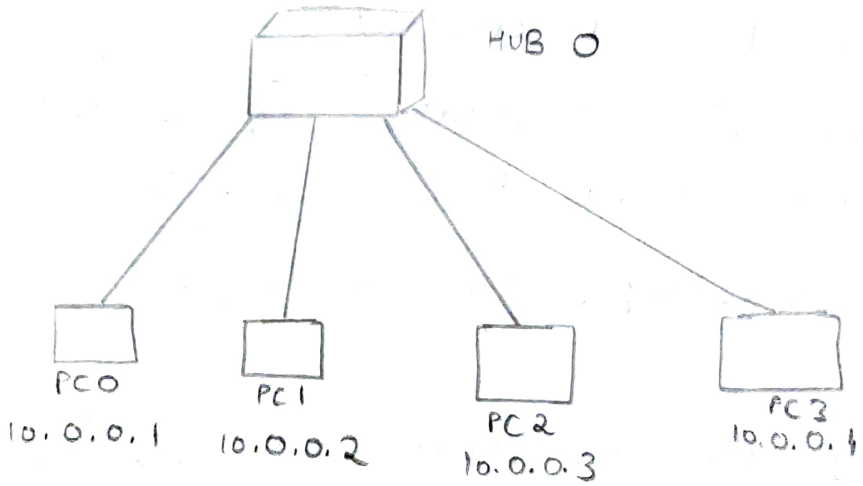
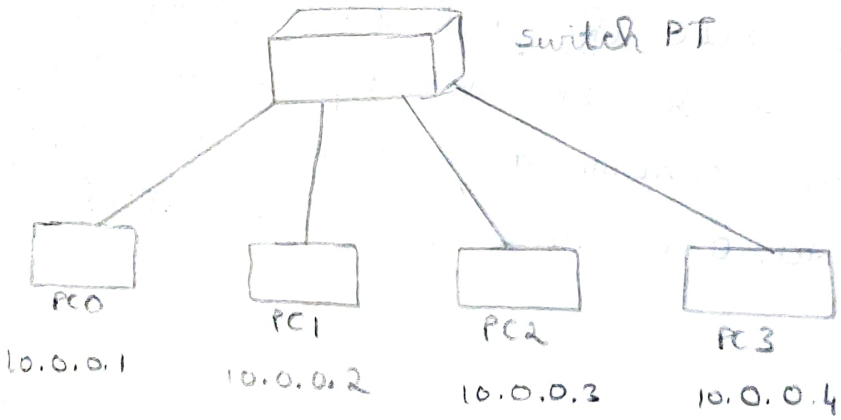
10/11/22

Experiment 1

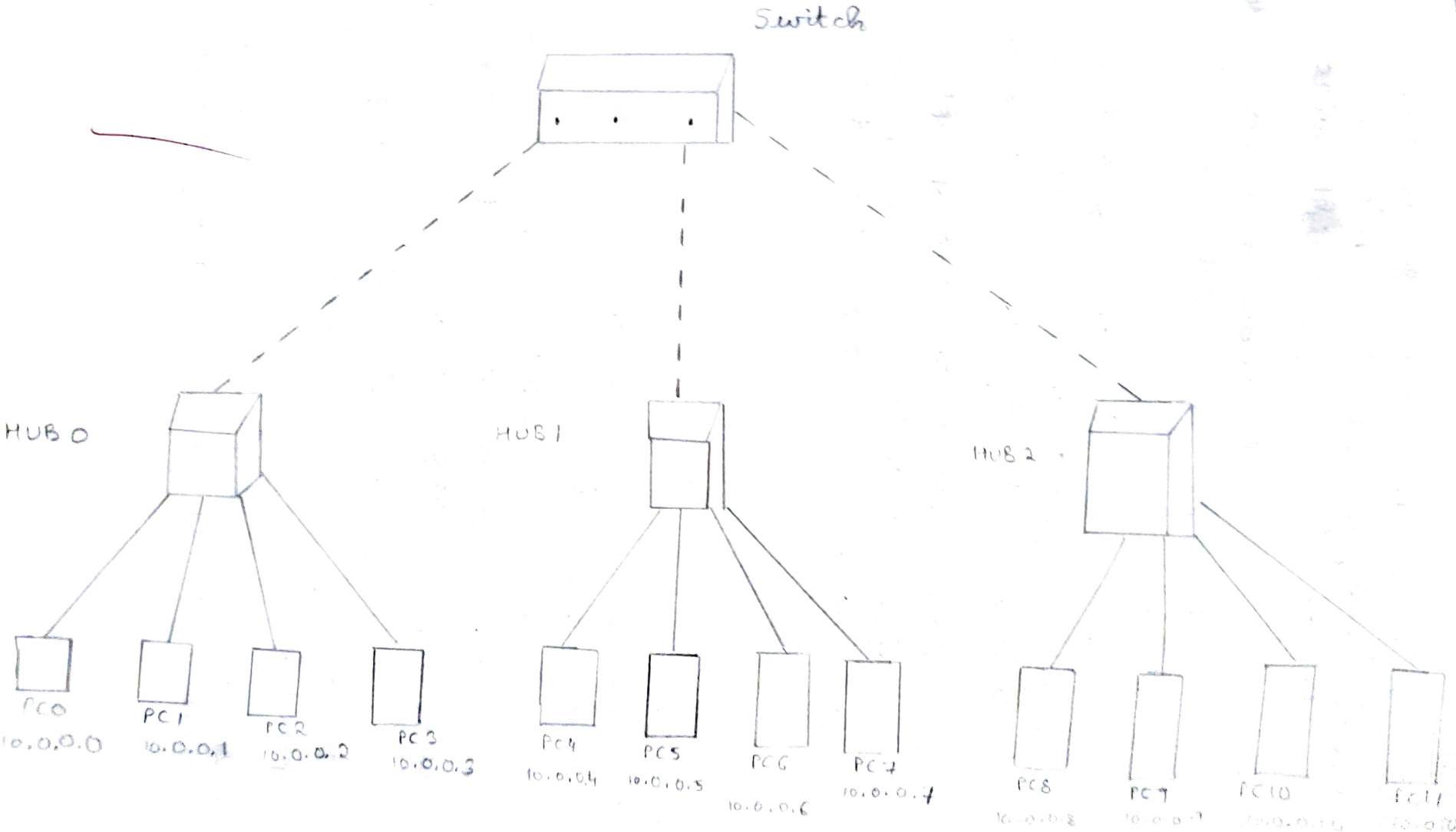
AIM:- Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.

Topology:-





Hybrid :-


Switch
HUB 0
HUB 1
HUB 2
PC0
10.0.0.0
PC1
10.0.0.1
PC2
10.0.0.2
PC3
10.0.0.3
PC4
10.0.0.4
PC5
10.0.0.5
PC6
10.0.0.6
PC7
10.0.0.7
PC8
10.0.0.8
PC9
PC10
PC11

Procedure :-

* HUB

→ Place 7 generic PC's and 1 generic hub in logical workspace and all 7 PC's are connected to hub, by copper straight wire.

→ Set each PC's with IP addresses from 10.0.0.0 to 10.0.0.6 respectively and connect each PC to hub by copper straight wire.

→ A simple PDU is placed on any 2 devices and message / packet passing can be seen in simulation mode, by clicking autocapture.

→ In realtime mode a command prompt is opened for certain PC and following command is given to transfer message PING destination IP - addresses.

* Switch

→ 4 generic PC's and one generic switch is placed on logical workspace.

→ Set IP address for each PC from 10.0.0.7 to 10.0.0.10 and connect each PC to switch using copper straight wire.

→ In simulation mode after placing simple PDU to any 2 PC's click auto - capture and packet transfer can be seen.

→ In real time mode click on any PC and open command prompt and type 'PING dest IP'

Hybrid :-

→ 12 PC's, 3 hubs, 1 switch all generics are placed on to logical workspace.

→ 3 generic hubs are connected to switch using copper cross over wire and 12 PC's are connected to 3 hubs, 4 PC each using copper straight wire after assigning IP address for each PC from 10.0.0.0 to 10.0.0.11 respectively.

→ After selecting 2 PC's from different hubs with simple PDU's and clicking on auto-capture, packets passing simulation can be seen in simulation mode.

→ In real time mode open command prompt by clicking any PC → devices → command prompt

Observations:-

* Hub

Learning outcome :-

→ After source sends message to hub its broadcasted to all end devices but only destination device reads and send response back to hub for source to get response.

→ Hub establishes connection to end-devices quickly and signals by green light.

Result:-

PING 10.0.0.3

PINGING 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 with bytes = 32 times 0ms

PING Statistics for 10.0.0.3

Details of how many packets sent and received.

Learning observation :-

→ Unlike hub, switch does not give green signal immediately but takes some amount of time called learning time and the packets can be sent once green signal is generated.

Result :-

```
PING 10.0.0.5
PINGING 10.0.0.5 with 32 bytes of data
  :
PING STATISTICS FOR 10.0.0.3:
```

Hybrid :-

Learning outcomes :-

Message sent by one PC of one hub to switch is sent to destination hub which broadcast to all devices of that hub and only destined end devices sends back response to source of other hub.

