time mode, simple PDU is established one PC is selected and command Prompt is opened and Ping command is given

Switch

1) 4 generic PC's and a generic switch are inserted into the logical workspace

2) Each PC is configured by a specific IP address and IP address is given by clicking on a specific PC. The switch and PC are connected by copper straight wire

Date :
Page No.:

3) In simulation mode simple PDU is established two end devices and auto capture is clicked the Packet transfer can be seen b/w the switch and PC's

4) In real-time mode, any PC can be selected and command prompt is opened and ping IP-address is given

Hybrid

1) 12 generic PC's, 3 Hubs and switch is taken 4 generic PC's are connected to each hub and all 3 hubs are connected to switch.

2) 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 address of all PC's are given by clicking on them.

3) In simulation mode simple PDU is established between two end devices and Packet transfer can be seen from the source to the destination.

4) In real-time mode, a PC can be selected and the command Prompt can be opened and ping command is given and output can be seen on the command prompt screen.

Date :
Page No.:

Observation:

1) Hub

Learning Outcome:

The hub sends message to all the end devices except to the one it receives the message from but the message is read only by the specified destination and destination responds by sending a packet

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 = 1ms 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 = 1ms, Average = 0ms

2) Switch

Learning Outcome:

+) switch does not establish connection immediately there is a certain time called learning time and message cannot be done until the green light connection is established

Date :
Page No.:

*) Switch only sends the message to the end device (Reciever)

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 = 1ms 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 = 1ms, Average = 0ms

3) Hybrid

Learning Outcome:

The switch first sends the message to all the hubs then learns about the IP address of the end system. Then it only sends message to the hub to which the end device is connected

Result:

Ping 10.0.0.12

pinging 10.0.0.12 with 32 bytes of data

:

Ping statistics for 10.0.0.12:

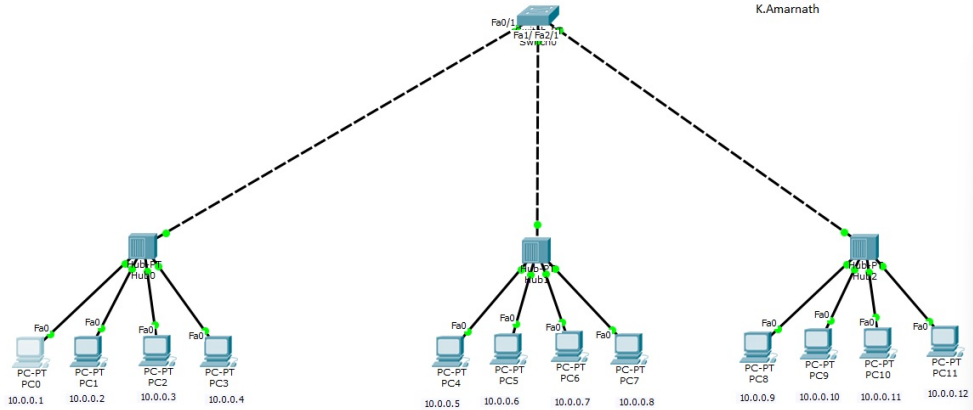
Packets: sent = 4, Received = 4, Lost = 0

Cisco Packet Tracer Student

File Edit Options View Tools Extensions Help

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

1BM20CS074
K.Amarnath



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.12

Pinging 10.0.0.12 with 32 bytes of data:

Reply from 10.0.0.12: bytes=32 time=0ms TTL=128
Reply from 10.0.0.12: bytes=32 time=0ms TTL=128
Reply from 10.0.0.12: bytes=32 time=0ms TTL=128
Reply from 10.0.0.12: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

Time: 00:13:37 Power Cycle Devices Fast Forward Time

Realtime

Connections

Copper Cross-Over

Scenario 0 | New | Delete | Toggle PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC11	ICMP		0.000	N	0	(edit)	(delete)

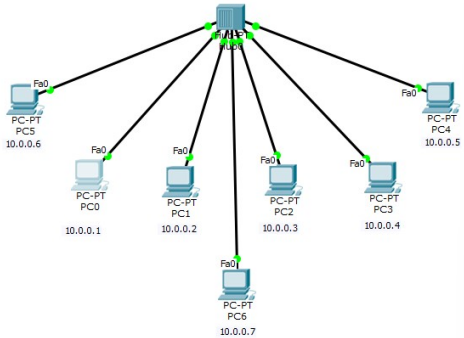
Cisco Packet Tracer Student

File Edit Options View Tools Extensions Help

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

1BM20CS074

K.Amarnath



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

```
Reply from 10.0.0.1: bytes=32 time=4ms TTL=128

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 = 12ms, Average = 6ms

PC>clr
Invalid Command.

PC>clear
Invalid Command.

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

PC>
```

Time: 01:37:57 | Power Cycle Devices | Fast Forward Time | Realtime

Connections

Copper Straight-Through

Scenario 0 | New | Delete | Toggle PDU List Window

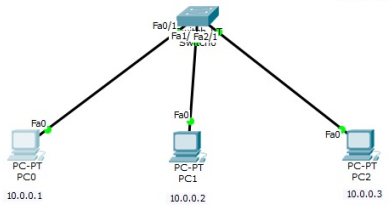
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC4	ICMP		0.000	N	0	(edit)	(delete)

Cisco Packet Tracer Student

File Edit Options View Tools Extensions Help

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

1BM20CS074
K.Amarnath



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

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

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms 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 = 1ms, Average = 0ms

