Dayananda Sagar College of Engineering

NAME: Hiya Namdeo
USN : 1DS18CS163
Course: Computer Networks Laboratory with mini-Project

Code : 18CS5DLCNL

SEM : 5th

Section: D

PART - 'A'

LAB PROGRAM #1.

a. Analyze VLAN communication using CPT

(a) Sketch and Simulate 3 VLANS

(b) Setup an extended VLAN using Trunk Interface

(c) Inter VLAN Routing

AIM: To setup 3 vlans & simulate them using CPT

DESIGN :

→ A VLAN is a group of devices or a subnetwork which can group together a collection of devices on separate physical local area network, configured to communicate as if they were attached to the same wire.

→ It is a concept in which we can divide the devices logically on layer

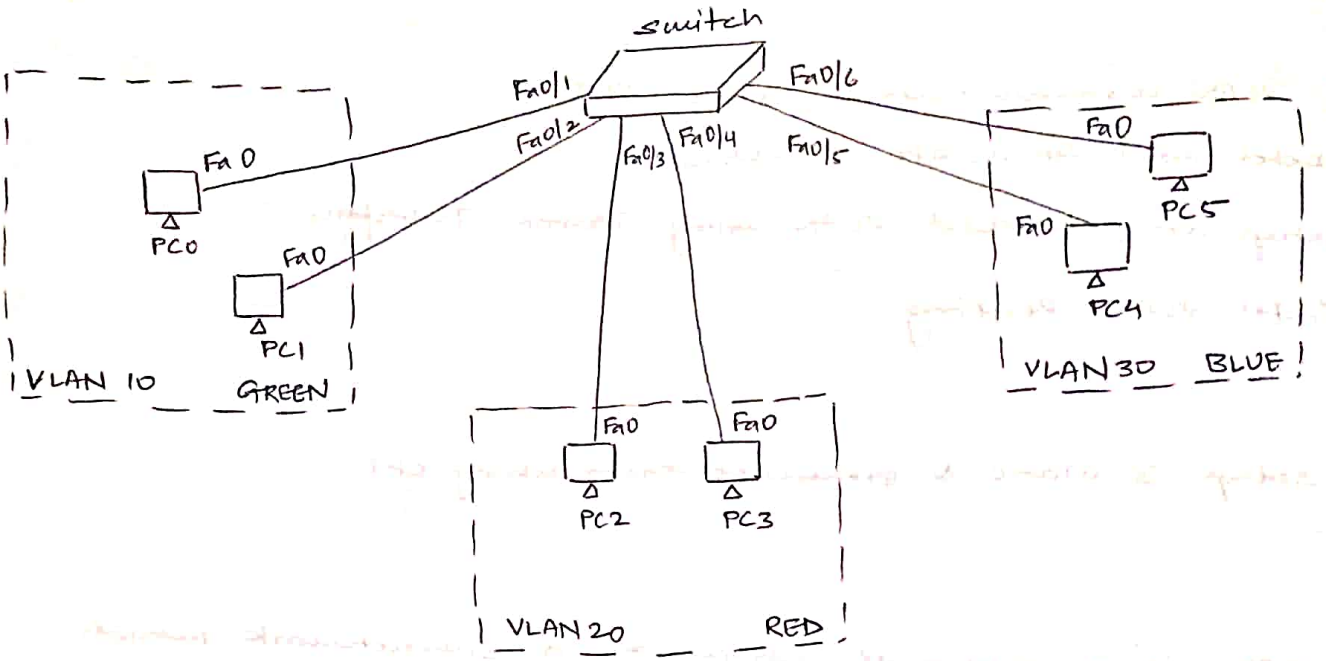
→ Increases no. of broadcast domains while decreasing their size.

→ Reduce security risks by reducing no. of hosts that receive copies of frames that the switches flood.

CONFIGURATION:

```
switch> enable
switch # config t
Switch (config) # vlan 10
switch (config-vlan) # name green
switch (config-vlan) # exit
Switch (config) # vlan 20
Switch (config-vlan) # name red
Switch (config-vlan) # exit
switch (config) # vlan 30
```

TOPOLOGY :

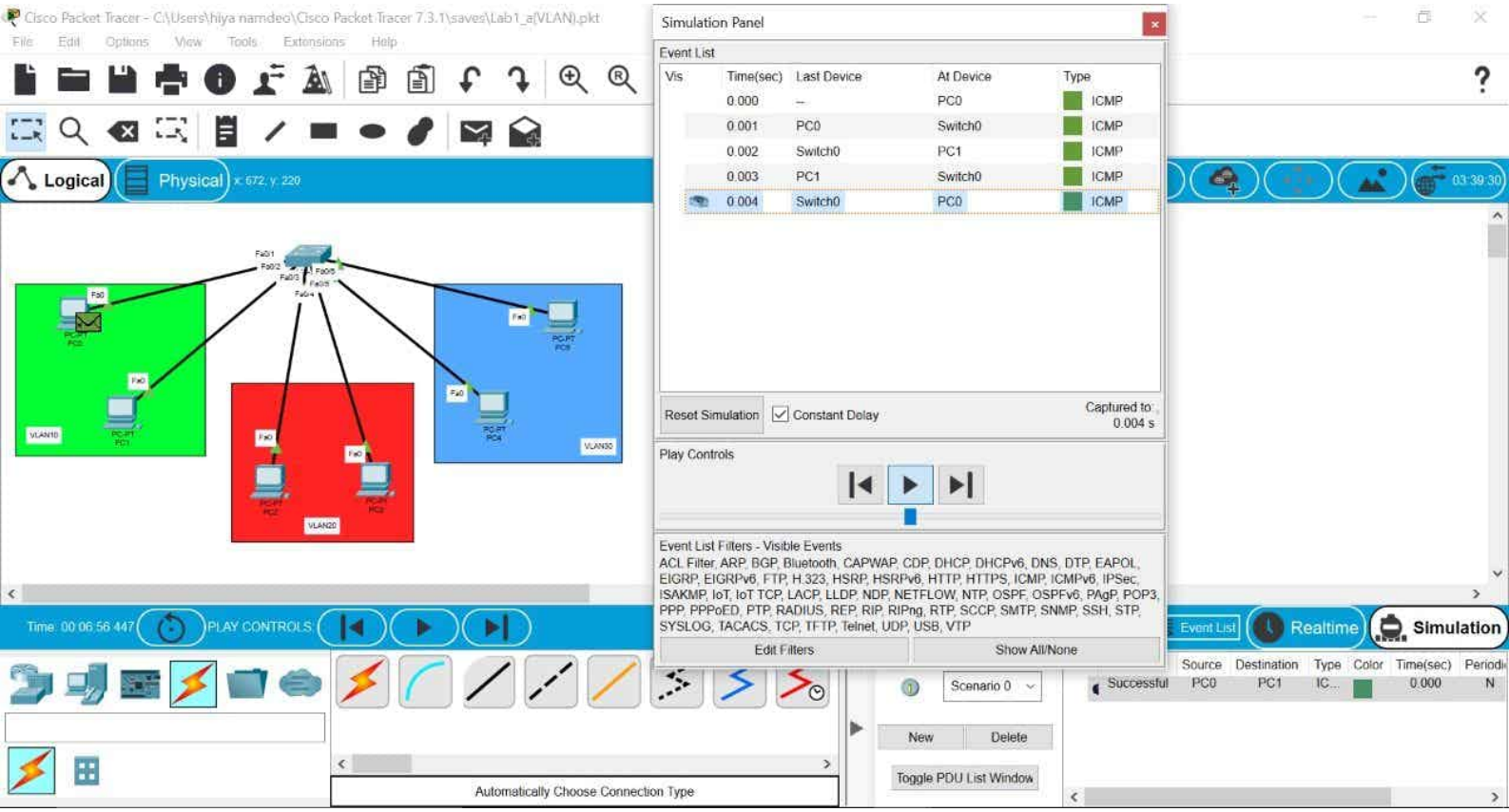


output :

switch (config-vlan) # name blue
switch (config-vlan) # exit
switch (config) # exit
switch #
switch # config t
switch (config) # interface range fo/1-2
switch (config-if-range) # switchport mode access
switch (config-if-range) # switchport access vlan 10
switch (config-if-range) # exit
switch (config) # interface range fo/3-4
switch (config-if-range) # switchport mode access
switch (config-if-range) # switchport access vlan 20
switch (config-if-range) # exit
switch (config) # interface range fo/5-6
switch (config-if-range) # switchport mode access
switch (config-if-range) # switchport access vlan 30
switch (config-if-range) # exit

OUTPUT: OBSERVATION:

The packets can be sent between any 2 hosts even though they are present in 2 different vlans.


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab1_a(VLAN).pkt
File Edit Options View Tools Extensions Help
Logical
Physical
x: 672, y: 229
03:39:30
Simulation Panel
Event List
Vis Time(sec) Last Device At Device Type
0.000 -- PC0 ICMP
0.001 PC0 Switch0 ICMP
0.002 Switch0 PC1 ICMP
0.003 PC1 Switch0 ICMP
0.004 Switch0 PC0 ICMP
Reset Simulation
Constant Delay
Captured to: 0.004 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
VLAN10
VLAN20
VLAN30
Time: 00:06:56.447
PLAY CONTROLS
Event List
Realtime
Simulation
Source Destination Type Color Time(sec) Periodi
Successful PC0 PC1 IC... 0.000 N
Scenario 0
New
Delete
Toggle PDU List Window
Automatically Choose Connection Type

LAB PROGRAM #3

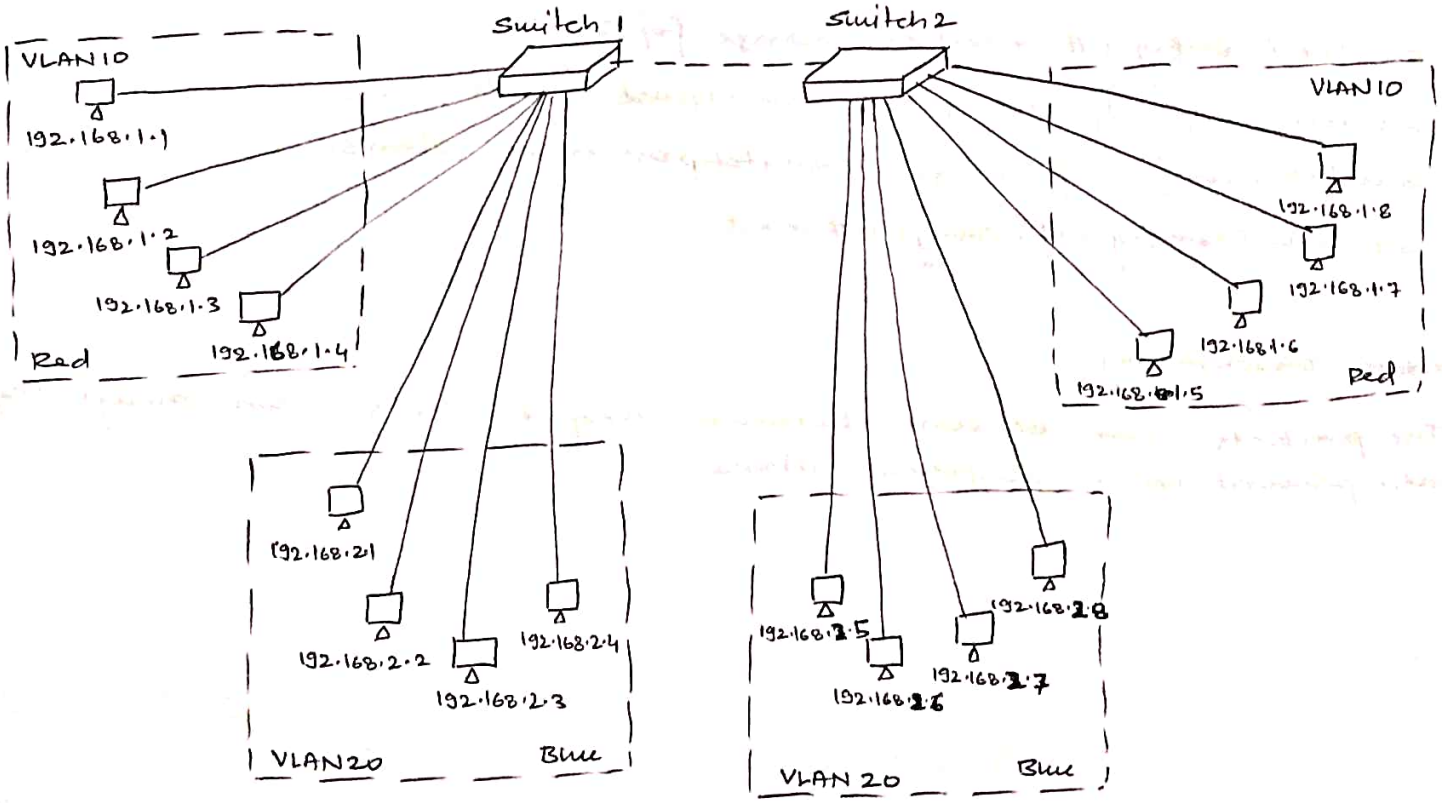
AIM: (b) To setup an extended VLAN and use trunk interface.

