30/11/12

Lab: week-0

Interface Overview:

Initial interface has 10 components.

1) Menu Bar
2) Main Tool Bar
3) Common

Workspaces and Modes

Packet tracer — 2 workspaces
1) Logical
2) Physical

2 modes
1) Realtime
2) Simulation

Logical/Physical Workspace & Navigation Bar: You can toggle b/n. Physical workspace & Logical workspace.

In Physical workspace, this bar allows you to navigate through physical locations.

In Logical workspace, this bar allows you to go back to previous level in clusters.

workspace: this area is where you will create network and watch simulations etc.

Realtime/Simulation Bar: You can toggle b/n. realtime mode & simulation mode with tabs on this bar. It has clock that displays relative time in realtime & simulation mode.

III) Sending Simple Test

1) Launch packet tracer
2) create a network with generic PC & generic server to the workspace.
3) Under connections select copper straight cable & connect both PC & servers.
4) Now after connecting both PC & server send simple PDU from PC to server.

- click on Auto Capture/Play & hence we can view the animation of packet transfer.
- In real-time mode open command prompt & send Ping using commands. & destination IP addresses.

Routes: A router is networking device that is used to transfer data packets between Computer networks.

Switches: Switches is networking device used to segment the network into different subnetworks called lan networks.

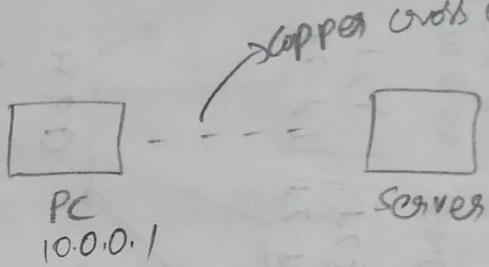
- IPv4 is address that consists of network address and host address
- Copper cross over is wire that is used to connect two wired connected devices of the same network level.
- Open (drive → Prog files (x86) → Cisco packet Help → default → index.htm

10 | 11 | 22

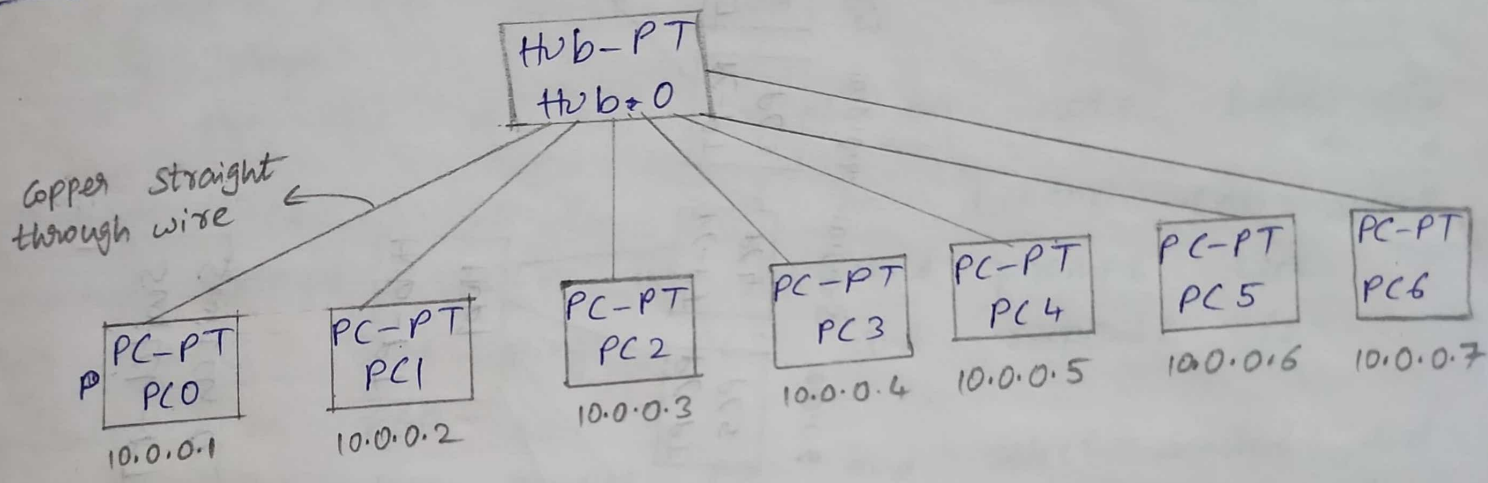
10/11/22 Lab - week - 1

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

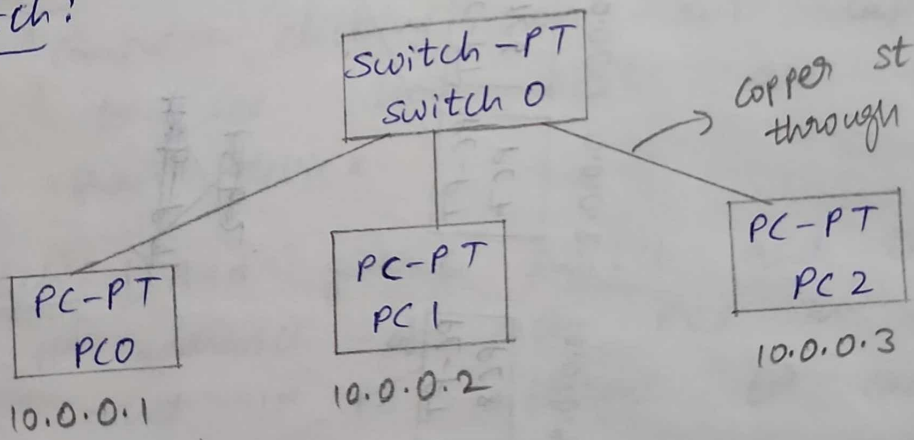
Topology:



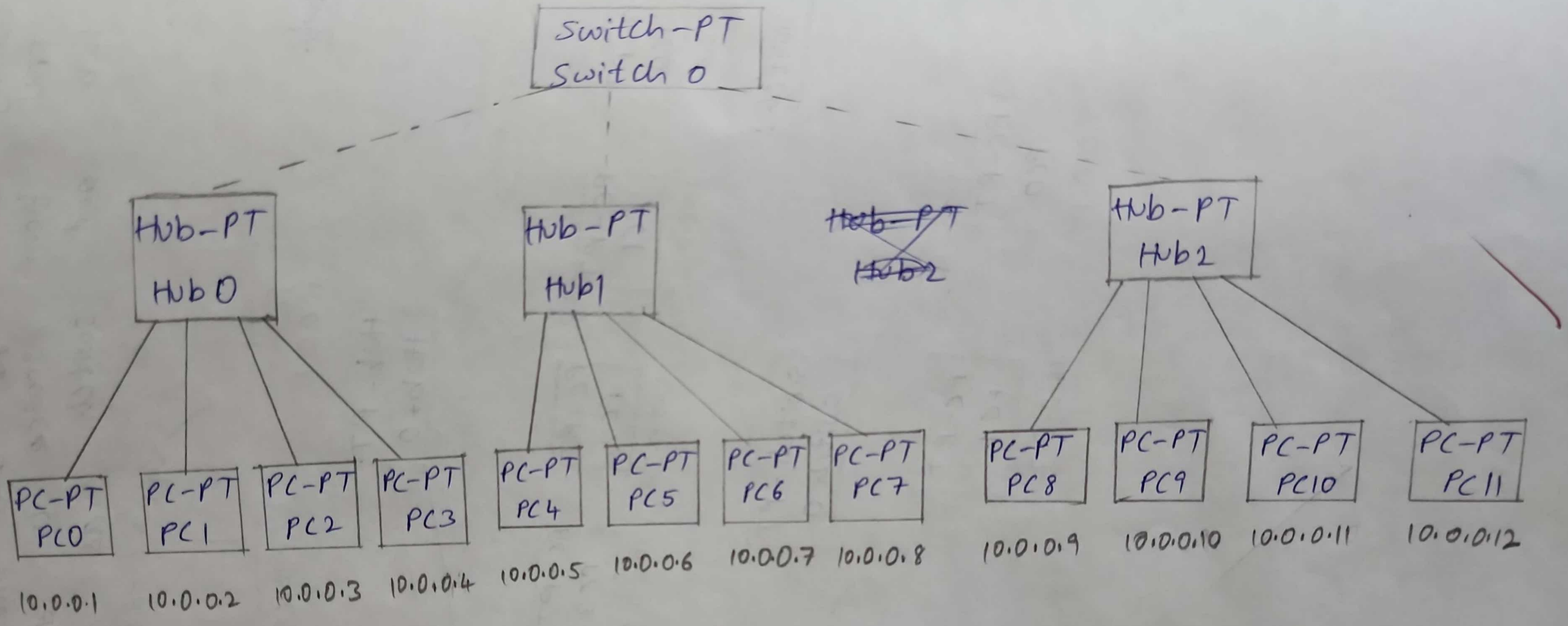
using hub:



Using switch:



Hybrid (using hubs and switches)



Procedure:

Using Hub: 1) Add generic hub and seven PCs to workspace
2) Configure the IP address of each PC in configuration tab. Ensure that IP is different for each device.
3) Connect all PC's to hub using copper straight wire.
4) Hub and PC is connected to each other's fast ethernet connection.
5) If no. of ports is insufficient then add extra port by clicking on device. Turn off device and add necessary ports
6) Write the IP's of all devices in note below the device.

Real time: Select source PC and in desktop tab, select command prompt option. In command prompt type ping 10.0.0.3. This pings PC2 and response is generated in PC0.

Simulation time: Select simple PDU and select source and destination computer. clicking on auto capture option allows us to see how ports are transferred to and from device.

Using switch: i) Add generic switch and then PCs to workspace.
ii) Configure IP addresses of each PC's in the configuration tab. Ensure that IP is different for each device.
iii) Connect all PCs to switch using copper straight through wire.
iv) If no. of ports are insufficient then add extra ports by clicking on device. Turn off device and add necessary ports.
v) write IP's of all devices in note below the device.

Real time: Select source PC and in the desktop tab. Select Command prompt option. In Command prompt option, Ping destination PC by specifying its IP.

Simulation time: Select simple PDU and select source and destination Computer. clicking on auto Capture option allows us to see how packets are transferred.

Hybrid mode: i) Add a switch, 3 hubs and 12 PC's to workspace.

ii) Connect three hubs to switch and 4 PC's to each of the hubs using copper cross over and copper straight through wires respectively.

iii) Configure the IP of each of the PC in Configure and add a note below each PC containing IP addresses

iv) Real time mode: Select PC you want to send packet from and open its command prompt. Specify destination PC by specifying its IP address. A response is sent by destination PC to source PC.

