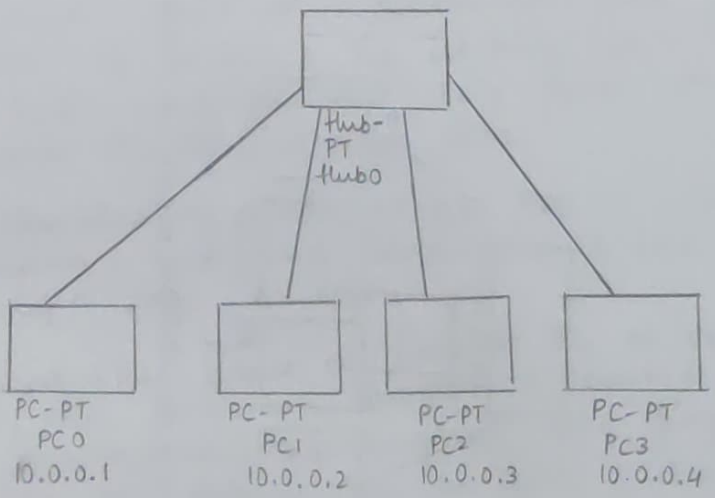
10/11/22

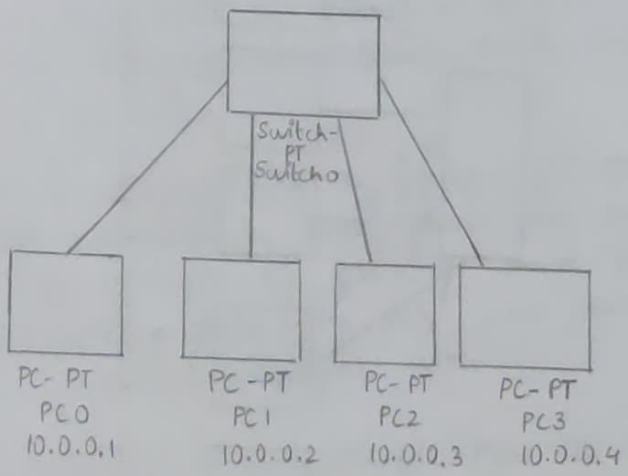
Aim: Creating a topology and stimulate sending a simple PDU from source to destination using hub and switch as connecting devices.

Topology:

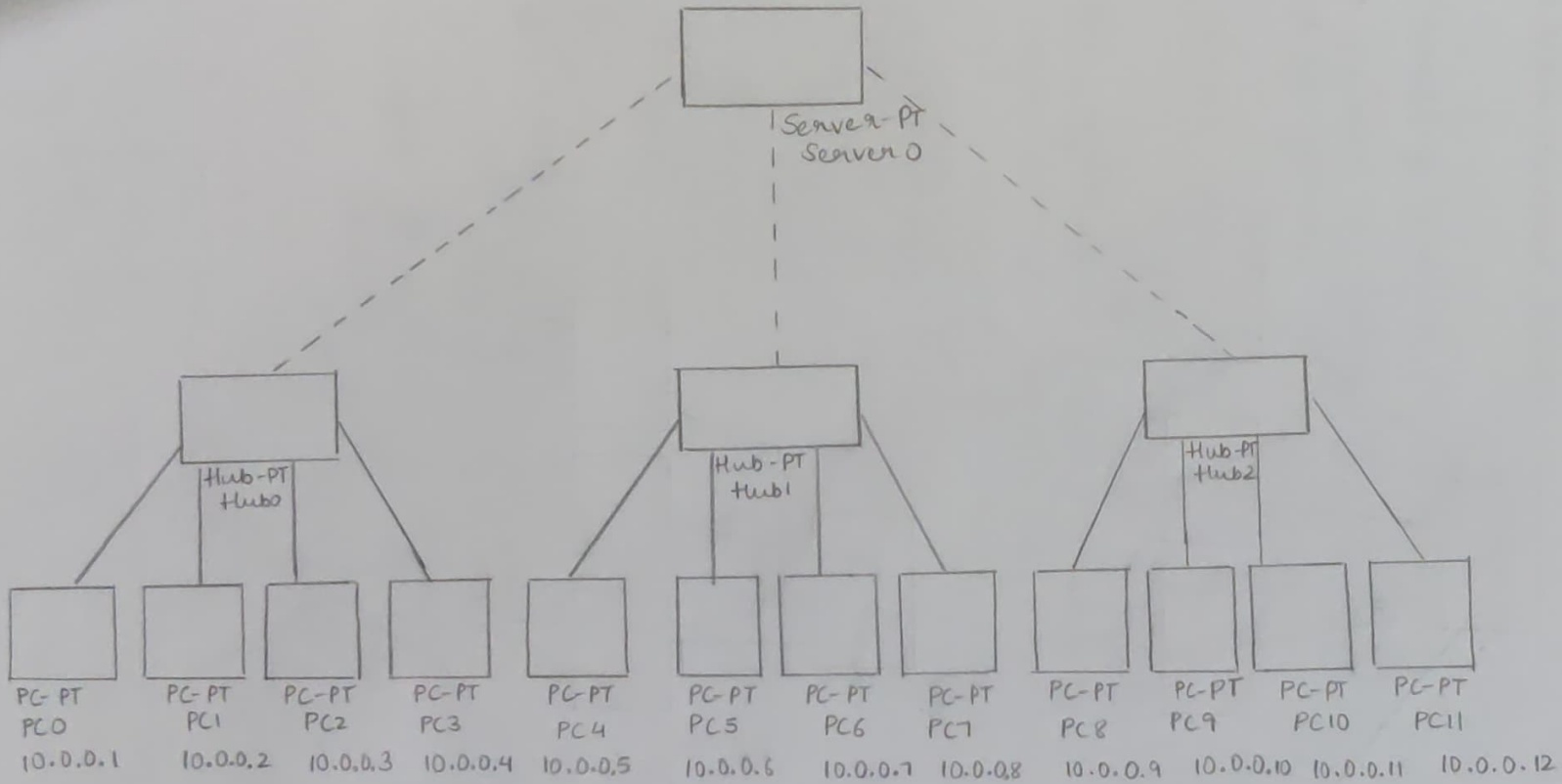
Hub:



Switch:



Hybrid:


Server-PT
Server 0
Hub-PT
Hub0
Hub-PT
Hub1
Hub-PT
Hub2
PC-PT
PC0
10.0.0.1
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
10.0.0.8
PC-PT
PC8
10.0.0.9
PC-PT
PC9
10.0.0.10
PC-PT
PC10
10.0.0.11
PC-PT
PC11
10.0.0.12

Procedure:

Hub

→ 7 Generic PC's and a generic hub are inserted into the logical workspace and the nodes for PC's are placed.

→ Each PC is configured by a specific IP address and the IP addresses are given from 10.0.0.1 to 10.0.0.7 by clicking on a specific PC. The hub and all the PC's are connected by copper straight through wire

→ In simulation mode, simple PDU is established between two end devices and the packet transfer can be seen.

→ In real-time mode a certain PC is selected and the command prompt is opened and the ping command can be given like ping IP address and the output can be seen.

Switch

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

→ Each PC is configured by a specific IP address and the IP addresses are given for the PC's from 10.0.0.1 to 10.0.0.4 by clicking on a specific PC. The Switch and all the PC's are connected by using copper straight wire

→ In simulation mode, simple PDU is established two end devices and auto capture is clicked. The packet transfer can be seen between the Switch and the PC's

→ In real-time mode, a PC is selected and command prompt is opened from the desktop option and the ping command can be given or sent to any other end devices and the output can be seen

Hybrid:

→ 12 generic PC's are inserted on the logical work space where 4 PC's are connected to one hub where there are 3 hubs each with 4PC's and the three hubs are connected to a single switch

→ All the nodes for 12 PC's are placed and the PC's are connected to their respective hub by copper-straight wire and the three hubs are connected to their switch by copper cross-over wire. The IP address are specified for all the 12 PC's from 10.0.0.1 to 10.0.0.12

→ In simulation mode simple PDU is established between two end devices and packet transfer can be seen from the source to the destination

→ In real-time mode, a PC can be selected and the command prompt can be opened for pinging another PC from the present PC and the output can be seen on the command prompt screen.

Observations:

Hub:

Learning outcome:

1) The hub broadcasts the message recieved by the source to all the other end devices but the message is read only by the specified destination and the destination responds back by sending a packet.

2) If the ports are not enough, the hub has to be clicked, the power has to be switched off and extra ports can be kept in the blank spaces and the power is switched on.