DESIGN:

Trunk ports can carry traffic of multiple VLANs thus allowing us to extend vlans across entire network

~~CONFIG~~

TOPOLOGY:



CONFIGURATION:

Trunk interface at S1

Switch> enable

Switch#config t

Switch(config)# interface fa0/15

switch (config-if) # switchport mode trunk

switch (config-if) # exit

~~switch~~

trunk interface at s2:

switch > enable

switch # config t

switch (config) # interface fa 0/15

switch (config-if) # switchport mode trunk

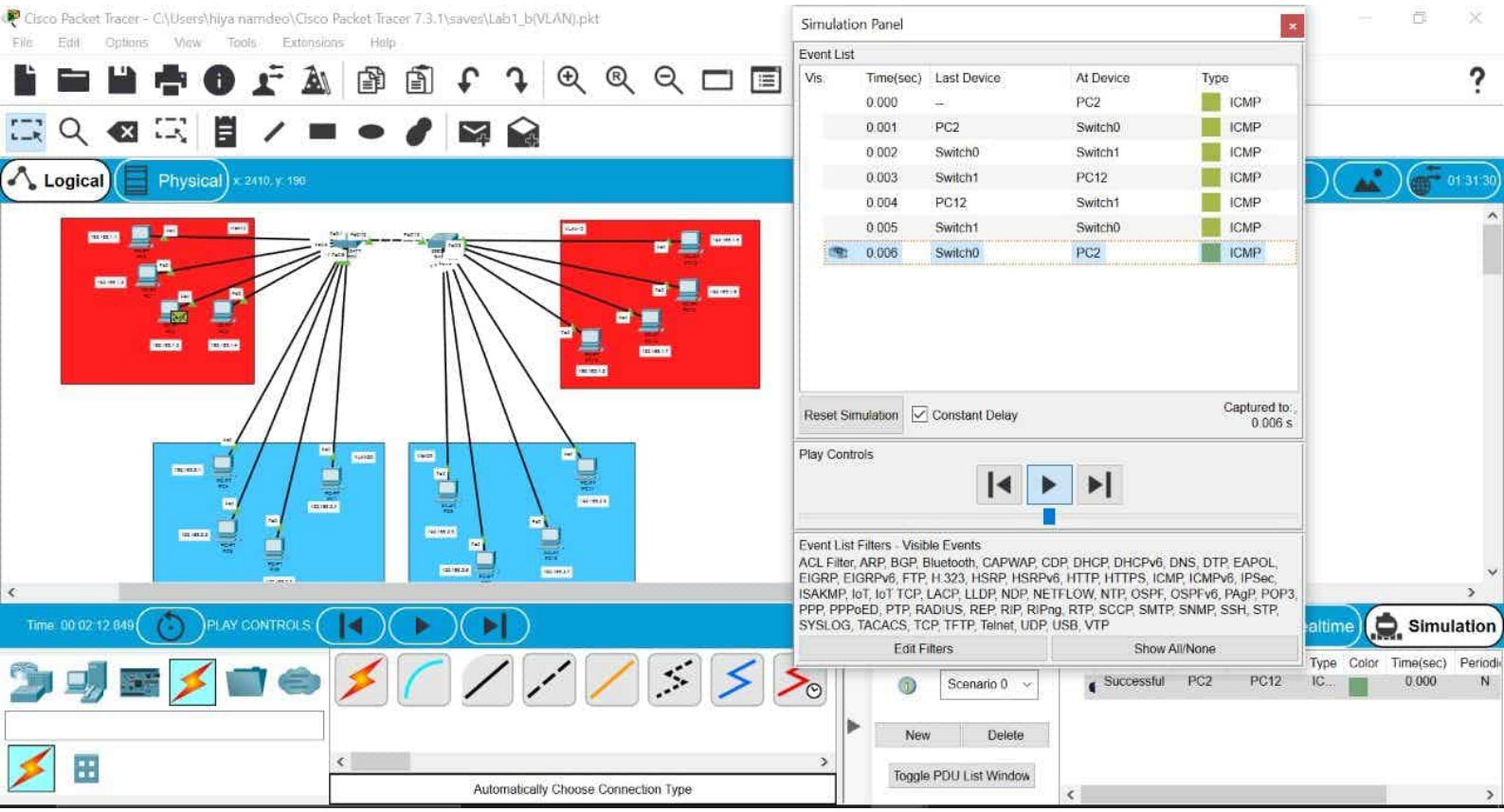
switch (config-if) # exit

~~OUTPUT~~

OBSERVATION & OUTPUT :

Packets can travel between hosts belonging to same vlan ~~in different~~ connected to different switch. using trunk interface.

~~OUTPUTS~~:


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab1_b(VLAN).pkt
File Edit Options View Tools Extensions Help
Logical
Physical
x: 2410, y: 190
Simulation Panel
Event List
Vis Time(sec) Last Device At Device Type
0.000 -- PC2 ICMP
0.001 PC2 Switch0 ICMP
0.002 Switch0 Switch1 ICMP
0.003 Switch1 PC12 ICMP
0.004 PC12 Switch1 ICMP
0.005 Switch1 Switch0 ICMP
0.006 Switch0 PC2 ICMP
Reset Simulation
Constant Delay
Captured to: 0.006 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:02:12.649
PLAY CONTROLS
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Successful PC2 PC12 IC... 0.000
Automatically Choose Connection Type

LAB PROGRAM #1

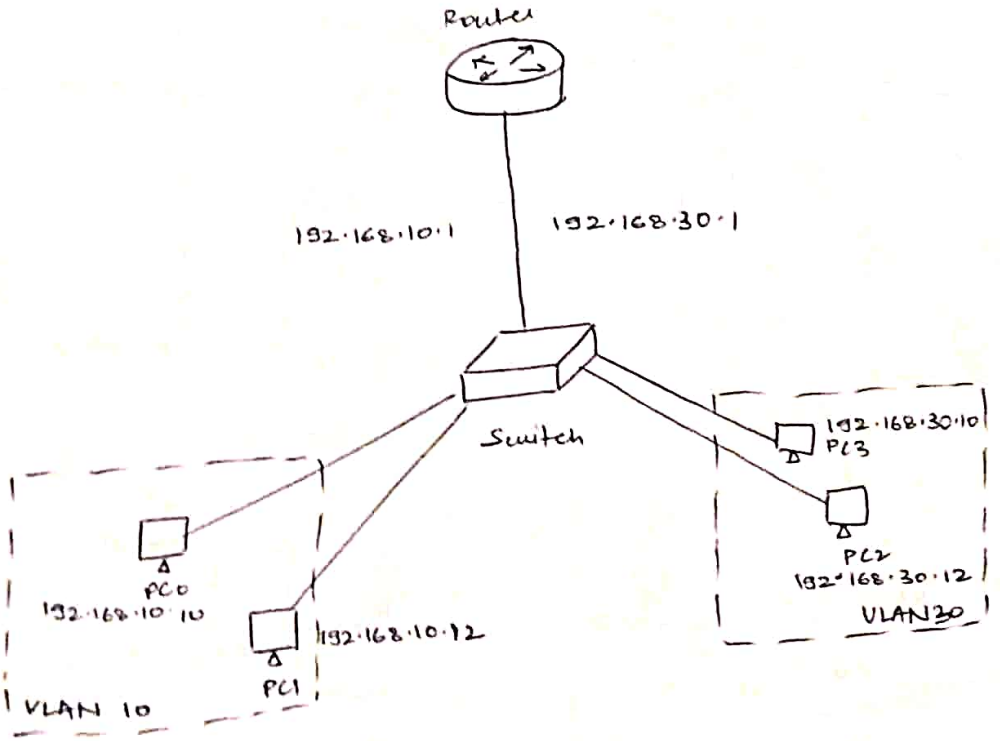
(C) AIM: To implement Intervlan Routing.

DESIGN:

Inter-vlan routing can be defined as a way to forward traffic between different VLAN by implementing a router in the network.

It can be accomplished by: Traditional inter-vlan routing or "Router-on-a-stick"

TOPOLOGY:



CONFIGURATION:

switch (config) # vlan 10

switch (config-vlan) # vlan 30

switch (config-vlan) # interface fa0/5

switch (config-if) # switchport mode trunk

switch (config-if) # exit

switch (config)# exit

switch #

configuration in router

router (config)# interface g0/0.10

router (config-subif)# encapsulation dot1q 10

router (config-subif)# ip address 192.168.10.1 255.255.255.0

router (config-subif)# interface g0/0.30

router (config-subif)# encapsulation dot1q 30

router (config-subif)# ip address 192.168.30.1 255.255.255.0

router (config-subif)# exit

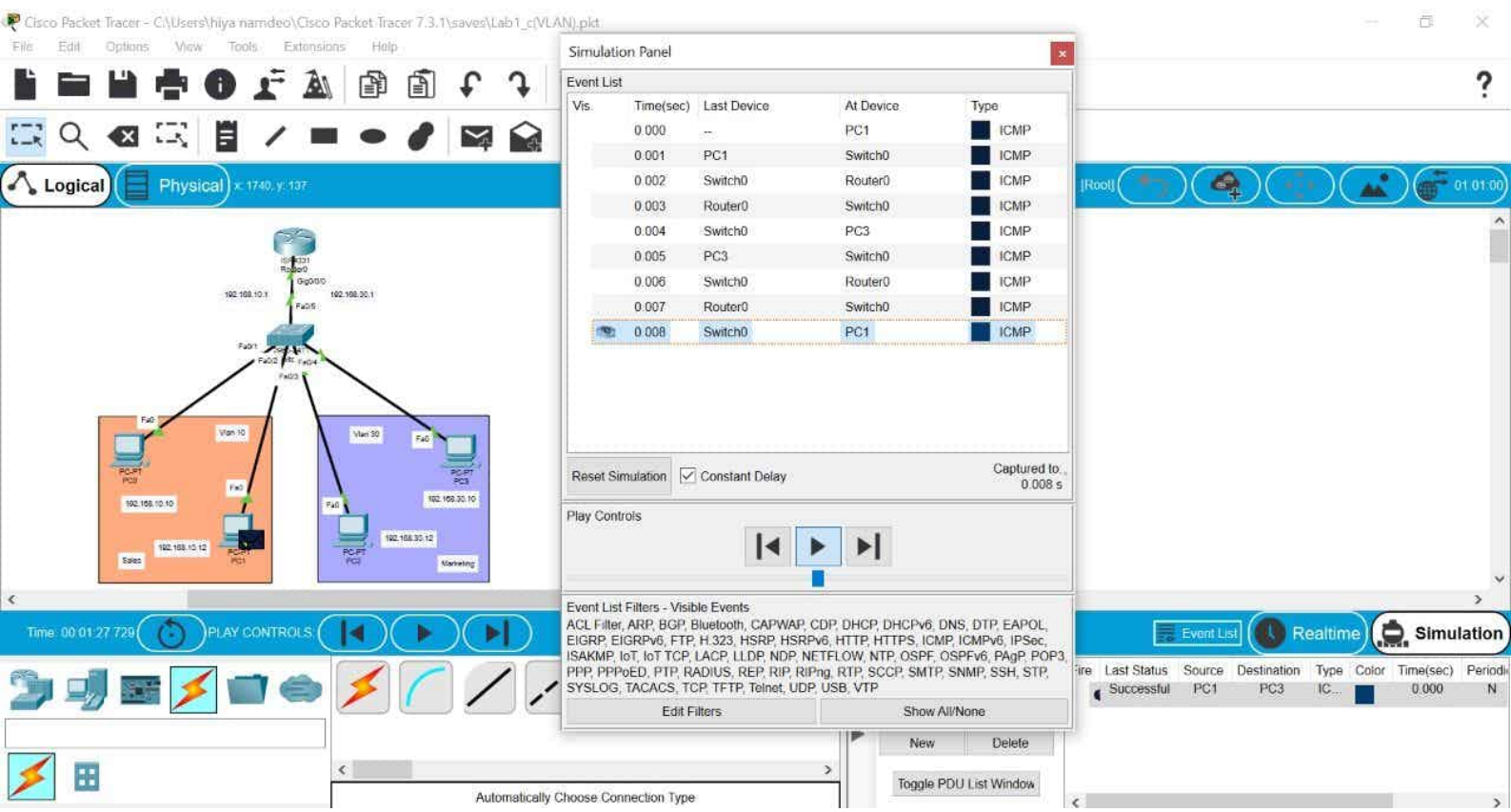
router (config)# interface g0/0

router (config-subif)# no shutdown

OBSERVATION & OUTPUT :

Packets travel between 2 different hosts belonging to different vlans using router-on-a stick or inter-vlan routing.

~~OUTPUT:~~


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab1_c(VLAN).pkt
File Edit Options View Tools Extensions Help
Logical
Physical
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- PC1 ICMP
0.001 PC1 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Switch0 ICMP
0.004 Switch0 PC3 ICMP
0.005 PC3 Switch0 ICMP
0.006 Switch0 Router0 ICMP
0.007 Router0 Switch0 ICMP
0.008 Switch0 PC1 ICMP
Reset Simulation
Constant Delay
Captured to: 0.008 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:01:27.729
PLAY CONTROLS
Event List
Realtime
Simulation
Last Status Source Destination Type Color Time(sec)
Successful PC1 PC3 IC... 0.000
New
Delete
Toggle PDU List Window
Automatically Choose Connection Type

LAB PROGRAM #3

Q. Setup a Router based wide area Network using Dynamic Routing (RIP, EIGRP, OSPF)

Aim: To Setup a network of 3 & 4 routers, Configure routing in each router and test the network.

DESIGN:

Dynamic Routing is a process where a router can forward data via a different route as given destination based on the current conditions of the communication circuits within a system.

(i) Routing Information Protocol (RIP):
- it is a dynamic routing protocol which uses hop count as a routing metric to find the best path between the source & destination network.

(ii) Enhanced Interior Gateway Routing Protocol (EIGRP):
- it is a dynamic routing protocol which is used to find the best path between any 2 layer 3 device to deliver the packet (distance vector protocol).

(iii) Open shortest path First (OSPF):
- It is link state routing protocol that is used to find the best path between source & destination router using its own shortest path first.