Simulation mode: Add a simple PDU by selecting the pair of PC and click on auto capture from right panel.

Observation:

→ Hub:

Learning outcomes: i) when source sends a packet in network the hub ~~servic~~. source the packet and ends broad cast over the network, i.e., it sends data to all the end devices in network and mode where it matches with the specified address accepts the packet and acknowledge it. Remaining nodes ignore the message.

ii) Comm^n. b/n. hub and end devices is established through copper straight through wire as they belong to different layers.

iii) No. of ports can be added if needed by clicking on the device and adding the necessary ports.

Result: PC > ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : byte = 32 time = 0ms

Reply from 10.0.0.3 : byte = 32 time = 0ms

Reply from 10.0.0.3 : byte = 32 time = 0ms

Reply from 10.0.0.3 : byte = 32 time = 0ms

Ping statistics for 10.0.0.3

packet sent = 4, received = 4, lost = 0

→ Switches:

Learning outcomes: i) when source device sends a message to the switch once a connection is established, which takes some time called learning time, the switch receives the packet. It initially broadcasts the packet to all connected devices to locate the destination. Once the destination is located the message is sent only to that device.

ii) Connection between the switch and end device is established using copper straigut though as they belong to different network layers.

iii) No. of ports can be added if needed by clicking on device and adding the necessary ports.

Result:

PC > Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : bytes = 32 time = 0 ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0 ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Ping statistics for 10.0.0.3

Packets : Sent = 4, Received = 4, lost = 0

→ Hybrid mode:

Learning outcomes: i) Switch and hub are connected through copper cross over as they belong to the same network layer but PC and hubs are connected through copper straight through as they belong to different network layers.

ii) message from source PC to destination is sent through the hub which then sends to all its connected PC's and the switch. The switch then sends the message to all its connected PC. The destination PC acknowledges that it has received the message by sending a acknowledged back to the source PC.

iii) The no. of ports can be added if needed by clicking on device and adding the necessary ports.

Result

PC > Ping 10.0.0.6

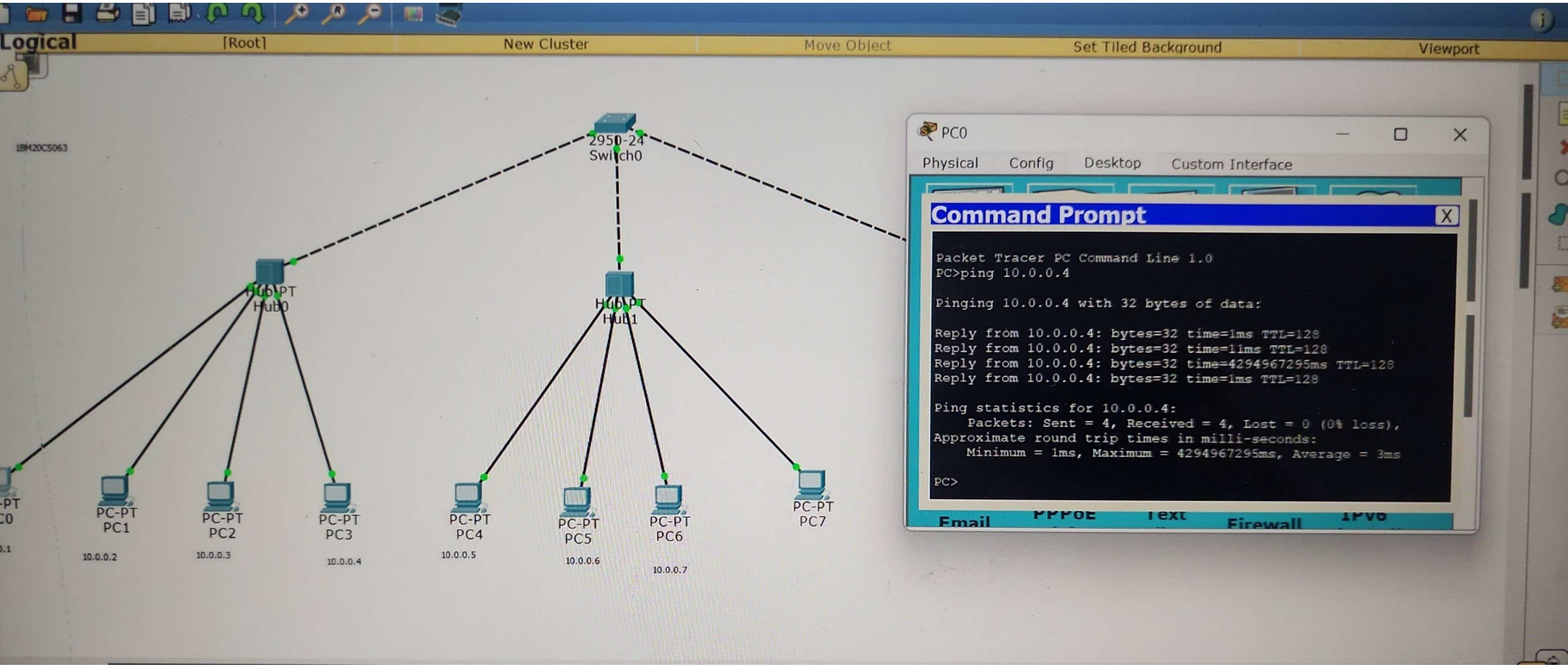
Pinging 10.0.0.6 with 32bytes of data

Reply from 10.0.0.6 bytes = 32

Ping statistics for 10.0.0.4

"Details of numbers of packets send and received".

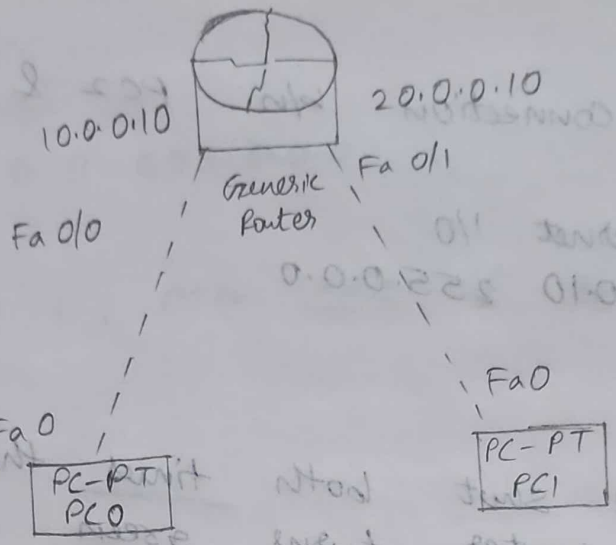
17/11/22


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
1BM20CS063
2950-24
Switch0
Hub-PT
Hub0
Hub-PT
Hub1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3
PC-PT
PC3
10.0.0.4
PC-PT
PC4
10.0.0.5
PC-PT
PC5
10.0.0.6
PC-PT
PC6
10.0.0.7
PC-PT
PC7
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=11ms TTL=128
Reply from 10.0.0.4: bytes=32 time=4294967295ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 4294967295ms, Average = 3ms
PC>
Email
Firewall

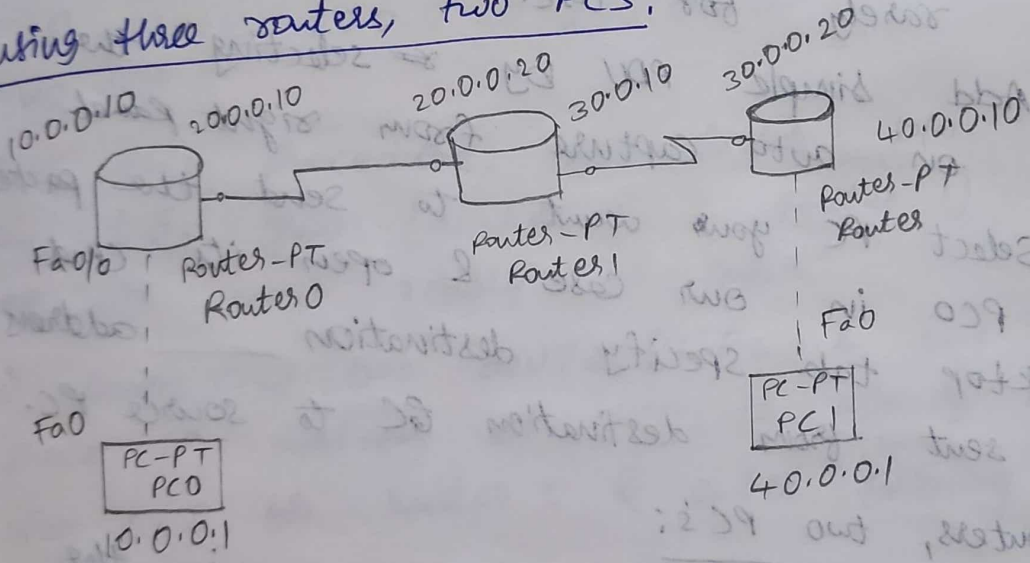
Lab: week 2

Aim: Configuring IP addresses to router in packet tracer to explore following messages: Ping- responses, destination unreachable, request timed out, reply.

Topology: using single router, 2 PC's



using three routers, two PC's:



Procedure:

using single router, two PC's

i) Place a generic router & add two generic PC's in your workspace.

ii) Connect router & PC's using copper cross over wire

iii) Configure IP address of each PC & in the configuration tab under settings set gateways for both PCs to router.

iv) Click on generic router and go to CLI tab. Enter the following commands to set up connection b/n PC$(1) and generic router through gateway (0.0.0.10

→ No
enable
config t
(Config) # interface fast ethernet 0/0.
(Config-if)# ip address 10.0.0.10 255.0.0.0
#no shut
exit
Now to set up connection b/n PC2 & router through gateway 20.0.0.10
interface fastethernet 1/0
IP address 20.0.0.10 255.0.0.0
no shut
exit
Once we enter no shut both times the amber light b/n the PC & router turns green indicating that two devices are ready for use.
Simulation mode: Add simple PDU by selecting the PCs and click on auto capture from right panel.
Real time mode: Select PC you want to send the packet from which is PC0 in our case & open its cmd prompt from desktop tab. specify destination ____ address.
A response is sent from destination PC to source PC.