Result:

ping 10.0.0.7

pinging 10.0.0.7 with 32 bytes of data:

Reply from 10.0.0.7 : bytes = 32 time = 0 ms TTL = 0.8

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

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

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

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

Switch

Learning outcome:

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

Result:
ping 10.0.0.3
pinging 10.0.0.3 with 32 bytes of data
Reply time 10.0.0.3 : bytes = 32 time = 0 ms TTL = 128
Reply from 10.0.0.3 : bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3 : bytes = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.3:
Packets Sent = 4, Recieved = 4, Lost = 0 (0% Loss)
Approximate round trip times in millisecond
Minimum = 0ms, Maximum = 0ms, Average = 0ms

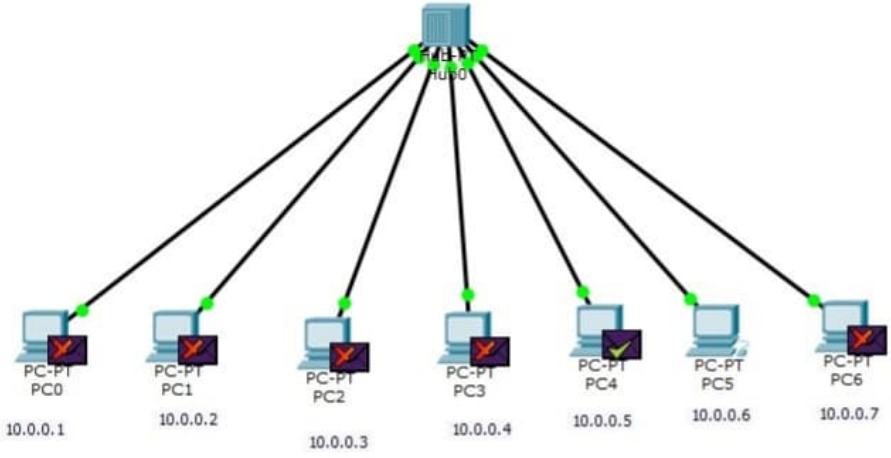
Hybrid

Learning outcomes:

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

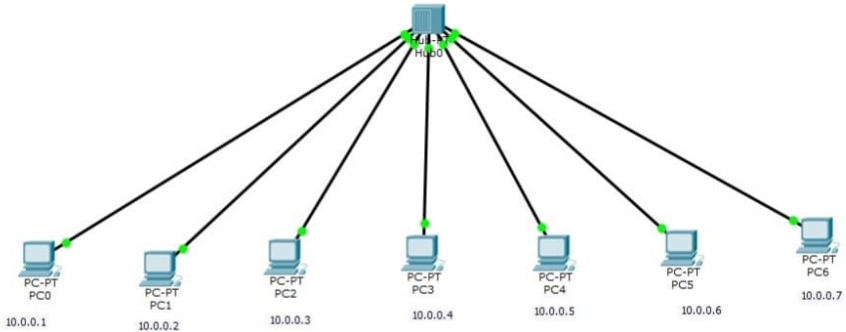
Result:
ping 10.0.0.11
Pinging 10.0.0.11 with 32 bytes of data
Reply from 10.0.0.11 : bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.11 : bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.11 : bytes = 32 time = 1 ms TTL = 128
Reply from 10.0.0.11 : bytes = 32 time = 0ms TTL = 128

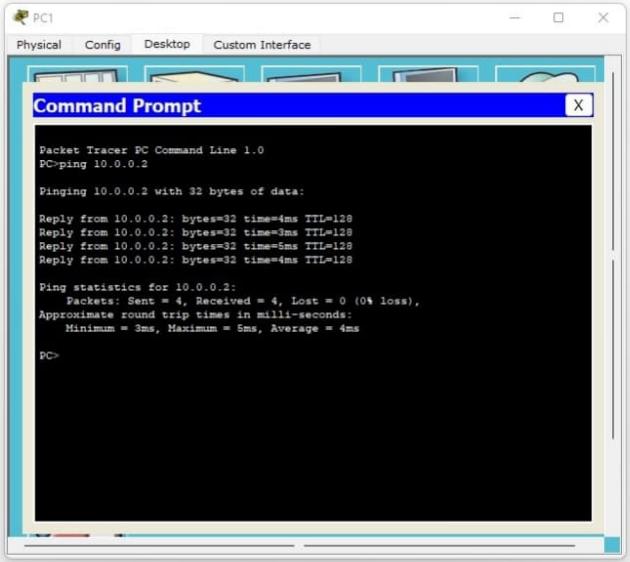
Ping statistics for 10.0.0.11:
Packets : Sent=4, Recieved =4, Last = 0 (0% Loss),
Approximate round trip times in milli seconds
Minimum=0ms, Maximum= 1ms, Average = 0ms