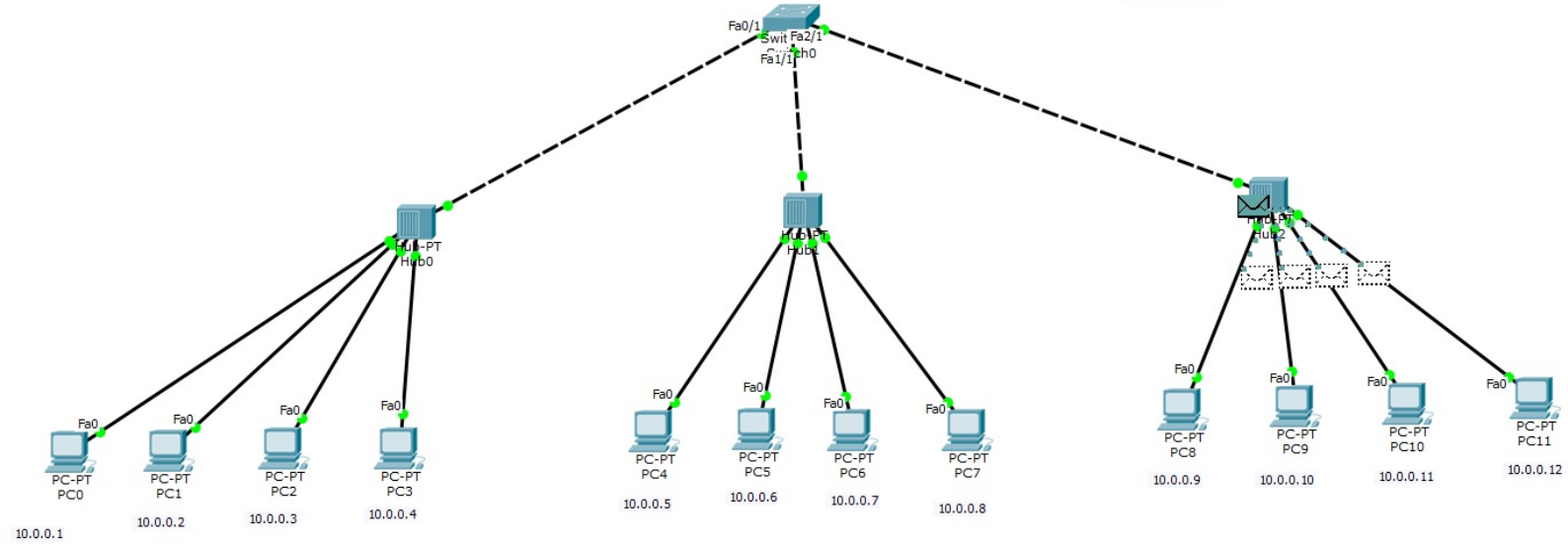
Result :-

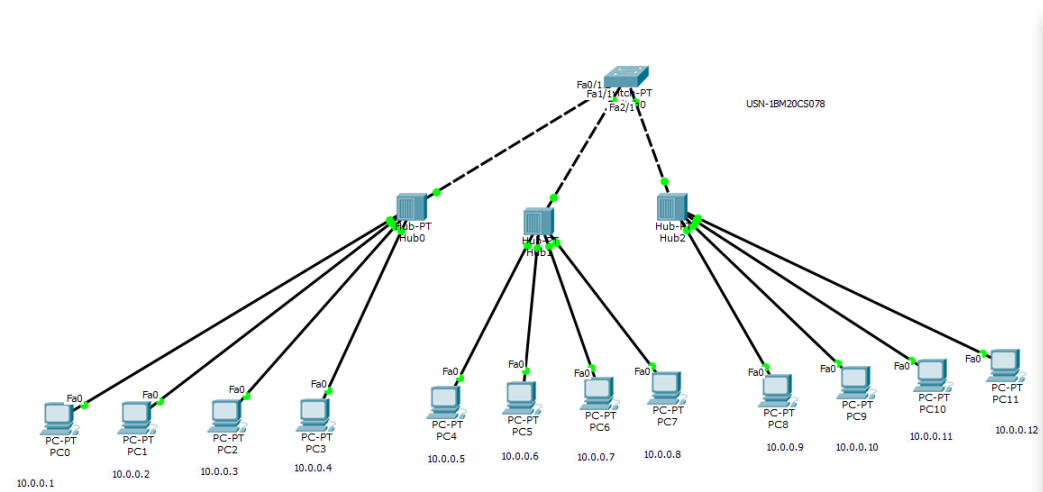
```
PING 10.0.0.4
PINGING 10.0.0.4 with 32 bytes of data
  Reply from 10.0.0.4 : bytes = 32

  PING statistics for 10.0.0.4
"Details of number of packets sent and received"
```

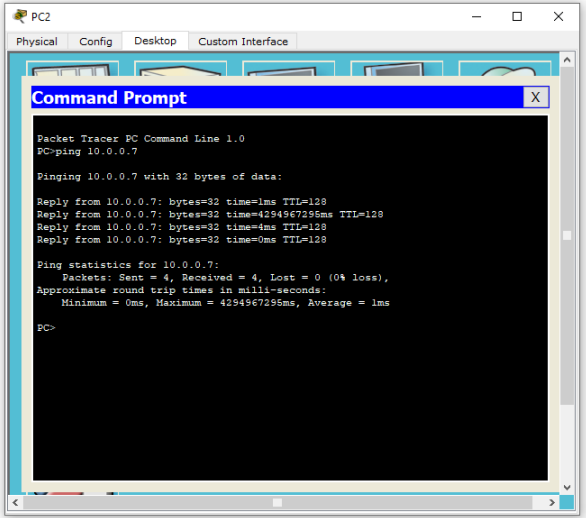
elabsca6

17/11/22

USN-1BM20CS078




Fa0/1
Fa1/1
Fa2/1
USN-1BM20CS078
Hub-PT
Hub0
Hub-PT
Hub1
Hub-PT
Hub2
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC6
PC-PT
PC7
PC-PT
PC8
PC-PT
PC9
PC-PT
PC10
PC-PT
PC11
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
10.0.0.8
10.0.0.9
10.0.0.10
10.0.0.11
10.0.0.12

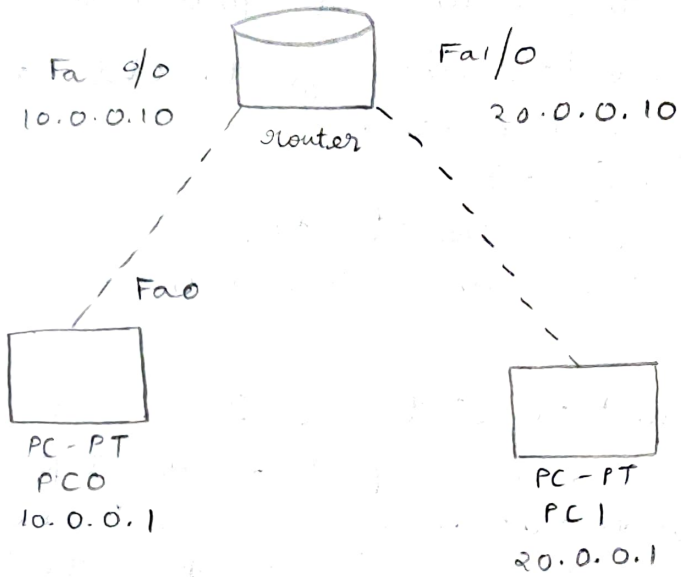

PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.7
Pinging 10.0.0.7 with 32 bytes of data:
Reply from 10.0.0.7: bytes=32 time=1ms TTL=128
Reply from 10.0.0.7: bytes=32 time=4294967295ms TTL=128
Reply from 10.0.0.7: bytes=32 time=4ms TTL=128
Reply from 10.0.0.7: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.7:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4294967295ms, Average = 1ms
PC>

17/4/22

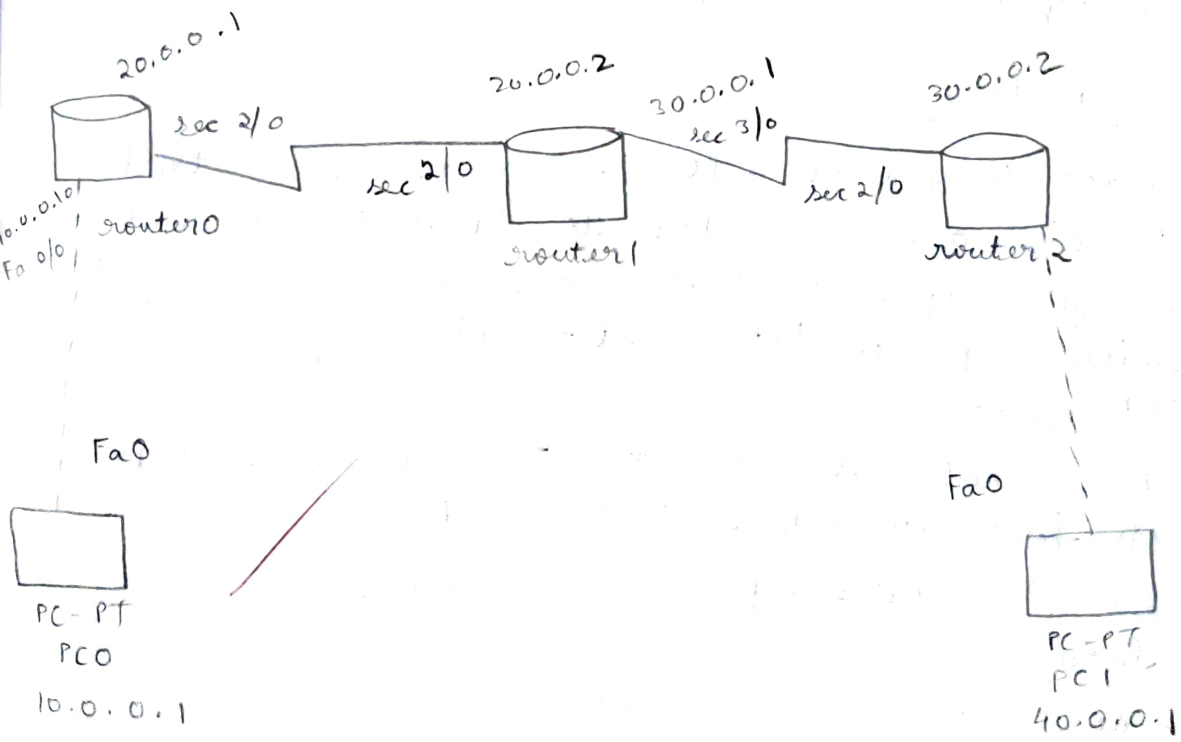
AIM:- Configuring IP address to routers in packet tracer. Explore the following messages: PING, response, destination unreachable, request timed out, reply.

Topology :-

One - router :-



Three Router :-



Procedure :-

One Router :-

→ Place 2 generic PC, a generic router and notes with IP address and connect using copper cross over

→ Click on both PC's and set IP addresses as 10.0.0.1 and 10.0.0.2 with gateway for each as 10.0.0.10 and 10.0.0.20.

→ Click on router, then go to CLI and execute following commands.

→ Enable → config - t
→ interface fast ethernet 0/0
→ ip address 10.0.0.10 255.0.0.0
→ no shut

Now connection between PC_0 and router turns green and repeat above steps for PC_1 until connection turns green. Route table can be seen by using "show ip route" command

→ Now ping a PC by selecting one of the PC → desktop → command prompt.

Three Routers

→ Place 2 generic 2 PC's, 3 routers such that 2 PC's are connected to 1st and 3rd router using copper cross and 3 routers are connected with sequence 1st → 2nd → 3rd using serial DCE cable.

→ Place notes to indicate IP addresses and ports of fast ethernet and serial ports

→ Set IP address, gateway, subnet mask for both PC's.

→ Now configure routers
* Click → no
* Enable
* Config t

* interface fastethernet 0/0.
* ip address 10.0.0.10 255.0.0.0
* no shut

By this first PC and left half of router connection is established.

* config t → interface serial 2/0 →
  ip address 20.0.0.1 255.0.0.0 → no shut

Now right side connection is established for 1st router.

→ Do the same steps for router 2 and three with corred IP addresses and port selection to establish connection on both sides of each of the routers successfully.

→ After all connections green lights are shown as a result of successful completion, now if we ping we get "destination unreachable".

→ If we ping to the other router 1 we get "request timed out" as routers are not trained for indirect LAN's.

* Train router 1 by :-
  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

* Train router 2 by :-
  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

* Train router 3 by :-
  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

→ Now pinging from PC0 to PC1 is successful.

Observations

One - router :

Pinging output for first time

Ping 20.0.0.1

request timed out.
Reply from 20.0.0.1 : bytes = 32
Reply from 20.0.0.1 : bytes = 32
Reply from 20.0.0.1 : bytes = 32

Ping statistics for 20.0.0.1 : packets sent = 4, received = 3, lost = 1

But when PC0 pings PC1 again or if reverse ping happens, we get reply all 4 times

3 routers :-

* Output when PC0 is pinged by PC1 or vice versa before routers are trained of unknown IP's

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.
Reply from 10.0.0.10 : Destination host unreachable.
Reply from 10.0.0.10 : destination host unreachable.