→ using three routers, two PC's:

i) Place 3 generic routers and 2 generic PCs in the workspace.
ii) Place note for each device [PC & router] and specify IP address.
iii) Connect router & PC using copper cross over
iv) ~~edit~~ Connect routers using serial ____
v) click on each PC, go to the configure tab. set the IP address & subnet mask in fastethernet.
vi) Next click on settings in config tab. set gateway on IP address of the next router [eg 10.0.0.10]

vii) IP address of PC and its gateway address should belong to the same network.

For connecting two routers:

click on Router 0. Go to CLI and enter following commands

- no
- enable
- config t
- interface serial 2/0
- ip address 20.0.0.10 255.0.0.0
- no shut

click on router 1, open CLI & enter following commands.

- no
- enable
- config t
- interface serial 2/0
- ip address 20.0.0.20 255.0.0.0
- no shut

After this procedure, red lights b/n 2 routers will now turn green [router 0 & router 1] indicating that they are now ready for comm^n.

For connecting two devices [1 PC & one router]

i) Since IP address of PC0 is already configured, go to router.

ii) open CLI for router 0 & enter following commands.

- no
- enable
- config t
- interface fastethernet 0/0
- ip address 10.0.0.10 255.0.0.0
- no shut

Red light turn green, which means ready for comm^n.

Teaching router 0 of network 30:

- no
- enable
- config t
- interface serial 2/0
- IP route 30.0.0.0 255.0.0.0 20.0.0.20
- exit
- show ip route

Teaching Router0 of network 40:

- no
- enable
- Config t
- interface serial 2/0
- IP route 40.0.0.0 255.0.0.0 20.0.0.20
- exit
- show ip route

Similarly, repeat for routers 1 & 2.

Simulation mode: Add simple PDU by ~~execute~~ selecting PC 2 & click on auto capture from right panel.

Realtime mode: Select PC 0 & go to its cmd prompt & Ping router 0. Once message has been sent successfully Repeat this with routers 1 & 2 as well. Finally, Ping PC1.

Observation:

Learning outcomes

1 Router: when PC0 pings PC1 for first time, we get Packet as request timed out.

Now, if we ping PC1 again from PC0 we get all 4 packets without any loss. Now reverse Pinging of PC0 from PC1 will also not lead to any loss, all Packets are acknowledged.

- 3 routers: Before training routers, we get results as destination not unreachable. After training routers, we get clear statistics the result.

Result: using 1 router, two PC's.

<1> 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=127

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

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

Ping statistics for 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

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

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

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

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

Ping statistics for 20.0.0.1:

Packets: sent = 4, received = 4, lost = 0 (0% loss)

- using 3 routers, 2 PC's:

1) 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

[3 more times]

Ping statistics for 40.0.0.1:

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

2) PC> Ping 10.0.0.10 with 32 bytes of data.

Reply from 10.0.0.10: bytes=32 time = 0ms TTL=255

[3 more times]

Ping statistics for 10.0.0.10

Packets: sent = 4, Received = 4, lost = 0 (0% loss)

3) PC > Ping 20.0.0.10 with 32 bytes of data

Reply from 20.0.0.10: bytes = 32 time = 1ms TTL = 255

[3 more times]

Ping statistics for 20.0.0.10:

Packets: sent = 4, received = 4, lost = 0 (0% lost)

4) PC > Ping 30.0.0.10

Pinging 30.0.0.10 with 32 bytes of data

Reply from 30.0.0.10: bytes = 32 time = 1ms

[3 more times]

Ping statistics for 30.0.0.10:

Packets: sent = 4, received = 4, lost = 0 (0% lost)

5) 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 = 10 ms TTL = 125

" " " : " = 32, " = 13ms TTL = 125

" " " : " = 32, " = 8 ms TTL = 125

Ping statistics for 40.0.0.1:

Packets: Sent = 4, received = 3, lost = 1 (25% loss)

6) 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 = 2ms TTL = 125

Reply from 40.0.0.1: bytes = 32 time = 24ms TTL = 125

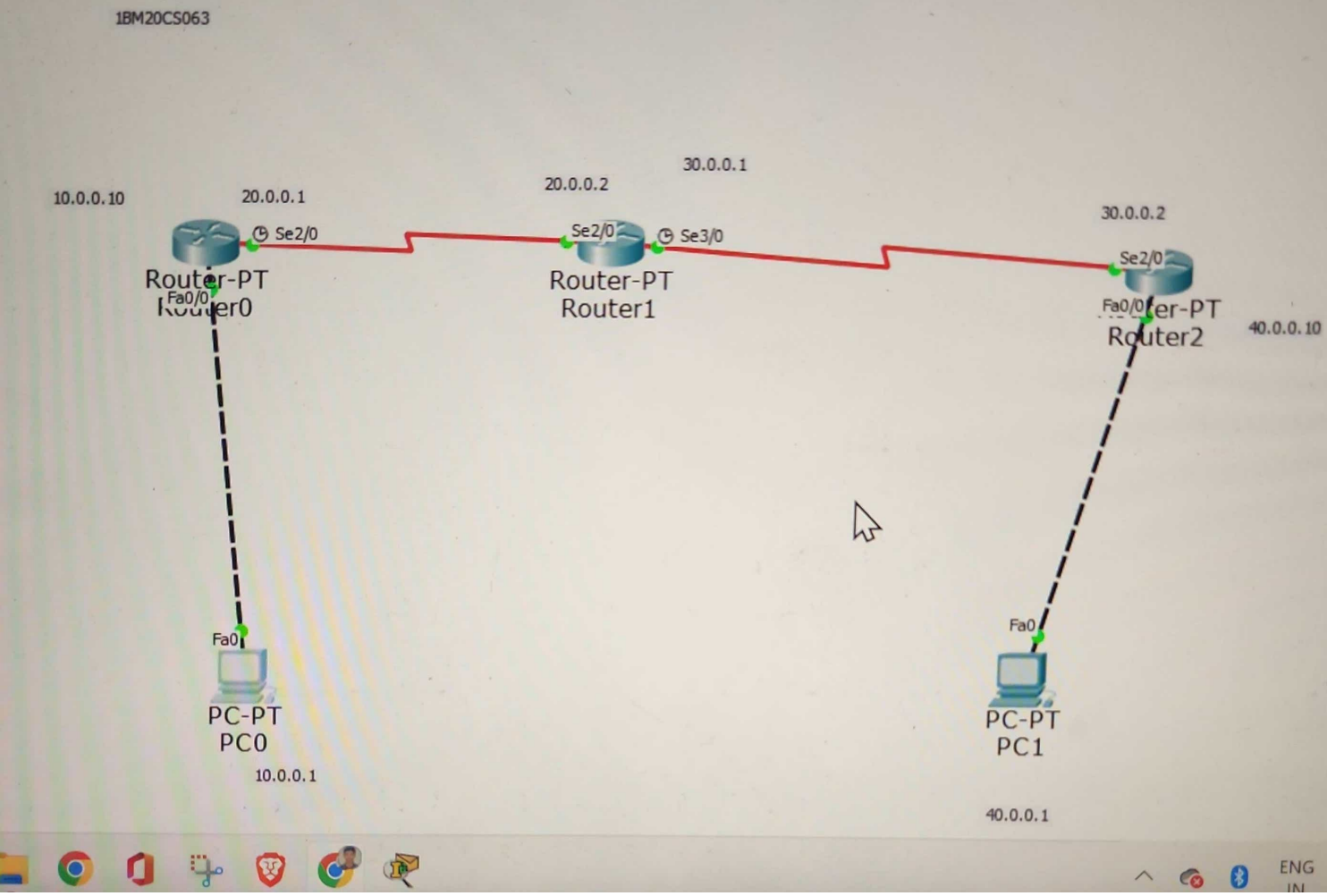
Reply from 40.0.0.1: bytes = 32 time = 9ms TTL = 125

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

Ping statistics for 40.0.0.1:

Packets: Sent = 4, received = 4, lost = 0 (0% loss)

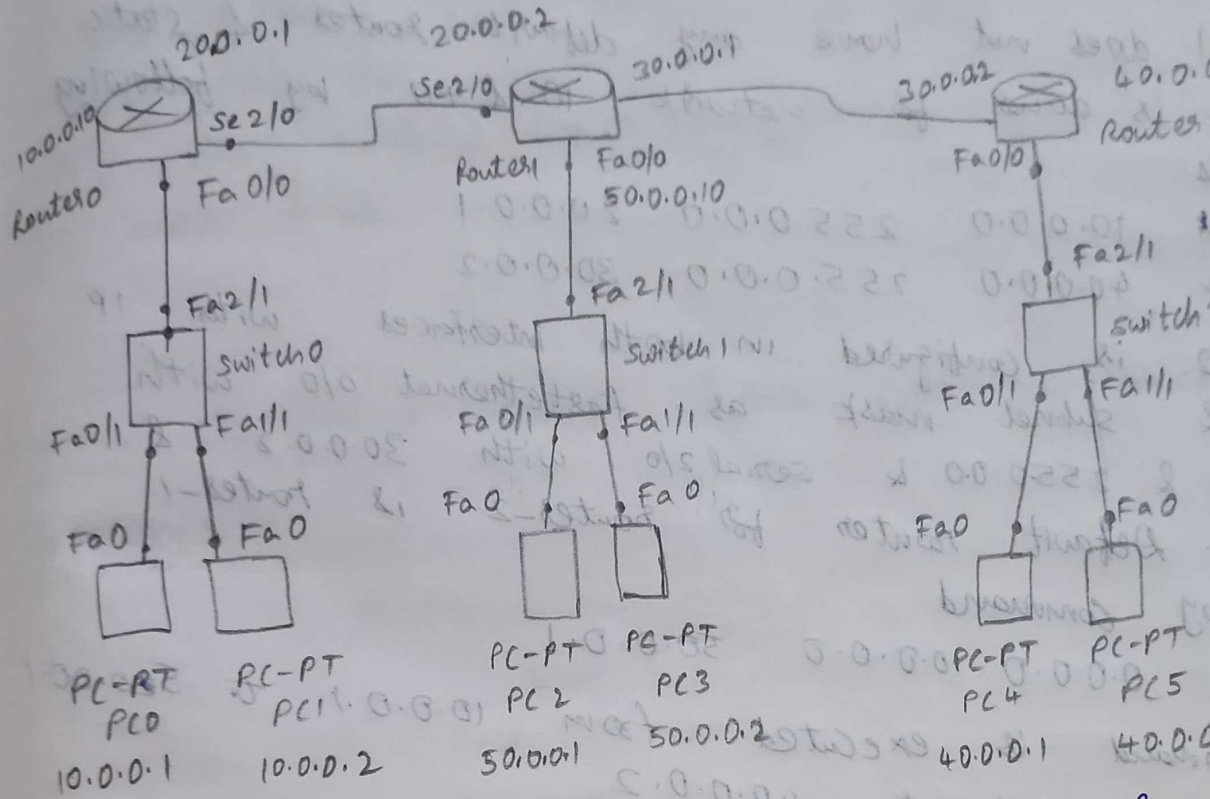
24 / 11 / 2



25/11/22

3) Aim: Configuring default route to routers

Topology:



Procedure: 1) First, Put 6 PC's, 3 switches & 3 routers & Connect two PC's to each switch with copper straight through wire and each switch is connected to one router with copper straight through wire, & three routers are connected by serial cables and nodes are placed for all devices & networks.