CONFIGURATION:

RIP:

router 0 : router(config) # router rip
router (config-router) # version 2
router (config-router) # network 10.0.0.0
router (config-router) # network 11.0.0.0
router (config-router) # network 192.168.1.0
router (config-router) # exit

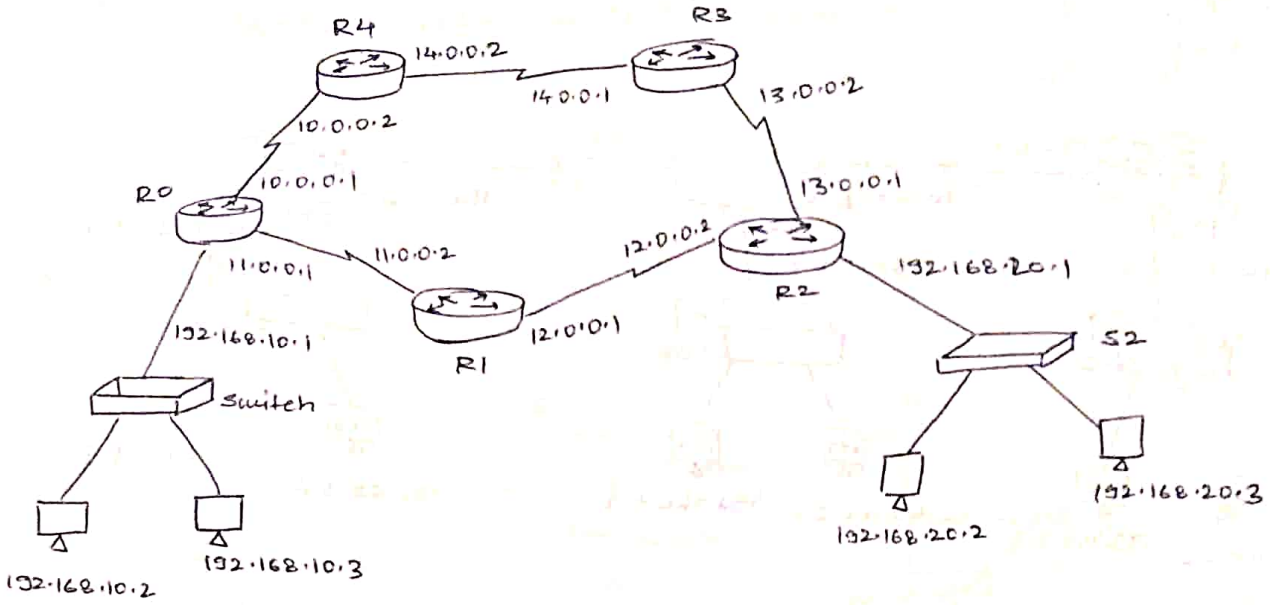
router 1 : router (config) # router rip
router (config-router) # version 2
router (config-router) # network 11.0.0.0
router (config-router) # network 12.0.0.0
router (config-router) # exit

router 2 : router (config) # router rip
router (config-router) # version 2
router (config-router) # network 12.0.0.0
router (config-router) # network 13.0.0.0
router (config-router) # network 192.168.20.0
router (config-router) # exit

router 3 : router (config) # router rip
router (config-router) # version 2
router (config-router) # network 13.0.0.0
router (config-router) # network 14.0.0.0
router (config-router) # exit

router 4 : router (config) # router rip
router (config-router) # version 2
router (config-router) # network 14.0.0.0
router (config-router) # network 10.0.0.0
router (config-router) # exit.

TOPOLOGY:



CONFIGURATION:

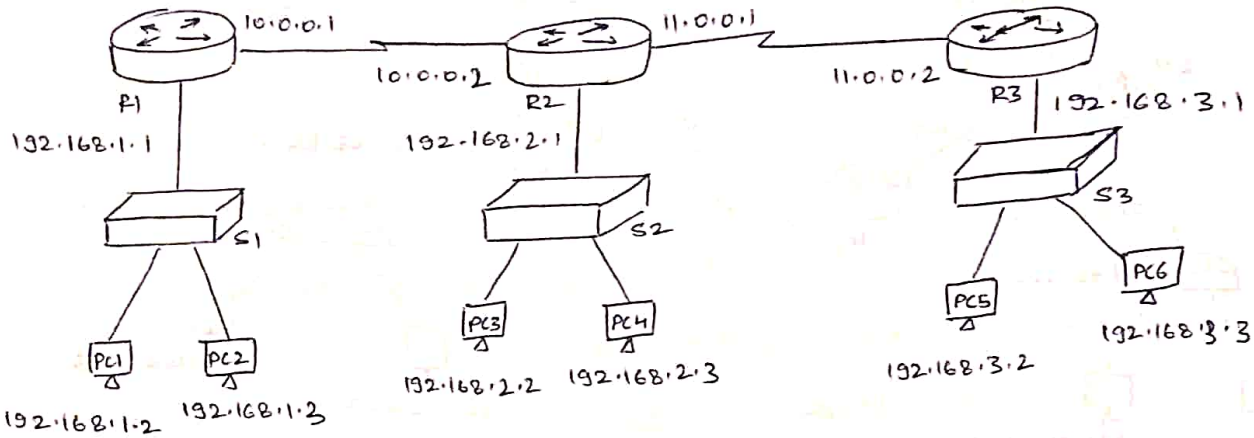
EIGRP

Router 0 : Router (config) # router eigrp 1
router (config-router) # network 192.168.1.0
router (config-router) # network 10.0.0.0
router (config-router) # exit
router (config) # exit

Router 1 : Router (config) # router eigrp 1
router (config-router) # network 10.0.0.0
router (config-router) # network 11.0.0.0
router (config-router) # network 192.168.2.0
router (config-router) # exit.

Router 2 : router (config) # router eigrp 1
router (config-router) # network 11.0.0.0
router (config-router) # network 192.168.3.0
router (config-router) # exit.

TOPOLOGY :



CONFIGURATION :

OSPF :

Router 0 : router (config) # router ospf 1

router (config-router) # network 10.0.0.0 0.255.255.255 area 0

router (config-router) # network 192.168.1.0 0.0.0.255 area 0

router (config-router) # exit

router (config) # exit

Router 1 : router (config) # router ospf 1

router (config-router) # network 10.0.0.0 0.255.255.255 area 0

router (config-router) # network 11.0.0.0 0.255.255.255 area 0

router (config-router) # network 192.168.2.0 0.0.0.255 area 0

router (config-router) # exit

Router 2 : router (config) # router ospf 1

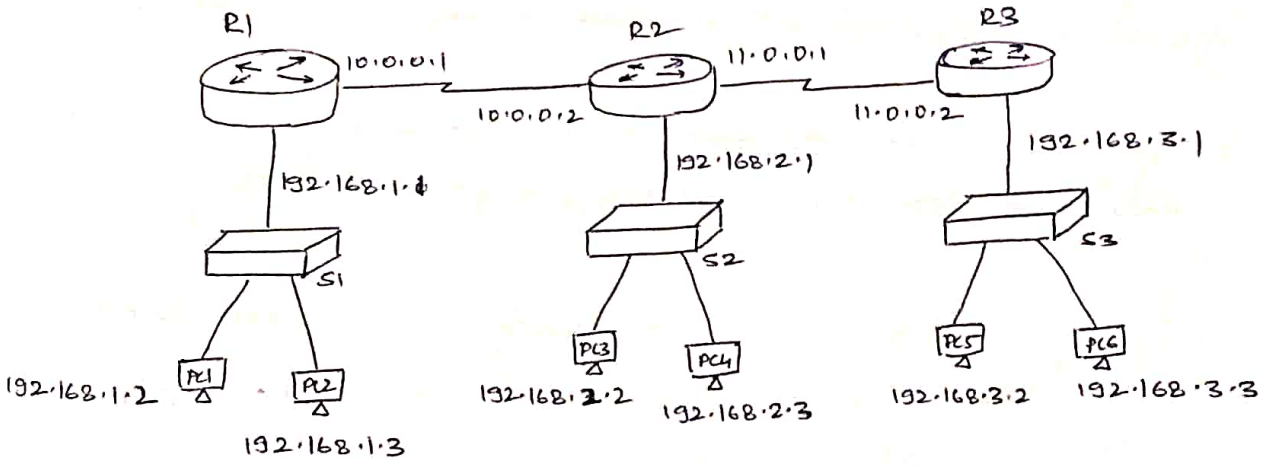
router (config-router) # network 192.168.3.0 0.0.0.255 area 0

router (config-router) # network 11.0.0.0 0.255.255.255 area 0.

router (config-router)# exit.

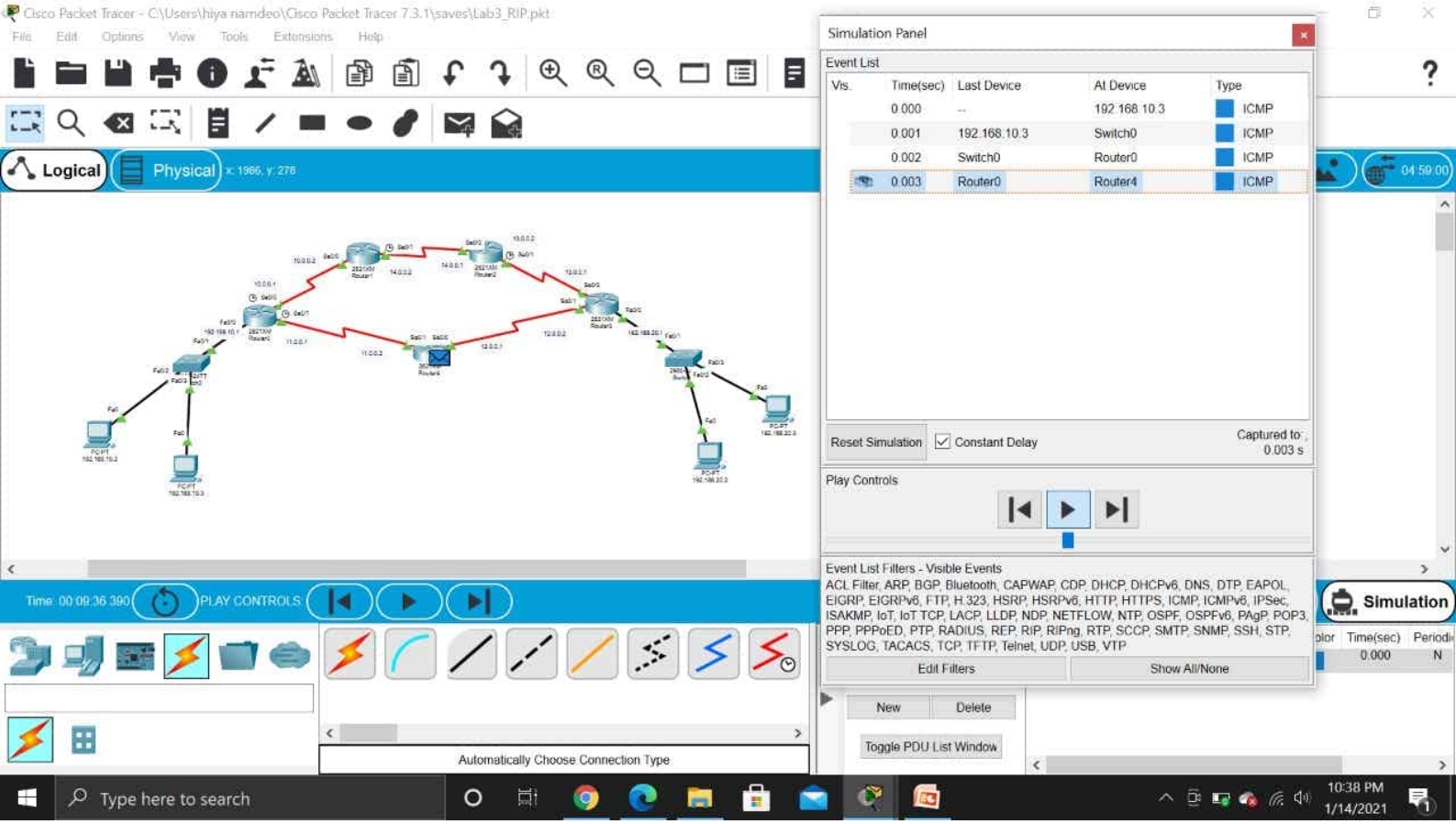
router (config)#-exit

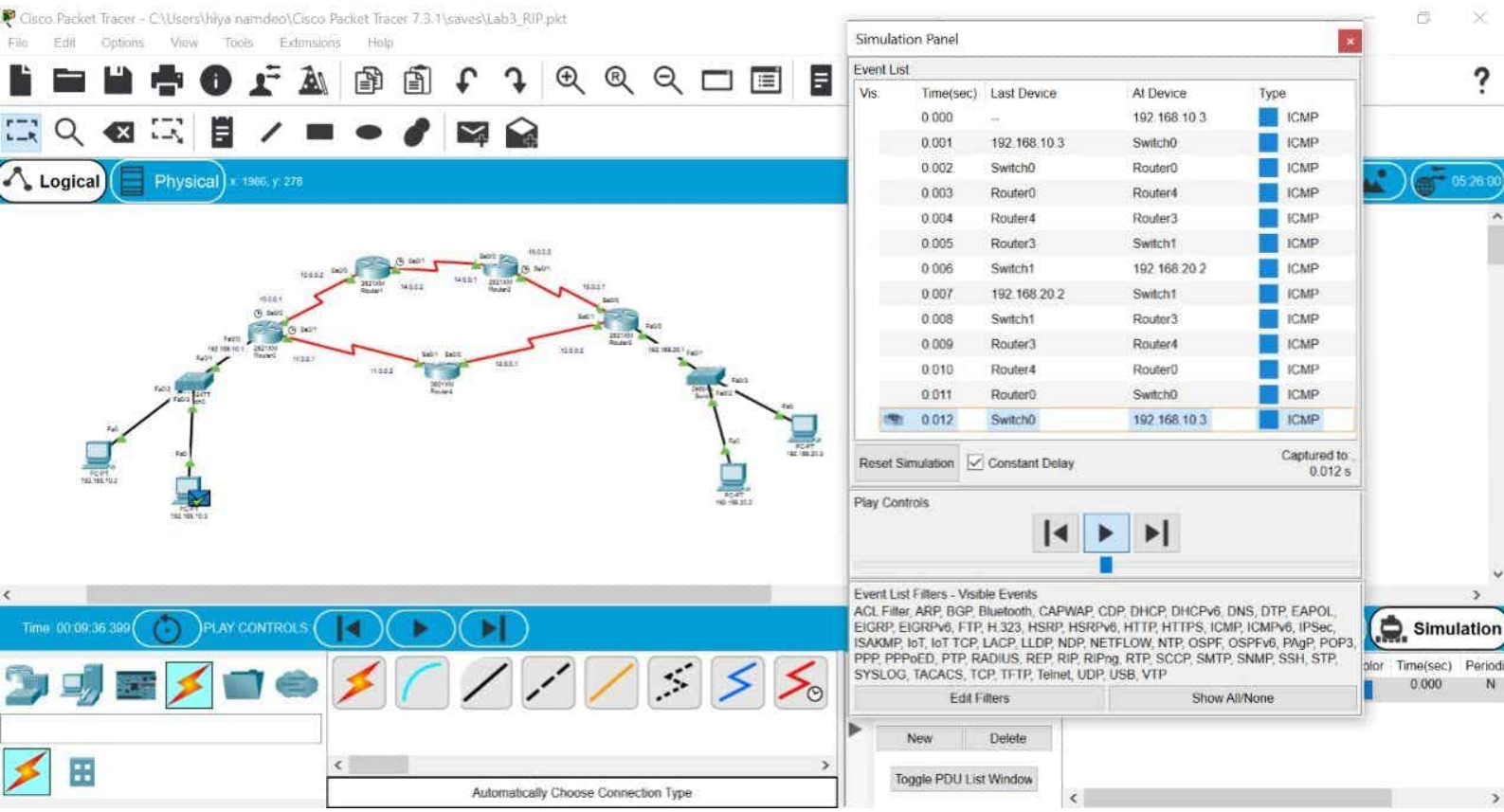
TOPOLOGY :