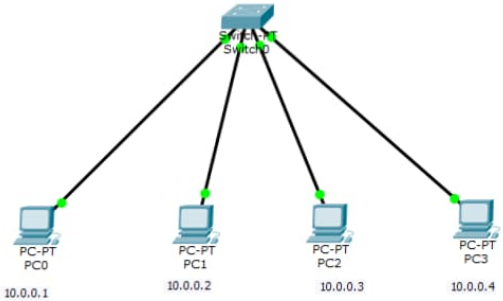
1BM20CS071
Keerthana N P

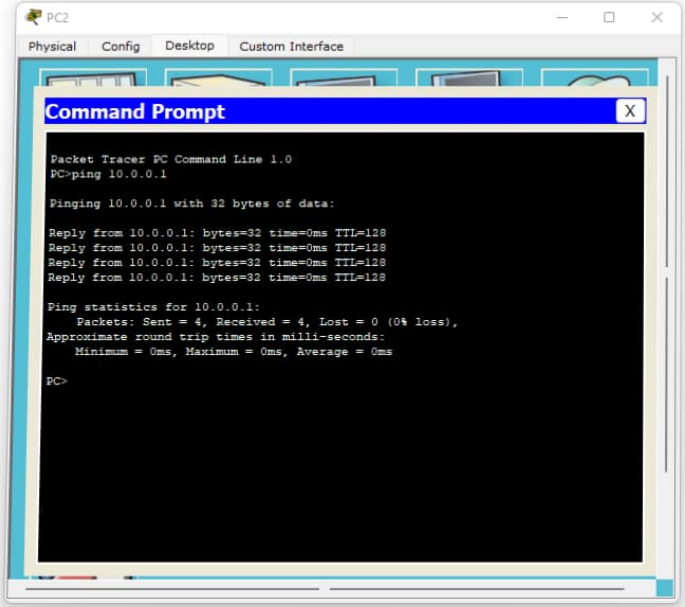
1BM20CS071
Keerthana N P



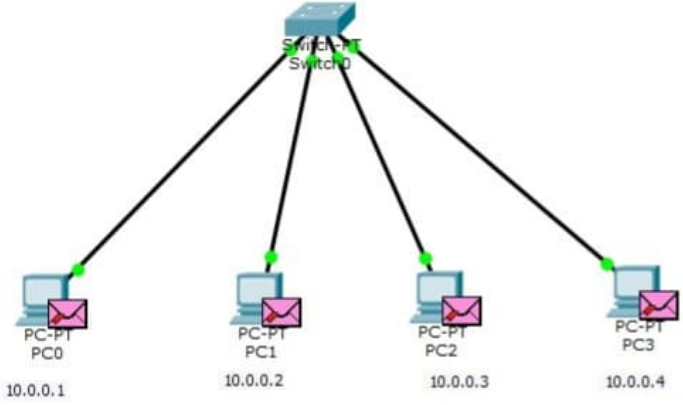


1BM20CS071
Keerthana N P

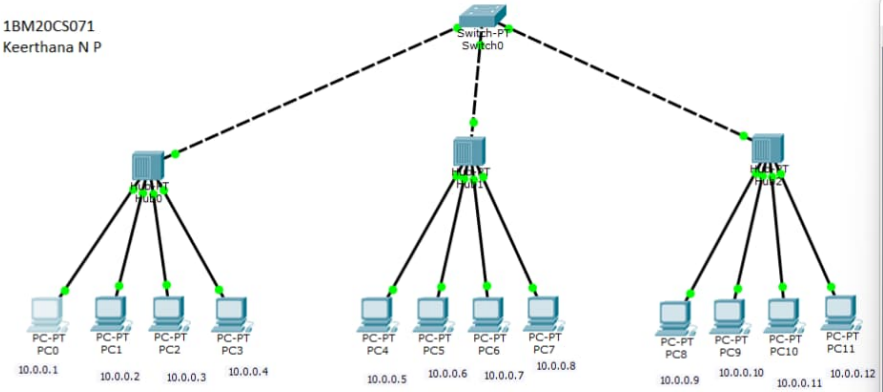


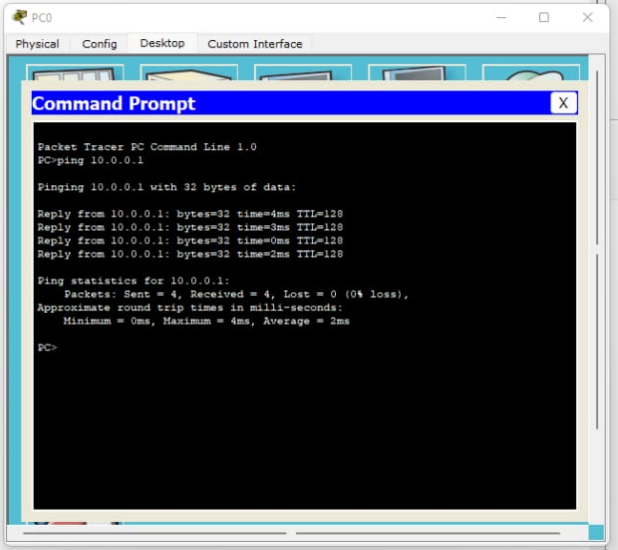


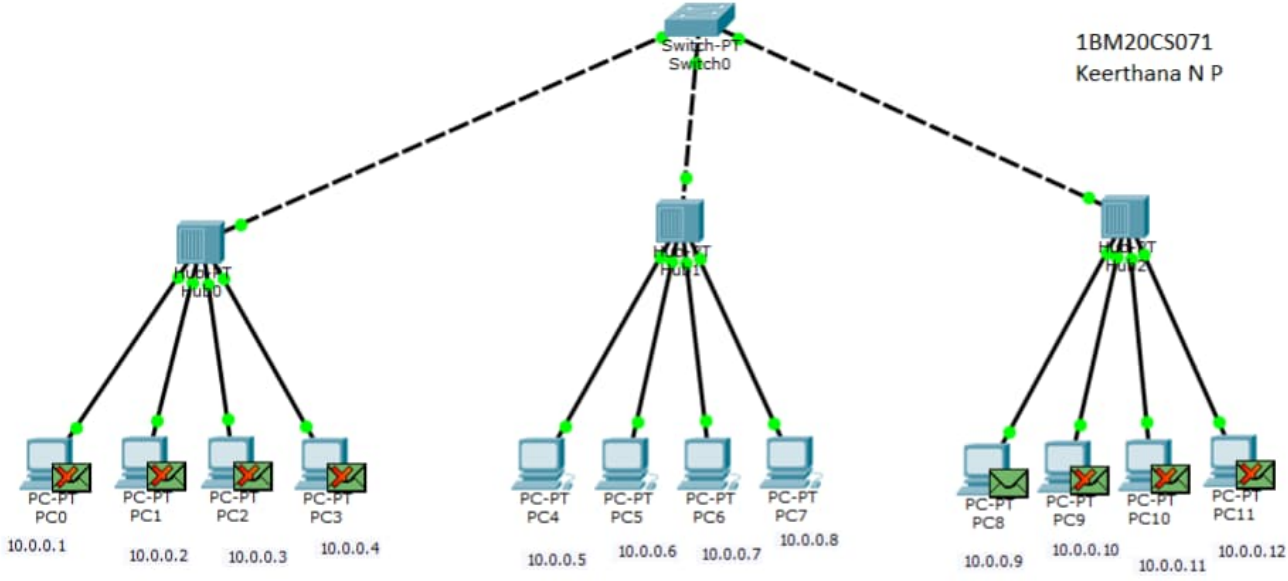
1BM20CS071
Keerthana N P



1BM20CS071
Keerthana N P





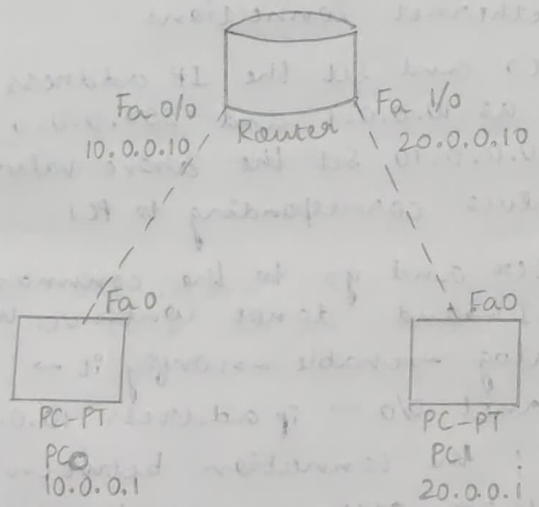

1BM20CS071
Keerthana N P
Switch-PT
Switch0
Hub-PT
Hub0
Hub-PT
Hub1
Hub-PT
Hub2
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3
PC-PT
PC3
10.0.0.4
PC-PT
PC4
10.0.0.5
PC-PT
PC5
10.0.0.6
PC-PT
PC6
10.0.0.7
PC-PT
PC7
10.0.0.8
PC-PT
PC8
10.0.0.9
PC-PT
PC9
10.0.0.10
PC-PT
PC10
10.0.0.11
PC-PT
PC11
10.0.0.12

17/11/22 LAB-2

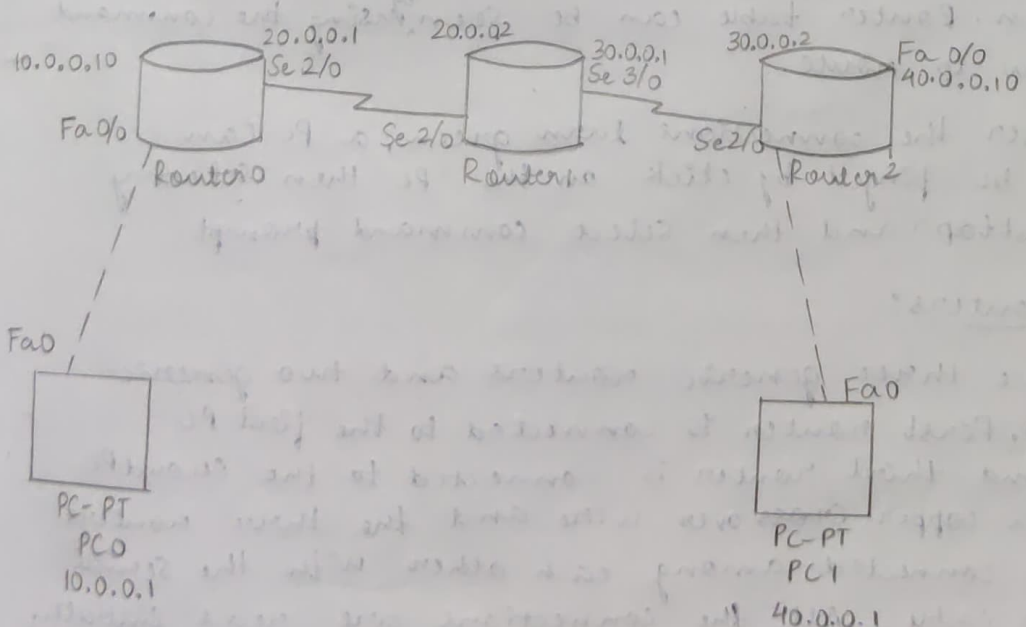
AIM: Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.

Topology:

One-Router:



Three-router:



Procedure:

One - Router

→ Place two generic PC's and a generic router and the router is connected to each of the PC's with a copper cross wire. The connections will be read initially.

→ Place the nodes for each of the PC's and each of the fast ethernet connections

→ Click on the PC0 and set the IP address and the Subnet mask as 10.0.0.1 and 255.0.0.0 and set the gateway as 10.0.0.10. Set the above values for PC1 with the values corresponding to PC1

→ Click on the router and go to the command line interface (CLI) → and do not continue with configuration dialog → enable → config t → interface fastethernet 0/0 → ip address 10.0.0.10 255.0.0.0 → no shut ; the connection between router and PC0 turn green when the above process is repeated for the connection between router and PC1 that connection also turns green. Router table can be seen by using the command show ip route.

→ After the connections turn green, a PC can be pinged by click on the PC then selecting desktop and then select command prompt.

3 - routers:

→ Place three generic routers and two generic PC's. First router is connected to the first PC and third router is connected to the second PC by a copper cross over wire and the three routers are connected among each other with the serial DCE cable. All the connections are read initially

→ Place the nodes and the router and PC is connected through fast ethernet while the routers are connected through serial

→ Each of the PC is clicked and the IP address, Subnet mask and gateway is set for each of the PC with the corresponding values

→ Router 1 is clicked > CLI > "no" > enable > config t > interface fastethernet 0/0 > ip address 10.0.0.10 255.0.0.0> no shut → with these the first connection is established.
config t > interface serial 2/0 > ip address 20.0.0.10 255.0.0.0> no shut → second connection is established

→ Router 2 is clicked > CLI > "no" > enable > config t > interface serial 2/0 > ip address 10.0.0.2 255.0.0.0> no shut → with these first connection is established.
Config t> interface serial 3/0 > ip address 30.0.0.1 255.0.0.0 > no shut → with these second connections established.