2) A PC is clicked to set attributes for PC & each PC has 3 attributes which are IP address subnet mask & gateway & all three are set according to nodes placed. This process is done for all PC's.

3) For router-0, configurations are done in CLI. IP address & subnet mask are set for both interfaces - fast ether net 0/0 as 10.0.0.10 & 255.0.0.0 & serial 2/0 as 40.0.0.1 and 255.0.0.0. Router-1 is default router for router-0 & this is done by command ip route 0.0.0.0 0.0.0.0 20.0.0.2.

4) For router-1, IP address & Subnet mask are set to all three interfaces - fastethernet 0/0 as 50.0.0.10 & 255.0.0.0 & serial 2/0 as 20.0.0.2 & 255.0.0.0 & serial 3/0 as 30.0.0.1 & 255.0.0.0.

5) Router-1 does not have any default router & static routing - is done for network 10 & 40 by following commands.

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

6) Router-2 is configured in both interfaces with IP address & subnet mask as fastethernet 0/0 with 40.0.0.10 & 255.0.0.0 & serial 2/0 with 30.0.0.2 & 255.0.0.0. Default router for router-2 is router-1 & set by 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 50.00.0.1 & from 10.0.0.1 to 40.0.0.2

Observations:

Learning Outcomes:

- One router can't have 2 default routers
- Default routers for first router is middle routes as any packets which have to be delivered will go to middle routers.
- Default routers for right router is middle routers for the same reason
- Middle router doesn't have any default router as if one of the routes is made default then there is a chance that packets which are to be sent to switch may send to routers.

Result:

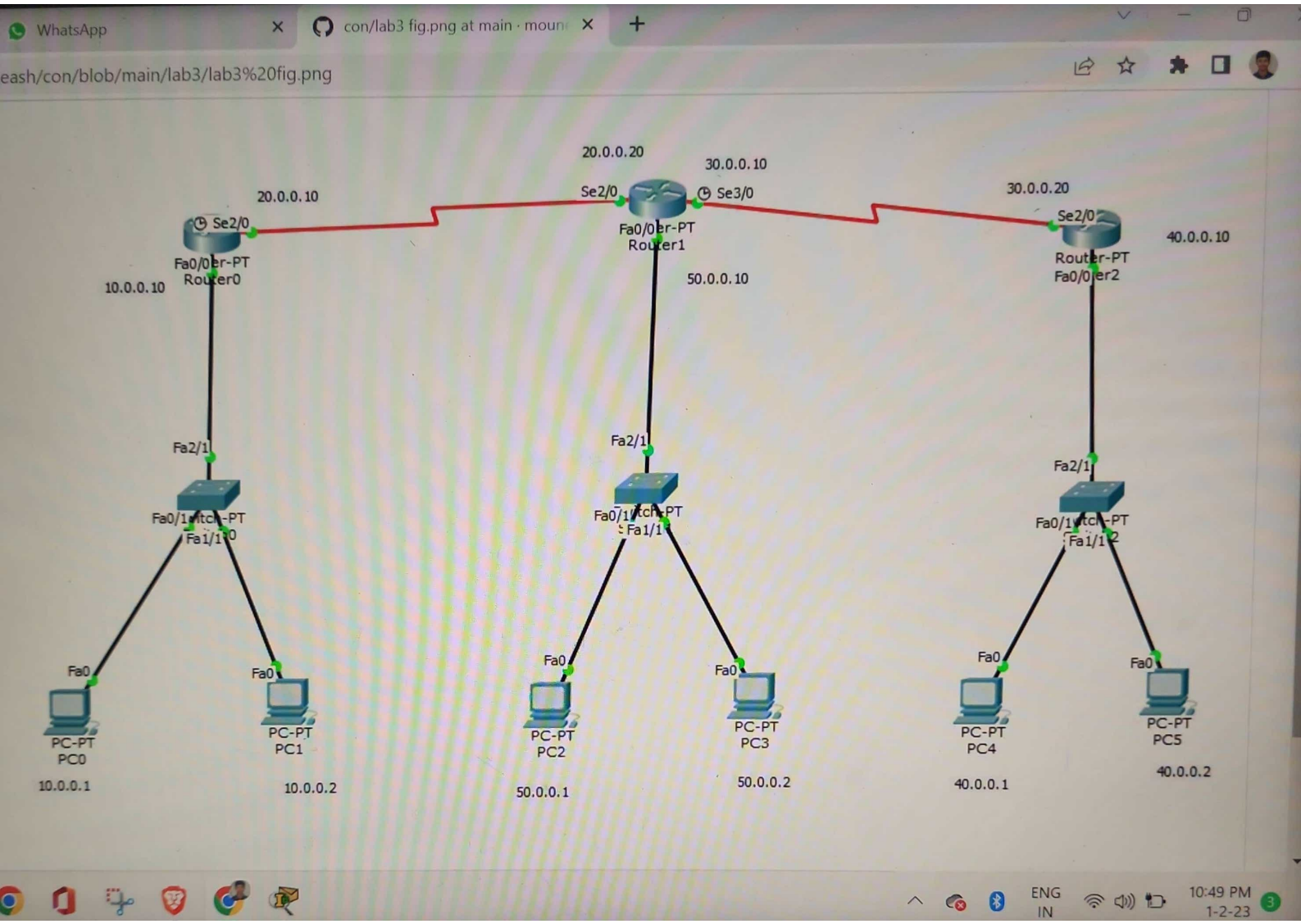
Ping 50.0.0.1

Pinging 50.0.0.1 with 32 bytes of data

Request timed out

Reply from 50.0.0.1 : bytes = 32, time = 1ms, TTL = 126
[2 more times

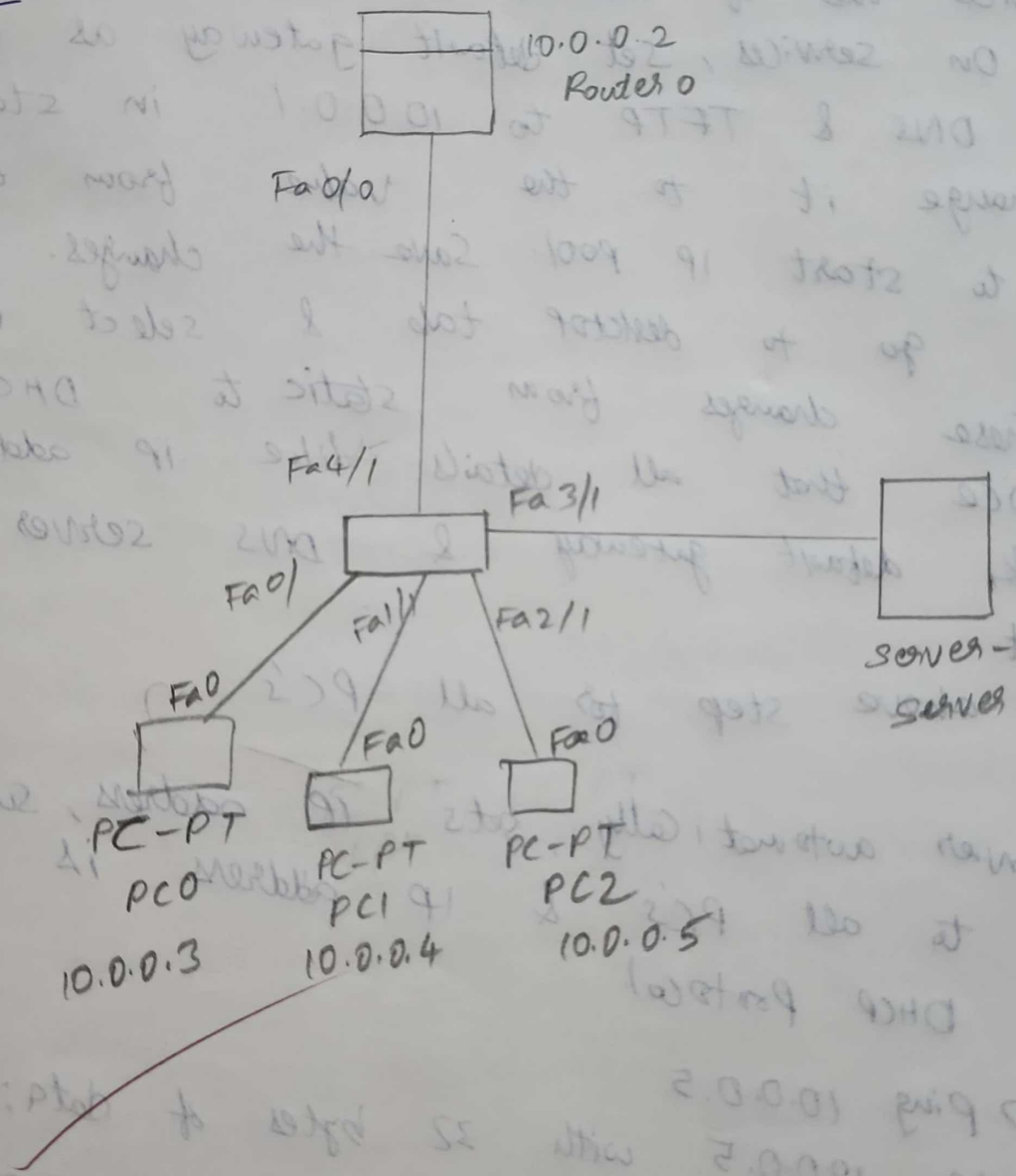
Ping 40.0.0.2
Pinging 40.0.0.2 with 32 bytes of data
Request timed out
Reply from 40.0.0.2 : bytes = 32, time = 4ms, TTL = 125
[2 more times


