WINTER – SEMESTER
Course Code: MCSE505P
Course-Title: – Computer Network Lab Component
DIGITAL ASSIGNMENT – IV

(LAB) Slot- L35+L36
Faculty: - SRIMATHI C (SCOPE)
Name: Nidhi Singh
Reg. No: 22MAI0015

Implementation of RIP Routing in Cisco for Connecting Two Routers

Routing Information Protocol (RIP) is an active routing protocol that operates hop count as a routing metric to find the most suitable route between the source and the destination network. It is a distance-vector routing protocol that has an AD value of 120 and works on the Network layer of the OSI model.

Steps to Configure and Verify Two Router Connections in Cisco Packet Tracer:

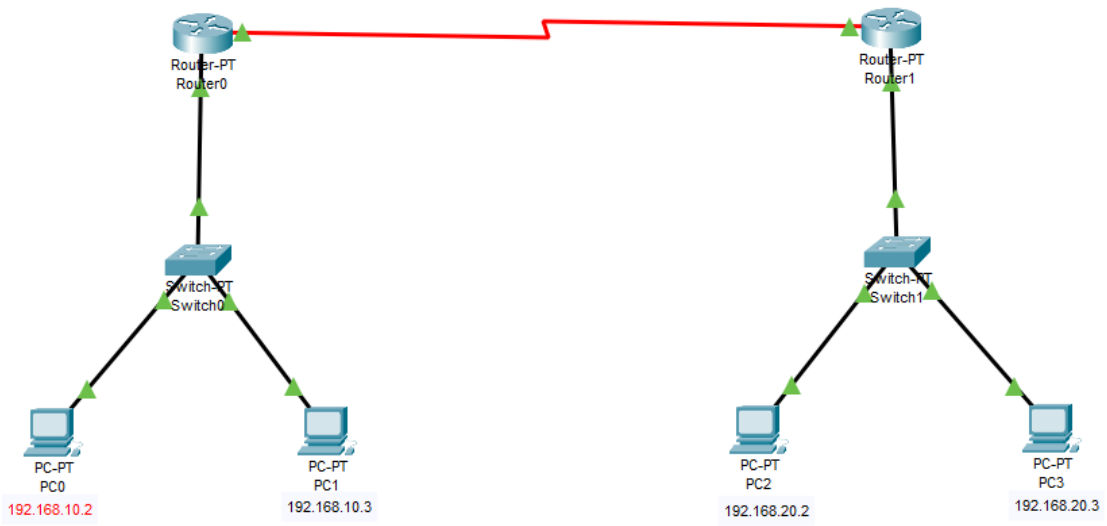
Step 1:
First, open the Cisco packet tracer desktop and select the devices given below:

S.NO	Device	Model Name	Qty.
1.	pc	pc	4
2.	switch	PT-Switch	2
3.	router	PT-Router	2

IP Addressing Table :-

S.NO	Device	IPv4 Address	Subnet-Mask	Default-Gateway
1.	pc0	192.168.10.2	255.255.255.0	192.168.10.1
2.	pc1	192.168.10.3	255.255.255.0	192.168.10.1
3.	pc2	192.168.20.2	255.255.255.0	192.168.20.1
4.	pc3	192.168.20.3	255.255.255.0	192.168.20.1