→ After all the above steps all green lights are glown and when the PC1 is pinged from PC0 → we get the reply, Destination unreachable

→ When the router 20.0.0.2 is pinged by PC0 the reply, request timed out is seen. The above replies are seen because the routers are not trained for the non-connected LAN's

→ Router 1 is trained by using
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
Router 2 is trained 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
Router 3 is trained 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, the correct reply is seen when PC1 is pinged by PC0

Observations:

One-Router

When PC0 pings PC1 for the first time we get
ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:
Request timed out
Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127
Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127
Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127

Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 3, host = 1 (25% loss)
Approximate round trip times in ms
Minimum = 0 ms, Maximum = 4ms, Average = 1ms

But when PC0 pings PC1 again or if PC1 reverse pings PC0 we get the output when where all the 4 times reply is observed

3-Routers:

Before the routers are trained and PC1 is pinged by PC0 and Router2 we get

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

Pinging Statistics 40.0.0.1
Packets: Sent = 4 Received = 0 host = 4 (100% Loss)

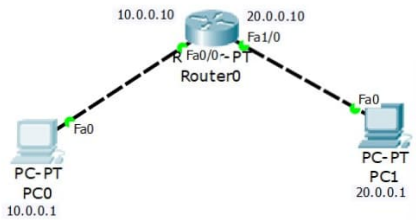
ping 20.0.0.2
Request timed out
Request timed out
Request timed out
Request timed out

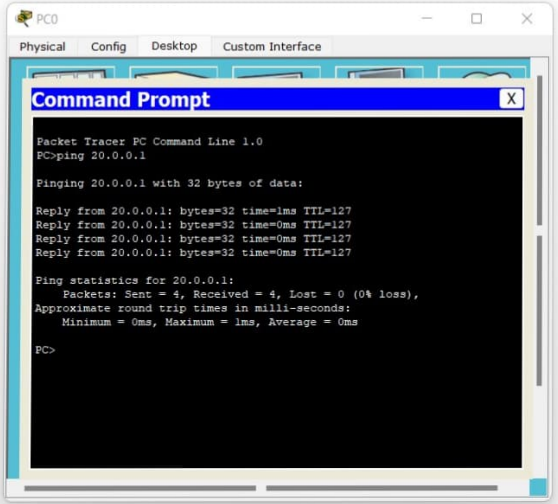
Pinging Statistics 20.0.0.2
Packets: Sent = 4 Received = 0 Lost = 4 (100% Loss)

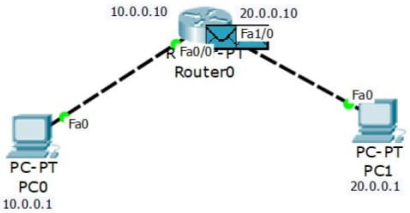
After the routers are trained:

ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data
Request timed out
Reply from 40.0.0.1: bytes = 32 time = 2 ms TTL = 125
Reply from 40.0.0.1: bytes = 32 time = 2 ms TTL = 125
Reply from 40.0.0.1: bytes = 32 time = 2ms TTL = 125

Pinging Statistics for 40.0.0.1
Packets: Sent = 4 Received = 9 Loss = 1 (25% Loss)


10.0.0.10
20.0.0.10
Fa1/0
Fa0/0
Router0
Fa0
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
20.0.0.1


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
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=1ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>



Simulation Panel

Event List

Vis.	Time(sec)	Last Devi	At Devi	Type	Info
	0.002	--	PC0	ICMP	
	0.003	PC0	Router0	ICMP	
	0.003	--	Router0	ARP	
	0.004	Router0	PC1	ARP	
	0.005	PC1	Router0	ARP	
	3.769	--	Router0	CDP	
	3.769	--	Router0	CDP	

Reset Simulation | Constant Delay | Capturing...

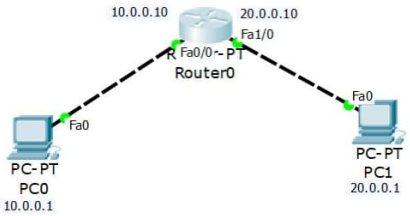
Play Controls

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



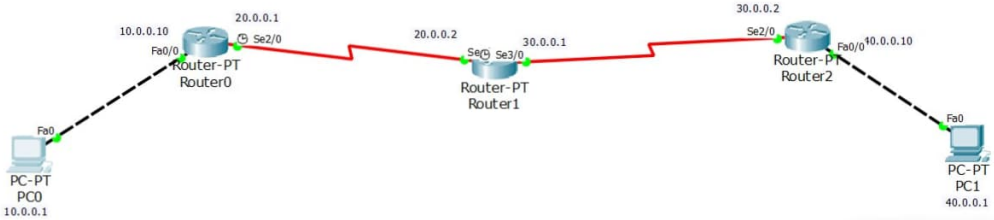
Router0

Physical Config CLI

IOS Command Line Interface

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shut

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

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

Router(config-if)#exit
Router(config)#
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed
state to up

Router(config-if)#exit
Router(config)#
```

Copy Paste



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

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 = 3ms, Average = 2ms

PC>
```

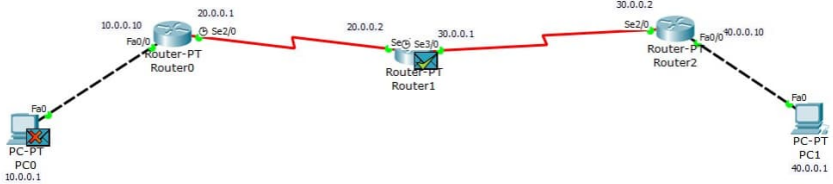
Time: 00:30:43 | Power Cycle Devices | Fast Forward Time

Connections

Automatically Choose Connection Type

Scenario 0 | New | Delete | Toggle PDU List Window

Fire | Last Status



Simulation Panel

Event List

Vis.	Time(sec)	Last Devi	At Devi	Type	Info
	0.722	--	Router1	CDP	
	0.723	Router1	Router0	CDP	
	0.723	Router1	Router2	CDP	
	18.504	--	Router0	CDP	
	18.504	--	Router0	CDP	
	18.505	Router0	PC0	CDP	
	18.505	Router0	Router1	CDP	

Reset Simulation | Constant Delay | Captured to: 18.505 s

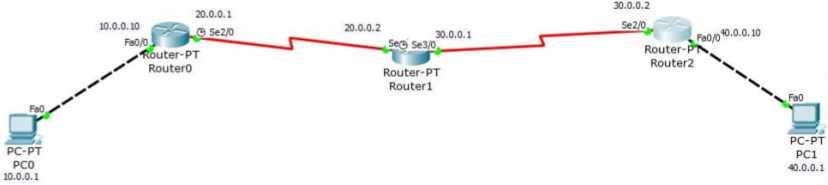
Play Controls

Back | Auto Capture / Play | Capture / Forward

Event List Filters - Visible Events

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

Edit Filters | Show All/None



Router2

Physical Config CLI

IOS Command Line Interface

```
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
```

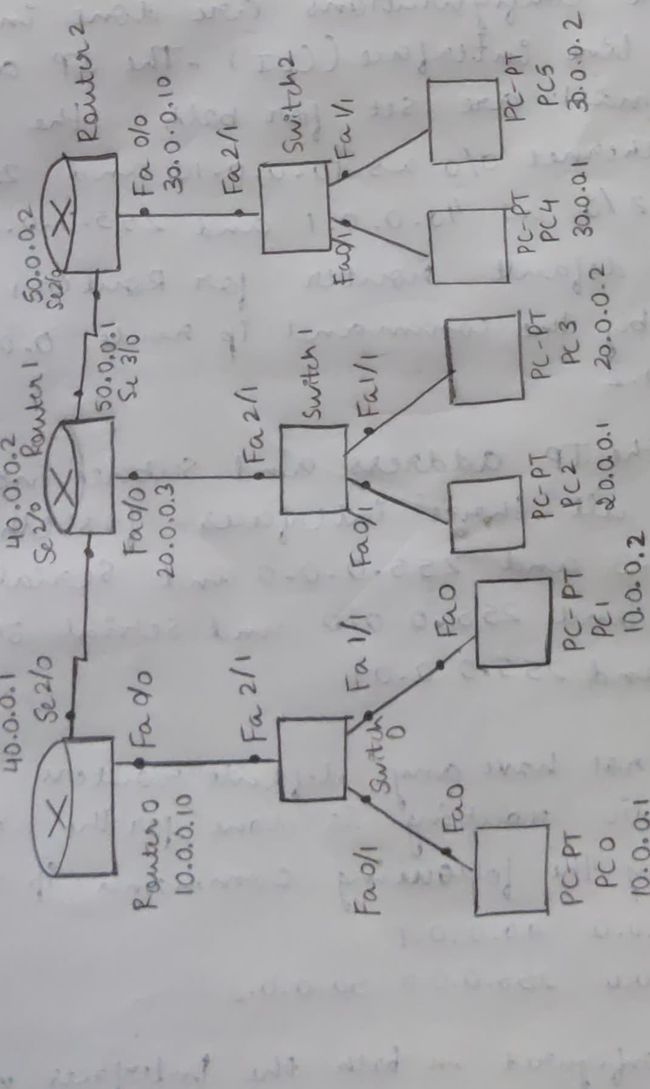
Copy Past

24/11/22

LAB-3

AIM: Configuring default route to the router

Topology:



Procedure:

→ Place 6 generic PC's, 3 switches and 3 routers and connect two PC's to each switch with copper straight through wire and each switch is connected to one router with a copper straight through wire and the three routers are connected among themselves by serial DCE cable and the nodes are placed for all the devices and networks

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

→ For router 1, the configurations are done in the command line interface (CLI). The IP address and Subnet mask are set for both the interfaces - fastethernet 0/0 as 10.0.0.10 and 255.0.0.0 and serial 2/0 as 40.0.0.1 and 255.0.0.0. Router 2 is the default router for Router 1 and this done by the command ip route 0.0.0.0 0.0.0.0 40.0.0.2

→ For Router 2, the IP address and Subnet mask are set for all three interfaces - fastethernet 0/0 as 20.0.0.3 and 255.0.0.0 and 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 2 does not have any default routers and static routing is done for the network 10 and 40 by the following command ip route 10.0.0.0 255.0.0.0 40.0.0.1
ip route 30.0.0.0 255.0.0.0 50.0.0.2

→ Router 3 is configured in both the interfaces with IP address and subnet mask as fastethernet 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 3 is Router 2 and this set by the command ip route 0.0.0.0 0.0.0.0 50.0.0.1

→ Ping command is executed from 10.0.0.1 to 20.0.0.1 and from 10.0.0.1 to 30.0.0.2