WhatsApp
con/lab3 fig.png at main · moun
eash/con/blob/main/lab3/lab3%20fig.png
20.0.0.20
30.0.0.10
20.0.0.10
Se2/0
Se3/0
30.0.0.20
Se2/0
Fa0/0er-PT
Router1
Router-PT
Fa0/0er2
40.0.0.10
Se2/0
Fa0/0er-PT
Router0
10.0.0.10
50.0.0.10
Fa2/1
Fa2/1
Fa2/1
Fa0/1
Fa1/1
Fa0/1
Fa1/1
Fa0/1
Fa1/1
Fa0
Fa0
Fa0
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
10.0.0.1
10.0.0.2
50.0.0.1
50.0.0.2
40.0.0.1
40.0.0.2
ENG
IN
10:49 PM
1-2-23

1/12/22

4) Aim: Configuring DHCP within LAN in Packet tracer

Topology:



Procedure: 1) Place generic router, generic switch, 1 server & 3 PCs in the work station & connect them as: Router to switch & switch to all 3 PCs & Server

2) Open Server Config tab, set its IP address & Subnet mask, from settings tab set its gateway

3) For server IP address is 10.0.0.1, gateway is 10.0.0.2

4) Open CLI of router → no → enable → config t → interface fastethernet 0/0 → ip address 10.0.0.2 255.0.0.0 → no shut

5) Go to services tab of server, & select DHCP from left panel. On services, set default gateway as 10.0.0.2 & change DNS & TFTP to 10.0.0.1 in start IP address change it to the value from where you want to start IP pool. Save the changes.

6) Select PC, go to desktop tab & select IP Configuration. These changes from static to DHCP, we can notice that all details like IP address, Subnet mask, default gateway & DNS server is set by default.

7) Do the above step for all PC's

Observations:

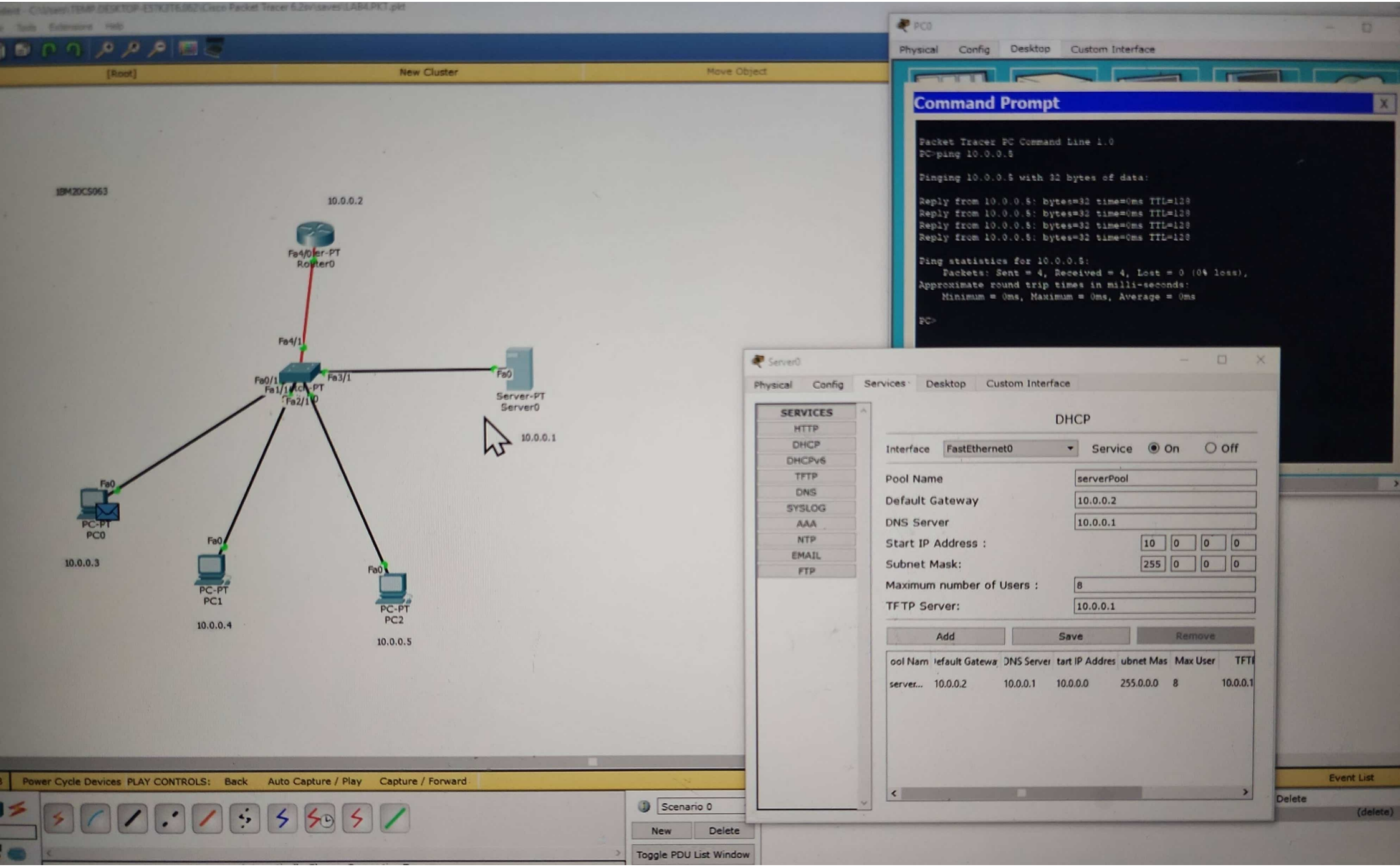
Learning: Server automatically sets IP address, subnet mask & gateway to all PC's & IP address is allocated serially in DHCP Protocol.

Result: 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

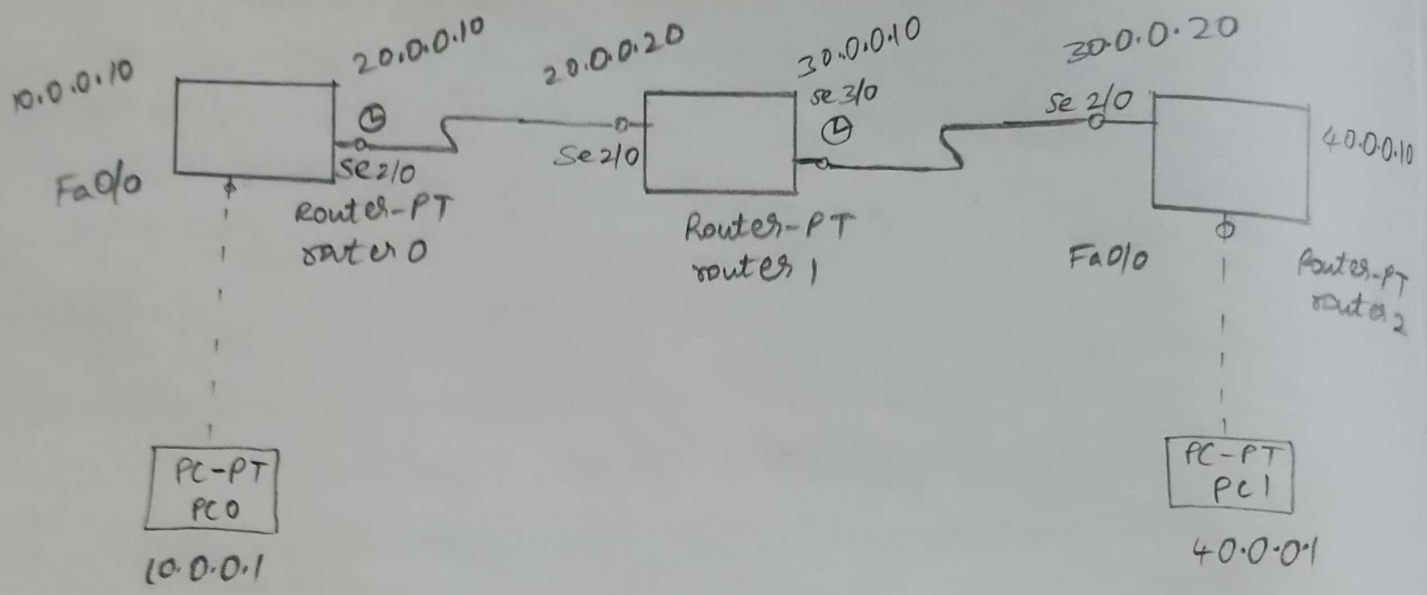
[3 more times


[Root]
New Cluster
Move Object
1BM20CS063
10.0.0.2
Fa4/0er-PT
Router0
Fa4/1
Fa0/1
Fa1/1
Fa2/1
Fa3/1
Fa0
Server-PT
Server0
10.0.0.1
Fa0
PC-PT
PC0
10.0.0.3
Fa0
PC-PT
PC1
10.0.0.4
Fa0
PC-PT
PC2
10.0.0.5
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Scenario 0
New
Delete
Toggle PDU List Window
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
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>
Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
serverPool
Default Gateway
10.0.0.2
DNS Server
10.0.0.1
Start IP Address :
10 0 0 0
Subnet Mask:
255 0 0 0
Maximum number of Users :
8
TFTP Server:
10.0.0.1
Add
Save
Remove
ool Nam
efault Gatewa
DNS Server
tart IP Addres
ubnet Mas
Max User
TFTI
server...
10.0.0.2
10.0.0.1
10.0.0.0
255.0.0.0
8
10.0.0.1
Event List
Delete
(delete)

8/12/22

5. Aim: Configuring RIP routing Protocol in routers.

RIP: Topology:



Procedure: 1) Place 3 generic routers & 2 generic PCs in workspace

2) Connect routers & two PCs using Copper cross over

3) Connect the routers using serial DCI with clock ~~sig~~ symbol

4) Place notes near PCs & routers

5) Select IP address of PC0 & PC1 as well as their subnet mask & default gateway.

6) Go to CLI of router 0 & enter following commands

- enable
- config t
- interface fastethernet 0/0
- ip address 10.0.0.10 255.0.0.0
- no shut

Connection should show green. Repeat for PC1 & router 2 as well.

Connection should be
Now, open CLI of router 0 & enter the following commands.