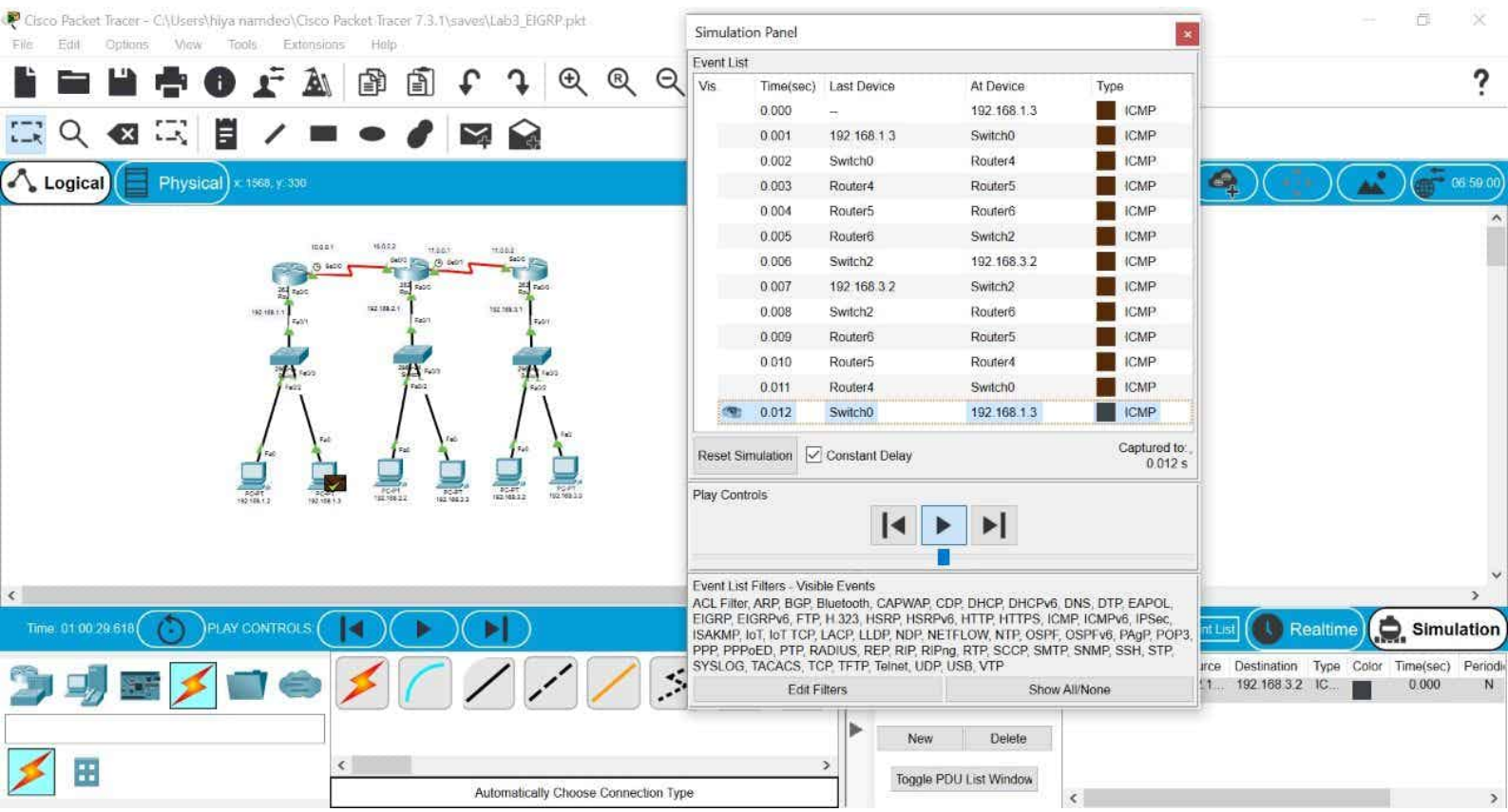


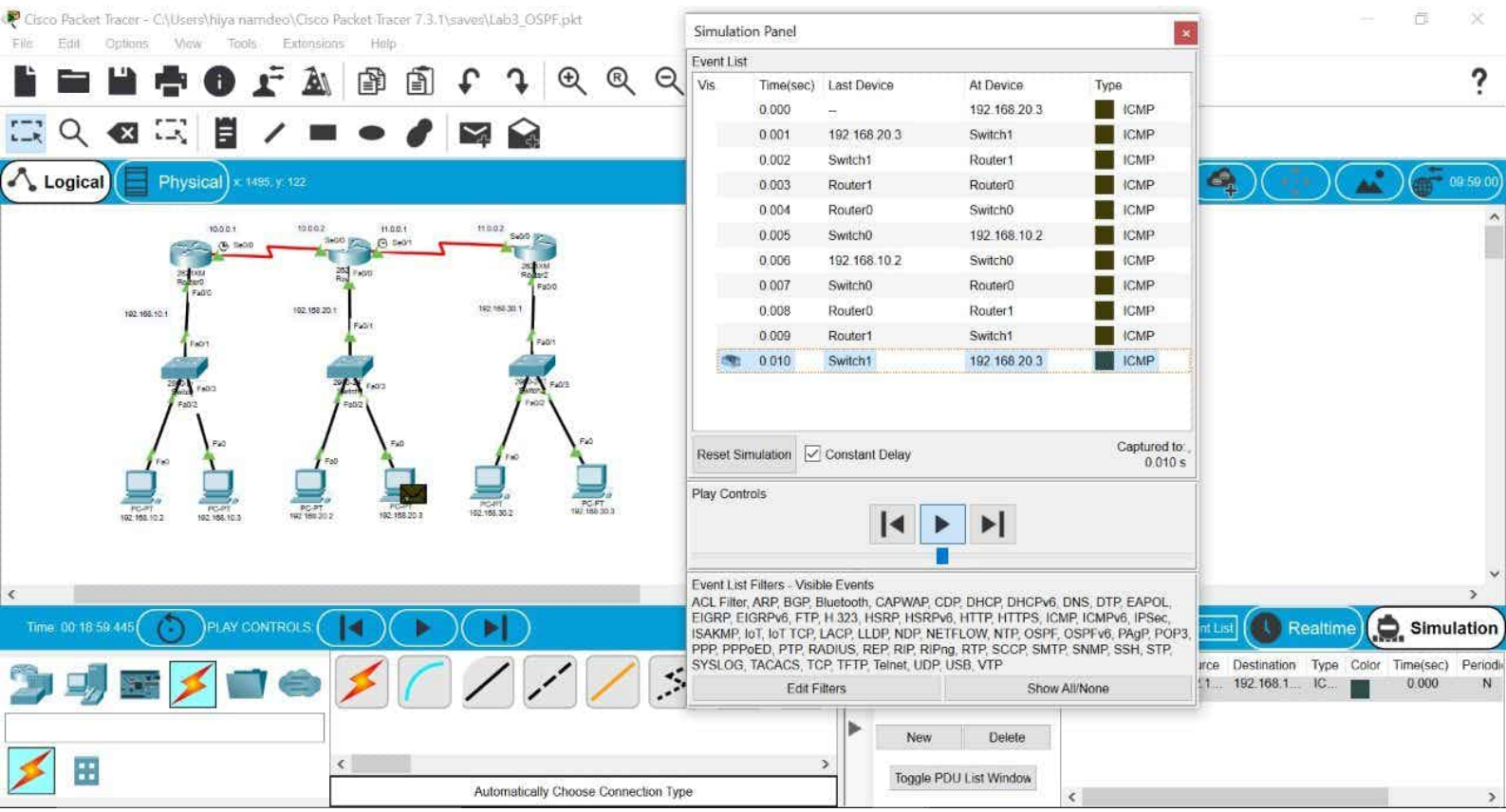
OBSERVATION AND OUTPUT :

Packets can be routed between source and destination by the router using appropriate interior gateway protocols within an ~~at out~~ autonomous system.


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab3_RIP.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
x: 1966, y: 278
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- 192.168.10.3 ICMP
0.001 192.168.10.3 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Router4 ICMP
Reset Simulation
Constant Delay
Captured to: 0.003 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
New
Delete
Toggle PDU List Window
Time: 00:09:36.390
PLAY CONTROLS
Simulation
Automatically Choose Connection Type
Type here to search
10:38 PM
1/14/2021


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab3_RIP.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- 192.168.10.3 ICMP
0.001 192.168.10.3 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Router4 ICMP
0.004 Router4 Router3 ICMP
0.005 Router3 Switch1 ICMP
0.006 Switch1 192.168.20.2 ICMP
0.007 192.168.20.2 Switch1 ICMP
0.008 Switch1 Router3 ICMP
0.009 Router3 Router4 ICMP
0.010 Router4 Router0 ICMP
0.011 Router0 Switch0 ICMP
0.012 Switch0 192.168.10.3 ICMP
Reset Simulation
Constant Delay
Captured to: 0.012 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
New
Delete
Toggle PDU List Window
Time: 00:09:36.399
PLAY CONTROLS:
Simulation
Time(sec)
0.000
Automatically Choose Connection Type


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab3_EIGRP.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
x: 1568, y: 330
Simulation Panel
Event List
Vis Time(sec) Last Device At Device Type
0.000 -- 192.168.1.3 ICMP
0.001 192.168.1.3 Switch0 ICMP
0.002 Switch0 Router4 ICMP
0.003 Router4 Router5 ICMP
0.004 Router5 Router6 ICMP
0.005 Router6 Switch2 ICMP
0.006 Switch2 192.168.3.2 ICMP
0.007 192.168.3.2 Switch2 ICMP
0.008 Switch2 Router6 ICMP
0.009 Router6 Router5 ICMP
0.010 Router5 Router4 ICMP
0.011 Router4 Switch0 ICMP
0.012 Switch0 192.168.1.3 ICMP
Reset Simulation
Constant Delay
Captured to: 0.012 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 01:00:29.618
PLAY CONTROLS
Realtime
Simulation
Destination Type Color Time(sec) Periodic
192.168.3.2 IC... 0.000 N
New
Delete
Toggle PDU List Window
Automatically Choose Connection Type


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab3_OSPF.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
Simulation Panel
Event List
Vis Time(sec) Last Device At Device Type
0.000 -- 192.168.20.3 ICMP
0.001 192.168.20.3 Switch1 ICMP
0.002 Switch1 Router1 ICMP
0.003 Router1 Router0 ICMP
0.004 Router0 Switch0 ICMP
0.005 Switch0 192.168.10.2 ICMP
0.006 192.168.10.2 Switch0 ICMP
0.007 Switch0 Router0 ICMP
0.008 Router0 Router1 ICMP
0.009 Router1 Switch1 ICMP
0.010 Switch1 192.168.20.3 ICMP
Reset Simulation
Constant Delay
Captured to: 0.010 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:18:59.445
PLAY CONTROLS
Realtime
Simulation
Destination Type Color Time(sec) Periodic
192.168.1... IC... 0.000 N
New
Delete
Toggle PDU List Window
Automatically Choose Connection Type

LAB PROGRAM # 4

4 Practice IP addressing Principles:-

(a) Setup a Subnet (N1) comprising 4 nodes. Change the subnet masks in some of the nodes & test the network; set up another subnet (N2) of 4 nodes, connect these 2 subnets using router.

AIM: To create 2 subnets (N1 & N2) having 4 nodes each connected with each other through a router.

DESIGN: A subnet is a logical partition of an IP network into multiple, smaller network segments.