Observations:

Learning outcome:

→ One router cannot have two default routers

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

→ The default router for third router is the middle router for the same reason

→ The middle router does not have any default router because if one of the router is made default then there is a chance that the packets which are to be sent to the switch are sent to the router

Result:

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

ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data
Request timed out
Reply from 30.0.0.2: bytes = 32, time = 4ms, TTL = 125
Reply from 30.0.0.2: bytes = 32, time = 4ms, TTL = 125
Reply from 30.0.0.2: bytes = 32, time = 4ms, TTL = 125
```

Router0

Physical | Config | CLI

IOS Command Line Interface

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shut

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

%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 t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 40.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

Gateway of last resort is 40.0.0.2 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    40.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 40.0.0.2
Router#
```


Copy | Paste

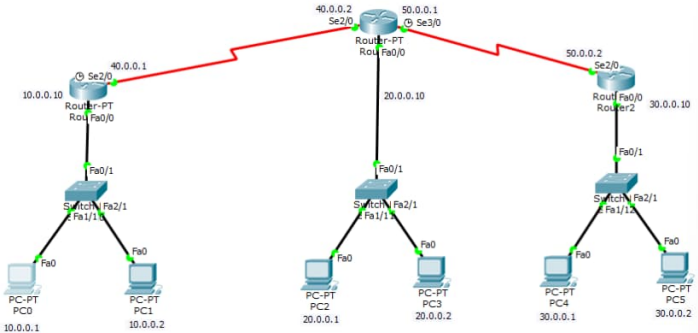
Router1

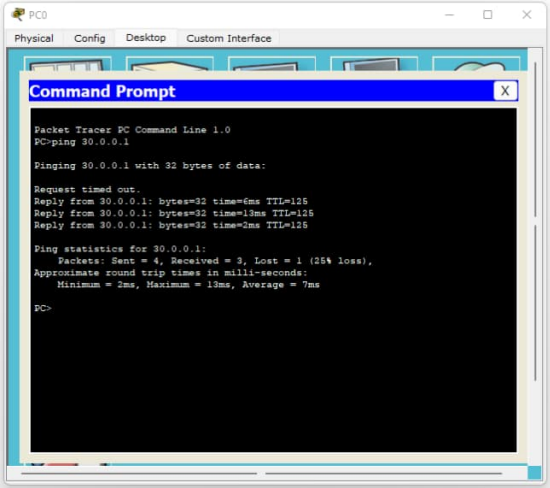
Physical | Config | CLI

IOS Command Line Interface

```
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface 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)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

%LINK-5-CHANGED: Interface Serial3/0, changed state to up

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

Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 30.0.0.0 255.0.0.0 50.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

Gateway of last resort is not set

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


Copy | Paste


40.0.0.2
Se2/0
50.0.0.1
Se3/0
Router-PT
Rou Fa0/0
40.0.0.1
Se2/0
10.0.0.10
Router-PT
Rou Fa0/0
20.0.0.10
50.0.0.2
Se2/0
Rout Fa0/0
Router2
30.0.0.10
Fa0/1
Switch Fa2/1
Fa1/1
Fa0/1
Switch Fa2/1
Fa1/1
Fa0/1
Switch Fa2/1
Fa1/1
Fa0
Fa0
Fa0
Fa0
Fa0
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
20.0.0.1
PC-PT
PC3
20.0.0.2
PC-PT
PC4
30.0.0.1
PC-PT
PC5
30.0.0.2

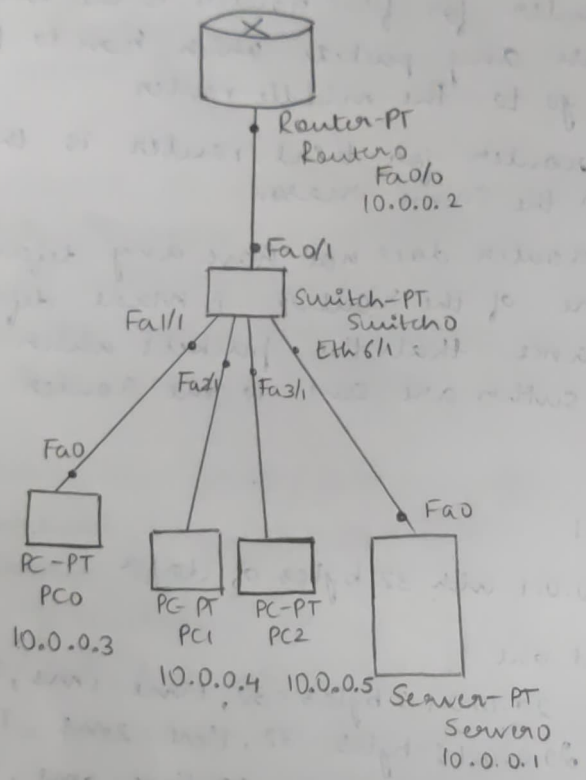

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 30.0.0.1: bytes=32 time=6ms TTL=125
Reply from 30.0.0.1: bytes=32 time=13ms TTL=125
Reply from 30.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 13ms, Average = 7ms
PC>

1/12/22

LAB - 4

AIM: Configuring DHCP within a LAN in a packet tracer

TOPOLOGY:



PROCEDURE:

→ Place a generic router, generic switch, 1 Server and 3 generic PC's into the workstation and connect them as: Router to switch and switch to all 3 PC's and 1 server

→ Open server config tab, set its IP address and subnet mask as 10.0.0.1 and 255.0.0.0 and from settings tab set the gateway as 10.0.0.2

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

→ Go to the services tab of server and select DHCP from the left panel. On the services, set default gateway as 10.0.0.2 and change DNS and TFTP to 10.0.0.1

In start IP address change it to the value from where you want to start the IP pool. Save the changes

→ Select the PC, go to the Desktop tab and select IP configuration. There change from static to DHCP. We can notice that all details like IP address, subnet mask, default gateway and DNS server is set by default.

→ Do the above step for all the PC's

OBSERVATIONS:

LEARNING OUTCOME:

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

RESULT:

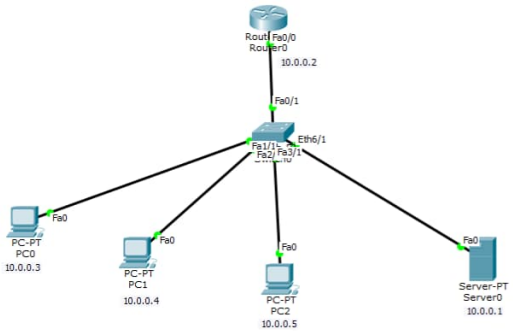
ping 10.0.0.5

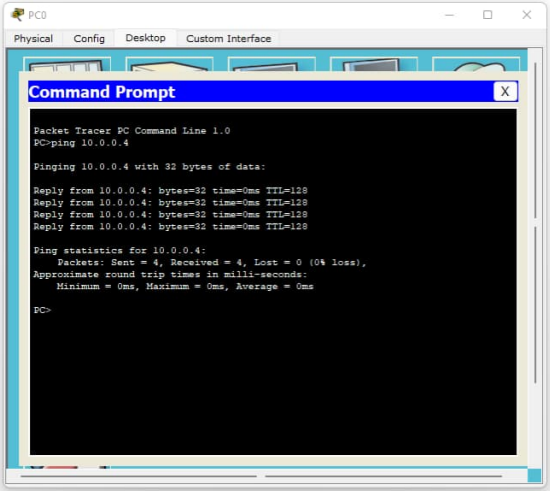
Pinging 10.0.0.5 with 32 bytes of data:

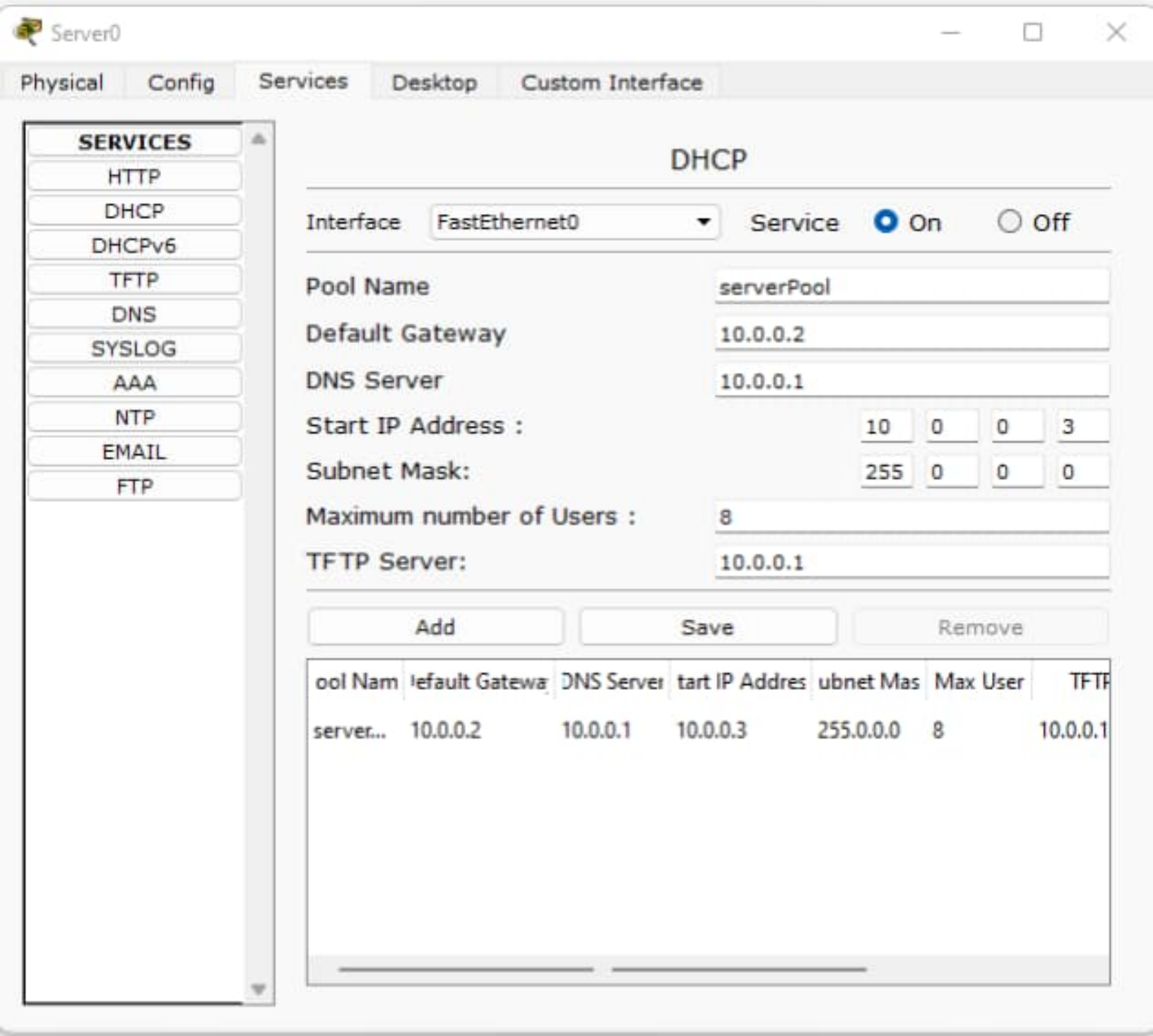
Reply from 10.0.0.5: bytes=32, time=0ms TTL=128
Reply from 10.0.0.5: bytes=32, time=0ms TTL=128
Reply from 10.0.0.5: bytes=32, time=0ms TTL=128
Reply from 10.0.0.5: bytes=32, time=0ms TTL=128

Ping statistics for 10.0.0.5:
Packets: Sent=4, Recieved=4, Loss=0 (0% Loss)

Approximate round trip times in ms:
Minimum =0ms, Maximum=4ms, Average=1ms


Router0
Fa0/0
10.0.0.2
Fa0/1
Eth6/1
Fa1/1
Fa2/1
Fa3/1
Fa0
PC-PT
PC0
10.0.0.3
Fa0
PC-PT
PC1
10.0.0.4
Fa0
PC-PT
PC2
10.0.0.5
Fa0
Server-PT
Server0
10.0.0.1


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms 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 = 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 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

Router0

Physical | Config | CLI

IOS Command Line Interface