- enable
- config t
- interface serial 2/0
- encapsulation PPP
- clock rate 64000
- no shut

Open CLI of router 1
- enable
- config t
- interface serial 2/0
- ip address 20.0.0.20 255.0.0.0
- encapsulation PPP
- no shut

Connection will turn green.

Open CLI of router 1 again.

- enable
- config t
- interface serial 3/0
- ip address 30.0.0.10 255.0.0.0
- encapsulation
- clock rate 64000
- no shut

Open CLI for router 2:

- enable
- config t
- interface serial 2/0
- ip address 30.0.0.20 255.0.0.0
- encapsulation PPP
- no shut

Now,
open CLI of router 0 & enter following commands.

(Config)# router rip
&Config-router # network 10.0.0.0
Config-router # network 20.0.0.0
Config-router # exit
show ip route

Similarly repeat for router 1 & router 2 with networks 20 & 30 and 30 & 40.

Simulation mode: Add simple PDU by selecting the PCs & click on auto capture from right panel.

Real time mode: simple PC0 go to command prompt and select destination address 40.0.0.1

Observation:
Learning outcome: Routing information protocol is a protocol that routers use to exchange information of topology among the network.

It is used when in place of static IP routing because with help of rip protocol routing it becomes easy when large scale of routers is present

Result:
PC> Ping 40.0.0.1
Request timed out
Reply from 40.0.0.1 : bytes=32 time=6ms TTL=125
~~Reply from~~ 2 times

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

2) PC>Ping 40.0.0.1
Ringing 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1 : bytes=32 time=19ms TTL=125

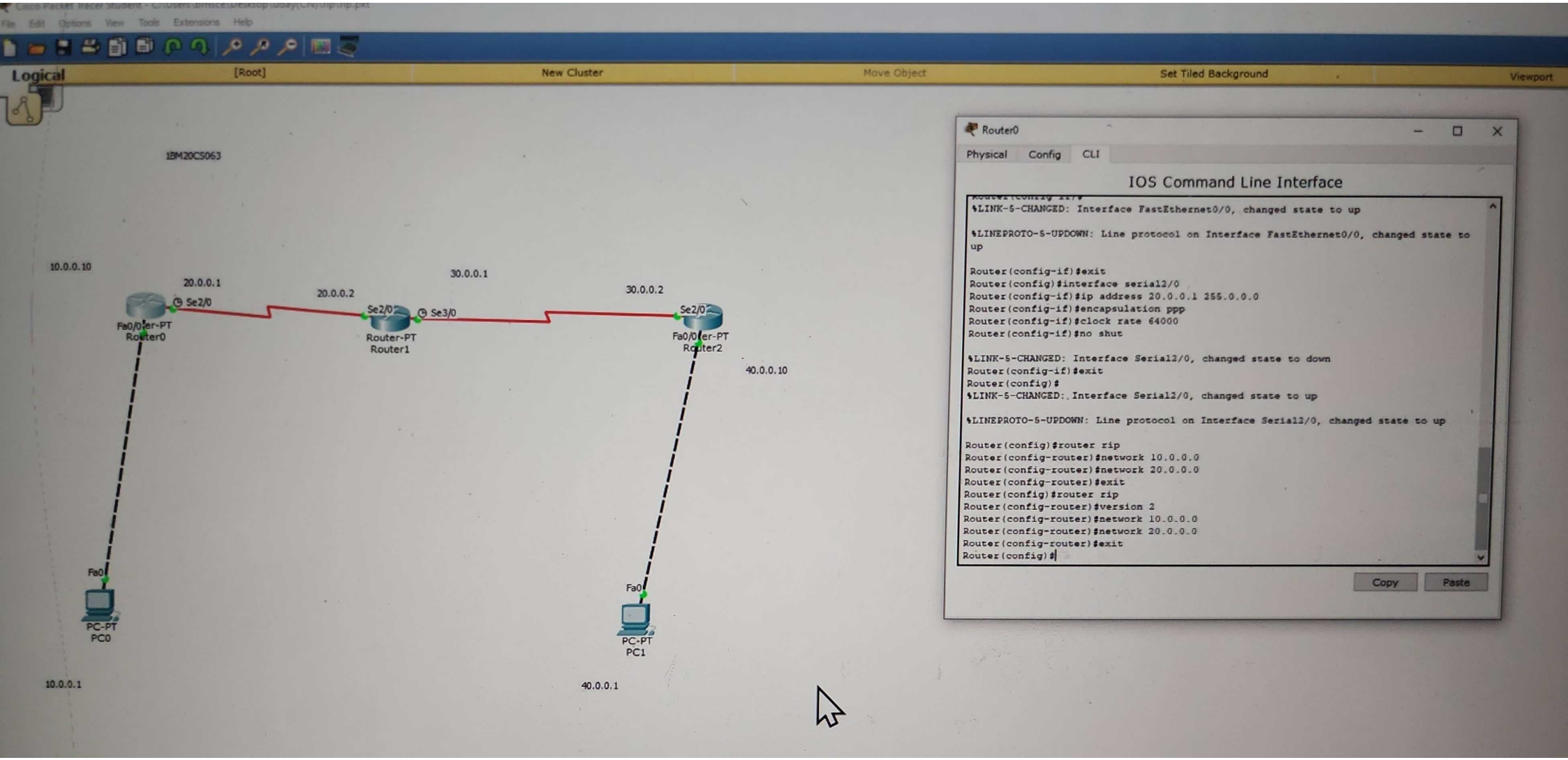
[3 more times

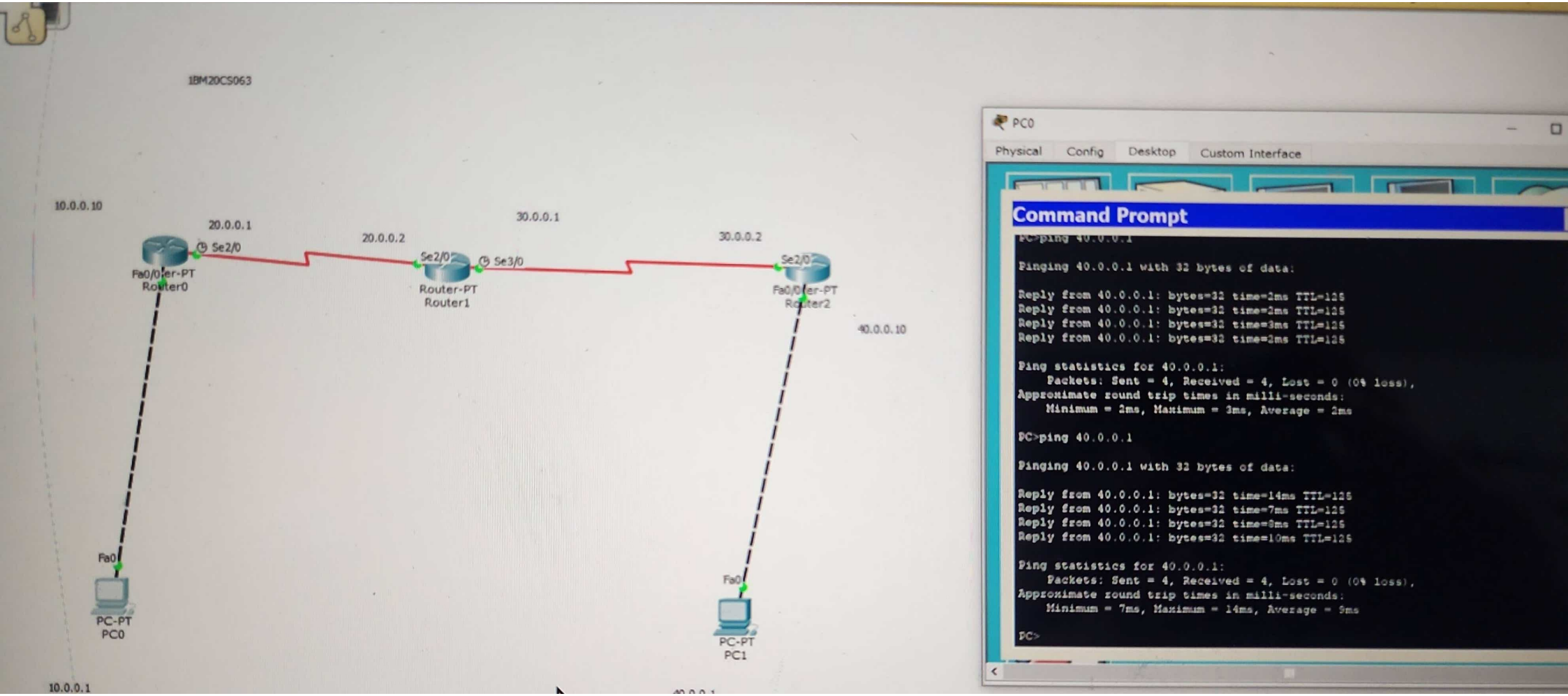
Ping statistics for 40.0.0.1:

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

Approximate round tip times in milli-seconds:

Minimum = 2ms, maximum = 19ms, Average = 10ms

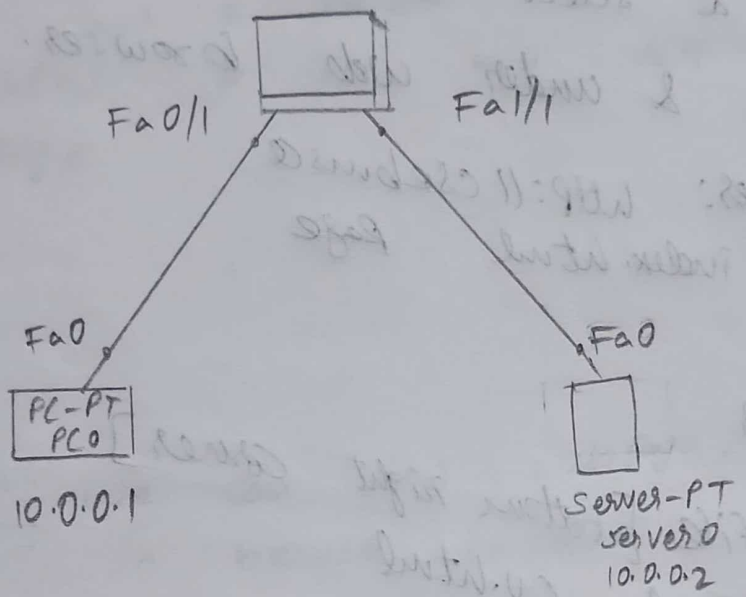

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
1BM20CS063
10.0.0.10
20.0.0.1
Se2/0
20.0.0.2
Se2/0
Se3/0
30.0.0.1
30.0.0.2
Se2/0
Fa0/0
Router-PT
Router0
Router-PT
Router1
Fa0/0
Router-PT
Router2
40.0.0.10
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
10.0.0.1
40.0.0.1
Router0
Physical
Config
CLI
IOS Command Line Interface
%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 serial2/0
Router(config-if)$ip address 20.0.0.1 255.0.0.0
Router(config-if)$encapsulation ppp
Router(config-if)$clock rate 64000
Router(config-if)$no shut
%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)$exit
Router(config)$
%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)$router rip
Router(config-router)$network 10.0.0.0
Router(config-router)$network 20.0.0.0
Router(config-router)$exit
Router(config)$router rip
Router(config-router)$version 2
Router(config-router)$network 10.0.0.0
Router(config-router)$network 20.0.0.0
Router(config-router)$exit
Router(config)$
Copy
Paste


1BM20CS063
10.0.0.10
20.0.0.1
Se2/0
20.0.0.2
Se2/0
Se3/0
30.0.0.1
30.0.0.2
Se2/0
Router-PT
Router0
Router-PT
Router1
Router-PT
Router2
40.0.0.10
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
10.0.0.1
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
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=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=3ms 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 = 3ms, Average = 2ms
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=7ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms 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 = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 14ms, Average = 9ms
PC>

15/12/22 6) DNS

Aim: Demonstration of web server & DNS using Packet tracer.