PC>
```

Time: 00:06:52 | Power Cycle Devices | Fast Forward Time | Realtime

Connections

Copper Straight-Through

Scenario 0 | New | Delete | Toggle PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC2	ICMP		0.000	N	0	(edit)	(delete)

26°C Partly sunny

Search

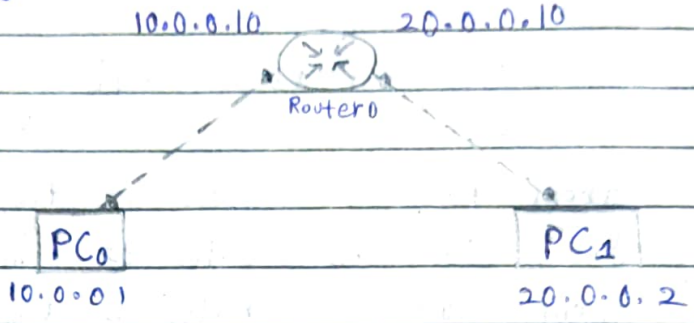
ENG IN 12:21 10-11-2022

Date: 17/11/22
Page No.:

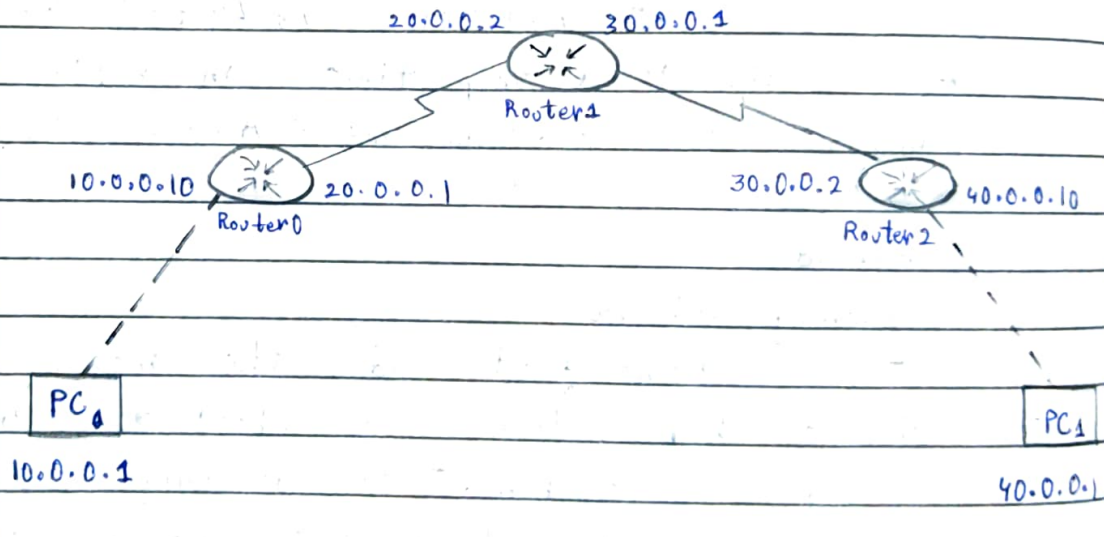
Lab-2

AIM: Configure IP address to Router in packet tracer. Explore the following messages ping Responces, Destination unreachable, Request time out, Reply

Topology:



Final Topology:



Date :
Page No.:

Procedure:

For single Router and Two PC's

1) 2 PC's and one router are inserted to the workspace, connect them by using copper cross-over

2) Each PC is configured by a specific IP address and IP address is given by clicking on a specific pc, after entering IP address now enter Gateway for both PC's

3) In the router go to CLI and type the following commands

*) First enter no
*) Router > enable
Router # Config t
Router (config) # interface Fasteternet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # interface Fasteternet 1/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # exit
Router # exit
Router >

4) After entering these commands the lights between PC and Router will become green

For Three Router and Two PC's

1) Add three Router's and two PC's to the workspace and connect PC's with the Router by using copper cross over and connect Router's by using Serial DCE.

2) Each PC is configured by a specific IP address and IP address is given by clicking specific PC after that enter gateway for both PC's

3) After entering IP address and gateways enter the following commands for all three Routers. In the router0 go to CLI and follow these commands