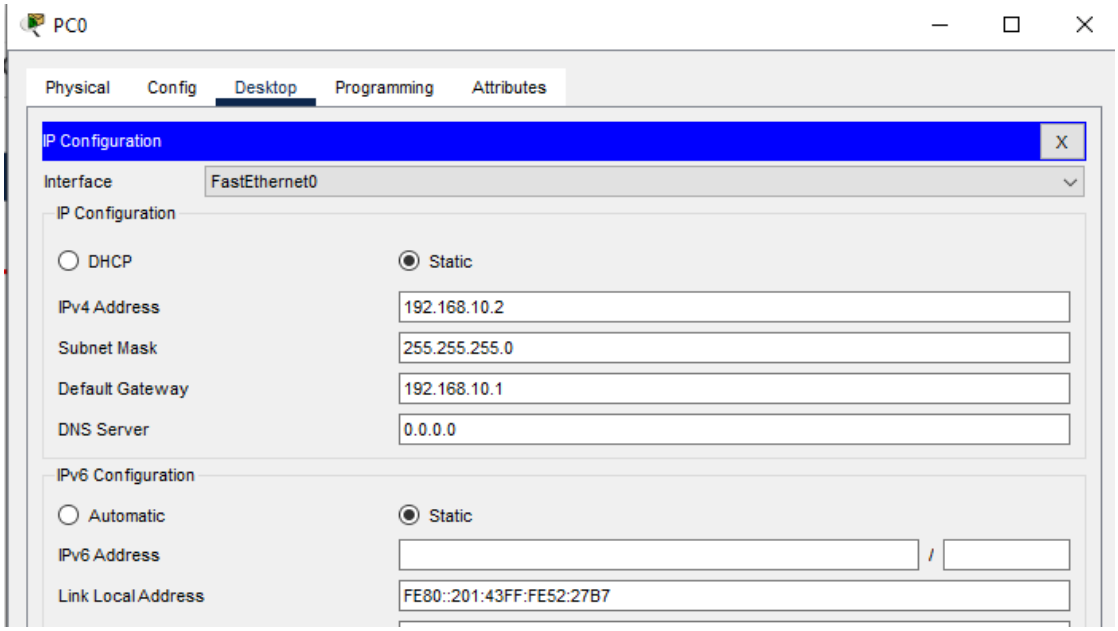
Use an Automatic connecting cable to connect the devices with others:-

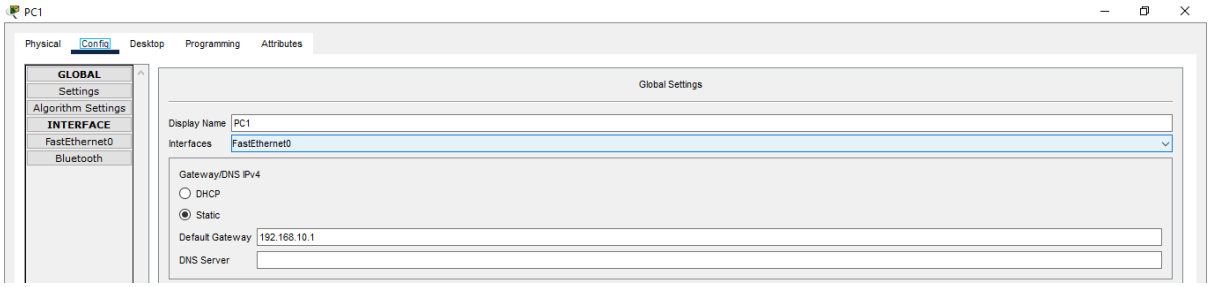


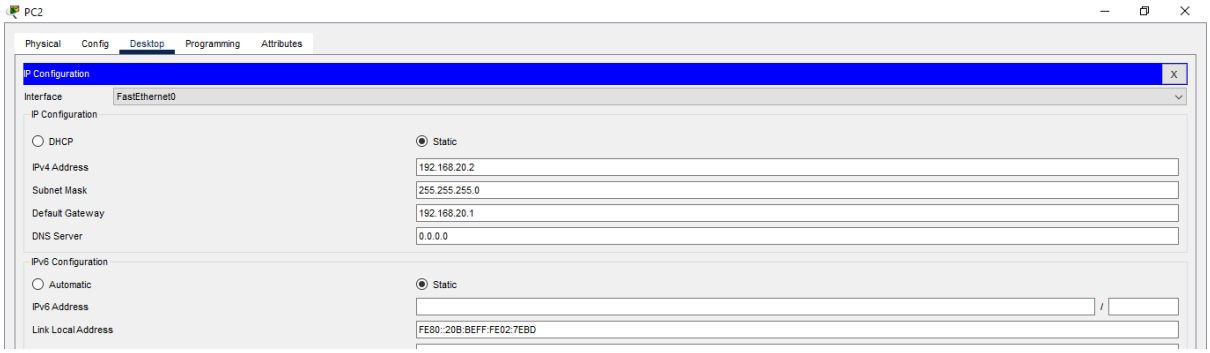
Step 2:

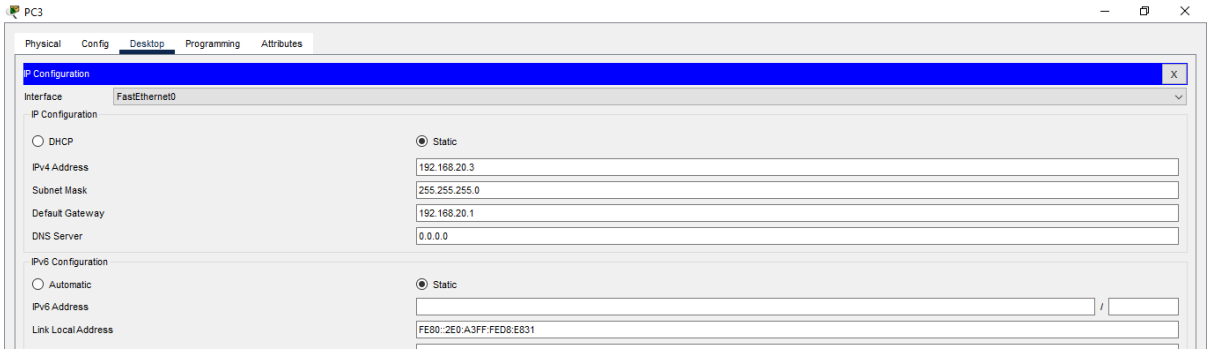
Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.