PING statistics 40.0.0.1
Packets sent = 4, received = 0, lost = 4

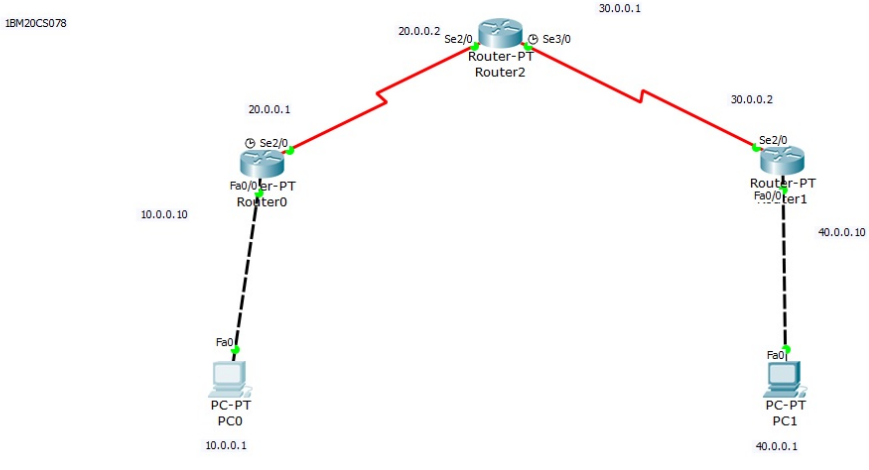
* PING 20.0.0.2
request timed out
request timed out
request timed out
request timed out

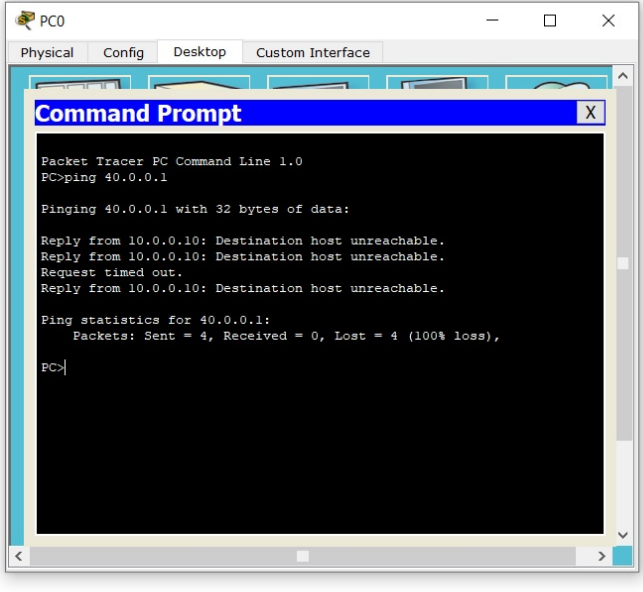
24/11/22

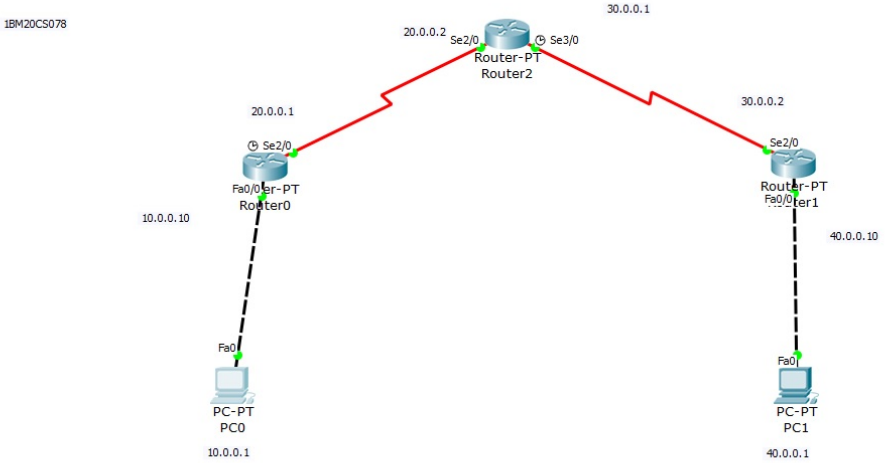
Pinging statistics 20.0.0.2
Packets sent = 4, received = 0, lost = 4

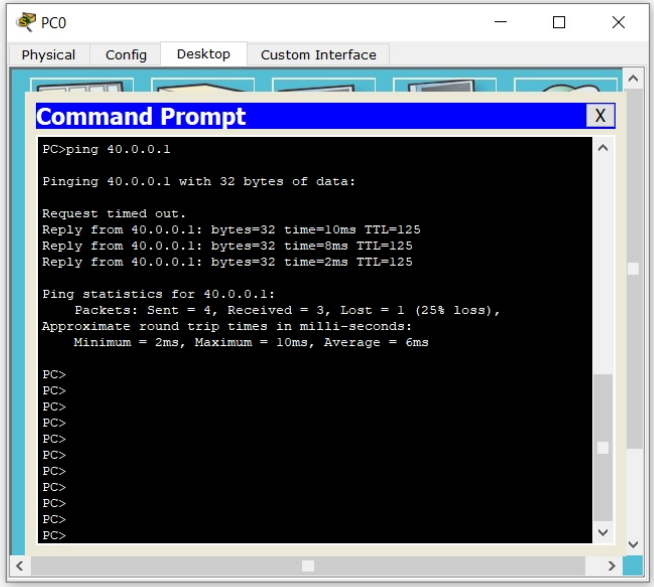
* When routers are trained
PING 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data
Request timed out
Reply from 40.0.0.1 : bytes = 32 times = 2 ms
Reply from 40.0.0.1 : bytes = 32 times = 2 ms
Reply from 40.0.0.1 : bytes = 32 time = 2 ms

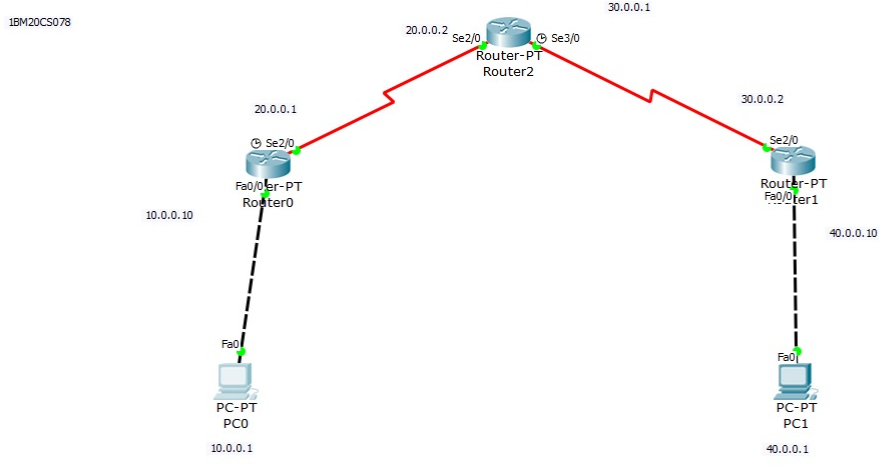
Pinging statistics for 40.0.0.1 packets sent = 4,

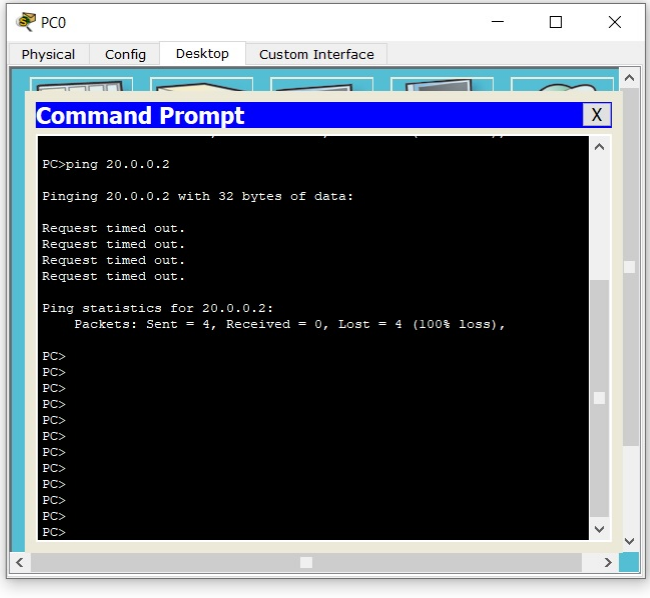

1BM20CS078
30.0.0.1
20.0.0.2
Se2/0
Se3/0
Router-PT
Router2
20.0.0.1
30.0.0.2
Se2/0
Se2/0
Fa0/0
Router-PT
Router0
Router-PT
Router1
10.0.0.10
40.0.0.10
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
10.0.0.1
40.0.0.1


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

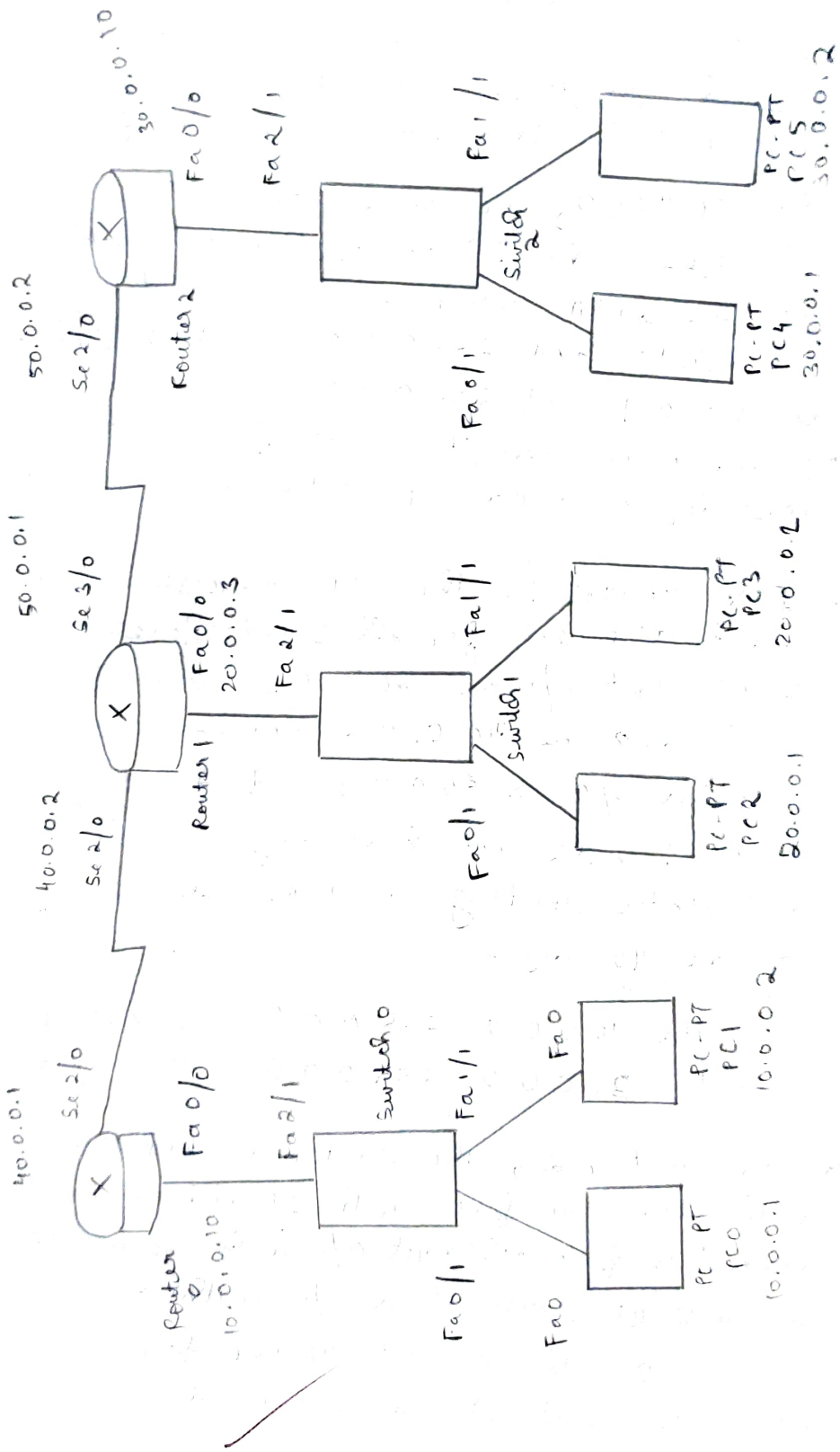

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 40.0.0.1:
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>

Aim :- Configuring default route to the router.