```
Router >enable
Router # config t
Router(config) # interface Fastethernet 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) # interface serial 2/0
Router(config-if) # ip address 20.0.0.1 255.0.0.0
Router(config-if) # no shut
Router(config-if) # exit
Router(config) # exit
Router # exit
Router >
```

Date :
Page No.:

4) Now for router1 follow the similarly follow the the same above commands with ip address as 20.0.0.2 for se 2/0 and 30.0.0.1 for se 3/0

5) Now for router2 follow same commands with ip address as 30.0.0.2 for se 3/0 and 40.0.0.10 for fastethernet 0/0

6) After entering all these commands all the lights are turned green. so that circuit is completed.

7) In next step we have to teach the router of all the networks (static routing)

For Router 0;

ip route 30.0.0.0 255.0.0.0 20.0.0.2
ip route 40.0.0.0 255.0.0.0 20.0.0.2

For Router 1;

ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 40.0.0.0 255.0.0.0 30.0.0.2

For Router 2;

ip route 10.0.0.0 255.0.0.0 30.0.0.1
ip route 20.0.0.0 255.0.0.0 30.0.0.1

8) After these commands pinging can be done between PC0 and PC1

Date :
Page No.:

Observation

1) For single Router

Learning Outcome:

We used router to set a connection b/w two end devices at first it shows "Request timed out" after we ping from one end device to another before showing the result.

Result

Ping 20.0.0.2

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 = 0ms TTL = 127
Reply from 20.0.0.1: bytes = 32 time = 0ms TTL = 127

2) Ping statistics for 20.0.0.1:

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

Approximate round trip time in milli-seconds:

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

2) For three Routers

Learning outcome:

*) when we first try to ping 40.0.0.1 from 10.0.0.1 we get a message destination host unreachable

*) when we try to ping 20.0.0.2 from PC0 we get Request timed out.

Result:

Initially only 3 packets is sent 1 packet is lost after that Reply can be seen

Ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes = 32 Time = 7ms TTL = 125

Reply from 40.0.0.1: bytes = 32 Time = 8ms TTL = 125

Reply from 40.0.0.1: bytes = 32 Time = 5ms TTL = 125

Reply from 40.0.0.1: bytes = 32 Time = 11ms TTL = 125

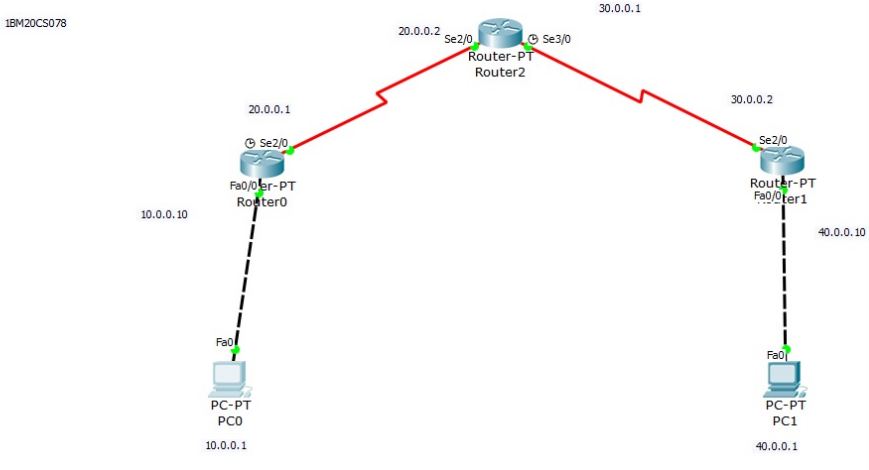
Ping statistics for 40.0.0.1:

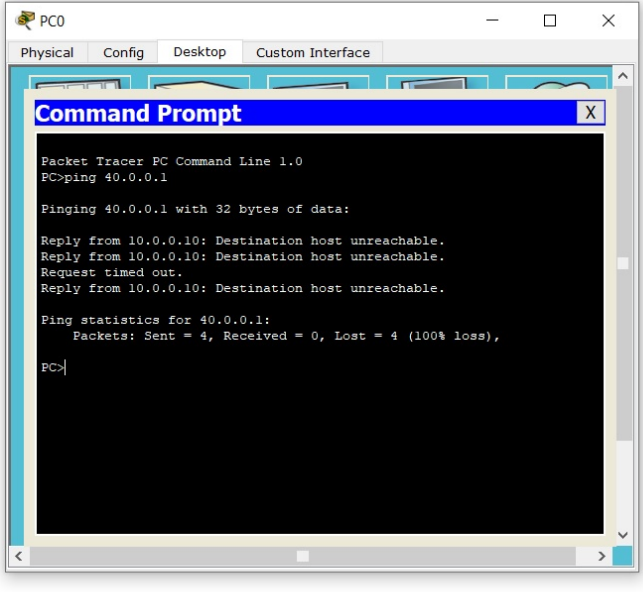
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss),

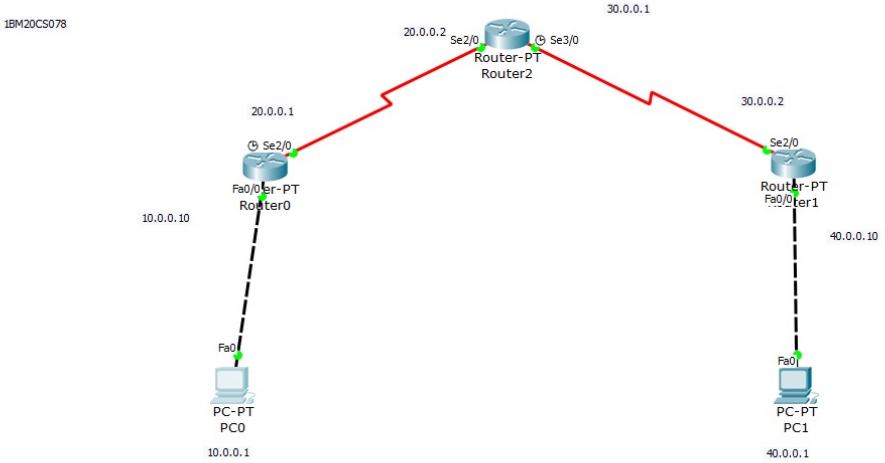
Approximate round Trip time in milli-seconds:

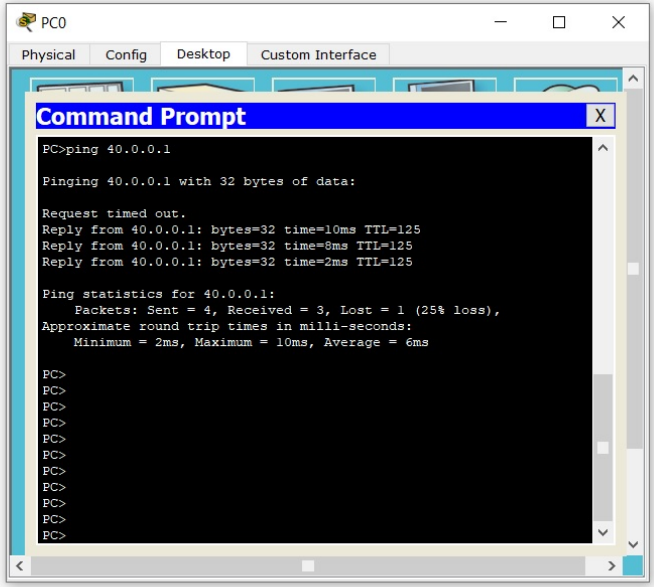
Minimum = 5ms, Maximum = 11ms, Average = 7ms

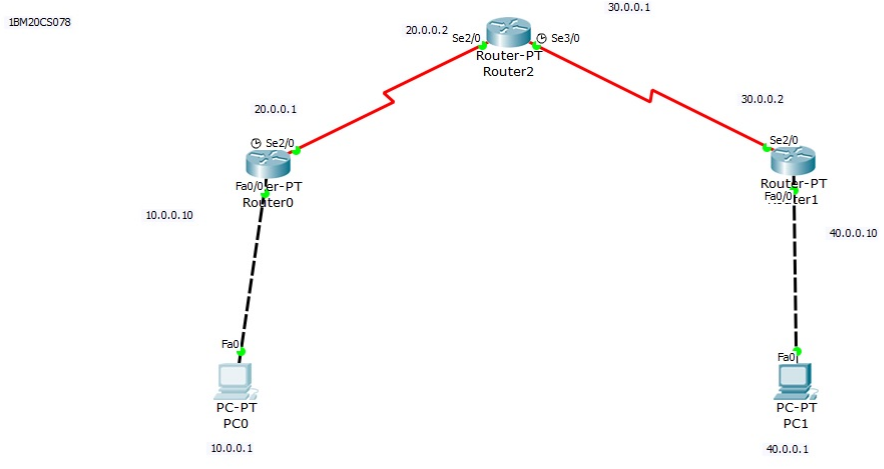
24/11/22

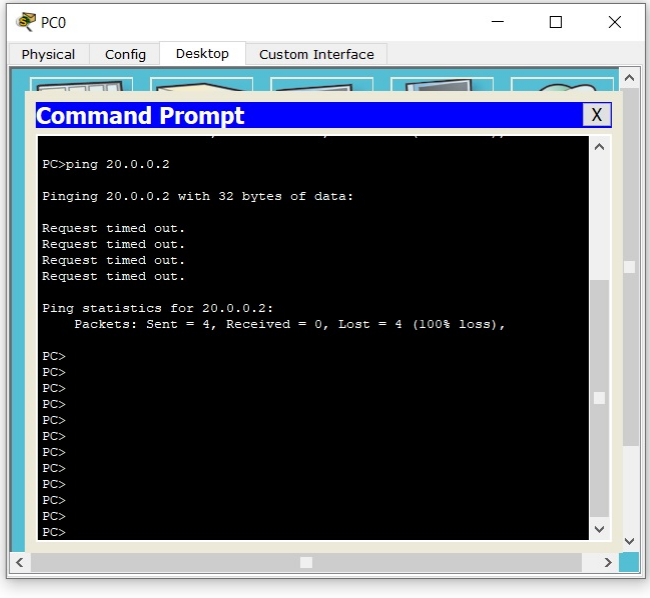

1BM20CS078
30.0.0.1
20.0.0.2
Se2/0
Se3/0
Router-PT
Router2
20.0.0.1
30.0.0.2
Se2/0
Se2/0
Fa0/0
Router-PT
Router0
Router-PT
Router1
10.0.0.10
40.0.0.10
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
10.0.0.1
40.0.0.1


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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>


1BM20CS078
20.0.0.2
Se2/0
Se3/0
Router-PT
Router2
30.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
30.0.0.2
Se2/0
Router-PT
Fa0/0
Router1
10.0.0.10
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 10ms, Average = 6ms
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>


1BM20CS078
20.0.0.2
Se2/0
Se3/0
Router-PT
Router2
30.0.0.1
20.0.0.1
Se2/0
30.0.0.2
Se2/0
Fa0/0
Router-PT
Router0
Router-PT
Fa0/0
Router1
10.0.0.10
40.0.0.10
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
10.0.0.1
40.0.0.1

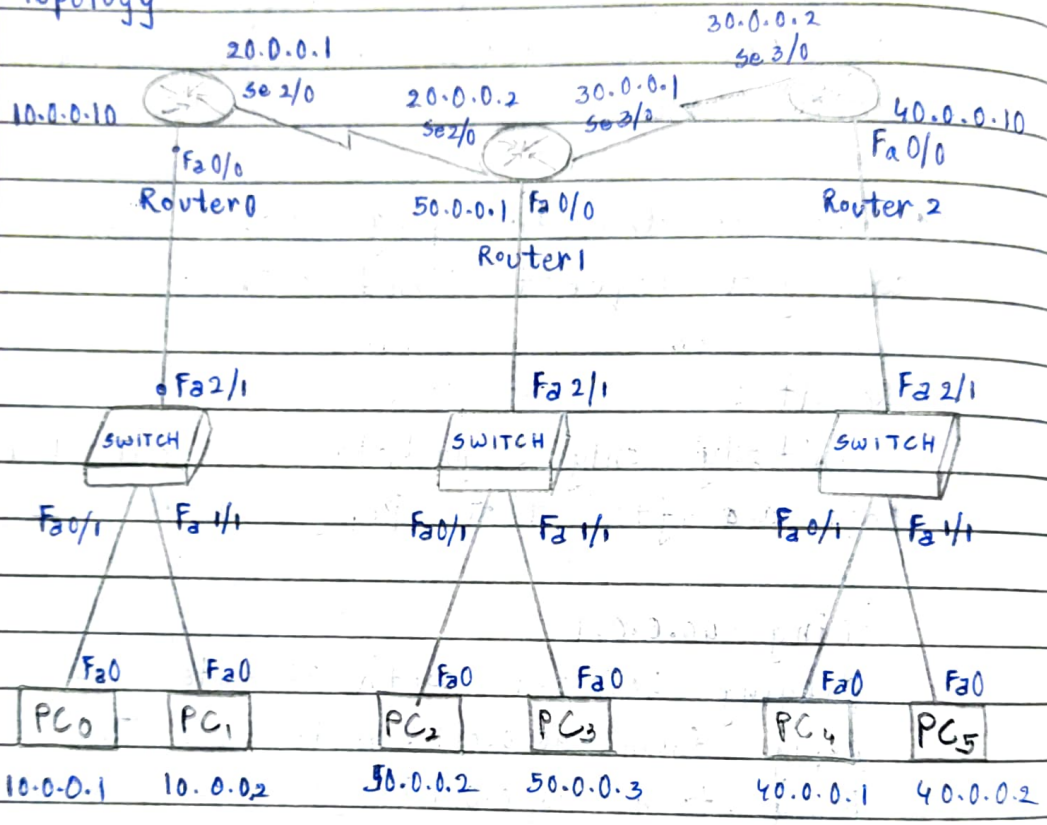

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>

Lab - 3

Date :
Page No.:

AIM: Configuare default route to the routers

Topology



Procedure:

1) Add three routers, three switches and six PC's to the workspace

2) Configuare ~~all~~ IP's of each PC's and also give gateways for all pc's. All connection between ~~PC's and~~ PC-switch and switch-Router are made using copper straight through. Connections between two routers are made using serial DCE

3) After config this in the router0 go to CLI and then follow these commands