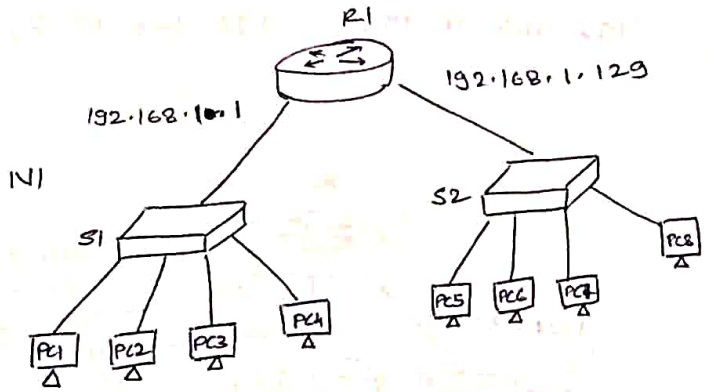
Subnet N1	Start IP	End IP	Broadcast IP
192.168.10.0/25	192.168.10.1	192.168.10.126	192.168.10.127

subnet N2	Start IP	End IP	Broadcast IP
192.168.10.128/25	192.168.10.129	192.168.10.254	192.168.10.255

Subnet mask - 255.255.255.128

~~TOPO~~
~~CONFIGURATION:~~

TOPOLOGY:



CONFIGURATION:

router (config) # interface fa 0/0
router (config-if) # ip address 192.168.10.1 255.255.255.128
router (config-if) # exit
router (config) #interface fa 1/0

change the subnet mask of PC2 from 255.255.255.128 to 255.255.255.248

> Ping PC2 from PC1

Reply from 192.168.10.3 byte - 32 time 1ms TTL: 128

⋮

Packets Sent = 4 Received = 4 Loss = 0 (% 0)

(b) Create 4 equal sized subnets in the subnet N1 & test the network

DESIGN:

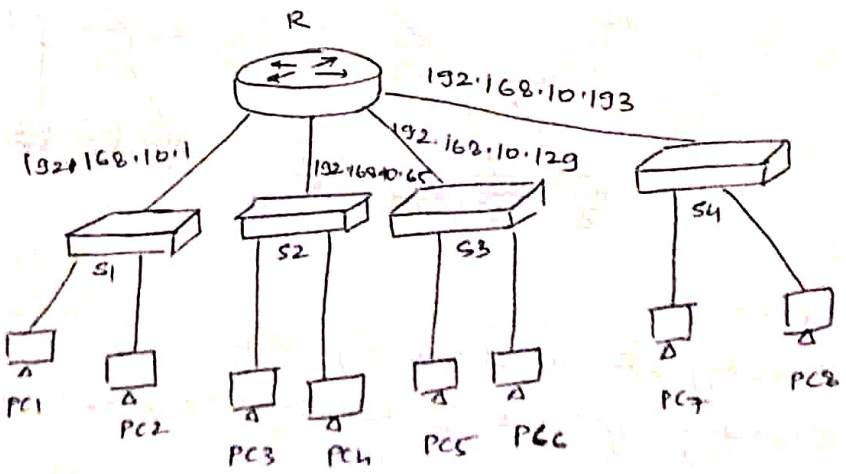
4 subnets ∴ borrow 2 bit

IP 192.168.10.0/26

Subnet mask = 255.255.255.192

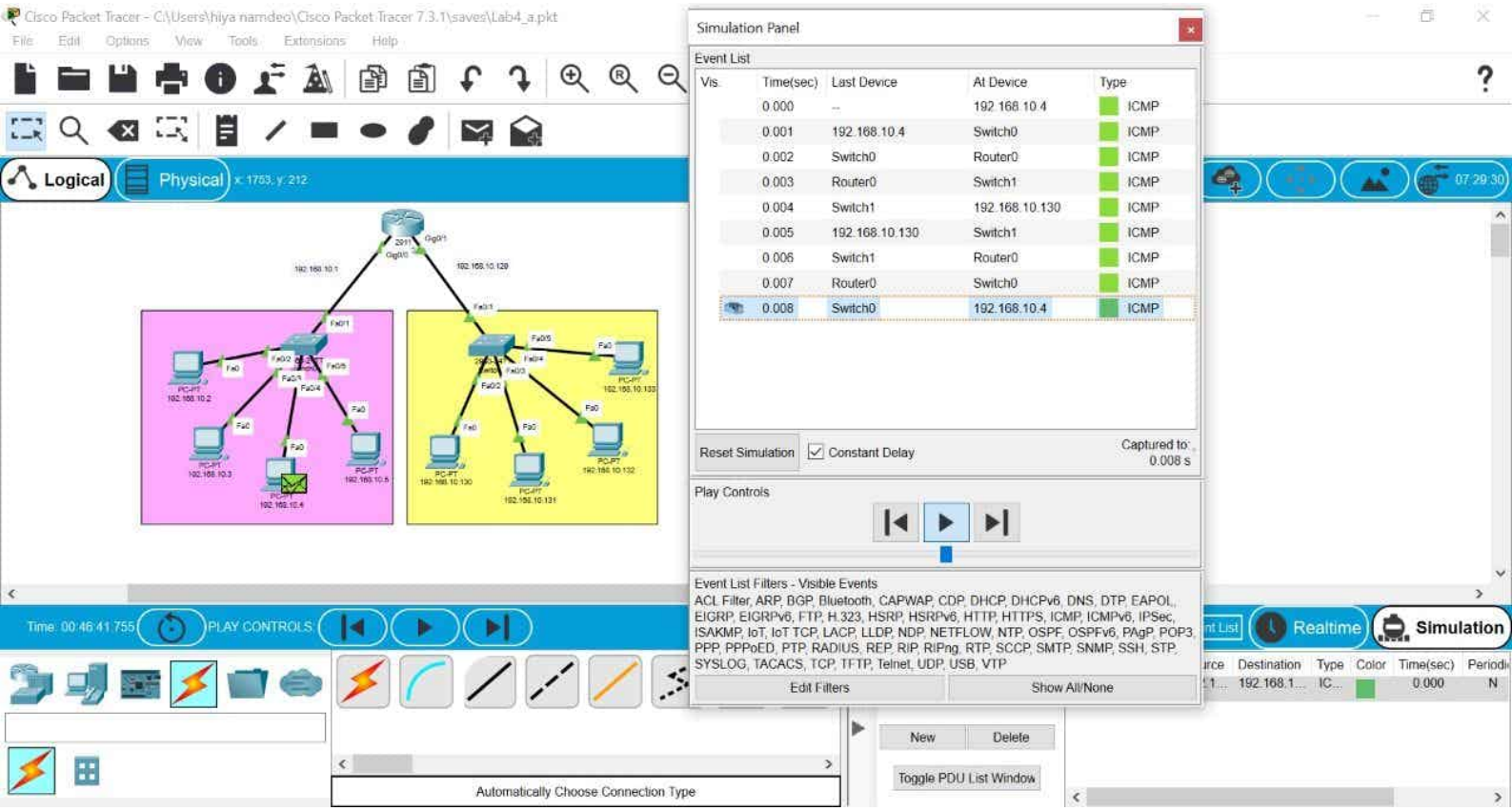
Subnets	start IP	End IP	Broadcast IP
192.168.10.0	192.168.10.1	192.168.10.62	192.168.10.63
192.168.10.64	192.168.10.65	192.168.10.126	192.168.10.127
192.168.10.128	192.168.10.129	192.168.10.190	192.168.10.191
192.168.10.192	192.168.10.193	192.168.10.254	192.168.10.255

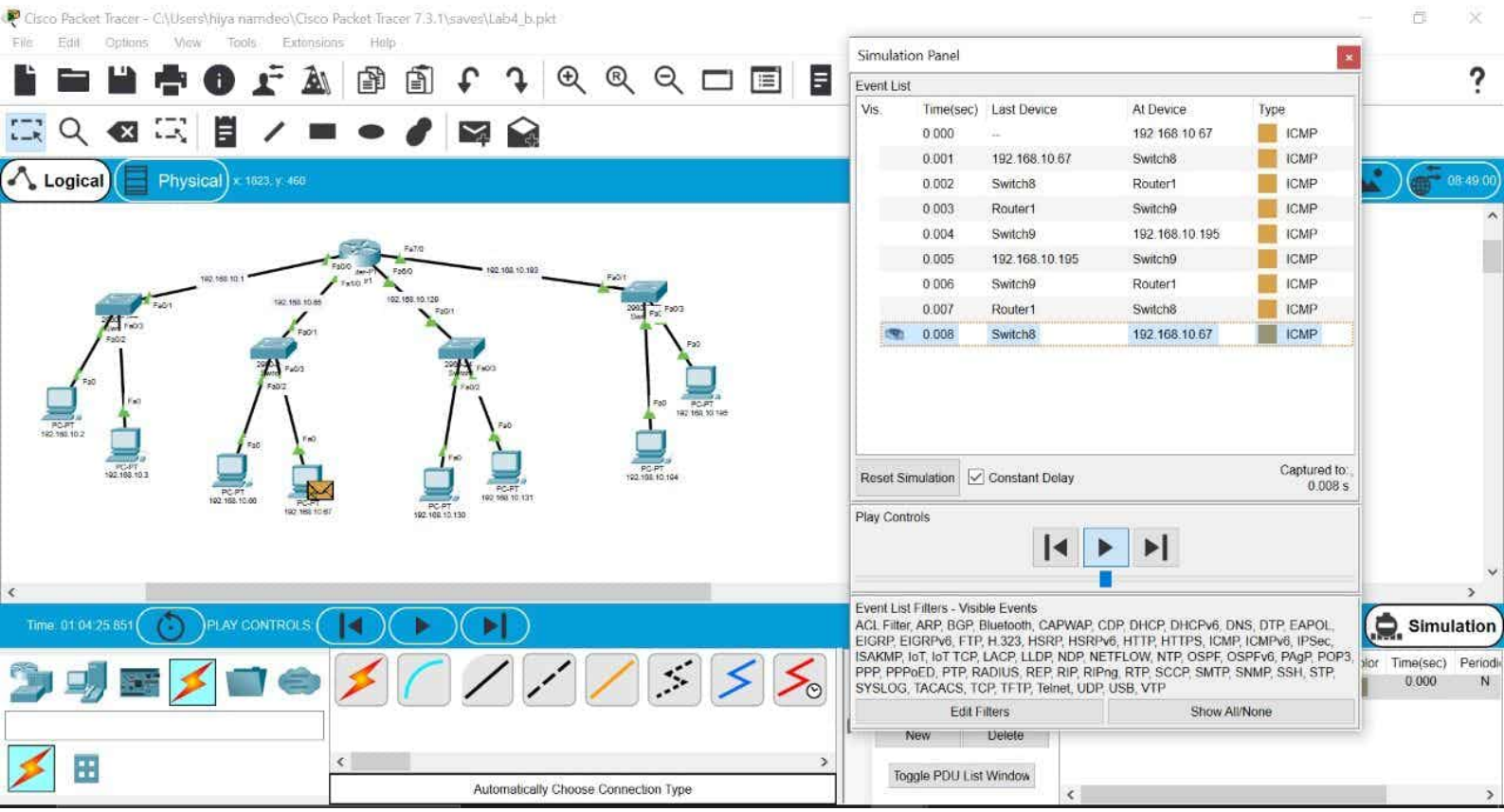
TOPOLOGY:



OBSERVATION & OUTPUT :

The packets sent from one subnet are being routed by the router to its respective destinations even though all systems are in the same network.


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab4_a.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
x: 1753, y: 212
07:29:30
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- 192.168.10.4 ICMP
0.001 192.168.10.4 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Switch1 ICMP
0.004 Switch1 192.168.10.130 ICMP
0.005 192.168.10.130 Switch1 ICMP
0.006 Switch1 Router0 ICMP
0.007 Router0 Switch0 ICMP
0.008 Switch0 192.168.10.4 ICMP
Reset Simulation
Constant Delay
Captured to: 0.008 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:46:41.755
PLAY CONTROLS
Realtime
Simulation
Destination Type Color Time(sec) Periodic
192.168.1... IC... 0.000 N
New
Delete
Toggle PDU List Window
Automatically Choose Connection Type


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab4_b.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
x: 1623, y: 460
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- 192.168.10.67 ICMP
0.001 192.168.10.67 Switch8 ICMP
0.002 Switch8 Router1 ICMP
0.003 Router1 Switch9 ICMP
0.004 Switch9 192.168.10.195 ICMP
0.005 192.168.10.195 Switch9 ICMP
0.006 Switch9 Router1 ICMP
0.007 Router1 Switch8 ICMP
0.008 Switch8 192.168.10.67 ICMP
Reset Simulation
Constant Delay
Captured to: 0.008 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 01:04:25.851
PLAY CONTROLS
Simulation
New
Delete
Toggle PDU List Window
Automatically Choose Connection Type

LAB PROGRAM #5

Implement DHCP and DNS

(a) A client, single DNS server and a web server.

(b) A client, 2 DNS servers and a web server.