```
63488K bytes of ATA CompactFlash (Read/Write)


         --- System Configuration Dialog ---

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


Press RETURN to get started!



Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shut

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

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

Router(config-if)#exit
Router(config)#
```

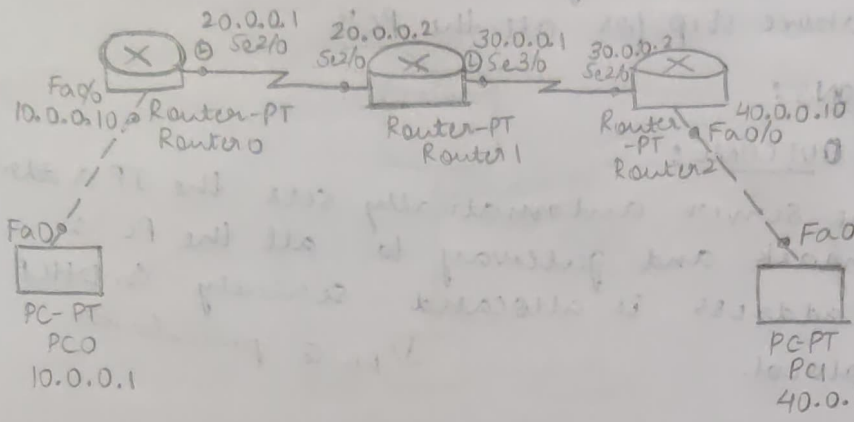
Copy | Paste

8/12/22

LAB-5

AIM: Configuring RIP routing protocol in router

TOPOLOGY:



PROCEDURE:

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

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

Procedure:

→ Place two PC's and three routers and connect the PC and router with copper cross over cable and the routers are connected with each other with serial DCE cable.

→ Click on the PC's and set the IP address, subnet mask and the gateway for each of the PC's.

PC0 → ip address: 10.0.0.1, subnet mask: 255.0.0.0
gateway: 10.0.0.10

PC1 → ip address: 40.0.0.1, subnet mask: 255.0.0.0
gateway: 40.0.0.10

→ Click on the first router → go to CLI → type the command
> enable ; #config t ; #interface fastethernet 0/0, #ip address 10.0.0.10 255.0.0.0 ; #no shut ; #exit ; #interface serial 2/0 ; #ip address 20.0.0.1 255.0.0.0 ;

#encapsulation PPP ; #clock rate 64000 ; #no shut ;

→ Click on the second router → go to CLI → type the commands > enable ; #config t ; #interface serial 2/0; #ip address 20.0.0.2 255.0.0.0; #encapsulation PPP; #no shut ; #exit ;
#interface serial3/0 ; #ip address 30.0.0.1 255.0.0.0 ; #encapsulation PPP ; #clock rate 64000 ; #no shut ;

→ Click on the third router → go to CLI → type the commands > enable ; #config t ; #interface serial 2/0; #ip address 30.0.0.2 255.0.0.0 ; #encapsulation PPP; #no shut ; #exit ;

#interface fastethernet 0/0 ; #ip address 40.0.0.10 255.0.0.0 ; #no shut

→ Now all the basic configuration are set for all the PC's and routers. All the lights are turned green

→ Now again click on first router and go to cli and type the following commands → #router rip #network 10.0.0.0
#network 20.0.0.0 ; #exit

→ Click on second router → go to CLI and execute
#router rip
network 20.0.0.0
#network 30.0.0.0 ; #exit

→ Click on third router → go to CLI and execute
router rip
network 30.0.0.0
network 40.0.0.0

→ Ping the PC 40.0.0.1 from 10.0.0.1

Observations:

Learning outcome:

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

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

Result:

ping 40.0.0.1

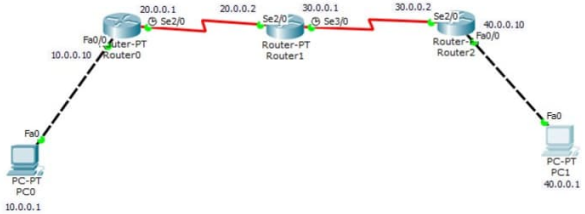
Pinging 40.0.0.1 with 32 bytes of data:

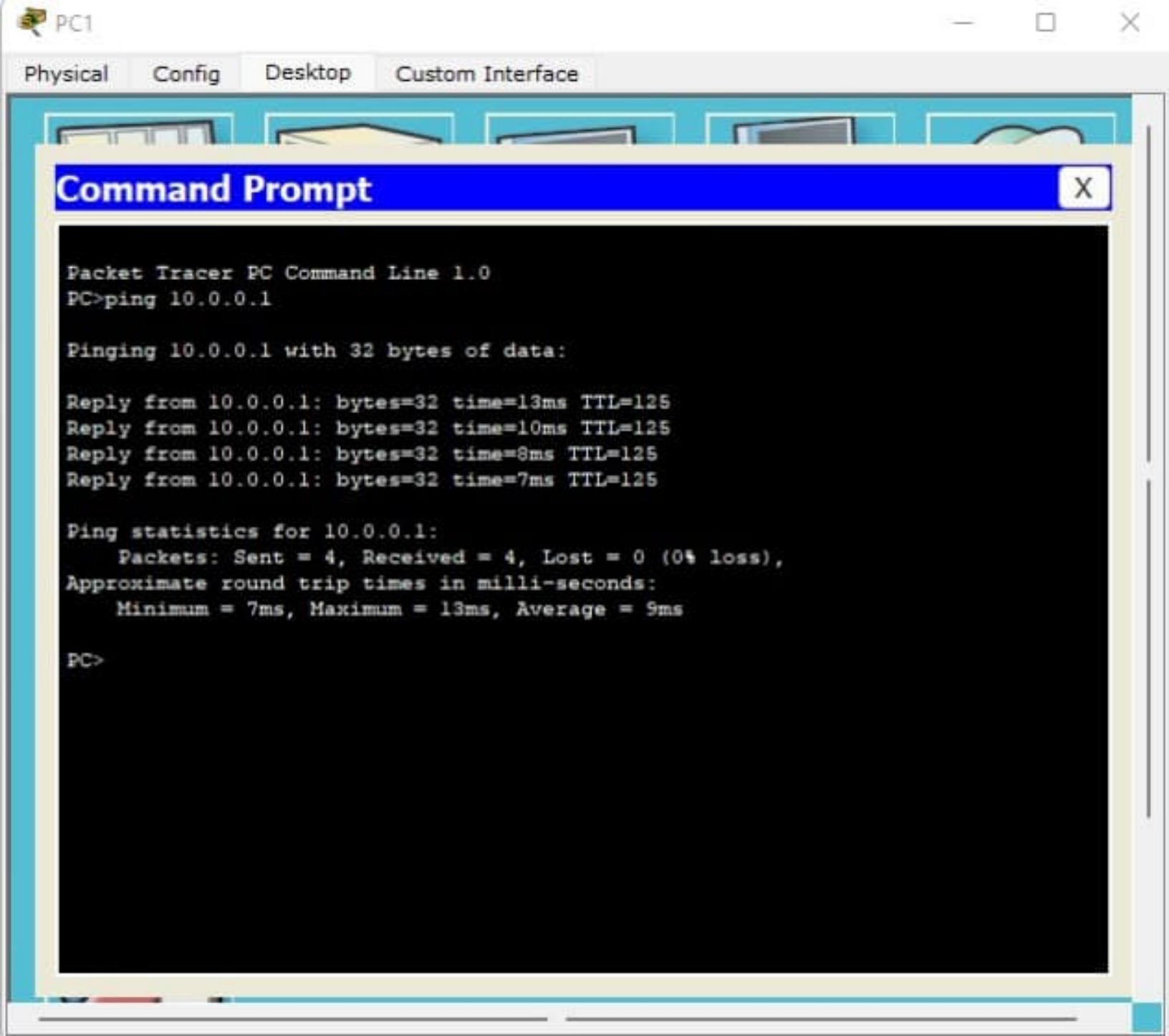
Request timed out

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

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

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


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


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=13ms TTL=125
Reply from 10.0.0.1: bytes=32 time=10ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 13ms, Average = 9ms
PC>

Physical Config CLI