```
Router>enable
Router # config t
Router(config) # interface fa 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) # interface serial 2/0
Router(config-if) # ip address 20.0.0.1 255.0.0.0
Router(config-if) # no shut
Router(config-if) #exit
Router(config) #exit
Router #exit
Router>
```

4) Follow same commands for Both Router2 and Router1, After performing all these operations on these routers all the lights will be turned green.

5) For Router0 and Router2 default routing can be done but for Router1 doesn't have any default routing & static routing is done for the Router1

Date :
Page No.:

Observation :-

Learning Outcome :-

→ Router cannot have two default routing.

→ The default router for ~~Routers~~ first router is the middle router because any packets which have to be delivered will go to the middle router.

→ Middle router doesn't have any default router because if one of the router is made default then there is a chance that the packets which are sent to the switch are sent to the router.

→ ~~Router~~ Router1 is default Router for 3rd Router because ~~packets~~ will be delivered only to Router1

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=12
Reply from 40.0.0.1 : bytes = 32, Time = 2ms, TTL=12
Reply from 40.0.0.1 : bytes = 32, Time = 2ms, TTL = 125

Ping statistics for 40.0.0.1:
    Packets : Sent = 4, Received = 3, Lost = 1 (25% Loss),
```

ACHIEVER

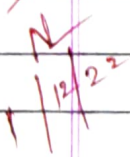
Approximate round trip time in milli-seconds:
Minimum = 2ms, Maximum = 23ms, Average = 9ms

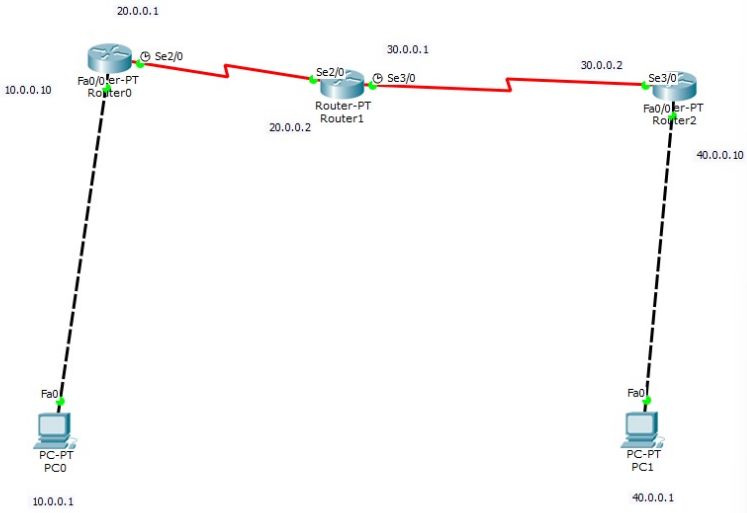
Ping 40.0.0.1

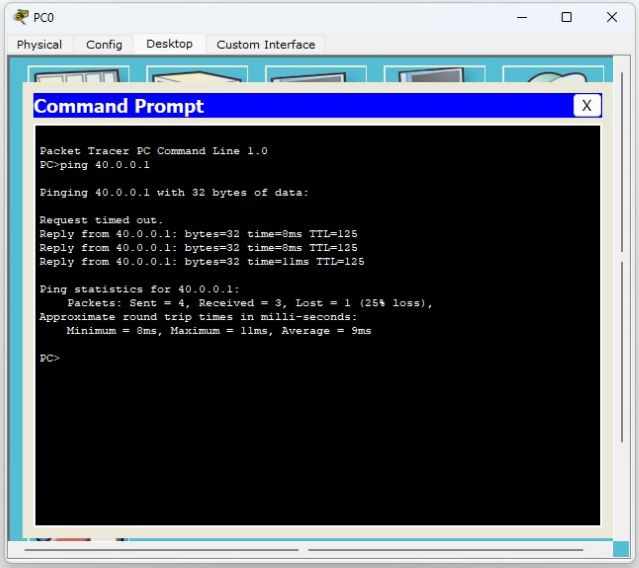
Pinging 40.0.0.1 with 32 bytes of data:

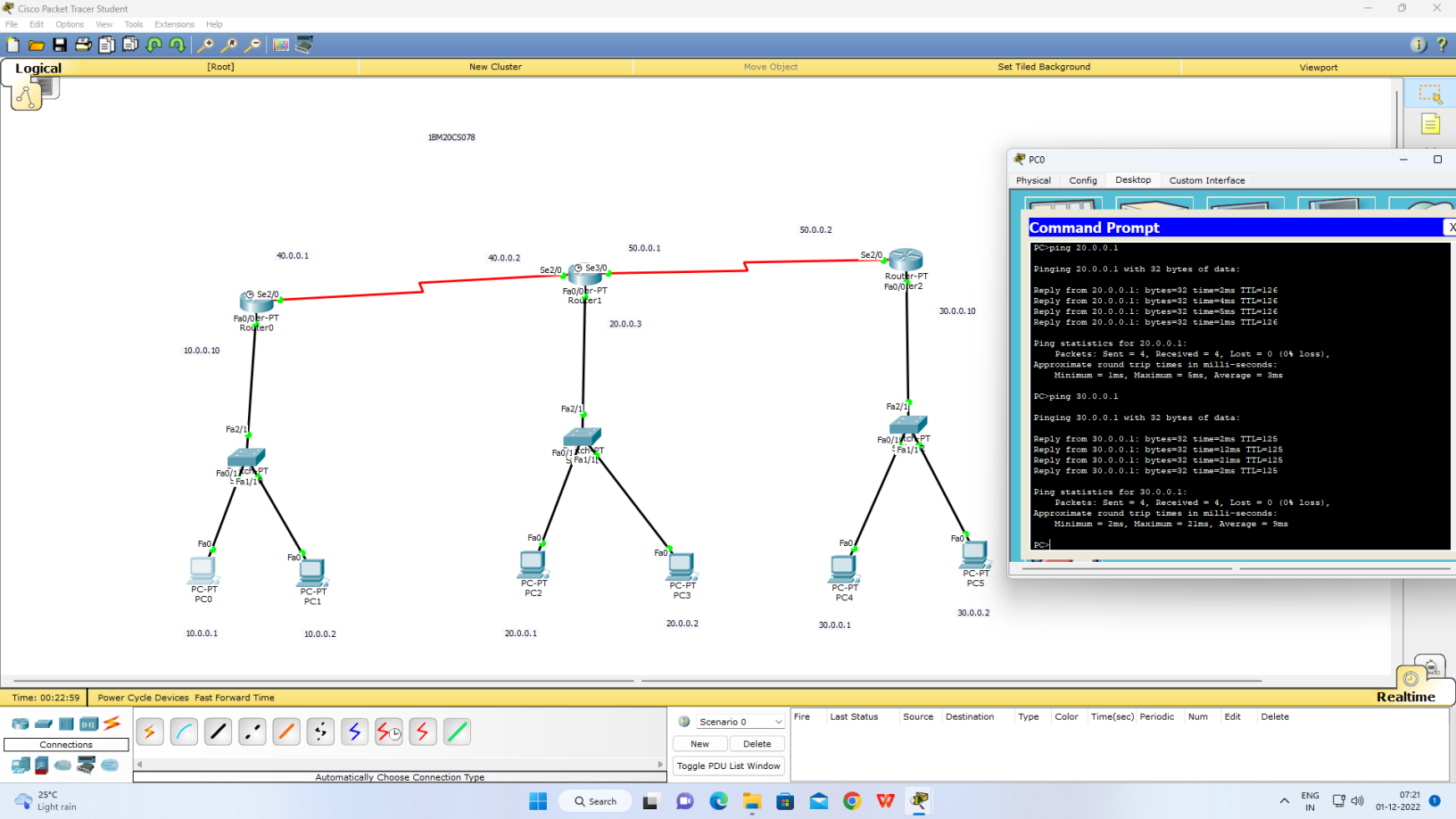
Reply from 40.0.0.1 : bytes=32 time=3ms TTL = 125
Reply from 40.0.0.1 : bytes=32 time=3ms TTL = 125
Reply from 40.0.0.1 : bytes=32 time=8ms TTL = 125
Reply from 40.0.0.1 : bytes=32 time=2ms TTL = 125

Ping statistics for 40.0.0.1
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 4ms




20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
10.0.0.10
30.0.0.1
Se2/0
Se3/0
Router-PT
Router1
20.0.0.2
30.0.0.2
Se3/0
Fa0/0
Router-PT
Router2
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=11ms 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 = 8ms, Maximum = 11ms, Average = 9ms
PC>

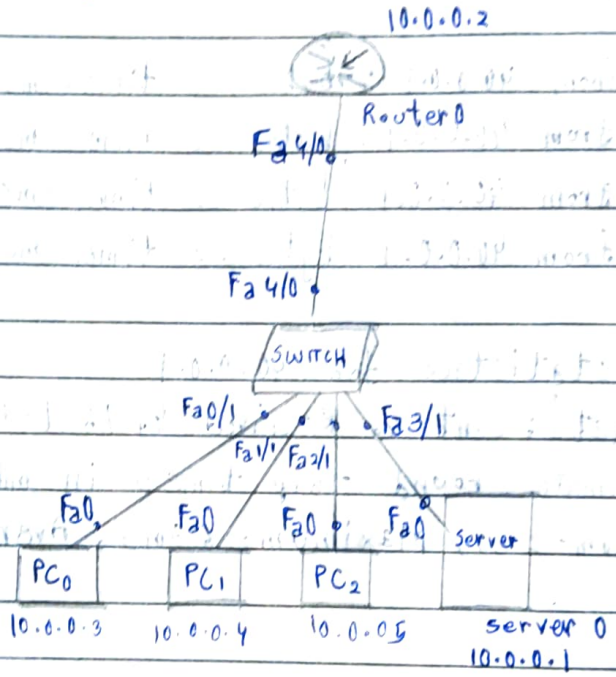

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
1BM20CS078
40.0.0.1
40.0.0.2
50.0.0.1
50.0.0.2
Se2/0
Se3/0
Router-PT
Router0
Router1
Router2
20.0.0.3
30.0.0.10
10.0.0.10
Fa2/1
Fa0/1
Fa1/1
PC-PT
PC0
PC1
PC2
PC3
PC4
PC5
Fa0
10.0.0.1
10.0.0.2
20.0.0.1
20.0.0.2
30.0.0.1
30.0.0.2
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
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=2ms TTL=126
Reply from 20.0.0.1: bytes=32 time=4ms TTL=126
Reply from 20.0.0.1: bytes=32 time=5ms TTL=126
Reply from 20.0.0.1: bytes=32 time=1ms TTL=126
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 5ms, Average = 3ms
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=2ms TTL=125
Reply from 30.0.0.1: bytes=32 time=12ms TTL=125
Reply from 30.0.0.1: bytes=32 time=21ms TTL=125
Reply from 30.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 21ms, Average = 9ms
PC>
Time: 00:22:59
Power Cycle Devices Fast Forward Time
Realtime
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
25°C
Light rain
Search
ENG
IN
07:21
01-12-2022

Date: 1/12
Page No:

Lab-4

AIM: Configuring DHCP within a LAN in a Packet Tracer

Topology:



Procedure:

1) Add 3 PC's, 1 server, 1 switch and 1 Router to workspace. Connect PC, server to swith by using copper straight-through wire.
2) Configure the server by giving the IP address and gateway.
3) Now go to Router and open CLI and follow these commands

Router > enable
Router # config t
Router (config) # interface Fastethernet 4/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # exit
Router # exit
Router -

4) Now go to services in Server0 go to DHCP change service from off to on. Now give start IP address 10.0.0.3 and give all servers as 10.0.0.1

5) Now open IP configuration in Desktop change IP configuration from static to DHCP Follow the same for all other PC's

6) Now Ping can be done from any PC.

Observation

Learning outcome:

Server automatically provides IP address for all the PC's.

Follows below procedure

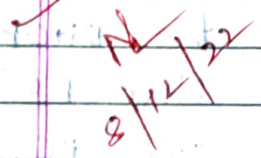
D - Discover
O - OFFER
R - Request
A - Acknowledgement

Result: Ping 10.0.0.05

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 time in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