Topology :-



Procedure:-

→ Place 6 generic PC's, 3 switches and 3 routers and connect 2 PC's to each switch with copper straight wire and each switch is connected to one router with a copper straight wire and 3 routers are connected among themselves by serial DCE cable

PC0 and PC1 gateway - 10.0
PC2 and PC3 gateway - 20.0
PC4 and PC5 gateway - 30.0

→ A PC is clicked to set the attributes for a PC and each PC has 3 attributes which are the IP address, subnet mask and the gateway and all the three are set according to the nodes placed. This process is done for all 6 PC's

→ For router 0, the configuration are done in the CLI. The IP address and subnet mask are set for both interface fastethernet 0/0 as 10.0.0.10 and 255.0.0.0 and serial 2/0 as 40.0.0.1 and 255.0.0.0
Router 1 is default router for Router 0 and done by command ip route 0.0.0.0 0.0.0.0 40.0.0

→ For router 1, the IP addresses and subnet mask are set for all 3 interfaces, fastethernet 0/0 as 20.0.0.3 and 255.0.0.0 serial 2/0 as 40.0.0.2 and 255.0.0.0 and serial 3/0 as 50.0.0.1 and 255.0.0.0.
Router 1 does not have any default routers and static routing is done for network 10 and 40 by:-

ip route 10.0.0.0 255.0.0.0 40.0.0.1
ip route 30.0.0.0 255.0.0.0 50.0.0.2

→ For router 2, the router is configured in both the interface with IP address and subnet mask as fast ethernet 0/0 with 30.0.0.10 and 255.0.0.0 and serial 2/0 with 50.0.0.2 and 255.0.0.0 The default router for router 2 is router 1 and 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 20.0.0.1 and from 10.0.0.1 to 30.0.0.2

Observations:-

Learning outcomes:-

→ One router cannot have 2 default routers.
→ The default router for first router is the middle router because any packets which have to delivered will go to the middle router.
→ The default router for third router is the middle router for same reason.
→ The middle router does not have any default router because if one of the router is made default then there is a chance that the packets which are to be sent to the switch and sent to the router.

Result:-

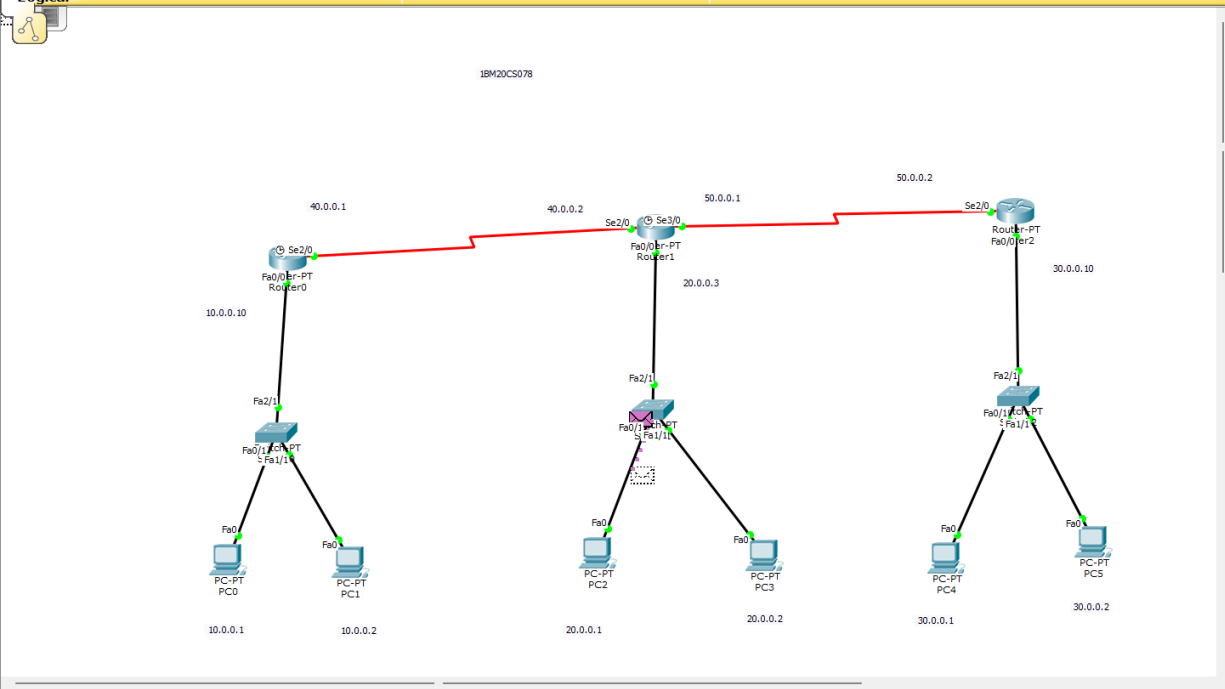
ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data
Request timed out
Reply from 20.0.0.1 bytes = 32, time = 1ms, TTL = 126
Reply from 20.0.0.1 bytes = 32, time = 2 ms, TTL = 126
Reply from 20.0.0.1 bytes = 32, time = 6ms, TTL = 126

24/11/22

Cisco Packet Tracer Student - C:\Users\bmsce\Desktop\lab3-udaya\lab3 program.pkt

File Edit Options View Tools Extensions Help

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



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	
	0.001	PC0	Switch0	ICMP	
	0.002	Switch0	Router0	ICMP	
	0.003	Router0	Router1	ICMP	
	0.004	Router1	Switch1	ICMP	
	0.005	Switch1	PC2	ICMP	

Reset Simulation | Constant Delay | Capturing...

Play Controls

Back | Auto Capture / Play | Capture / Forward

Event List Filters - Visible Events

ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Edit Filters | Show All/None

Time: 00:30:54.562 | Power Cycle Devices | PLAY CONTROLS: | Back | Auto Capture / Play | Capture / Forward | Event List | Simulation

Connections

Automatically Choose Connection Type

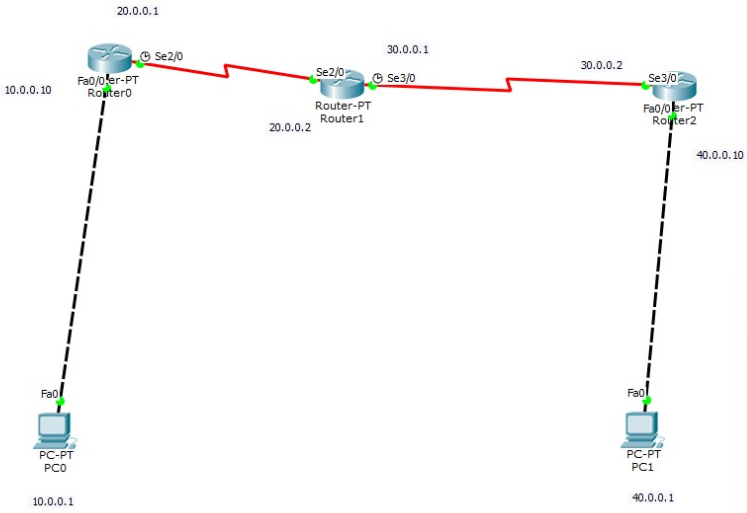
Scenario 0 | New | Delete | Toggle PDU List Window

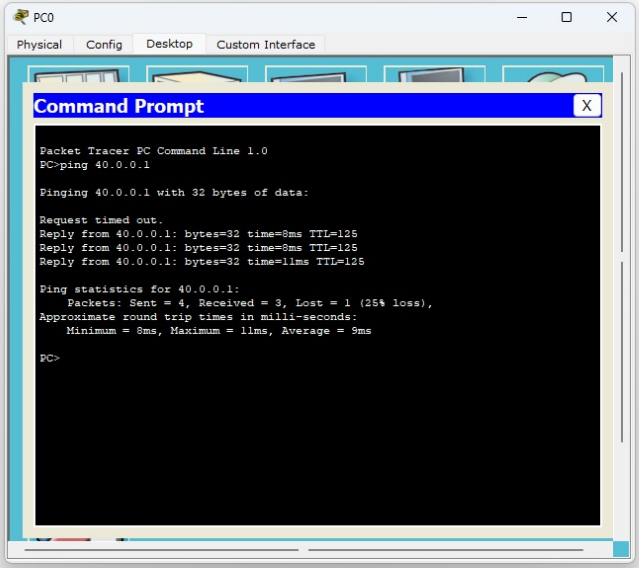
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	In Progress	PC0	PC2	ICMP		0.000	N	0	(edit)	(delete)