```
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)


         --- System Configuration Dialog ---

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


Press RETURN to get started!



Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shut

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
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)#interface fastethernet 0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#encapsulation PPP
                  ^
% Invalid input detected at '^' marker.

Router(config-if)#exit
Router(config)#interface serial 2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation PPP
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
%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)#
```

Physical Config CLI

```
         --- System Configuration Dialog ---

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


Press RETURN to get started!



Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#no shut

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

Router(config-if)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shut

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

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

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

Router(config)#interface serial 2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation PPP
Router(config-if)#no
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
% Ambiguous command: "n"
Router(config)#interface serial 3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation PPP
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to down

Router(config-if)#clock rate 64000
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

Router(config)#router rip
Router(config-router)#network 20.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#
```

Physical Config CLI

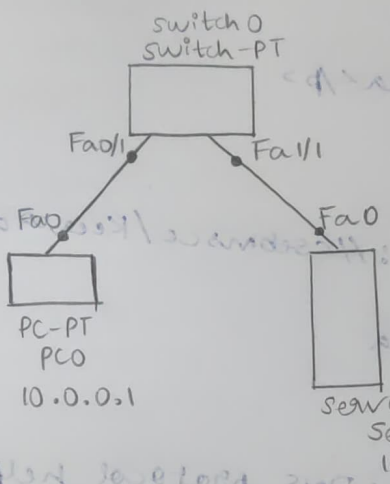
```
.
Processor board ID PT0123 (0123)
PT2005 processor: part number 0, mask 01
Bridging software.
X.25 software, Version 3.0.0.
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)


         --- System Configuration Dialog ---

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


Press RETURN to get started!



Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shut

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

Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shut

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

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

Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

Router(config)#interface serial 2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation PPP
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#exit
Router(config)#
```

15/12/22

LAB-6

AIM: Demonstration of Webserver and DNS using Packet Tracer.

Topology:



Procedure:

→ Set the IP Address and subnet mask of the PC and Server accordingly. From the PC invoke the web browser in the desktop tab and give the ip of the server. http://10.0.0.2 See that a home page gets displayed

→ Now go to services of services tab in the properties of the server, click on HTTP. The following window is seen. Click on the edit button of index.html and change text from CISCO to BMSCE. Save to overwrite. View the same as above from the PC to see the modified page.

Domain Name System:

→ To activate DNS, click on DNS service. Select on. Enter the name and IP. Click on Add button

→ As the name is fixed for the IP address, check the same by giving IP address in the web http://csebmsce browser window of the PC.

→ Make your own HTML page with a table displaying any details. Connect same with the hyperlink View the same from PC http://csebmsce/keerthana.html

Results:

The html file: Keerthana.html

```
<html>
<head>
Hello
</head>
<body>
<p> This is Keerthana</p>
</body>
</html>
```

Output: Url: http://csebmsce/Keerthana.html

Hello
This is Keerthana

Observations:

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

Server0

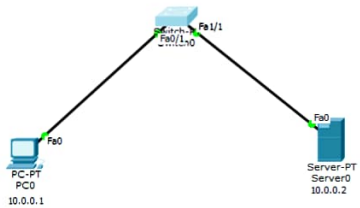
Physical Config Services Desktop Custom Interface

SERVICES

File Name: index.html

```
<html>
<center><font size='+2' color='blue'>BMSCE Packet
Tracer</font></center>
<hr>Welcome to Cisco Packet Tracer. Opening doors to new
opportunities. Mind Wide Open.
<p>Quick Links:
<br><a href='helloworld.html'>A small page</a>
<br><a href='copyrights.html'>Copyrights</a>
<br><a href='image.html'>Image page</a>
<br><a href='cscoptlogo177x111.jpg'>Image</a>
</html>
```

File Manager Save



PC0

Physical Config Desktop Custom Interface

Web Browser X

< > URL http://csebmsce Go Stop

BMSCE Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

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

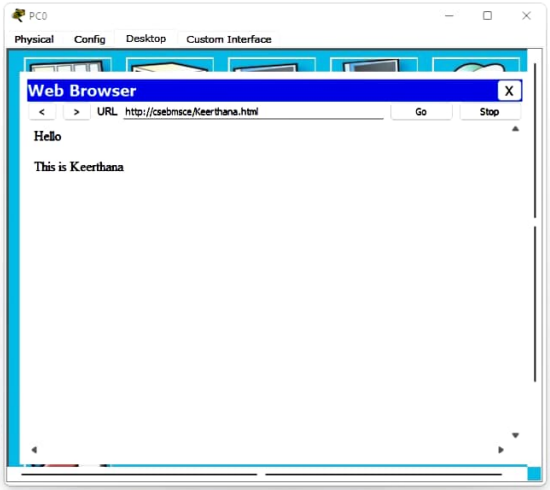
Name _____ Type A Record

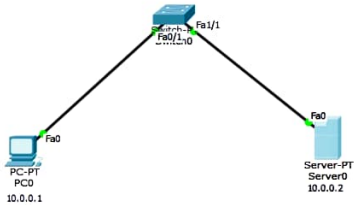
Address _____

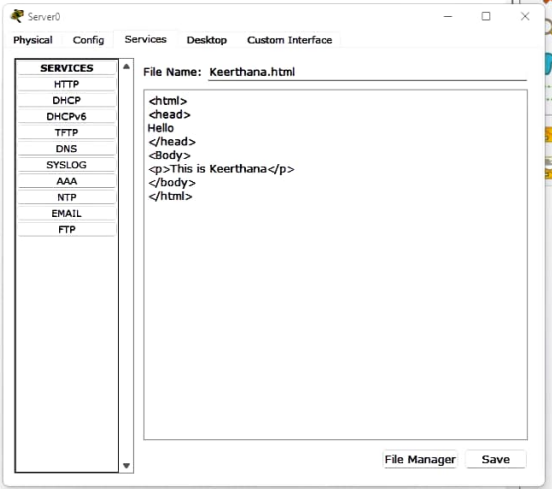
Add | Save | Remove

No.	Name	Type	Detail
0	csebmsce	A Record	10.0.0.2

DNS Cache


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL http://csebmsce/Keerthana.html
Go
Stop
Hello
This is Keerthana


Fa1/1
Fa0/1
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
File Name: Keerthana.html
<html>
<head>
Hello
</head>
<Body>
This is Keerthana
</body>
</html>
File Manager
Save

LAB-7

29/12/22

Write a program to Error detection using CRC-CCITT (16 bits)

```
#include<stdio.h>
#include<string.h>
#define N strlen(gen_poly)
char data[28];
char check_value[28];
char gen_poly[10];
int data_length, i, j;
void XOR(){
    for(j=1; j<N; j++)
    check_value[j] = ((check_value[j] == gen_poly[j])?
                                        '0':'1');
}

void receiver(){
    printf("Enter data received data");
    scanf("%s", data);
    printf("Data received: %s", data);
    crc();
    for(i=0; (i<N-1) && (check_value[i] != '1'); i++);
        if(i<N-1)
            printf("Error detected\n");
        else
            printf("NO error detected");

}

void crc()
{
    for(i=0; i<N; i++)
        check_value[i] = data[i];

    do{
        if(check_value[0] == '1')
            XOR();
        for(j=0; j<N; j++)
            check_value[j] = check_value[j+1];
        check_value[j] = data[i++];
    }while(i<= data_length+N-1);