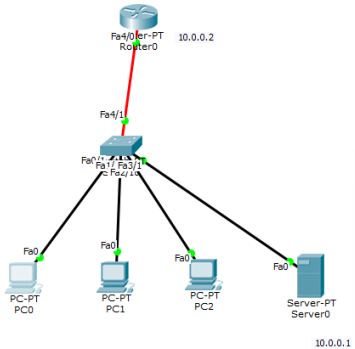


Cisco Packet Tracer Student - C:\amar\1-12-22\DYNAMIC HOST CONFIGURATION.pkt

File Edit Options View Tools Extensions Help

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

K.AMARNATH
1BM20CS074



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt X

```
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=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 milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

Time: 00:50:36 | Power Cycle Devices | Fast Forward Time | Realtime

End Devices

Scenario 0 | New | Delete | Toggle PDU List Window

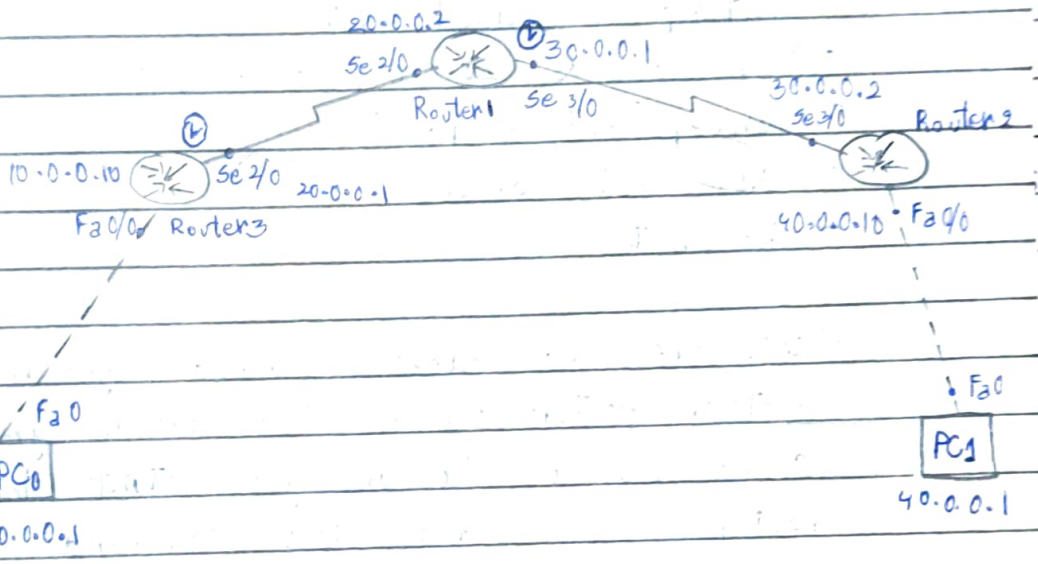
Fire | Last Status | Source | Destination | Type | Color | Time(sec) | Periodic | Num | Edit | Delete

Automatically Choose Connection Type

Lab-5

Aim: Configuring RIP Routing Protocol in Routers

Topology:



Procedure:

1) Place 3 Generic-routers, 2 Generic-PC's and place notes to indicate respective IP addresses.
2) Use copper cross over cable to connect PC's with routers and serial DCE to connect router's
3) Set IP address, gateway and subnet mask as 10.0.0.1, 10.0.0.10, 255.0.0.0 for PC0 and 40.0.0.1, 40.0.0.10 and 255.0.0.0 for PC1

4) For Interfaceing : PC0 and router3 and serial 2/0 of router3 use the following commands

→ enable
→ config-t
→ interface fastethernet 0/0
→ no shut
→ exit
→ interface serial 2/0
→ ip address 20.0.0.1 255.0.0.0
→ encapsulation ppp
→ clock rate 64000
→ no shut

5) Use the same above commands for interfacing another router which has clock symbol in one cable near to it and for other interfaces of routers use same above commands but don't use "clock rate 64000" command

6) Once all lights turn green you have set RIP protocal by using following commands

→ Router RIP
→ network 10.0.0.0
→ network 20.0.0.0
→ exit

7) Repeat above commands for all all routers with network address they known.

Date :
Page No.:

Observation

Learning Outcome :

→ Instead of using static IP routing for all routers by using RIP protocal routing becomes easy when large number of routers are present

Result : ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data :

Reply from 40.0.0.10 : bytes = 32 time = 14ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time = 14ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time = 12ms TTL = 125

Ping statistics for 40.0.0.1 :

Packets : Sent = 4, Received = 4, Lost = 0 (0% Loss)
Approximate round trip times in milli-seconds :
Minimum = 2ms, Maximum = 14ms, Average = 10ms

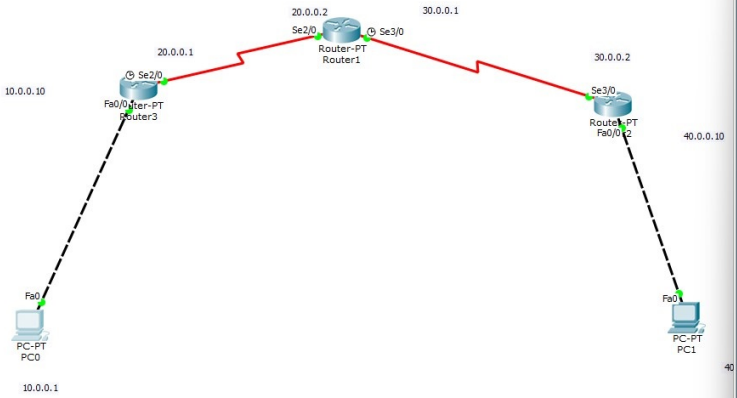
Cisco Packet Tracer Student

File Edit Options View Tools Extensions Help

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

K.Amarnath

1BM20CS074



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

```
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=14ms TTL=125
Reply from 40.0.0.1: bytes=32 time=7ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms 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 = 7ms, Maximum = 14ms, Average = 10ms

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time=14ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=14ms TTL=125
Reply from 40.0.0.1: bytes=32 time=12ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 14ms, Average = 10ms