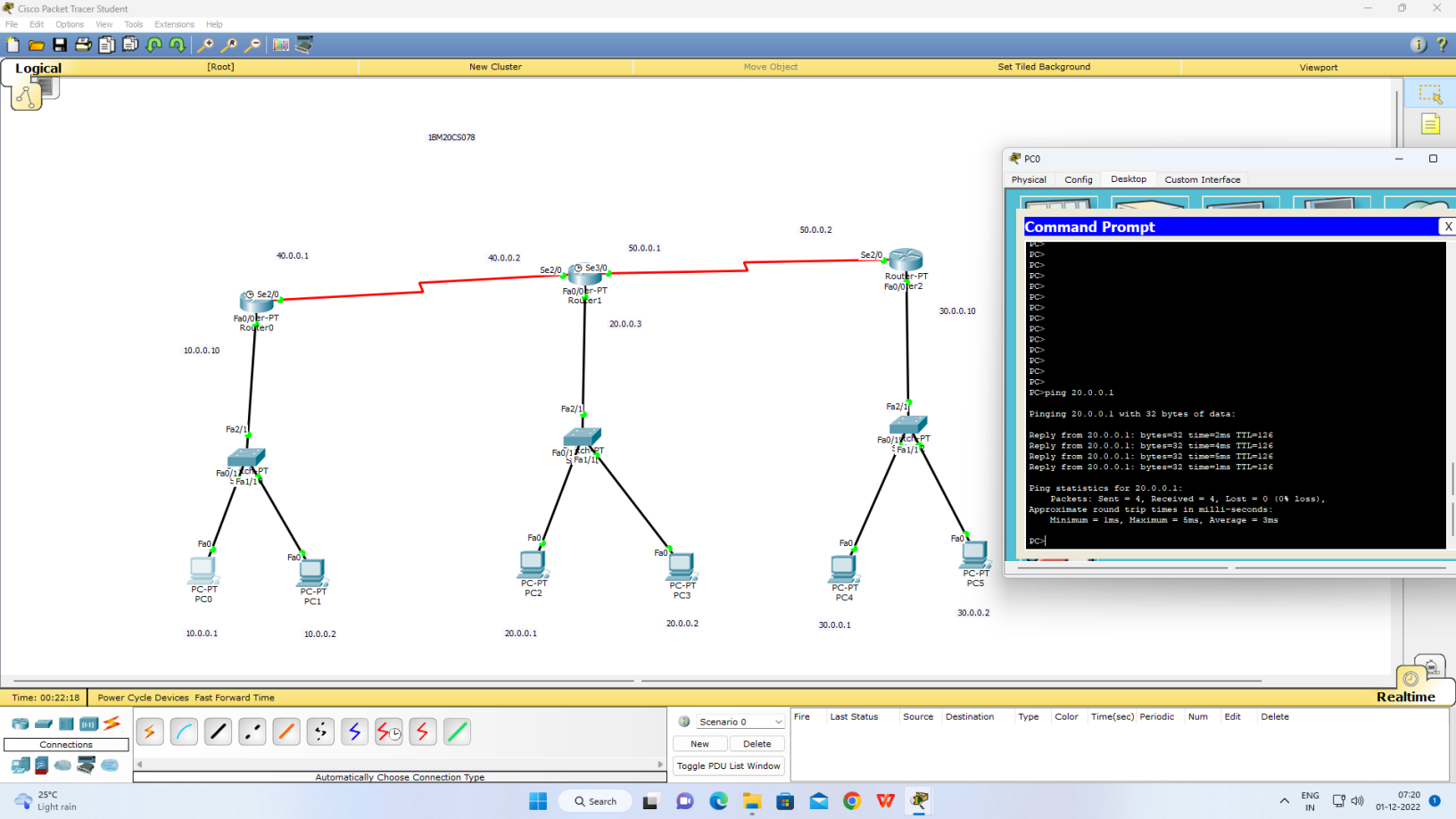
25°C Light rain

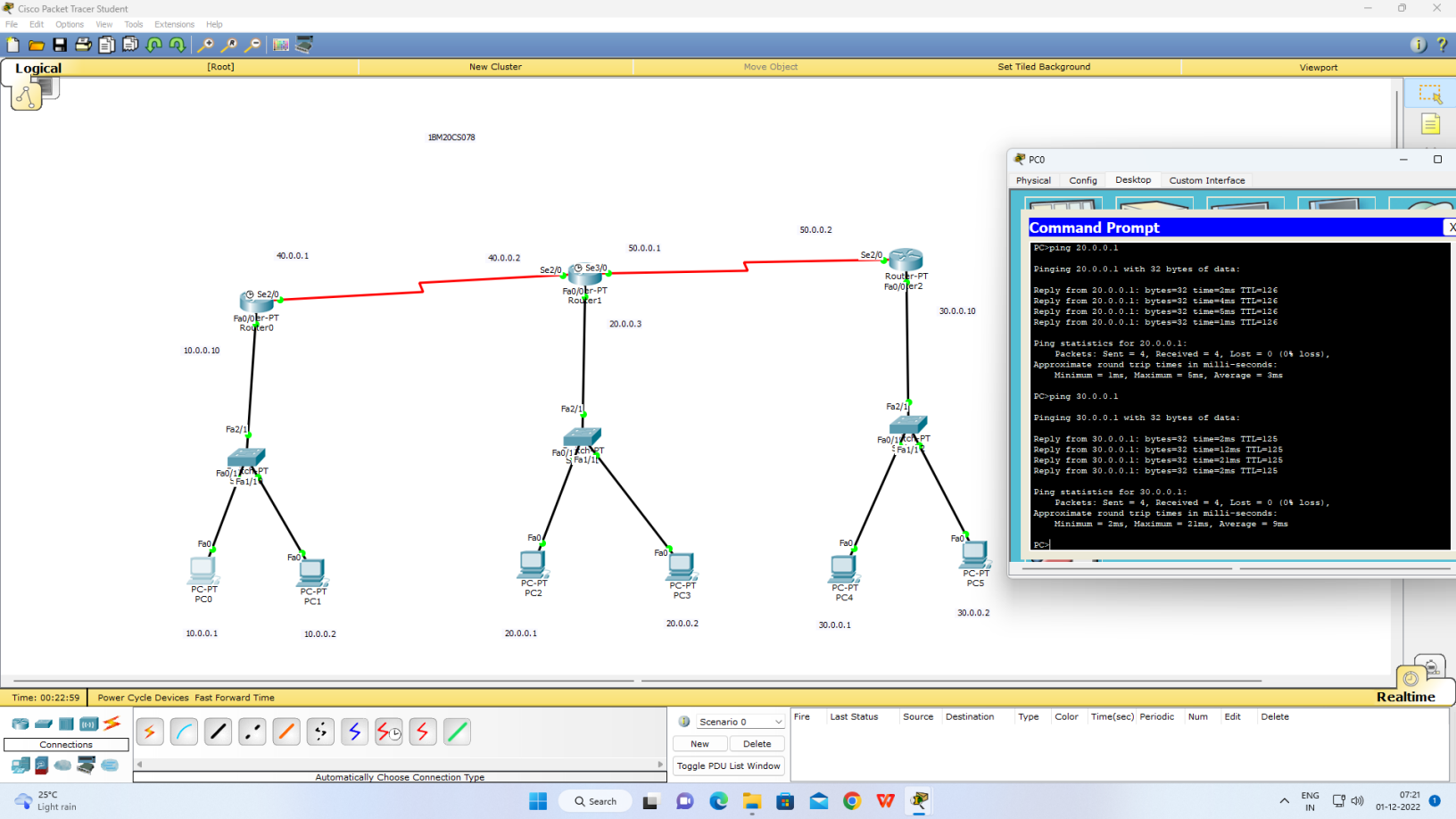
Search

ENG IN 07:29 01-12-2022


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

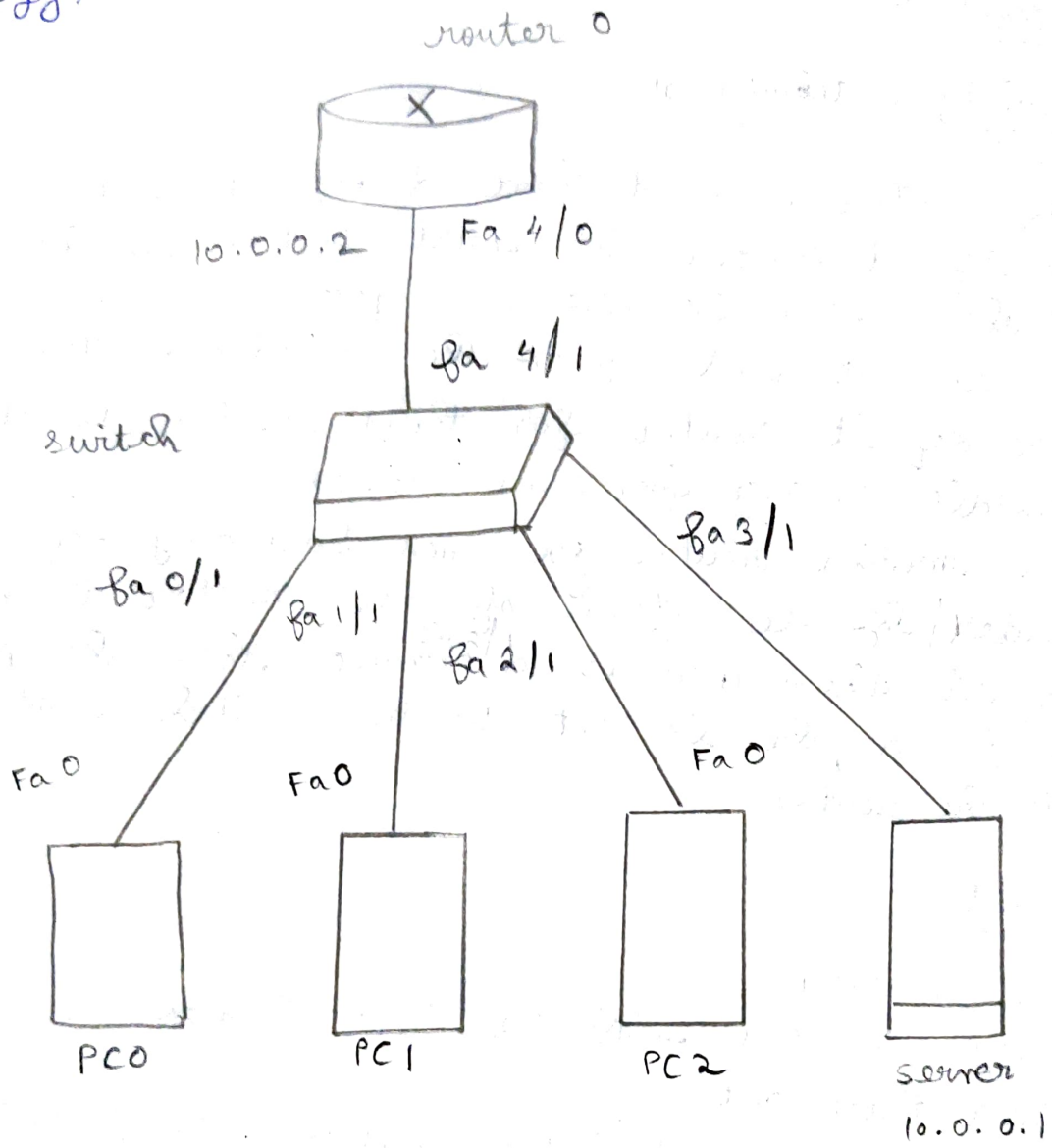

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=11ms 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
Switch-PT
Fa0
PC-PT
PC0
PC1
PC2
PC3
PC4
PC5
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
Time: 00:22:18
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
07:20
01-12-2022


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

AIM :- Configuring DHCP within a LAN in a packet tracer.

Topology :-



Procedure :-

* Place 3 PC's, one router, one generic switch, one server and connect everything with copper straight wire.
* Place note 10.0.0.2 for router indicating gateway and 10.0.0.1 for server indicating its IP address
* Interface the corresponding fastethernet interface for router using CLI :-

→ enable
→ config t
→ interface fastethernet 3/0
→ ip address 10.0.0.2 255.0.0.0
→ no shut

* Now configure server by clicking services tab in server :
→ set default gateway
→ set DNS server
→ set starting IP address as 10.0.0.3
→ Max. no of users is 8
→ TFTP server as 10.0.0.1
→ Then click on save.
* Then click on PC0, go to desktop, click on IP configuration and you click on DHCP.