}
```

```
int main()
{
    printf("Enter data to be transmitted ");
    scanf("%s",data);
    printf("Enter the generating polynomial ");
    scanf("%s", gen_poly);
    data_length = strlen(data);
    for(i=data_length; i<data_length+N-1; i++)
        data[i] = '0';

    crc();
    for(i=data_length; i<data_length+N-1; i++)
        data[i] = check_value[i-data_length];

    receiver();

    return 0;
}
```

Output:

```
Enter data to be transmitted : 1011010101
    Enter generating polynomial : 1011
    Data padded with n-1 zeros : 1011010101000
    CRC or check value is : 101
    Final data to be sent : 1011010101101
Enter the received data : 1011010101101
    Data received : 1011010101101
        No error detected.
```

29/12/22

```
Enter Codeword : 1011010101
Enter generating Polynomial : 1010
Data with zeros : 1011010101 0000
CRC value is 000
Final data to be sent is : 1011010101 0000
Enter data received : 1011010101 0000
    Error detected
```

```
Enter codeword: 1011010101

 Enter the Generating polynomial: 1010

 Data padded with n-1 zeros : 1011010101000
CRC or Check value is : 000
 Final data to be sent : 1011010101000
Enter the received data:
1011010101001
Data received: 1011010101001
Error detected
```

```
Enter codeword: 1011010101

 Enter the Generating polynomial: 1010

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

LAB-9

12/1/23

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

```
#include<stdio.h>
#include<stdlib.h>
int Bellman_Ford(int G[20][20], int V, int 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]+G[u][v]<distance[v])
                distance[v]=distance[u]+G[u][v];
                    parent[v]=u;
        }
    }
for (k=0;k<E;k++)
{
    u=edge[k][0], v=edge[k][1];
    if (distance[u]+G[u][v]<distance[v])
            flag=0;
}
if(flag)
for (i=0; i<v ; i++)
    printf("vertex %d → cost=%d, parent=%d\n",
                    i+1, distance[i], parent[i]+1);

    return flag;
}
int main()
{
  int V, edge[20][2], G[20][20], i, j, k=0;
    printf("Enter no of vertices");
```

```
    scanf("%d", &V);
printf("Enter graph in matrix form:\n");
    for (i=0; i<V; i++)
        for (j=0; j<V; j++)
        {
            scanf("%d", &G[i][j]);
            if (G[i][j] != 0)
                edge[k][0] = i,
                edge[k+1][1] = j;
        }
    if (Bellman_ford (G, V, k, edge))
        printf("\n No negative weight
                cycle \n");
    else
        printf("\n Negative weight cycle
                exists \n");
    return 0;
}
```

Output:

```
Enter no of vertices: 4
Enter graph in matrix form:
0 5 17 3
2 0 3 5
8 5 0 2
1 3 2 0
Enter source : 1
Vertex 1 → cost = 0 parent = 0
Vertex 2 → cost = 5 parent = 1
Vertex 3 → cost = 5 parent = 4
Vertex 4 → cost = 3 parent = 1
NO negative weight cycle.
```

12/1/23

```
BELLMAN FORD
Enter no. of vertices: 4
Enter graph in matrix form:
0 5 17 3
2 0 3 5
8 5 0 2
1 3 2 0
Enter source: 1
Vertex 1 -> cost = 0 parent = 0
Vertex 2 -> cost = 5 parent = 1
Vertex 3 -> cost = 5 parent = 4
Vertex 4 -> cost = 3 parent = 1

No negative weight cycle
```

LAB-10

12/1/23

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

```
#include<stdio.h>
#include<conio.h>
#define INFINITY 9999
#define MAX 10
void dijkstra(int G[MAX][MAX], int n, int startnode);
int main()
{
  int G[MAX][MAX], i, j, n, u;
  printf("Enter no of vertices");
  scanf("%d", &n);
  printf("\nEnter the adjacency matrix:\n");
  for(i=0; j<n; i++)
      for(j=0; j<n; j++)
          scanf("%d", &G[i][j]);
  printf("Enter the starting node");
  scanf("%d", &n);
  dijkstra(G, n, u);
  return 0;
}

void dijstra(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 the adjacency matrix:
   0   5   9999  9999
   2   0   4     9999
9999  9999  0     6
   4     7   5     0

Enter the starting node: 0
Distance of node 1 = 5
    Path = 1 <- 0
    Distance of node 2 = 9
    Path = 2 <- 1 <- 0
    Distance of node 3 = 15
    Path = 3 <- 2 <- 1 <- 0
```

N
12/1/2023

```
Enter no. of vertices:4

Enter the adjacency matrix:
0 5 9999 9999
2 0 4 9999
9999 9999 0 6
4 7 5 0

Enter the starting node:0

Distance of node1=5
Path=1<-0
Distance of node2=9
Path=2<-1<-0
Distance of node3=15
Path=3<-2<-1<-0
```

LAB-8 5/1/23

Write a program for congestion control using Leaky bucket algorithm.

```
#include<iostream>
using namespace std;
int main()
{
cout<< "Enter bucket size"<<endl;
 int bucket_size;
int filled=0;
 int outputrate ;
 int inputrate;
int choice;
 cin>> bucketsize ;
 cout<< "Enter output rate"<<endl;
 cin>> outputrate;
  do {
  cout<< "Enter packet size"<<endl;
   cin>> outputrate;
   do{
     cout<< "Enter packet size"<<endl;
       cin>> inputpacket ;
      if (inputpacket <= bucketsize){
           if (filled + input packet >bucketsize)
          {
            cout<< "packets too big for bucket"<<endl;
          }
          else{
          filled=filled+inputbucket;
          }
    }
  else
  {
  cout<<"packet is too big for bucket"<<endl;
   }
  if(filled<= outputrate)
   {
      filled=0 ;
   }
   else
   {
      filled = filled - outputrate; }
    cout<< "Amount of bucket filled"<< filled;
     cout<< "Do you want to enter another
              packet (9 for yes, 8 for no)"<<endl;
          cin>>choice; while(choice==9 ;}
```

Output:

```
Enter bucket size : 500
Enter output rate : 50
Enter packet size : 100
Packet too big for bucket
Do you want to enter packet (9 for yes, & No) : 9
Enter packet size : 200
    packet filled : 150
Do you want to enter packet (9 for yes, & No) : 9
    Enter packetsize : 250
    Bucket size : 350
    Do you want to enter packet : 9
    Enter packet size : 250
    Packet too big for bucket
    Amount of bucket filled 300
```

```
Enter bucket size
500
Enter  output rate
50
Enter packet size
700
Packets too big for bucket
Amount of bucket filled 0
Do you want to enter another packet(9 for yes, 5 for no)
9
Enter packet size
200
Amount of bucket filled 150
Do you want to enter another packet(9 for yes, 5 for no)
9
Enter packet size
250
Amount of bucket filled 350
Do you want to enter another packet(9 for yes, 5 for no)
9
Enter packet size
250
Packets too big for bucket
Amount of bucket filled 300
Do you want to enter another packet(9 for yes, 5 for no)
```

28/1/23

LAB-11

Socket (TCP/IP)

Using TCP/IP sockets, write a client server program to make client sending the file name and the server to send back the contents of the requested file if present

Client TCP.py

```
from socket import *
ServerName = '127.0.0.1'
ServerPort = 12000
ClientSocket = socket(AF_INET, SOCK_STREAM)
ClientSocket = connect((ServerName, ServerPort))
Sentence = input("\nEnter file name:")
Client Socket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("\n From Server:\n")
print(filecontents)
ClientSocket.close()
```

Server TCP.py

```
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
ServerSocket = socket(AF_INET, SOCK-STREAM)
ServerSocket.bind((ServerName, Server Port))
ServerSocket.listen(1)
while(1):
    print("The Server is ready to receive")
    connectionSocket, addr = ServerSocket.accept()
    Sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "y")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Send contents of" + Sentence)
    file.close()
    connectionSocket.close()
```

Output:

Server TCP

The server is ready to receive
Sent contents of Server TCP.py
The server is ready to recieve

Client TCP:

Enter file name : Server TCP.py

From Server:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket = bind((serverName, serverPort))
serverSocket.listen(1)

while(1):
    print("The server is ready to receive")
    connectionSocket, addr = serverSock.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Sent contents of" + sentence)
    file.close()
    connectionSocket.close()
```

>>>

29/1/23

LAB - 12

Sockets (UDP)

Using UDP sockets, write a client server program to make client sending the filename and the server to send back the contents of the request file if present.

Client UDP.py

```
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(fileContents.decode("utf-8"))
# for i in fileContents:
    #print(str(i), end='')
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to recive")
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 Send contents of ", end='')
    print(sentence)
    #for i in sentence
        #print(str(i), end='')
    file.close()
```

Output:

Server UDP:

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

Client UDP:

```
Enter the filename: serverUDP.py
Reply from Server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_PGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
while 1:
    print("The server is ready to receive").
    sentence. clientAddress = serverSocket.recv
                                from (2048)
    sentence = sentence.decode("utf-8")
    file = open(sentince, "r")
    l = file.read(2048)
    serverSocket.sendto(bytes(l, "utf-8"),
                                clientAddress)
    print("\n Send contents of " end = " ")
    print(sentence)
    file.close()
>>>
```