Topology:



Procedure:

i) Place generic PC, switch and server in workspace.
ii) Place note of ip address under PC as 10.0.0.1 & server as 10.0.0.2
iii) Connect PC to switch & switch to server.
iv) Set ip address of PC as 10.0.0.1 & server as 10.0.0.2.

- click on PC
- Go to desktop tab
- open webserver & enter ip address as 10.0.0.2
- Now, click on server
  Go to server.

under http, go to index.html. click on (edit)
& make some changes. Do overwrite it.

- Now go to PC & under web browser : →
  In the address url, give ip address as 10.0.0.2.
  click on go. It display the changes made in index.html

- Now go to server again
  under service, go to DNS
  Configure name as : csebmsce
  address : 10.0.0.2
  click on add & select & save.

- Now, go to PC & under web browser:
  In url : enter: http://csebmsce
  It displays index.html page

- click on server
  HTTP
  click on new file [bottom right corner]
  Enter file name as cv.html

```
<html>
<head><h1> CSE student details </h> </head>
<table>
  <tr>
    <th>Name </th>
    <th> Branch </th>
    <th > class </th>
```

```
<th> USN </th>
<th> DOB </th>
</tr>
<tr>
<td> Jayam Mounesh </td>
<td> CSE </td>
<td> 5-B </td>
<td> 1BM20CS063 </td>
<td> 27-06-2001 </td>
</tr>
</table>
```

- Now, go to PC & under web browser:

i) In url, enter: csebmsce

It will show a new link in the index.html page as cv.html. & it will display the student details page

Observation: we can view webpage, when we type 'csebmsce' in browser because 10.0.0.2 address is linked to name 'csebmsce' according to DNS.

- Since it's difficult for users to remember IP addresses, linking is required i.e., IP is linked with domain name.

Result:

web browser

< > URL http://csebmsce

BMSCE Packet tracer

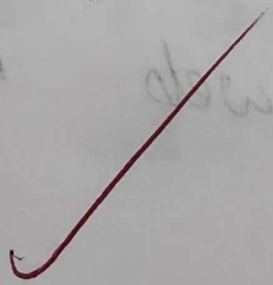
Welcome to BMSCE Packet tracer

Quick links

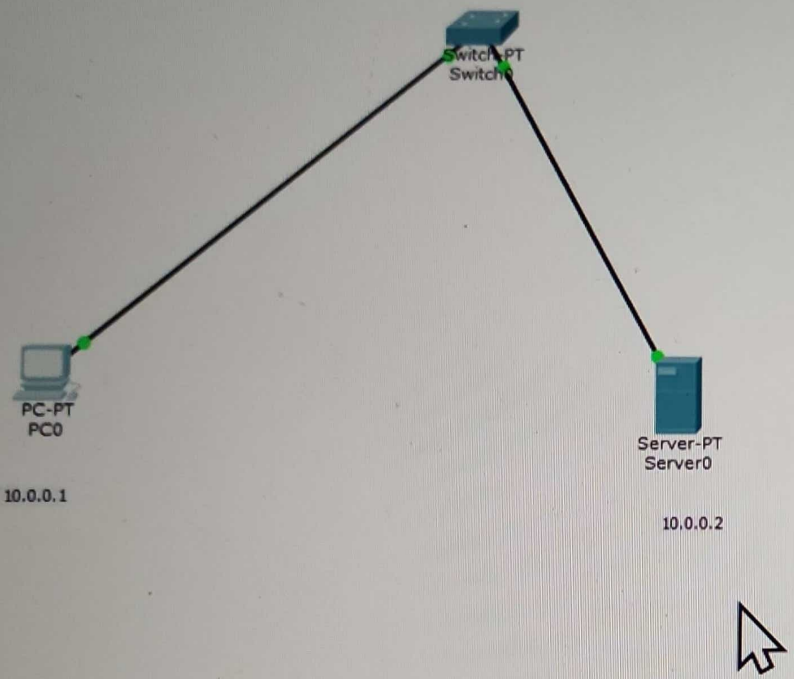
Cycle 1
29-12-2022

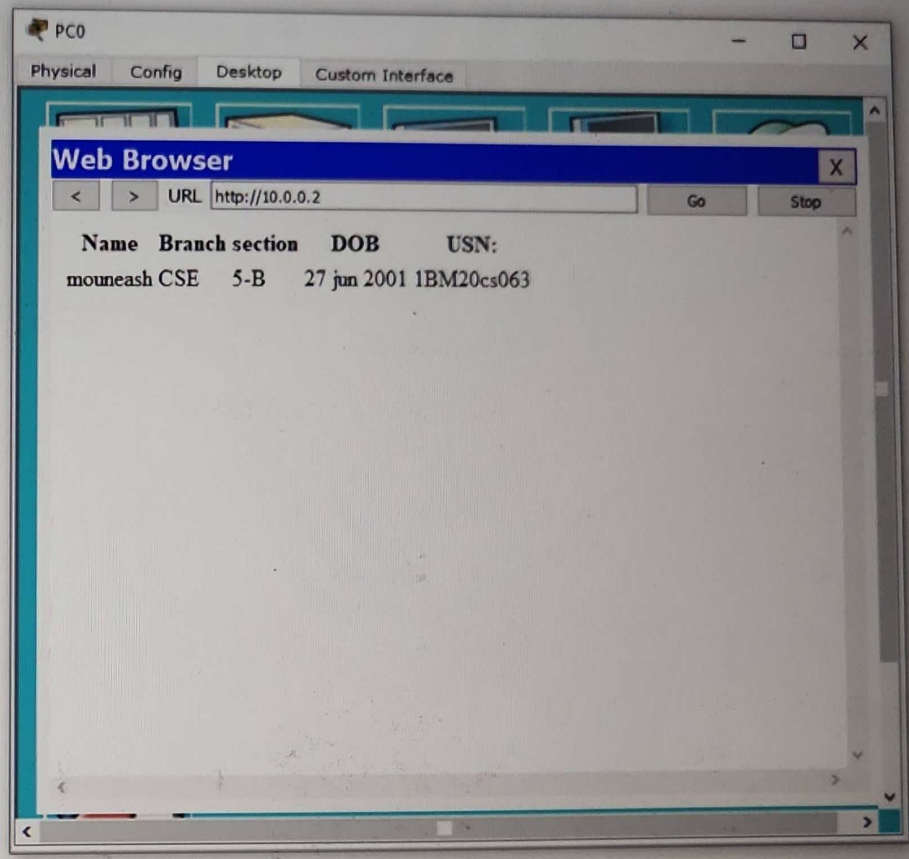
http://csebmsce.github.io/myc v.html

~~NAME~~	~~KRIPALDEEP~~
NAME	JAYAM MOUNEASH
AGE	20
DOB	27/06/2001
College	BMSCE
Branch	CSE
Year	3rd



1BM20CS063


Switch-PT
Switch0
PC-PT
PC0
10.0.0.1
Server-PT
Server0
10.0.0.2


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL
http://10.0.0.2
Go
Stop
Name Branch section DOB USN:
mouneash CSE 5-B 27 jun 2001 1BM20cs063

Cycle-2

① WAP for error detecting code using CRC-CCITT (16 bits)

```
#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("\n Enter data: ");
  scanf("%s", t);
  Printf("\n Enter generating polynomial: ");
```

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

void main(){
 int i,j,key,len,msglen;
 char input[100],key[30],temp[30],quot[100],rem[30],
 key1[30];

Printf("Enter data:");
gets(input);
Printf("Enter G(x):");
gets(key);
keylen = strlen(key);
 msglen = strlen(input);
 strcy(key1,key);
for(i=0;i<keylen-1;i++){
input[msglen+i]='0'
}
for(i=0;i<keylen;i++)
 temp[i]=input[i];
for(i=0;i<msglen;i++)
 quot[i]=temp[0]
 if(quot[i]=='0')
 for(j=0;j<keylen;j++)
 key[j]='0';
 else
 for(j=0;j<keylen;j++)
 key[j]=key1[j];
```

```
for ( j= keylen-1; j>0; j--) {
if ( temp [j]== key [j])
rem [ j-1] = '0'; else
 rem [j-1] = '1';

}
rem [keylen-1] = input [ i+keylen ];
strcpy (temp, rem);