Observations :-

When we select DHCP mode, it automatically fetches, gateway from server for which we had configured already.

Result :-

ping 10.0.0.5

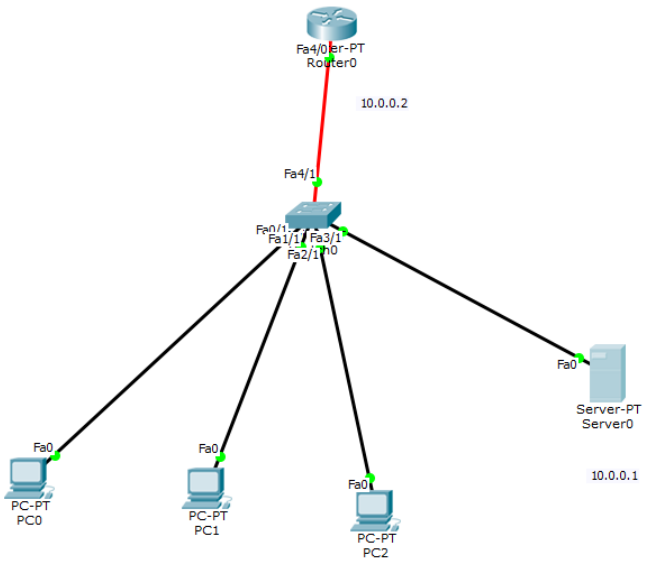
pinging 10.0.0.5 with 32 bytes of data :-

Reply from 10.0.0.5, bytes = 32 time = 1ms 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

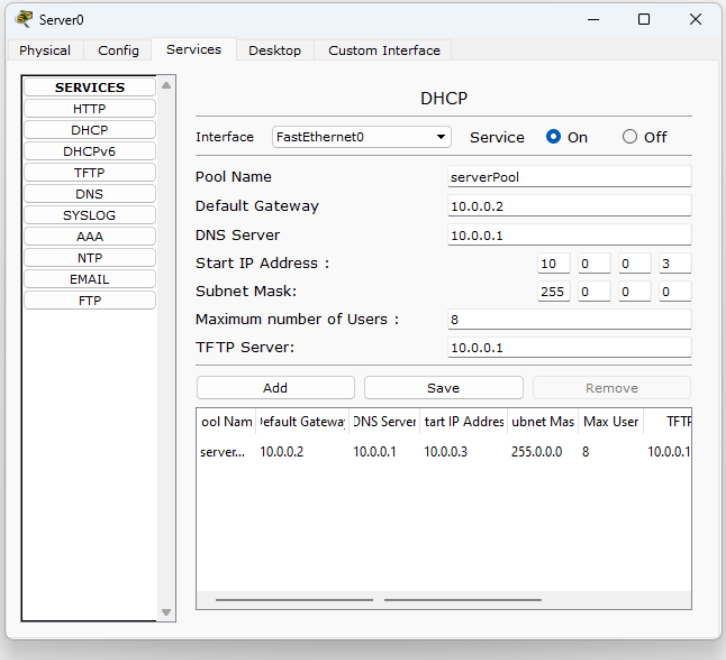
ping statistics for 10.0.0.5 :-

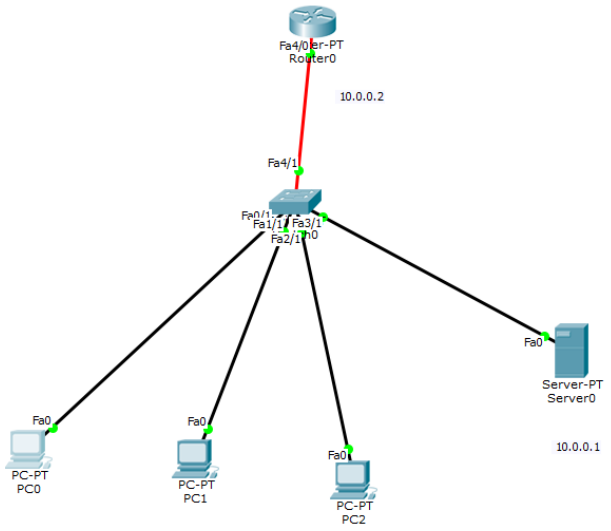
Packets sent = 4, Received = 4, Loss = 0 (0% loss)

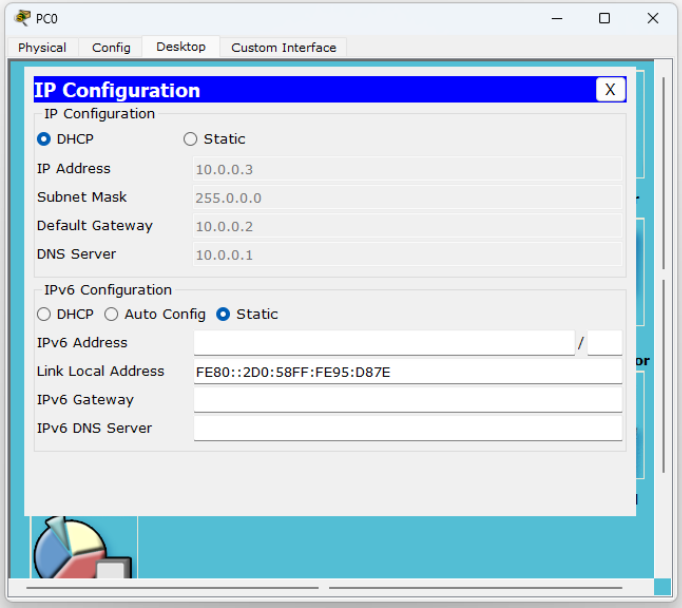
8/12/22

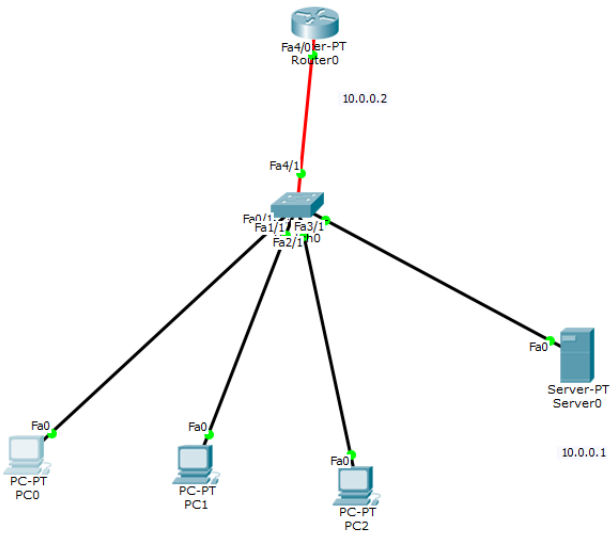

Fa4/0er-PT
Router0
10.0.0.2
Fa4/1
Fa0/1
Fa1/1
Fa3/1
Fa2/1
Fa0
Server-PT
Server0
10.0.0.1
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2

1BM20CS078

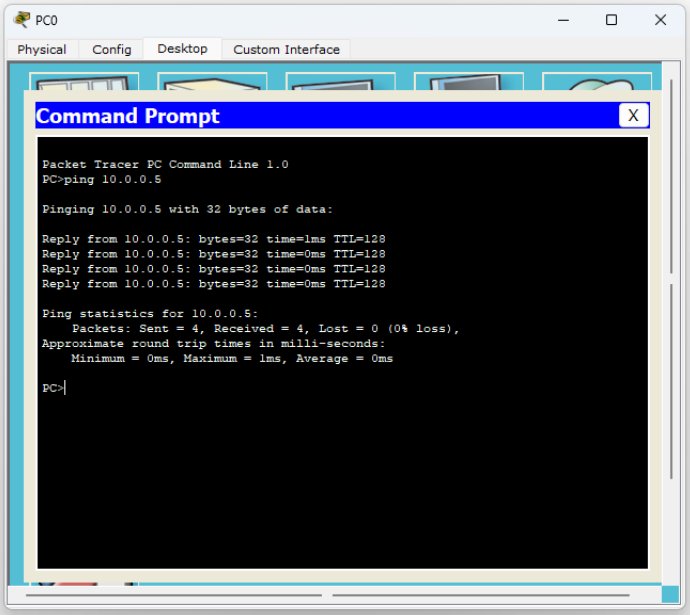

Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
serverPool
Default Gateway
10.0.0.2
DNS Server
10.0.0.1
Start IP Address :
10 0 0 3
Subnet Mask:
255 0 0 0
Maximum number of Users :
8
TFTP Server:
10.0.0.1
Add
Save
Remove
ool Nam
efault Gatewa
DNS Server
tart IP Addres
ubnet Mas
Max User
TFTP
server...
10.0.0.2
10.0.0.1
10.0.0.3
255.0.0.0
8
10.0.0.1

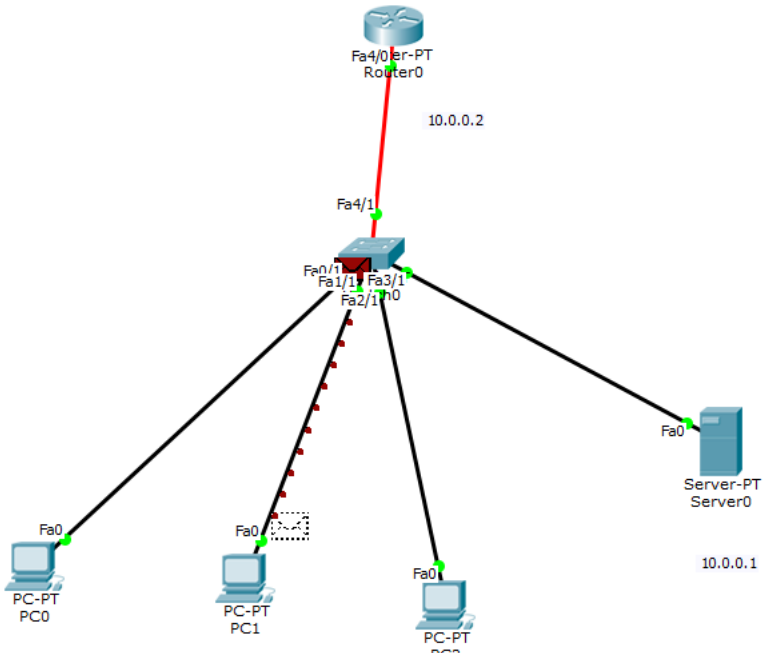

Fa4/0
Router0
10.0.0.2
Fa4/1
Fa0/1
Fa1/1
Fa3/1
Fa2/1
Fa0
Server-PT
Server0
10.0.0.1
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2
1BM20CS078


PC0
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.2
DNS Server
10.0.0.1
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2D0:58FF:FE95:D87E
IPv6 Gateway
IPv6 DNS Server


Fa4/0er-PT
Router0
10.0.0.2
Fa4/1
Fa0/1
Fa1/1
Fa2/1
Fa3/1
h0
Fa0
Server-PT
Server0
10.0.0.1
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2