(c) A client & a hierarchy of DNS servers & a web server

AIM: To setup a client, DNS servers and a web server to implement DHCP & DNS

DESIGN: A DNS is a type of name server that helps internet users discover websites using human readable addresses

In DNS Server

Resource Records

No.	Name	Type	Detail
0	www.google.com	Record	60.60.60.2

In local DNS → Services → DNS

No.	Name	Types	Details
0	root	A Record	20.20.20.2
1	www.google.com	NS	root

In root DNS → services → DNS

No.	Name	Types	Detail
0	tld	A Record	30.30.30.2
1	www.google.com	NS	tld

In TLD DNS → services → DNS

No.	Name	Types	Detail
0	authority	A record	40.40.40.2
1	www.google.com	NS	authority

In authority

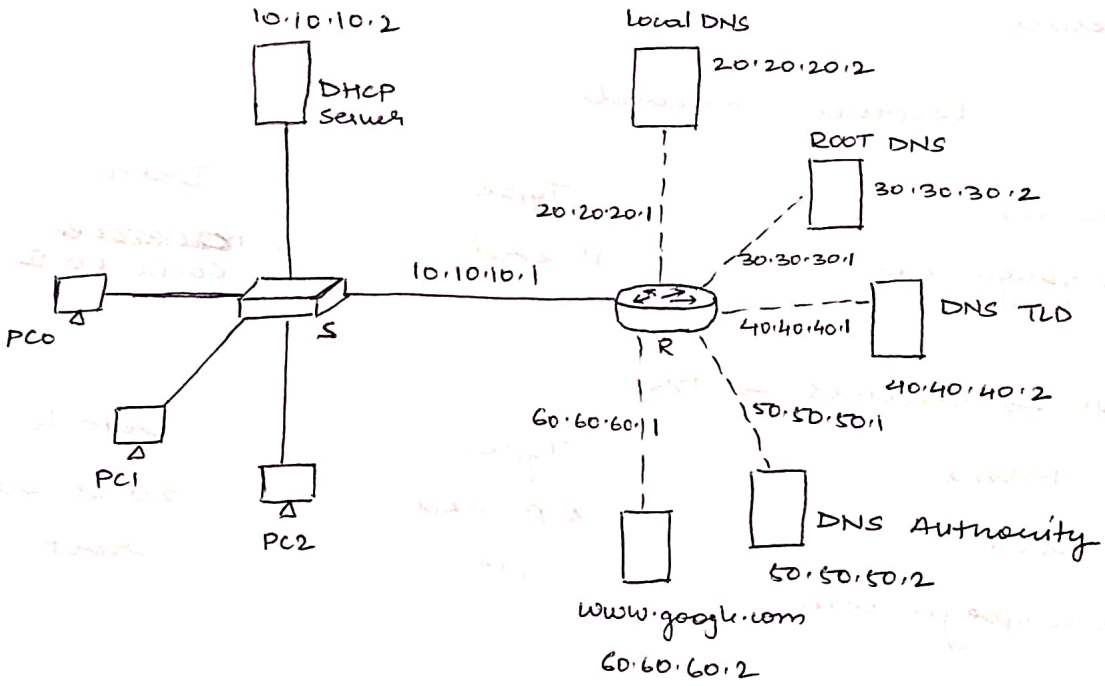
No.	Name	Type	Detail
0	www.google.com	A record	50.50.50.2

DHCP services → DHCP

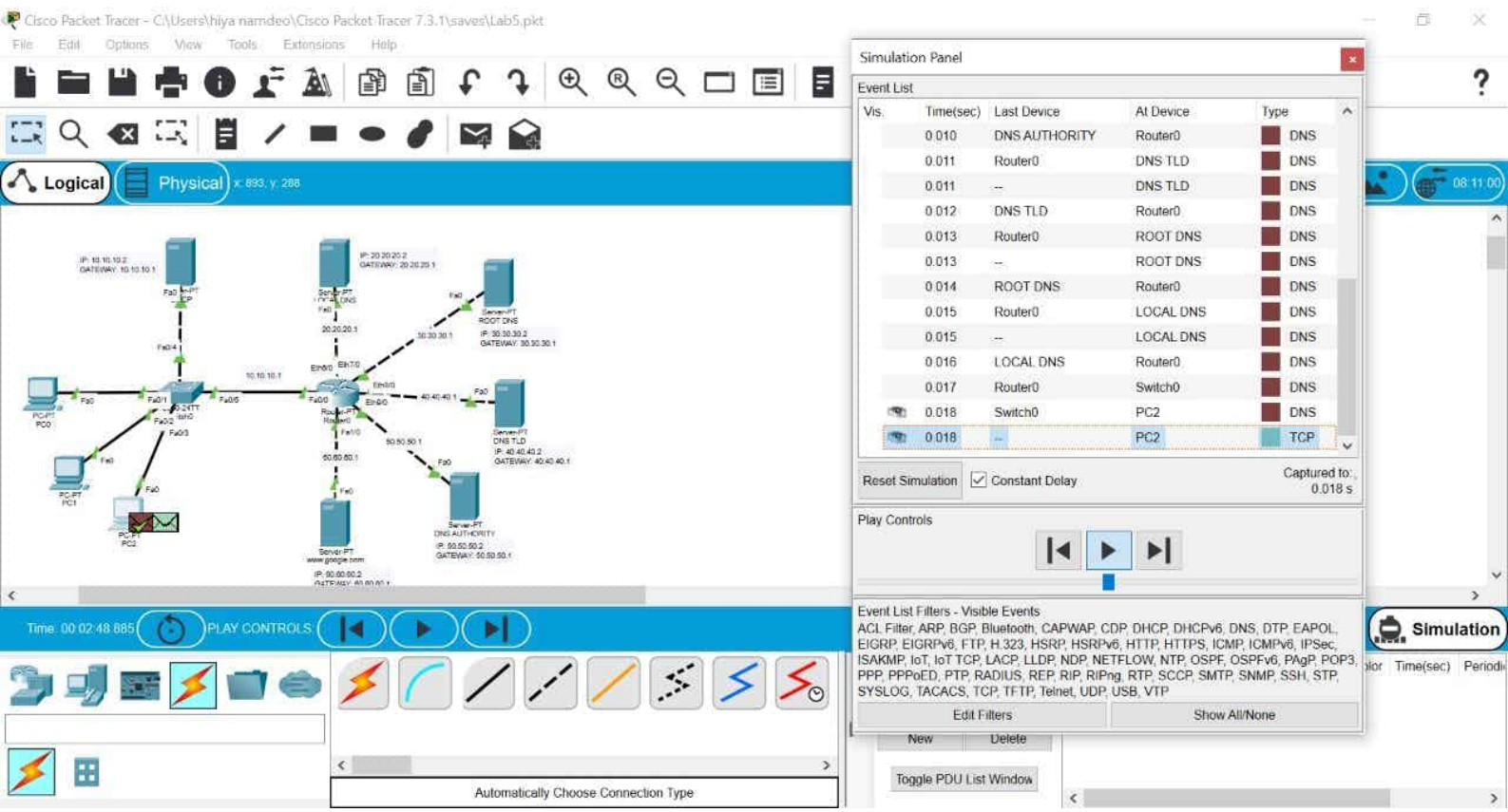
Pool Name	My_POOL				
Default Gateway	10.10.10.1				
DNS Server	20.20.20.2				
Start IP Address	:	10	10	10	3
Subnet Mask	:	255	0	0	0

Set PC's IP configuration to DHCP

TOPOLOGY :



OBSERVATION & OUTPUT


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab5.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.010 DNS AUTHORITY Router0 DNS
0.011 Router0 DNS TLD DNS
0.011 -- DNS TLD DNS
0.012 DNS TLD Router0 DNS
0.013 Router0 ROOT DNS DNS
0.013 -- ROOT DNS DNS
0.014 ROOT DNS Router0 DNS
0.015 Router0 LOCAL DNS DNS
0.015 -- LOCAL DNS DNS
0.016 LOCAL DNS Router0 DNS
0.017 Router0 Switch0 DNS
0.018 Switch0 PC2 DNS
0.018 -- PC2 TCP
Reset Simulation
Constant Delay
Captured to: 0.018 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
New
Delete
Toggle PDU List Window
Time: 00:02:48.885
PLAY CONTROLS
Simulation
Time(sec)
Automatically Choose Connection Type

LAB PROGRAM #6

Q. Implement Static NAT, Dynamic NAT and PAT

AIM : (a) To setup a Network and implement STATIC NAT

DESIGN : NAT :

→ NAT stands for 'Network Address Translation".

→ It is a method of Translation of private IP address into public IP address

→ common devices that can perform address translation include firewalls, routers, and servers.

→ Typically address translation is done at the perimeter of the network by either a firewall or router.

→ There are certain addresses in each class of IP address that are reserved for Private Networks. These addresses are called private addresses.

- class A : 10.0.0.0 – 10.255.255.255
- class B : 172.16.0.0 – 172.31.255.255
- class C : 192.168.0.0 – 192.168.255.255

- NAT Terminology :

- Inside local Address :- Name of inside source address before translation (private IP)

- Inside Global Address :- Name of inside host after translation (public IP)

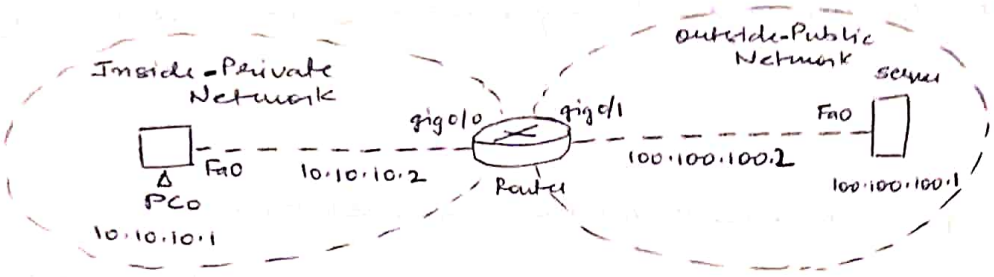
- Outside local Address :- Name of destination host before translation

- Outside Global Address :- Name of outside destination host after translation.

- STATIC NAT :

- This type of NAT is designed to allow one to one mapping between local & global address.

TOPOLOGY :



CONFIGURATION :

Router : router (config) # int gig 0/0
router (config-if) # ip nat inside
router (config-if) # int gig 0/1
router (config-if) # ip nat outside
router (config-if) # exit
router (config) # ip nat inside source static 10.10.10.1 100.100.100.2
router (config) # exit
router #

to verify NAT translation

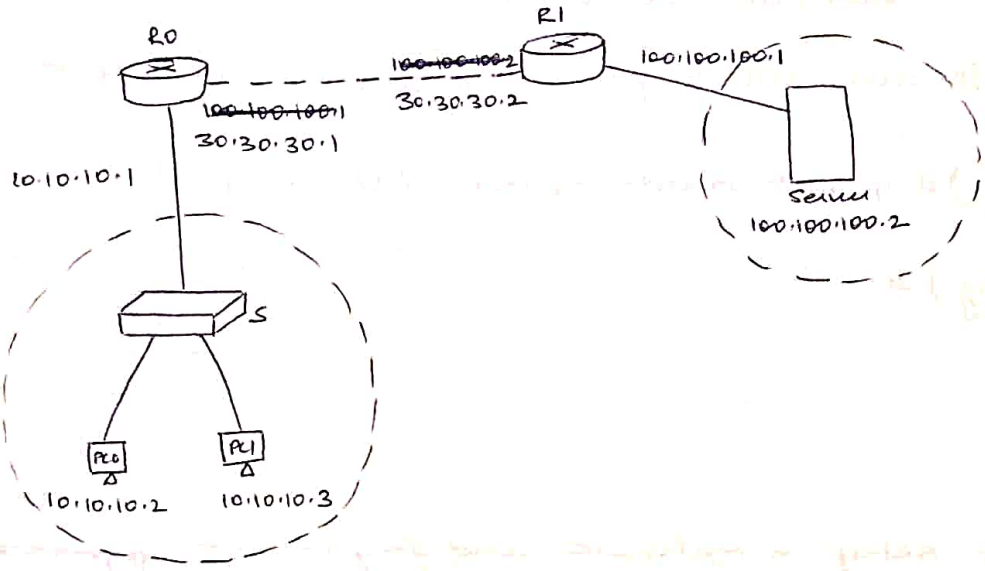
router # show ip nat translation

AIM : (b) To setup a network and Implement Dynamic NAT

DESIGN : Dynamic NAT :

- This version gives you the ability to map an unregistered IP address to a registered IP address from out of a pool of registered IP addresses.
- It is method of mapping of a private IP address to a public IP address from a group of public IP address called as NAT POOL.

TOPOLOGY:



CONFIGURATION:

In router 1:

```
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# network 100.0.0.0
Router(config-router)# network 30.0.0.0
Router(config-router)# exit
Router(config)# exit
```

In router 0:

```
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# network 30.0.0.0
Router(config-router)# exit
Router(config)# int gig 0/1
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# int gig 0/0
Router(config-if)# ip nat outside
Router(config-if)# exit
```

Router (config)#ip nat pool cse 30.30.30.4 30.30.30.10 netmask 255.0.0.0

Router (config)#access-list 10 permit 10.10.10.2 10.10.10.5

Router (config)# ip nat inside source list 10 pool cse

Router (config)# exit

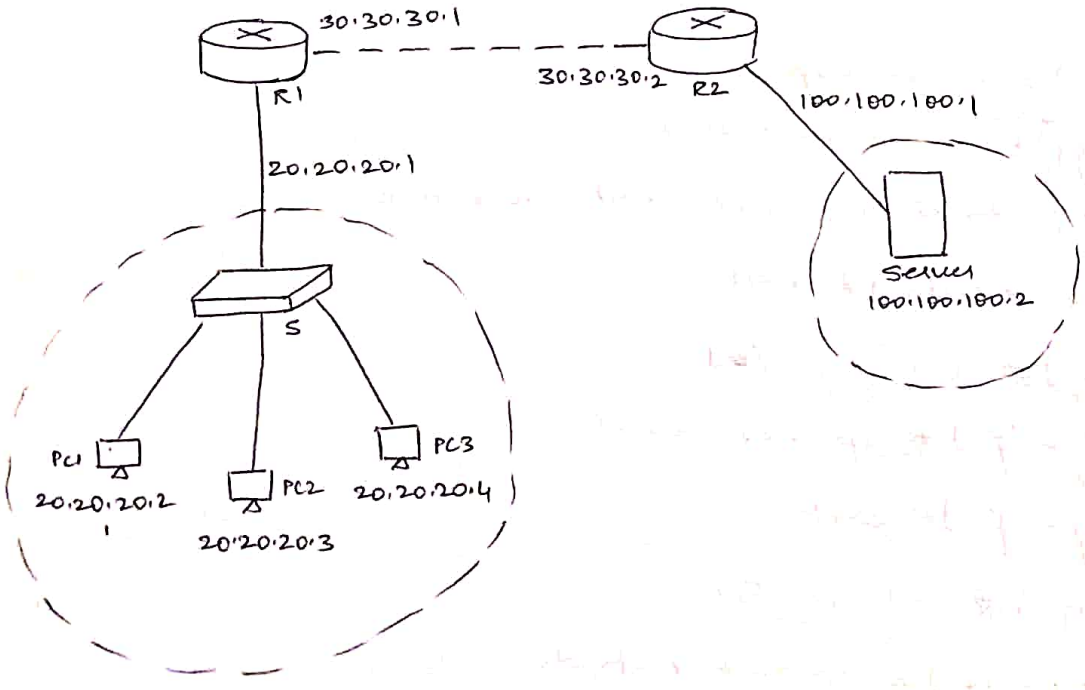
Router #

AIM: (C) To setup a network and Implement ~~Dynamic NAT~~ PAT

DESIGN: PAT: (NAT overload)

→ Port address Translation (PAT) is an extension to network address translation (NAT) that permits multiple devices on a local area network (LAN) to be mapped to a single public IP address. It's goal is to conserve IP addresses.