PC>
```

Time: 00:42:47 | Power Cycle Devices | Fast Forward Time | Realtime

Connections | Serial DCE

Scenario 0 | New | Delete | Toggle PDU List Window

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

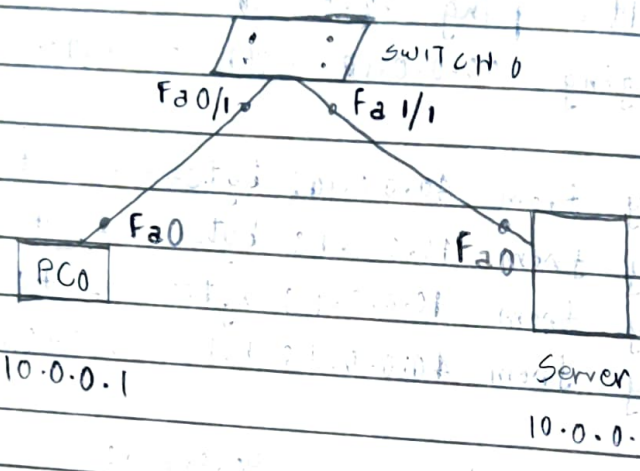
25°C Haze | Search | ENG IN | 12:20 08-12-2022

Lab-6

Date: 15/12/22
Page No.:

Aim: Demonstration of WEB server and DNS using packet Tracer

Topology:



Procedure:

1) Add a PC, switch and server in the workspace and connect them using copper straight-Through
2) Configure the IP address of PC and Server as 10.0.0.1 and 10.0.0.2
3) Now go to the server and open services Tab click on HTTP. Edit the index.html file and save.
4) To activate DNS go to DNS server turn it on give a Name and IP-address of server and press add

5) After Now open desktop in PC and go to web services / web Browser and search for the domain name you gave.

6) Now write html code for your CV

Observation:

Learning outcome: Humans are capable of remembering names better than numbers, so DNS helps is nameing the ip-addrees for ease of the use of the users.

Result:

Web Browser

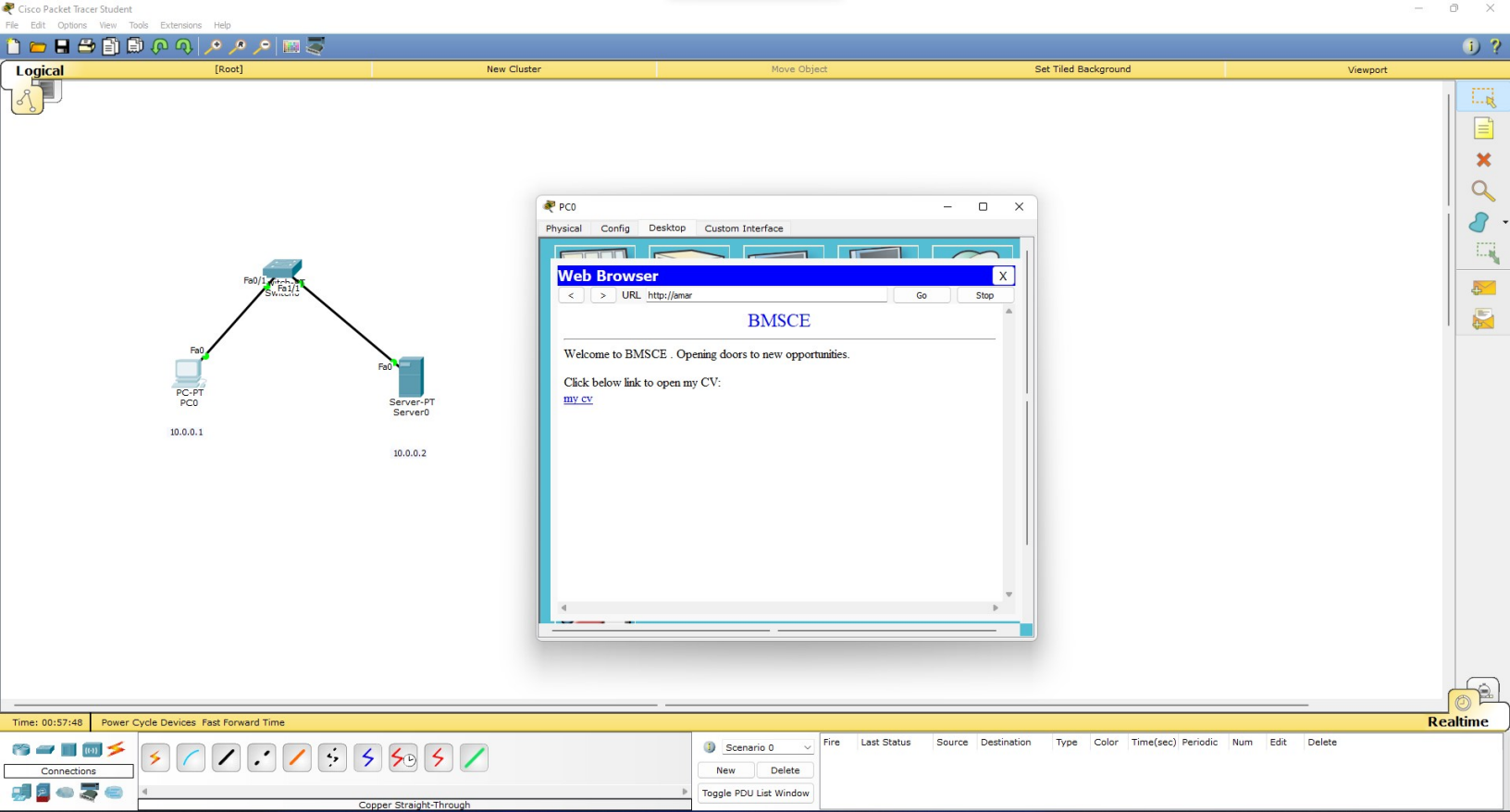
< > URL http://amax Go Stop

BMSCE

Welcome to Bmsce. Opening Door to new opportunities

Click below link to open my CV:

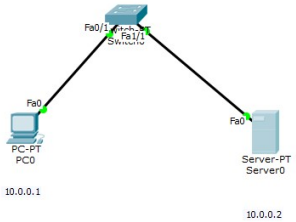
my CV


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/1
Fa1/1
Fa0
PC-PT
PC0
10.0.0.1
Fa0
Server-PT
Server0
10.0.0.2
PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL http://amar
Go
Stop
BMSCE
Welcome to BMSCE . Opening doors to new opportunities.
Click below link to open my CV:
my cv
Time: 00:57:48
Power Cycle Devices Fast Forward Time
Realtime
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete

Cisco Packet Tracer Student

File Edit Options View Tools Extensions Help

Logical [Root] New Cluster



Time: 01:03:30 Power Cycle Devices Fast Forward Time

Connections

Copper Straight-Through

New Delete Toggle PDU List Window

PC0

Physical Config Desktop Custom Interface

Web Browser X

< > URL http://amar/mycv.html Go Stop

Kotturu Amarnath AMARNATH.CS20@BMSCE.AC.IN

Educational Qualifications

Qualification	Board	Percentage / Grades	Year
S.S.C	G.S.E.B India	75.57%	2004
H.S.C (Science Stream)	G.S.E.B India	72.40%	2006
GDCA (Grad. Dipl in Computer Applications)	NIE (National Institute Of Engineering), Mumbai, India	A Grade	2008
B.Sc. (Hons) - Applied Accounting	Oxford Brookes University	2:1 ? Upper Second Class Honours	2012
Chartered Accounting - ACCA	ACCA Glasgow, UK	Passed (1st Attempt)	2013

Independent Courses

- **HTML & CSS for Beginners ? Web Fundamentals** ? Codecademy.com
- **Python ? Fundamentals and Dynamic Programming** - Codecademy.com
- **JavaScript ? Programming Basics, JS Apps and Build Games** - Codecademy.com
- **CS101: Introduction to Computer Science - Building a Web Crawler** - Udacity.com
- **CS50x ? Introduction to Computer Science I** ? edX.org & Harvard University
- **Calculus One** - Ohio State University & Coursera.org
- **Introduction to Finance** - Coursera.org & University of Michigan

Technical Skills

- **Operating Systems:** DOS, Windows 98, Windows 2000, Windows XP, Windows NT, Windows Server 2003, Windows Vista, Windows 7, Macintosh Computers (OS X), Linux (Ubuntu, Fedora)
- **Application Software:** Office 97-2003; Office XP, Office 2007, Office for Mac 2011, iWork ?09, Sage Accounting Software, Sage 50 (Accounting Software), Excel 2003/2007 for Financial Modelling spreadsheets.
- **Programming Skills:**HTML, CSS, Python, JavaScript, learning C and C++

Certifications / Awards:

- Scored highest in ACCA P1 ? Governance, Risk & Ethics exam ? June 2012 session amongst full time international students at Kaplan Financial, London.
- Interviewed by ACCA for ?international ACCA student in UK?, interview published in January 2012 edition of ACCA Student Accountant Magazine.

Date: 29/12/22
Page No.:

lab-7

① Write a Program for error detecting code using CRC-CCITT

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

char t[28], cs[28], g[10];
int a, e, c;

void xorfunction(){
    for (c=1; c<N; c++)
    cs[c] = ((cs[c] == g[c]) ? '0' : '1');
}

void crc(){
    for (e=0; e<N; e++)
      cs[e] = t[e];
    do{
       if (cs[0] == '1')
          xorfunction();
       for (c=0; c<N-1; c++)
          cs[c] = cs[c+1];
       cs[c] = t[e++];
    }
    while (e<=a+N-1);

int main()
  printf("Enter data: ");
  scanf("%s", t);
  printf("Enter generateing polynomial: ");
  scanf("%s", g);
  printf("------");
  printf("Generating polynomial: %s", g);
```

```
a = strlen(t);
for (e=a; e<a+N-1; e++)
    t[e] = '0';
printf("\n.....");
printf("\nPadded data is: %s", t);
printf("\n.....");
crc();
printf("\nCRC is: %s", cs);
for (e=a; e<a+N-1; e++)
    t[e] = cs[e-a];
printf("\n.....");
printf("\nFinal data to be sent: %s", t);
printf("\n.....");
printf("\nTest error detection 0(yes) 1(no)? :");
scanf("%d", &e);
if (e==0)
{
    do{
        Printf("\nEnter the position where error is to
                be Inserted");
        scanf("%d", &e);
    }
    while (e==0 || e>a+N-1);
    t[e-1] = (t[e-1]=='0')? '1' : '0';
    printf("\nErroneous data: %s\n", t);
}
crc();
for (e=0; (e<N-1) && (cs[e]!='1'); e++);
    if (e<N-1)
        Printf("\nError detected);
```

```
    else
    printf("No error detected");
    return 0;
}
```