1BM20CS078

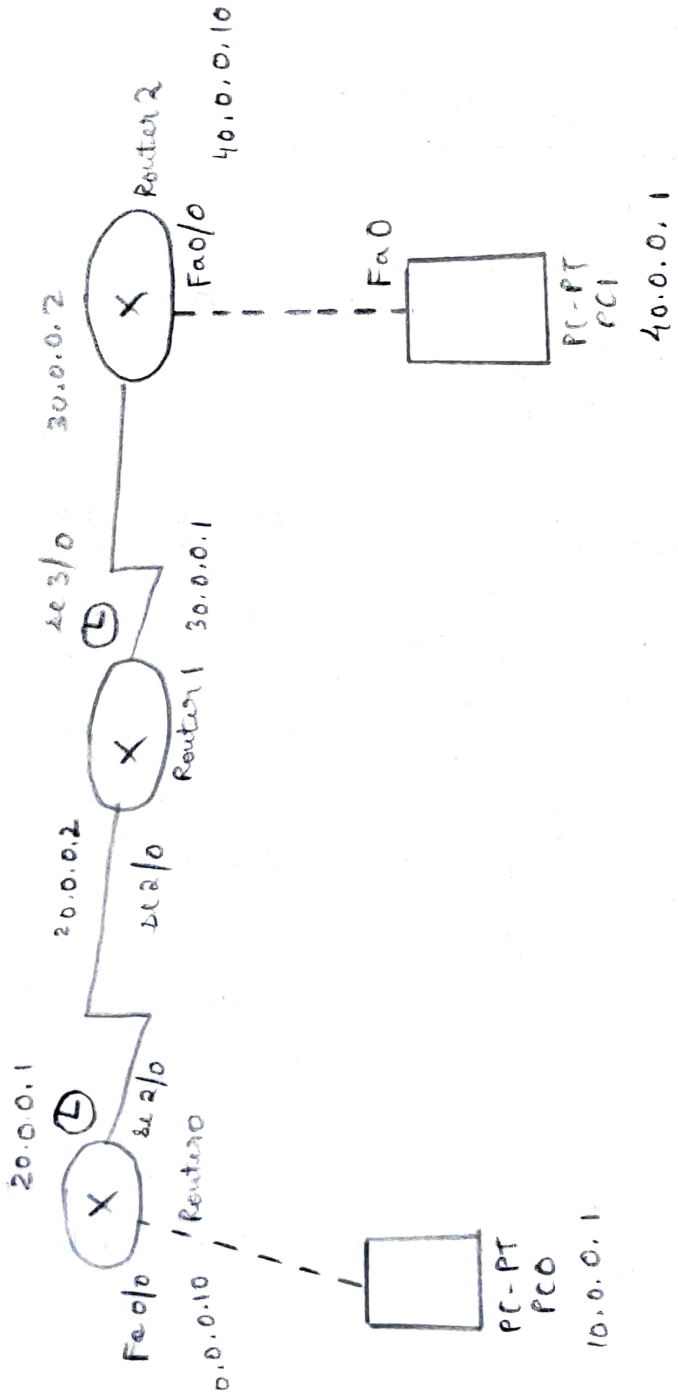

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time=1ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 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 = 1ms, Average = 0ms
PC>


Fa4/0
Router-PT
Router0
10.0.0.2
Fa4/1
Fa0/1
Fa1/1
Fa3/1
Fa2/1
Switch0
Fa0
Server-PT
Server0
10.0.0.1
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2
1BM20CS078

2/12/22

AIM :- Configuring RIP routing protocol in routers

RIP :- RIP is routing information protocol which finds best path between source and destination network. It is a distance vector routing protocol.



Procedure :-

* Place 2 generic PC's, 3 routers, and place notes to indicate respective IP address
* For PC-PT PC0 set IP address 10.0.0.1
  subnet 255.0.0.0
  and gateway as 10.0.0.10
* For PC-PT PC1 set IP address 40.0.0.1
  subnet 255.0.0.0
  and gateway as 40.0.0.10
* For router 0

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

⟹ interface serial 2/0
→ ip address 20.0.0.1 255.0.0.0
→ encapsulation ppp
→ clock rate 64000
→ no shut

* Use the above commands for interfacing router which has clock symbol in cable near it and for interfacing other routers use the same commands except "clock rate 64000"
* Place correct values while interfacing the routers.
* Once all the green lights are visible, follow the commands below for the routers,

→ router rip
→ network 10.0.0.0 } known networks
→ network 20.0.0.0 } are written here.
→ exit

* Repeat above commands for router 1 and router 2 with respective network addresses.

Observation :-

* In static IP routing we need to teach the routers independently, but we make use of RIP so that routing becomes easy when large number of routers are used.

DCE - ?

Result :-

Pinging 40.0.0.1 with 32 bytes of data :

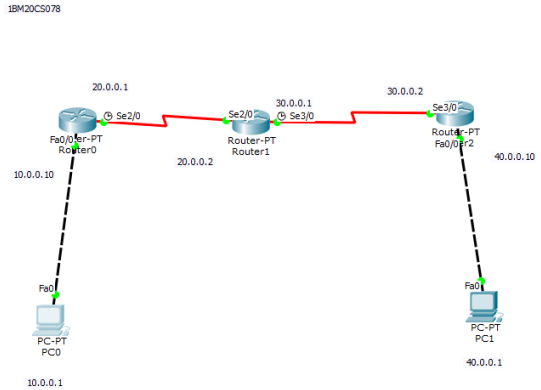
Reply from 40.0.0.1 bytes = 32 time = 14 ms TTL = 125
Reply from 40.0.0.1 bytes = 32 time = 15ms TTL = 125
Reply from 40.0.0.1 bytes = 32 time = 0ms TTL = 125
Reply from 40.0.0.1 bytes = 32 time = 16ms TTL = 105

Ping statistics for 40.0.0.1
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



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 40.0.0.1: bytes=32 time=14ms TTL=125
Reply from 40.0.0.1: bytes=32 time=15ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms 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 = 8ms, Maximum = 16ms, Average = 13ms

PC>
```

Time: 00:08:57 | Power Cycle Devices | Fast Forward Time | Realtime

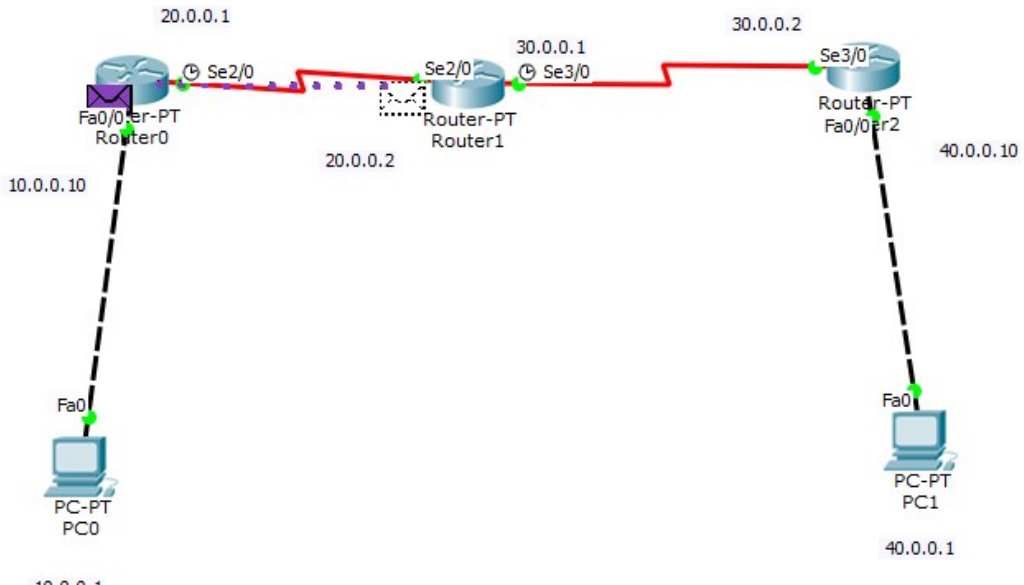
Connections | Serial DCE

Scenario 0 | New | Delete | Toggle PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)	(delete)

25°C Haze | Search | ENG IN | 12:17 08-12-2022

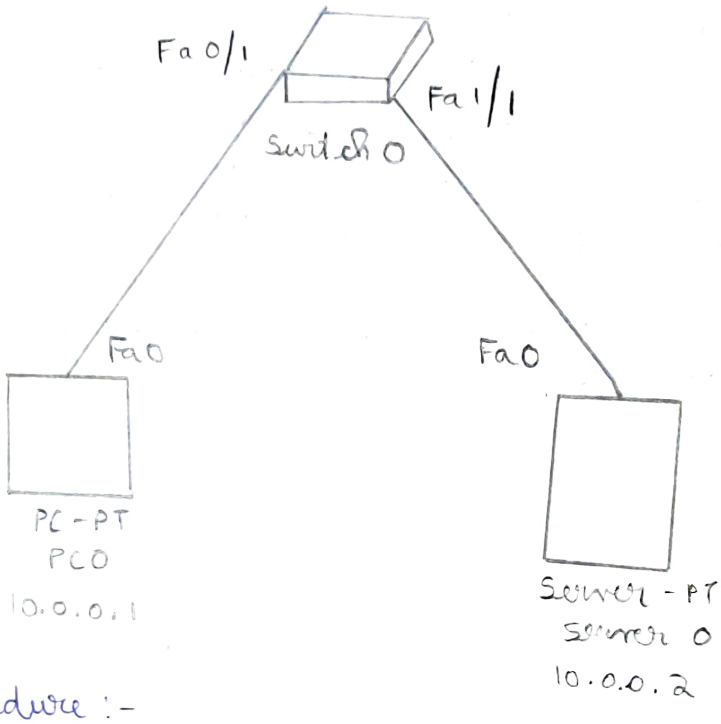
1BM20CS078


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

15/12/22

AIM:- Demonstration of WEB server and DNS using packet tracer.

Topology:-



Procedure:-

* Set the IP address of PC (10.0.0.1) and server (10.0.0.2) according. From the PC invoke the web browser in the Desktop tab and give ip of server.

* Now go to services tab in properties of server, click on HTTP. A window is seen, here click on edit button of index.html and change text from CISCO to BMSCE. Save to overwrite and see modified page.

* Now to activate DNS, click on DNS services. Select On, enter the name and IP as seen in the figure. Click on add button.
IP ⇒ 10.0.0.2

* Now give name in the web browser of PC0 to check if it is displaying index.html.

Custom-page

→ Create a new page resume.html and save it in http services.

→ Change hyperlink in index.html to link the above created file.

→ Check the output in web browser of PC0 by click on hyper-link.

Observations

→ We can view the webpage then we type "csebmsce" in browser because 10.0.0.2, address is mapped to the name "csebmsce" by domain name system concept.

→ Mapping is required its difficult to users to remember IP's, hence IP is mapped with a name.

Result :-

Web Browser ×

< > URL http://csebmsce

BMSCE Packet Tracer

Welcome to BMSCE packet tracer.

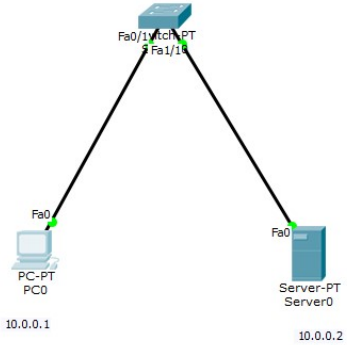
Quick links

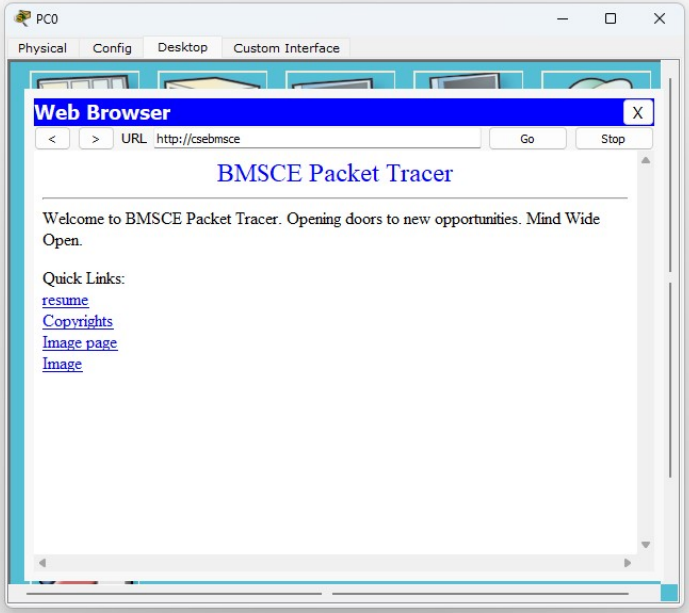
resume

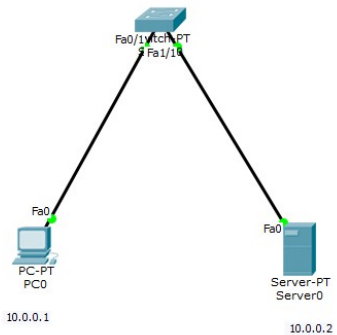
http://csebmsce/resume.html

NAME	Lokesh
DOB	18/3/2002
COLLEGE	BMSCE.

9-12-22


Fa0/1
Fa1/1
Fa0
Fa0
PC-PT
PC0
Server-PT
Server0
10.0.0.1
10.0.0.2


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL
http://csebmsce
Go
Stop
BMSCE Packet Tracer
Welcome to BMSCE Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
resume
Copyrights
Image page
Image



Server0

Physical | Config | Services | Desktop | Custom Interface

SERVICES

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP

HTTP

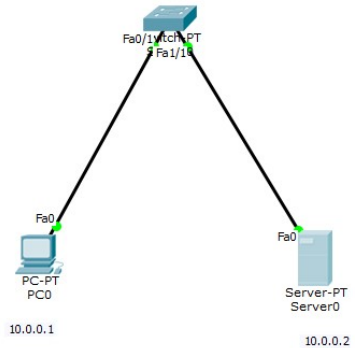
HTTP: (•) On ( ) Off

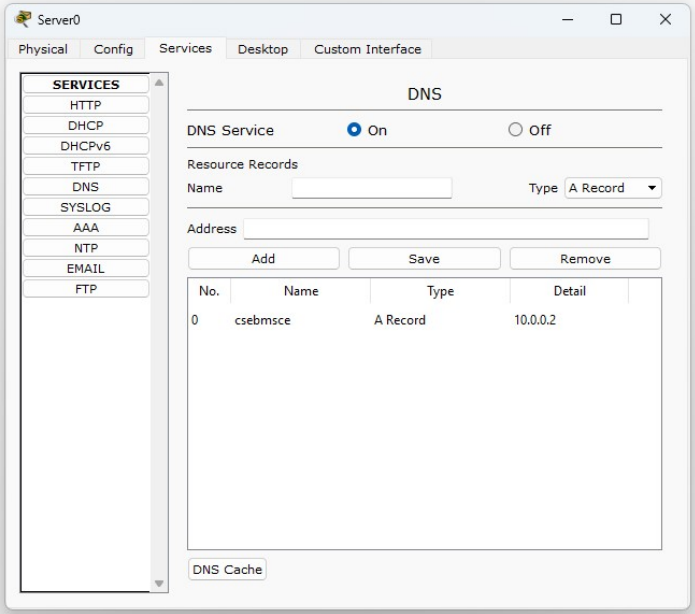
HTTPS: (•) On ( ) Off

File Manager

	File Name	Edit	Delete
1	copyrights.html	(edit)	(delete)
2	cscoptlogo177x...		(delete)
3	helloworld.html	(edit)	(delete)
4	image.html	(edit)	(delete)
5	index.html	(edit)	(delete)
6	login.html	(edit)	(delete)
7	login1.html	(edit)	(delete)
8	resume.html	(edit)	(delete)

New File | Import


Fa0/1
Fa1/1
Fa0
PC-PT
PC0
10.0.0.1
Fa0
Server-PT
Server0
10.0.0.2


Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DNS
DNS Service
On
Off
Resource Records
Name
Type
A Record
Address
Add
Save
Remove
No.
Name
Type
Detail
0
csebmsce
A Record
10.0.0.2
DNS Cache

AIM :- Write a program for error detecting using CRC