TOPOLOGY:



CONFIGURATION :

at Router 1 :

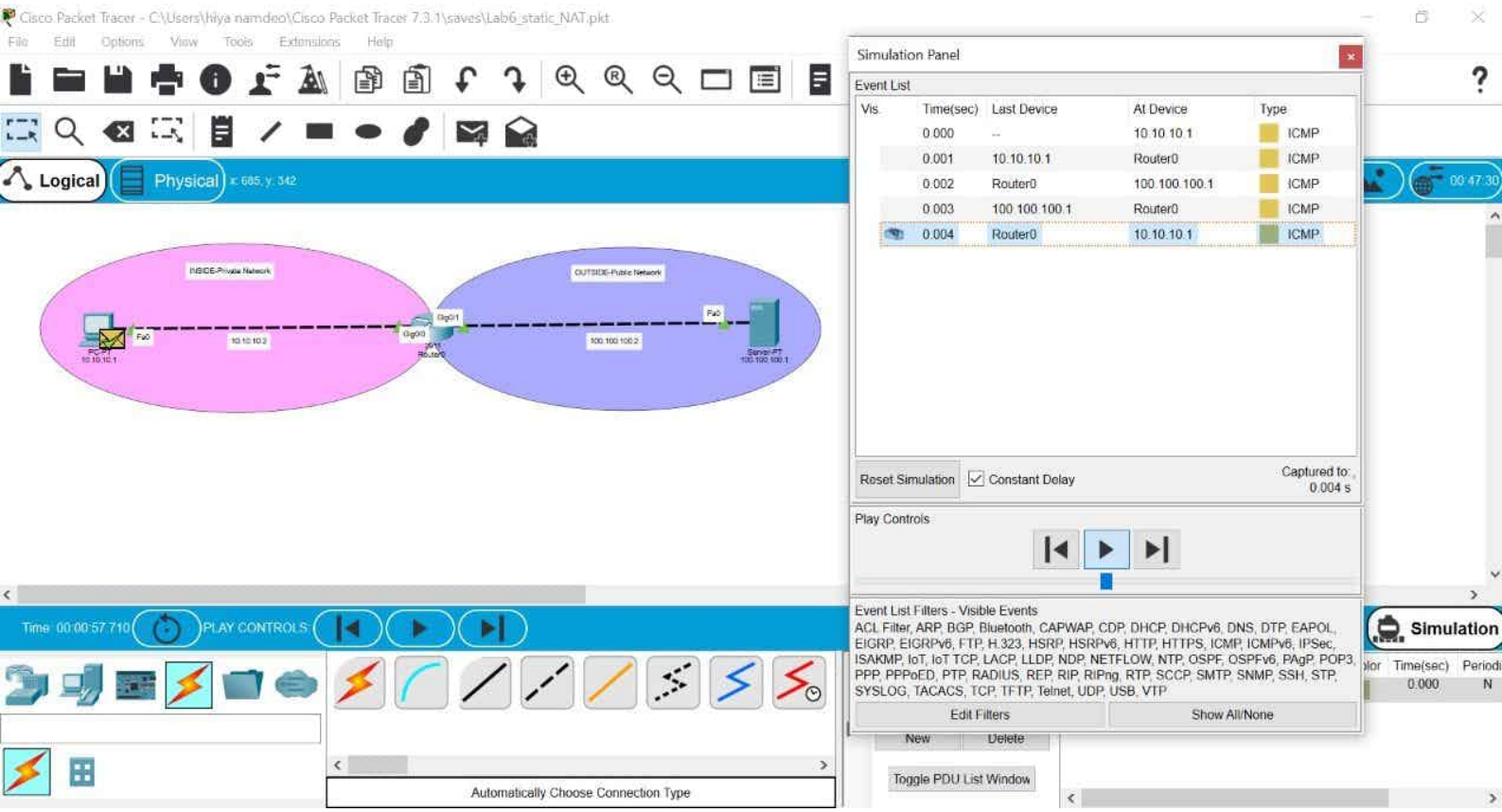
```
Router (config) # router rip
Router (config-router) # version 2
Router (config-router) # network 100.0.0.0
Router (config-router) # network 30.0.0.0
Router (config-router) # exit
Router (config) # exit
```

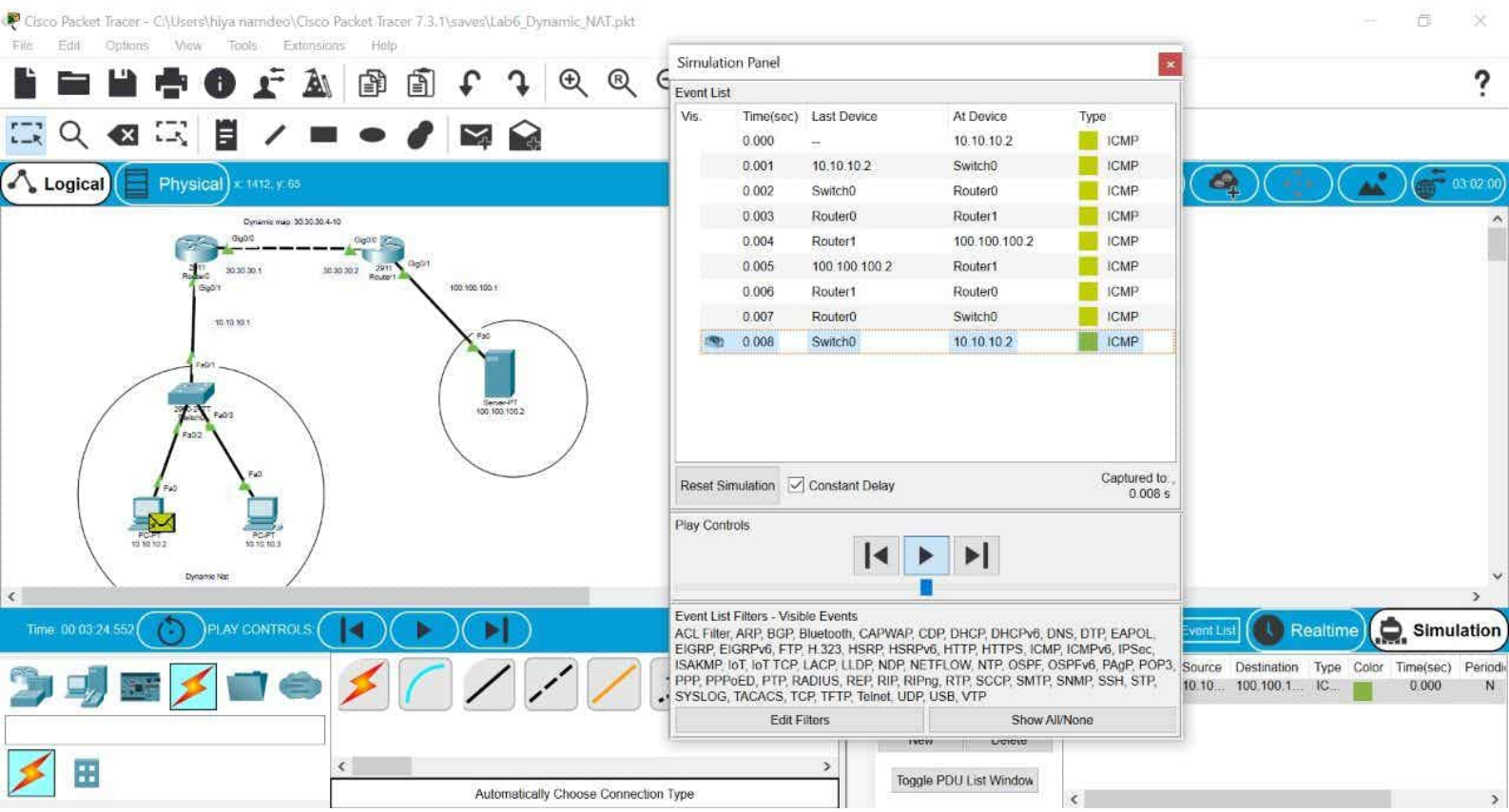
at Router 0 :

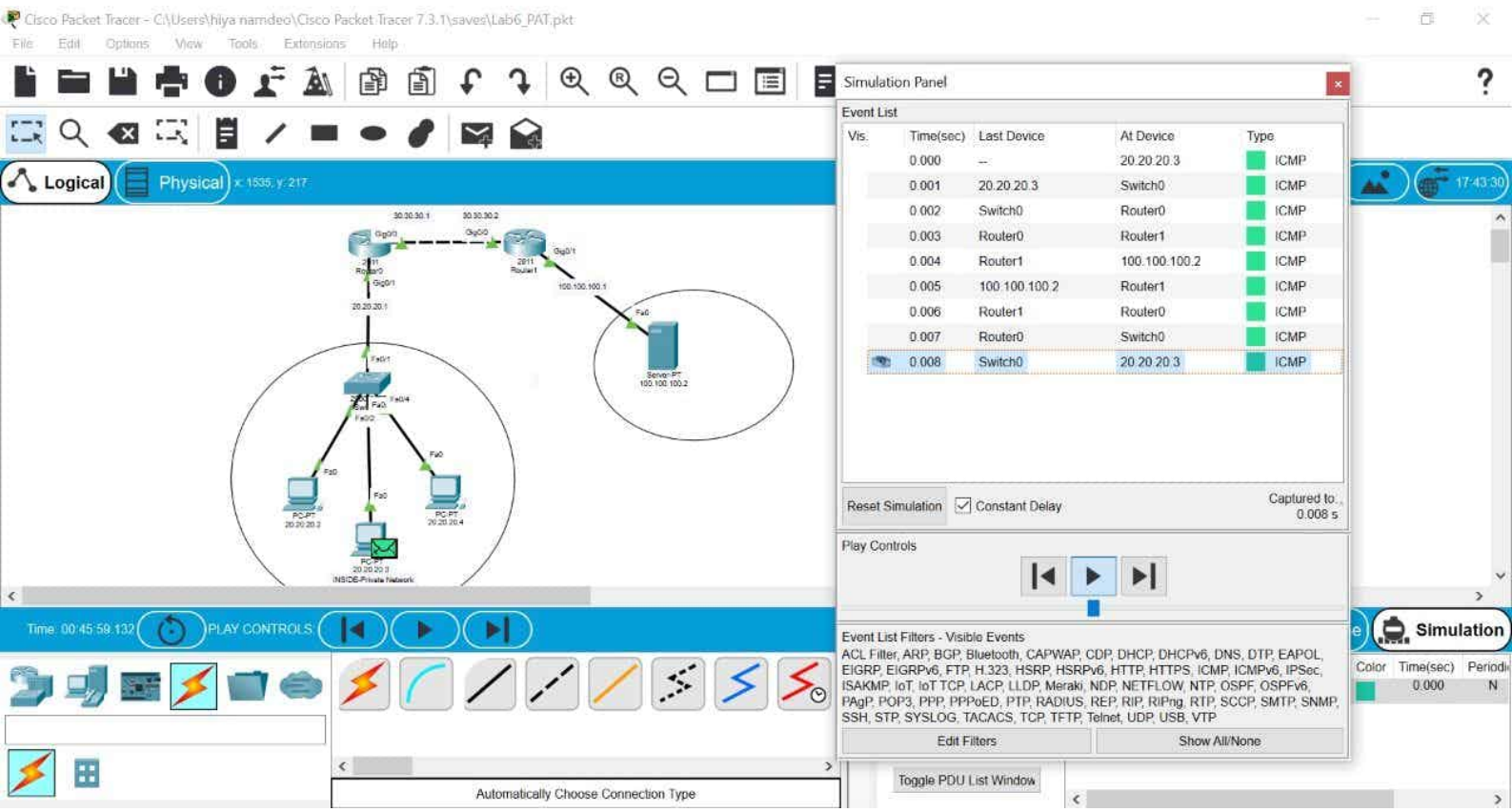
```
Router (config) # router rip
Router (config-router) # version 2
Router (config-router) # network 100.0.0.0 30.0.0.0
Router (config-router) # exit
Router (config) # int gig 0/0
Router (config-if ) # ip nat outside
Router (config-if) # exit
Router (config ) # int gig 0/1
Router (config-if ) # ip nat inside
Router (config-if) # exit
Router (config) # access-list 20 permit 20.20.20.0 0.0.0.255
Router (config) #ip nat inside source list 20 interface gig 0/0
                overload

Router (config) #exit

Router #
```

OBSERVATION & OUTPUT :

The private IP addresses within a system are converted to public IP addresses in the 3 cases respectively depending on whether it is a static NAT, dynamic NAT or PAT


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab6_static_NAT.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
INSIDE-Private Network
OUTSIDE-Public Network
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- 10.10.10.1 ICMP
0.001 10.10.10.1 Router0 ICMP
0.002 Router0 100.100.100.1 ICMP
0.003 100.100.100.1 Router0 ICMP
0.004 Router0 10.10.10.1 ICMP
Reset Simulation
Constant Delay
Captured to:
0.004 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
New
Delete
Toggle PDU List Window
Time: 00:00:57.710
PLAY CONTROLS
Simulation
Automatically Choose Connection Type


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab6_Dynamic_NAT.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- 10.10.10.2 ICMP
0.001 10.10.10.2 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Router1 ICMP
0.004 Router1 100.100.100.2 ICMP
0.005 100.100.100.2 Router1 ICMP
0.006 Router1 Router0 ICMP
0.007 Router0 Switch0 ICMP
0.008 Switch0 10.10.10.2 ICMP
Reset Simulation
Constant Delay
Captured to: 0.008 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:03:24.552
PLAY CONTROLS
Realtime
Simulation
Source Destination Type Color Time(sec) Periodi
10.10... 100.100.1... IC... 0.000 N
Automatically Choose Connection Type
Toggle PDU List Window