Output:

```
Enter data : 1011010101
Enter generating polynomial : 1010
Generateing polynomial: 1010
padded data is: 1011010101000
Crc is (Remainder) : 000
Final data to be sent : 1011010101000

Test error detection 0(yes) 1(no)? 0

Enter the position where error is to be inserted :2

Erroneous data: 1111010101000
Error detected
```


C:\amar\crc.exe
Enter data : 1011010101
Enter generating polynomial : 1010

Generating polynomial : 1010

Paded data is : 1011010101000

CRC is : 000

Final data to be sent : 1011010101000

Test error detection 0(yes) 1(no)? : 0
Enter the position where error is to be inserted : 2

Erroneous data : 1111010101000
Error detected

Process returned 0 (0x0) execution time : 855.580 s
Press any key to continue.

12-01-22

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

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

int Bellman_Ford(int G[20][20], int v, int i,
                 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;
}
```

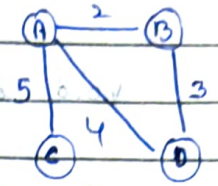
```
if (flag)
    for (i=0; i<v; i++)
        printf("Vertex %d -> cost = %d  parent = %d
                \n", i+1, distance[i], parent[i]+1);

    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][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;
}
```

Output:

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



0 2 5 4
2 0 999 3
5 999 0 999
4 3 999 0

Enter source: 1

Vertex 1 → cost = 0 parent = 0
Vertex 2 → cost = 2 parent = 1
Vertex 3 → cost = 5 parent = 1
Vertex 4 → cost = 4 parent = 1

No negative weight cycle

```
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
Enter matrix:1 99 99 99

Enter 1 if the corresponding router is adjacent to routerC else enter 99:
 A B D E
Enter matrix:1 99 1 1

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

Lab - 10 - Dijkstra

```
#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, v;
  Printf("Enter no of vertices");
  scanf("%d", &n);
  printf("\n 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;
  int i, j;
  for(i=0; i<n; i++)
     for(j=0; j<n; j++)
        if(G[i][j]==0)
```

```
        cost[i][j] = INFINITY;
    else
    cost[i][j] = G[i][j]
}
```

Output

```
Enter the graph
0 9 2 5
9 0 6 8
2 6 0 0
5 8 0 0
```

Vertex	Distance from Source
0	0
1	8
2	2
3	5

```
Enter the graph
0 9 2 5
9 0 6 8
2 6 0 0
5 8 0 0
Vertex                  Distance from Source
0                       0
1                       8
2                       2
3                       5


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

lab. 8

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
 int no_of_queries, storage = 0;
 output_pkt_size;
 int input_pkt_size, bucket_size,
size_left;
 cout << "Enter no of queries:";
 cin >> no_of_queries;
 cout << "Enter the bucket size:";
 cin >> bucket_size;
 cout << " Enter output packet
size:";
     cin >> output_pkt_size;
  for (int i = 0; i < no_of_queries; i++)
    {
     cout << "Enter input packet size:";
     cin >> input_pkt_size;
     size_left = bucket_size - storage;
     if (input_pkt_size <= size_left)
     {
      storage += input_pkt_size;
     }
     else
     {
      cout << " packet loss = " << input_pkt_
        size << endl;
     }
     cout << " Buffer size=" <<
```

Achi ver

```
storage << " out of bucket size=" <<
bucket_size << endl;
    storage -= output_pkt_size;
    if (storage < 0)
        storage = 0;
    cout << " after output new buffer size" <<
    storage << endl;
}
return 0;
}
```

Output

```
Enter no of queries: 3
Enter the bucket size: 100
Enter output packet size: 50
Enter input packet size: 100
Buffer size = 100 out of bucket size = 100
after output new buffer size = 50

Enter input Packet size: 150
Packet loss = 150
Buffer size = 50 out of bucket size = 100
after output newbuffer size = 50
Enter input packet size: 20
Buffer size = 20 out of bucket size = 100
after output new buffer size = 0
```

```
Enter no of queries:3
Enter the bucket size:100
Enter output packet  size:50
Enter input packet size:100
Buffer size=100 out of bucket size=100
after output new buffer size=50
Enter input packet size:150
Packet loss = 150
Buffer size=50 out of bucket size=100
after output new buffer size=0
Enter input packet size:20
Buffer size=20 out of bucket size=100
after output new buffer size=0


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

11. TCP Socket

Q) Using TCP/IP sockets, write a client server program to make client sending file name and server to send back the contents

→ A server has find() method which binds specific IP and it can listen to incoming requests.
→ Passing an empty string means that the server can listen to incoming connections from other computer as well.
→ Server is in listening mode.
→ At last, we make a while loop and start to accept all incoming connections and close those connections

Client.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentance = input("\n Enter file Name:")
clientsocket.send(sentance.encode())
filecontents = clientSocket.recv(1024).decode()
print("\n From Server: \n")
print(fileContents)
clientSocket.close()
```

Server.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 ()
    sentance = connectionSocket.recv (1024).decode()
    file = open (sentance, "r")
    l = file.read (1024)
    connectionsocket.send (l.encode())
    print ('\n Sent contents of ' + sentance)
    file.close()
    connectionSocket.close()
```

Output

For server.py

Server is Ready to Recive

For Client.py

Enter File name : server TCP.py

From server:

```
connectionsocket, addr = serverSocket.accept ()
sentance = connectionSocket.recv (1024).decode()
```

file = open (sentance, "r")
t = file.read (1024)

connectionSocket.send (t.encode())
print ("In Sent contents of " + sentance)
file.close()
connectionSocket.close()

```
Enter file name: serverTCP.py

From Server:
 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()
```

```
The server is ready to receive

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

12 • UDP

Using UDP socket, write client server program to make client sending name and server to send back the contents

Client UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentance = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentance, "utf-8"),
                    (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print("\nReply from Server:\n")
print(filecontents.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

Server UDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
Print("Server is ready to receive")
while 1:
    sentance, clientAddress = serverSocket.recvfrom(2048)
    sentance = sentance.decode("utf-8")
    file = open(sentance, "r")
```

```
l = file.read(2048)
serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
print('\n Sent contents of ', end = ' ')
print(sentance)
file.close()
```

Output :-

```
The server is ready to receive
Send contents of server.py
```

```
Enter filename : 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:
    sentance, clientAddress = serverSocket.recvfrom(2048)
    sentance = sentance.decode("utf-8")
    file = open(sentance, "r")
    l = file.read(2048)
    serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
    Print('\n Sent contents of ', end = ' ')
    Print(sentance)
    file.close()
```

```
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()
```

The server is ready to receive