```
#include <stdio.h>
void main()
{
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 );
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] == key[j] )
 rem[j-1] = '0';
 else
 rem[j-1] = '1';
 }
```

```
rem[keylen-1] = input[i+keylen];
strcpy(temp, rem);
}
strcpy(rem, temp);
printf("Quotient is ");
for(i=0; i<msglen; i++)
printf("%c", quot[i]);
printf("\nRemainder ");
for(i=0; i<keylen-1; i++)
printf("%c", rem[i]);
printf("Modified data is");
for(i=0; i<msglen; i++)
printf("%c", input[i]);
for(i=0; i<keylen-1; i++)
printf("%c", rem[i]);
getch();
}
```

Output :-

Enter data :- 1011010101

Enter G(x) :- 1010

Quotient is :- 1001000100

Remainder :- 000

Modified :- 1011010101000

Checking for error

Enter data :- 1011010101000

G(x) :- 1010

Remainder :- 0 0 0

```
Enter data to be transmitted: 1000100000100001

Enter the Generating polynomial: 1011101

Data padded with n-1 zeros : 100010000010000100000
CRC or Check value is : 010011
Final data to be sent : 100010000010000101011
Enter the received data: 100010000010000101011
Data received: 100010000010000101011
No error detected



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

Write a program to implement distance vector algorithm

```
#include <stdio.h>
#include <stdlib.h>
int bellmanFord (int G[20][20], int v, int edge[10][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");
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");
else
printf("\n Negative weight cycle \n");
return 0;
}
```

Output :-

Enter no of vertices = 5

Enter graph in matrix form

```
0  6  0  7  0
0  0  5  8 -4
0 -2  0  0  0
0  0 -3  0  9
2  0  7  0  0
```

Enter source : 1

Vertex 1 → cost = 0 parent = 0

Vertex 2 → cost = 2 parent = 3

Vertex 3 → cost = 4 parent = 4

Vertex 4 → cost = 7 parent = 1

Vertex 5 → cost = -2 parent = 2

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
```

Write a program to implement dijistras algorithm.

```
#include<stdio.h>
#include<conio.h>
#define INFINIY 9999
#define MAX 10
void dijikstra(int G[MAX][MAX], int n, int startnode)
int main()
{ int G[MAX][MAX], i, j, n, u;
  printf("Enter no of vertices");
  scanf("%d", &n);
  printf("\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", &n);
  dijikstra(G, n, u);
  return 0;
  }
void dijikstra(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,
visited[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]);
printf("\n path = %d",i);
j = i;
do
{ j = pred[j];
printf ("<- %d", j);
} while (j != startnode);
}}
```

Output :-

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

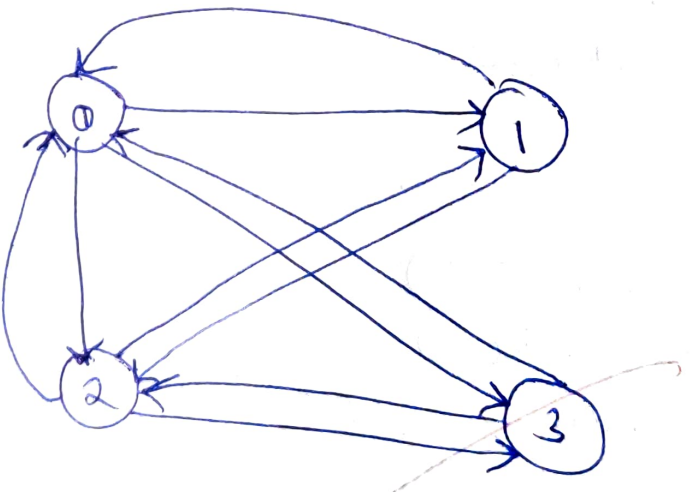
Path = 0 ← 1

Distance of 2 = 1

Path = 2 ← 1

Distance of 3 = 2

Path = 3 ← 0 ← 1



```
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
```

5/1/2023

Leaky Bucket

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  int input = 0;
  int i = 0, bs;
  printf("Enter bucket limit \n");
  scanf("%d", &bs);
  int op = 1;
  printf("Bucket limit is %d ", bs);
  printf("Rate is 50 mbps \n");
  while(op)
  { printf("Enter the input \n");
    scanf("%d", &i);
    if(i <= bs && input <= bs)
    { input = input + i;
      input = input - 50;
      if(input <= bs)
      { if(input < 0)
          input = 0;
        printf("qty in bucket %d", input);
      }
      else if(input > bs)
    { printf("Bucket limit exceeded");
      }
      else
      { printf("Bucket limit exceeded"); }
    printf("press 1 to add input, 0 to end");
    scanf("%d", &op);
    }
    return 0;
}
```

Output :-

```
Bucket limit is 400
Rate is 50 mbps
enter input
300
qty in bucket 250
press 1 to add input 0 to end
```

300

Bucket list exceeded.

press 1 to add

1

enter input

50

qty in bucket 25.

12-1-2023

```
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.
```

Socket Programming :-

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

Client TCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print('\n From server\n')
print(fileContents)
clientSocket.close()
```

Output is been later, Server TCP.py

→ The server codes is :-

```
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()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('sent contents' + sentence)
    file.close()
```

Output :-

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

} server side

```
Enter file name :- serverTCP.py
From Server:
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
connectionSocket.send(l.encode())
file.close()
connectionSocket.close()
```

File Edit Format Run Options Window Help

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

2
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

File Edit Shell Debug Options Window Help

```
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec  6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/mdsur/Desktop/client.py ==========

Enter file name: server.py

From Server:

from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()

>>>
```

File Edit Format Run Options Window Help

```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

File Edit Shell Debug Options Window Help

```
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec  6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
========== RESTART: C:/Users/mdsur/Desktop/server.py ==========
The server is ready to receive

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

Using UDP sockets, 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)
sentence = input("Enter file name")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom(2048)
print('\n reply from server')
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 ready")
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()
```

Output:-

The server is ready to receive

Send contents of server.py

Enter filename : Server UPP.py

Reply from Server:

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('127.0.0.1', serverPort))

serverSocket.sendto(bytes(1), "utf-8"), clientAddress)
file.close()
```

```
C:\Users\mdsur\Desktop\UDP>python -u serverUDP.py
The server is ready to receive

Sent contents of  serverUDP.py
```

```
C:\Users\mdsur\Desktop\UDP>python -u clientUDP.py

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()
```