Cisco Packet Tracer - C:\Users\hiya namdeo\Cisco Packet Tracer 7.3.1\saves\Lab6_PAT.pkt
File Edit Options View Tools Extensions Help
Logical
Physical
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- 20.20.20.3 ICMP
0.001 20.20.20.3 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Router1 ICMP
0.004 Router1 100.100.100.2 ICMP
0.005 100.100.100.2 Router1 ICMP
0.006 Router1 Router0 ICMP
0.007 Router0 Switch0 ICMP
0.008 Switch0 20.20.20.3 ICMP
Reset Simulation
Constant Delay
Captured to:
0.008 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Toggle PDU List Window
Time: 00:45:58.132
PLAY CONTROLS
Simulation
Color Time(sec) Periodi
0.000 N
Automatically Choose Connection Type

PART - B

LAB PROGRAM #1

Q1. A) Write a c/c++ program to implement the data link layer framing methods

(A) bit stuffing

```
# include <stdio.h>
#include <conio.h>
# include <string.h>
void main()
{
    int a[20], b[30], i, j, k, count, n;
    clrscr();
    printf("enter frame length : ");
    scanf("%d", &n);
    printf("enter input frame (0's & 1's only): ");
    for(i=0; i<n; i++)
      { scanf("%d", &a[i]);
     }
    i=0;
    count=1;
    j=0;
    while(i<n)
    {  if(a[i]==1)
       {
          b[j] = a[i];
          for(k=i+1; a[k] == 1 && k<n && count<5; k++)
          {
             j++;
             b[j]=a[k];
             count++;
             if(count == 5)
             {
                j++;
                b[j] = 0;
```

```
        i=k;
      }
    }
    else
      {
          b[j] = a[i];
      }
      i++;
      j++;
    }
    printf ("After stuffing the frame is: ");
    for (i=0; i<j ; i++)
      {
        printf ("%d ", b[i]);
      }
      getch();
}
```

OUTPUT

```
Enter the number of bits : 10
1
0
1
0
1
1
1
1
1
1
Data after stuffing : 10101111101
```

LAB PROGRAM #1

(b) Write a c/c++ program to implement the data link layer framing methods.

(B) character stuffing

```
// program for character stuffing
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <process.h>
void main()
{
    int i=0, j=0, n, pos;
    char a[20], b[50], ch;
    clrscr();
    printf("enter string : \n");
    scanf("%s", &a);
    n = strlen(a);
    printf("enter position \n");
    scanf("%d", &pos);
    if (pos > n)
    {
        printf("invalid position, Enter again: ");
        scanf("%d", &pos);
    }
    printf("enter the character \n");
    ch = getch();
    b[0] = 'd';
    b[1] = 'l';
    b[2] = 'e';
    b[3] = 's';
    b[4] = 't';
    b[5] = 'x';
    j=6;
```

```
while (i<n)
{
    if (i==pos-1)
    {
      b[j]='d';
      b[j+1]='l';
      b[j+2]='e';
      b[j+3]=ch;
      b[j+4]='d';
      b[j+5]='l';
      b[j+6]='e';
      j=j+7;
    }
    if (a[i]=='d' && a[i+1]=='l' && a[i+2]=='e')
    {
      b[j]='d';
      b[j+1]='l';
      b[j+2]='e';
      j=j+3;
    }
    b[j]=a[i];
    i++;
    j++;
  }
  b[j]='d';
  b[j+1]='l';
  b[j+2]='e';
  b[j+3]='e';
  b[j+4]='t';
  b[j+5]='x';
  b[j+6]='\0';
  printf("\n frame after stuffing : \n");
  printf("%s", b);
  getch();
}
```

OUTPUT

Enter string : haiarchana

Enter position : 4

Enter the character K

Frame after stuffing :

dlestxhaidlekdlearchanadleetx

LAB PROGRAM # 2

Q. Write a C/C++ program to implement Distance Vector Routing algorithm.

```
#include <stdio.h>
struct node
{
    unsigned dist [20];
    unsigned from [20];
} rt [10];
int main ()
{
    int costmat [20] [20];
    int nodes, i, j, k, count = 0
    printf ("\n Enter the no. of nodes : ");
    scanf ("%d", &nodes);
    printf ("\n Enter the cost matrix : \n");
    for (i=0; i < nodes; i++)
    {
        for (j=0; j < nodes; j++)
        {
            scanf ("%d", &costmat [i] [j]);
        }
        costmat [i] [i] = 0
        rt [i]. dist [j] = costmat [i] [j];
        rt [i]. from [j] = j;
    }
}
do
{
    count = 0;
    for (i=0; i < nodes; i++)
    {
        for (j=0; j < nodes; j++)
        {
            for (k=0; k < nodes; k++)
            {
                if (rt [i]. dist [j] > costmat [i] [k] + rt [k]. dist [j];
                {
                    rt [i]. dist [j] = rt [i]. dist [k] + rt [k] . dist [j];
                    rt [i]. from [j] = k;
                    count++;
                }
            }
        }
    }
}
```

```
3 while (count!=0);

  for (i=0; i<nodes; i++)
  {
   printf("\n\n For router %d\n", i+1);

    for (j=0; j<nodes; j++)
    {
     printf("\t\n node %d via %d Distance %d", j+1,
            rt[i].from[j]+1, rt[i].dist[j]);
    }
  }
  printf("\n\n");
  getch();
}
```

OUTPUT

Enter the cost matrix :

0 2 7
2 0 1
7 1 0

For router 1

node 1 via 1 Distance 0

node 2 via 2 Distance 2

node 3 via 3 Distance 3

For router 2

node 1 via 1 Distance 2

node 2 via 2 Distance 0

node 3 via 3 Distance 1

For router 3

node 1 via 1 Distance 3

node 2 via 2 Distance 1

node 3 via 3 Distance 0

LAB PROGRAM #3

Q Write a C/C++ program to implement stop and wait Flow control Protocol.

```
#include <iostream>
#include <time.h>
#include <cstdlib.h>
#include <ctime>
#include <unistd.h>
using namespace std;
class timer
{ private:
   unsigned long begTime;
  public:
   void start()
    { begTime = clock();
    }
  unsigned long elapsedTime()
  { return ((unsigned long) clock() - begTime) / CLOCKS_PER_SEC;
  }
  bool isTimeout(unsigned long seconds)
   { return seconds >= elapsedTime();
   }
 };
int main()
{  int frames[] = {1,2,3,4,5,6,7,8,9,10};
   unsigned long seconds=5;
   srand(time(NULL));
   timer t;
   cout<<"Sender has to send frames:";
   for(int i=0; i<10; i++)
      { cout<<frames[i]<<" ";
      }
```

```
cout << endl << "Sender \t \t \t \t \t Receiver " << endl;
do
{
   bool timeout = false;
   cout << " Sending Frame : " << frames [count];
   cout.flush();
   cout << " \t \t ";
   t.start();
   if (rand() % 2)
   {
      int to = 24600 + rand() % (64000 - 24600) + 1;
      for (int i = 0; i < 64000 ; i++)
      {
         for (int j = 0; j < to; j++)
         {
         }
      }
   }
   if (t.elapsedTime() <= seconds)
   { cout << " Received Frame : " << frames [count] << " ";
      if (delay)
      {
         cout << " Duplicate ";
         delay = false;
      }
      cout << endl;
      count ++;
   }
   else
   {
      cout << " --- " << endl;
      cout << " Timeout " << endl;
      timeout = true;
   }
   t.start();
   if (rand() % 2 || !timeout)
   { int to = 24600 + rand() % (64000 - 24600) + 1;
      for (int i=0; i < 64000 ; i++)
      {
```

```
{
  for (int j = 0; j < 10 ; j++)
   {
    }
  }
  if (t.elapsedTime() > seconds)
   {
     cout << "Delayed Ack" << endl;
     count--;
     delay = true;
   }
   else if (!timeout)
    cout << "Acknowledgement : " << frames[count]-1 << endl;
 }
 } while (count != 10);
 return 0;
}
```

OUTPUT

Sender has to send frames : 1 2 3 4 5 6 7 8 9 10

Sending Frame : 1 Received Frame 1

Acknowledge : 1

Sending Frame : 2 ---

Time out

Sending Frame : 2 Received Frame : 2

Acknowledge : 2

Sending Frame : 3 Received Frame : 3

Acknowledge : 3

Sending Frame : 4 Received Frame : 4

Acknowledge : 4

Sending Frame : 5 Received Frame : 5

Acknowledge : 5

Sending Frame : 6 Received Frame : 6

Acknowledge : 6

Sending Frame : 7 Received Frame : 7

Delayed Ack

Sending Frame : 7 Received Frame : 7 Duplicate

Delayed Ack

Sending Frame : 7 --

Time out

Sending Frame : 7 Received Frame : 7 Duplicate

Acknowledge : 7

Sending Frame : 8 ---

Time out

Delayed Ack

OUTPUT

Sending Frame : 7 Received Frame : 7

Acknowledge : 7

Sending Frame : 8 Received Frame : 8

Acknowledgement : 8

Sending Frame : 9 Received Frame : 9

Delayed Ack

sending Frame : 9 ---

Timeout

sending Frame : 9 Received Frame : 9 Duplicate

Delayed Ack

Sending Frame : 9 Received Frame : 9 Duplicate

Acknowledgement : 9

Sending Frame : 10 ---

Timeout

Sending Frame : 10 Received Frame : 10

Acknowledgement : 10

LAB PROGRAM #4

Q Write a c/c++ Program for ERROR detecting code using CRC-CCITT (16bit).

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int data[60], div[16], rem[16];
    int datalen, divlen, i, j, k;
    int ch;
    clrscr();
    printf("Enter the data : ");
    i=0;
    while((ch = fgetc(stdin)) != '\n')
    {
        if(ch == '1')
            data[i] = 1;
        else
            data[i] = 0;
        i++;
    }
    datalen = i;
    printf("\n Enter the divisor : ");
    i=0;
    while((ch = fgetc(stdin)) != '\n')
    {
        if(ch == '1')
            div[i] = 1;
        else
            div[i] = 0;
        i++;
    }
    divlen = i;
    for(i= datalen; i< datalen + divlen - 1; i++)
    {   data[i] = 0;
    }
    datalen = datalen + divlen - 1;
```

```
for (i=0; i<divlen; i++)
  rem [i] = data [i];
  k = divlen - 1;
  while (k < datalen)
  if (rem [0] == 1)
   {
      for (i=0; i<divlen; i++)
        rem [i] = rem [i] ^ div[i];
   }
   else
   {
      if (k== datalen-1)
      break;
      for (i=0; i<divlen-1; i++)
      { rem [i] = rem [i+1];
        printf (" %d", rem [i]);
      }
     rem [i] = data [++k];
     printf ("%d \n", rem [i]);
   }
   j=1;
   for (i= datalen - divlen +1 ; i< datalen; i++)
    {
      data [i] = rem [j++];
    }
    printf("\n The data to be sent is \n");
    for (i=0; i< datalen ; i++)
     printf ("%d", data [i]);
     getch ();
     return 0;
}
```

OUTPUT

Enter the data : 10101111

Enter the divisor : 1011

```
0011
 0111
 1111
 1001
 0100
 1000
 0110
```

The data to be sent is 10101111110

LAB PROGRAM # 5

Q write a c/c++ program for congestion control using Leaky Bucket Algorithm.

```
#include <stdlib.h>
#include <unistd.h>
#define NOF_PACKETS 10
int rand (int a)
{
    int rn = (random() % 10) % a;
    return rn == 0 ? 1 : rn;
}
int main()
{
  int packet_sz [NOF_PACKETS], i, clk, b_size, o_rate, p_sz_rm=0,
      p_sz, p_time, op;
  for (i=0; i< NOF_PACKETS, ++i)
     packet_sz [i] = rand (6) *10;
  for (i=0; i<NOF_PACKETS, ++i)
   printf ("\n packet [%d] : %d bytes \t", i, packet_sz [i]);
  printf("\n Enter the output rate: ");
  scanf ("%d", &o_rate);
  printf(" Enter bucket size : ");
  scanf ("%d", &b_size);
  for (i=0; i< NOF_PACKETS; ++i)
   { if ((packet_sz [i] + p_sz_rm) > b_size)
       if (packet_sz [i] > b_size)
         {printf ("\n\n Incoming packet size (%d bytes) is
             Greater than bucket capacity (%d bytes) - PACKET
             REJECTED", packet_sz [i], b_size);
         }
       else
```

```
printf ("\n\n Bucket capacity exceeded - PACKET REJECTED!!");
else
{
    p_sz_rm += packet_sz[i];
    printf("\n\n Incoming Packet size: %d", packet_sz[i]);
    printf ("\n Bytes remaining to Transmit : %d", p_sz_rm);
    p_time = rand (4) * 10;
    printf ("\n Time left for transmission : %d units", p_time);
    for (clk=10; clk <= p_time; clk += 10)
    {
        sleep (1);
        if( p_sz_rm)
        {
            if (p_sz_rm <= o_rate)
                op = p_sz_rm, p_sz_rm=0;
            else
                op = o_rate , p_sz_rm -= o_rate;
            printf (" \n Packet of size %d Transmitted ", op);
            printf(" ---- Bytes Remaining to Transmit: %d",
                        p_sz_rm);
        }
        else
        {
            printf (" \n Time left for transmission : %d units",
                        p_time-clk);
            printf("\n No packets to transmit!!");
        }
    }
}
```

OUTPUT

Enter The Bucket Size 5

Enter the operation Rate 2

Enter The No. of Seconds You want to Stimulate 3

Enter the size of the Packet Entering At 1 sec 5

Enter the size of the Packet Entering At 1 sec 4

Enter the size of the Packet Entering At 1 sec 3

~~Second Packet Rec~~

Second	Packet Received	Packet Sent	Packet left	Packet Dropped
1	5	2	3	0
2	4	2	3	2
3	3	2	3	1
4	0	2	1	0
5	0	1	0	1