}
strcpy (rem, temp);
Printf ("\n Quotient is ");
for (i=0; i<msglen ; i++)
Printf ("%c", quot [i]);
Printf ("\n Remainder is ");
for (i=0; i<keylen-1; i++)
Printf ("%c", rem [i]);
Printf ("\n Modified data is: ");
for(i=0; i<msglen; i++)
Printf ("%c", input [i]);
for (i=0; i<keylen-1; i++)
Printf ("%c", rem[i]);
   strcat (input, rem);

    keylen = strlen (key);
msglen = strlen (input);
strcpy (key1, key);
for (i=0; i<keylen-1; i++)
   input [msglen+i] = '0';

for(i=0; i<keylen; i++)
   temp [i]= input [i];

for (i=0; i<msglen; i++){
 quot[i]= temp [0];
```

```
if (quot[i] == '0')
for (j=0; j<keylen; j++)
key[j] = '0';
else
for (j=0; j<keylen; j++)
key[j] = key1[j];
for (j=keylen-1; j>0; j--)
if (temp[j-1] == key[j])
rem[j-1] = '0'
else
rem[j-1] = '1';
}
rem[keylen-1] = input[i+keylen];
strcpy(temp, rem);
}
strcpy(rem, temp);
Printf("\nRemainder is at receiver side");
for(i=0; i<keylen-1; i++)
Printf("%c", rem[i]);
getch();
}
```

O/P:

Case-1: Enter data: 10110101010

Enter g(x): 1010 → CRC-16 only $\frac{x^{16}+x^{12}+x^5+1}{\text{Fixed}}$

Quotient: 1001000100

Remainder: 000

modified: 10110101001000 → CW

checking for error

Case-2: Enter data: 10110101001000

G(x): 1010

Remainder: 000

D

DW

CW = R

CW = QP

R = 0 ✓

R ≠ 0 Error detected

3) Case - 3) Enter input : 1011010101

Transmitted message - 1011010101111110111010

Enter received message:

111111110000000111111111

Error in data transmission has occured

AL

5/1/23

```
Enter data to be transmitted: 1011010101

 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: 1011010101001
Data received: 1011010101001
Error detected
```

5/01/23

2) Leaky Bucket algm

Program for congestion control using leaky bucket algm.

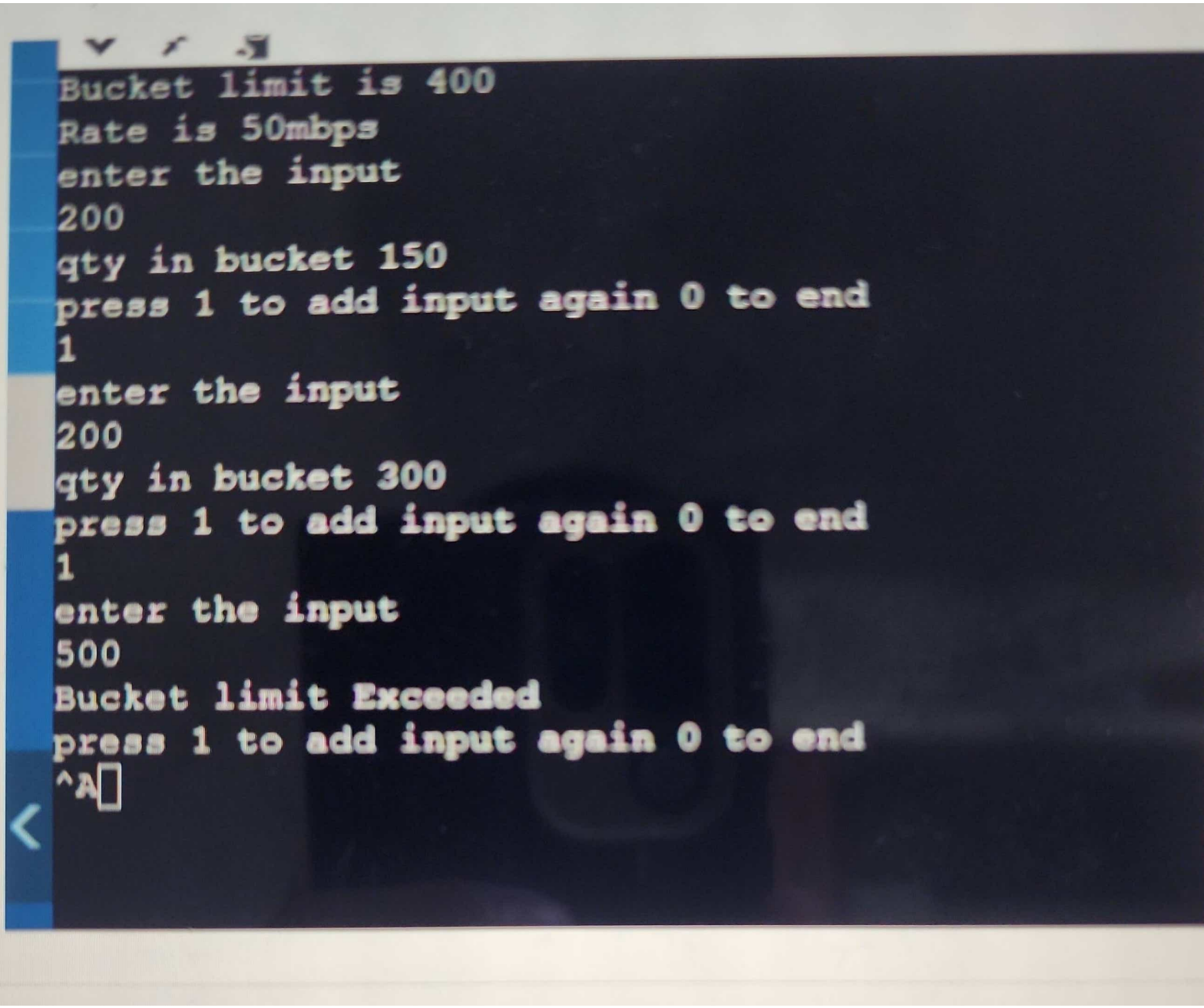
```
#include <stdio.h>
#include <stdlib.h>
int main(){
  int input=0, i=0, bucketlimit=400, op=1;
  Printf("Bucket limit is 400 \n");
  Printf("output rate is 50 mpps \n");
  while(op){
    Printf("Enter input: ");
    Scanf("%d",&i);
    if(i<=400 && input<=400){
      input+=i;
      input-=50;
```

```
if ( input <= 400) {
Printf ("qty in bucket : %d\n", input);
}
else if (input > 400)
 Printf ("Bucket limit Exceeded \n");
else
  Printf (" Bucket Limit Exceeded \n");
 Printf ("press 1 to add input \n 0 to end");
 scanf ("%d", &op);
 }
 return 0;
}
```

O/P :-

Bucket limit is 400
Rate is 50mbps
enter input
300
qty in bucket 250
Press 1 to add input, 0 to end
1
300
Bucket limit exceeded
Press 1 to add input 0 to end
1
Enter input
50
qty in bucket 250
Press 1 to add, input 0 to end

12/1/23


Bucket limit is 400
Rate is 50mbps
enter the input
200
qty in bucket 150
press 1 to add input again 0 to end
1
enter the input
200
qty in bucket 300
press 1 to add input again 0 to end
1
enter the input
500
Bucket limit Exceeded
press 1 to add input again 0 to end
^A

12/01/23

3) # Bellmann Ford algm: Program for distance vector algm to find suitable path for transmission

```
#include <stdio.h>
#include <stdlib.h>
int BellmanFord ( int G[20][20], int V, int E, 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] + dis 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;
 Pf("Enter no. of vertices:");
 Scanf("%d", &V);
 Pf("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("\nNo negative weight cycle \n");
else
   Printf("\n Negative weight cycle exists \n");
   return 0;
}
```

```
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 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
```

Q) Implement Dijkstras algm to find shortest path for given topology.

Dijkstras :-

```
#include<stdio.h>
#include<conio.h>
#define INFINITY 999
#define MAX 10
void dijkstra(int G[MAX][MAX], int n, int startnode);
int node(){
    int G[MAX][MAX], i, j, n, u;
    Printf("Enter no. of vertices");
    scanf("%d", &n);
    Printf("\n Enter adjacency matrix : \n");
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            scanf("%d", &G[i][j]);
    Printf("Enter starting node:");
    scanf("%d", &u);
    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)
            cost[i][j] = INFINITY;
        else
            cost[i][j]= G[i][j];

for(i=0; i<n; i++){
    distance[i]= cost[startnode][i];
    pred[i] = startnode;
    visted[i] = 0;
```

```
distance[startnode]=0;
visited[startnode]=1;
count=1;
while(count<n-1){
mindistance=INFINITY;
for(i=0;i<n;i++){
if(distance[i]<mindistance && visited[i]){
mindistance=distance[i];
nextnode=i;
}
visited[nextnode]=1;
for(i=0;i<n;i++)
if(!visited[i]){
if(mindistance(nextnode[i])<distance[i]){
distance[i]=mindistance+cost[nextnode][i];
Pred[i]=nextnode;
}
count++;
for(i=0;i<n;i++)
if(i!=startnode){
Printf("\n Distance of node %d = %d",i,
distance[i]);
j=i;
do{
j=Pred[i];
Printf("%d",j);
}while(j!=startnode);
}}
```

O/P ÷ Enter no. of vertices : 4

Enter adjacency matrix

	0	1	2	3
0	0	1	1	1
1	1	0	1	0
2	1	1	0	1
3	1	0	1	0

Enter starting node : 1

Distance of 0 = 1

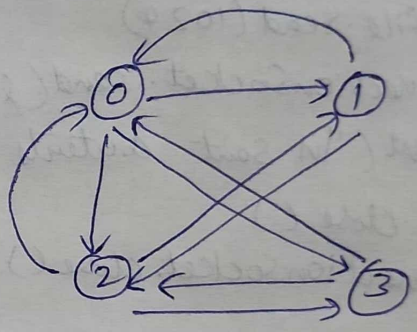
Path = 0 ← 1

Distance of 2 = 1

Path = 2 ← 1

Distance of 3 = 2

Path = 3 ← 0 ← 1



11) Using TCP/IP sockets, write client server Prgm to make client sending filename and server to send back contents of requested file if present

Sol:

client.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 filename: ")
clientSocket.send(sentence.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("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('In Sent contents of ' + sentence)
    file.close()
    ConnectionSocket.close()
```

O/P :

```
Enter filename: 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('In Sent contents of ' + sentence)
file.close()
ConnectionSocket.close()

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

12) Using UDP sockets, write client - server program to make the client sending file name and server to send back contents of requested file if present.

```
from socket import *
ServerName = "127.0.0.1"
Serverport = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
Sentence = input("\n Enter File name : ")
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))
```

```
fileContents, serverAddress = clientSocket.recvfrom(2048)
print('\n Reply from Server: \n')
print('\t', fileContents.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

```
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:
    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('\n Sent Contents of ', end = ' ')
    print(sentence)
    file.close()
```

Op: Enter file name: serverudp.py #client op
Reply from Server:

```
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:
    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('\n Sent Contents of ', end = ' ')
    print(sentence)
    file.close()
```

The server is ready to receive #server op