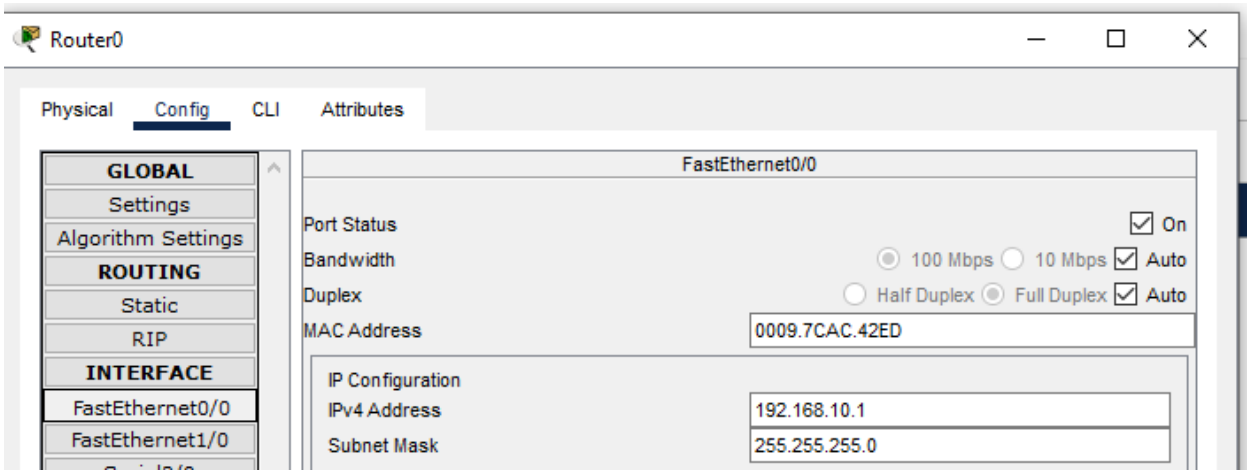


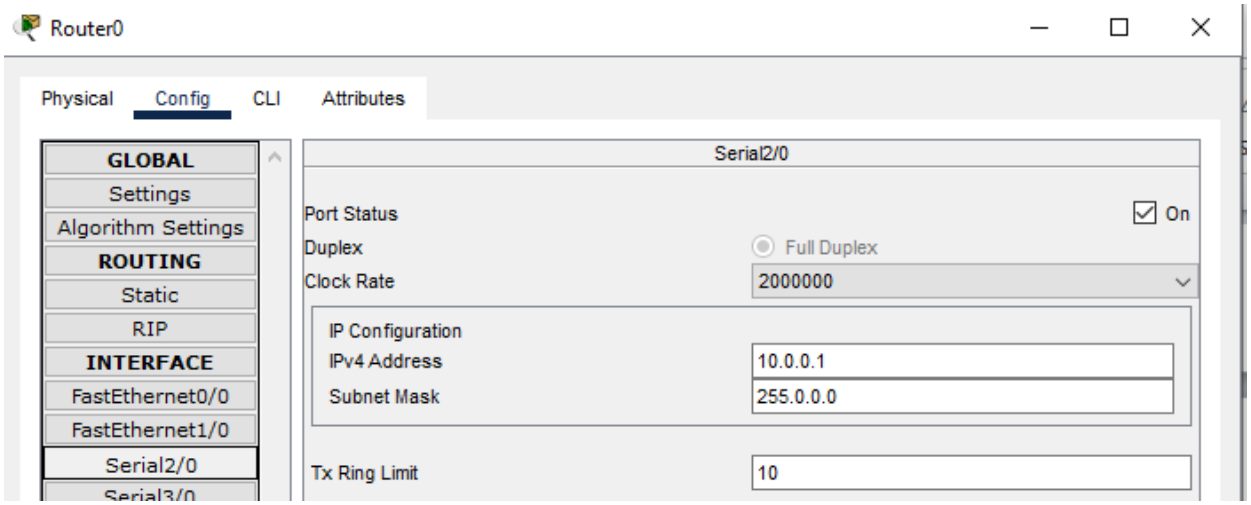
Step 3: Configure router with IP address and subnet mask.

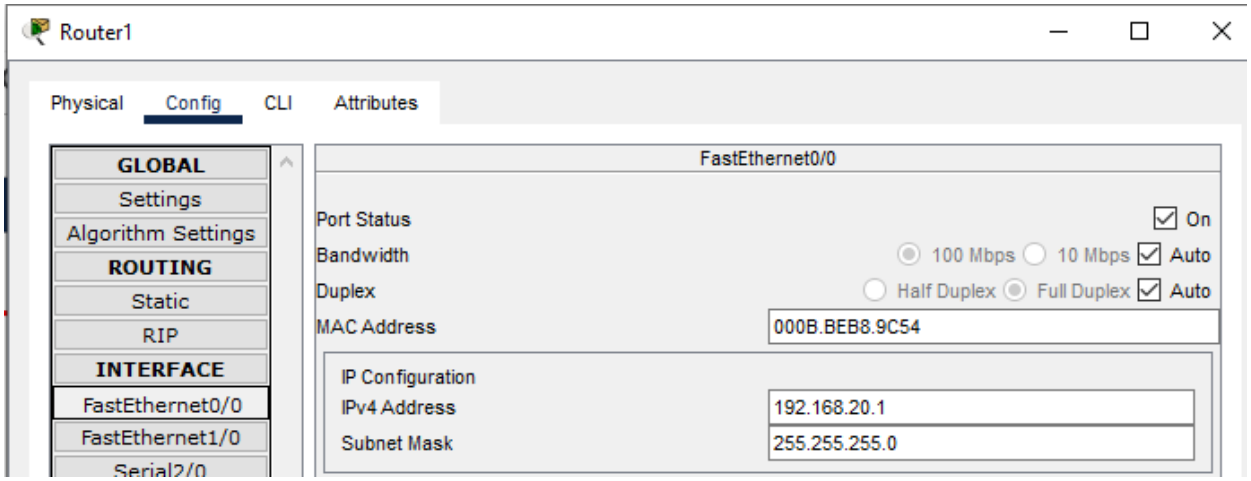
IP Addressing Table Router:-

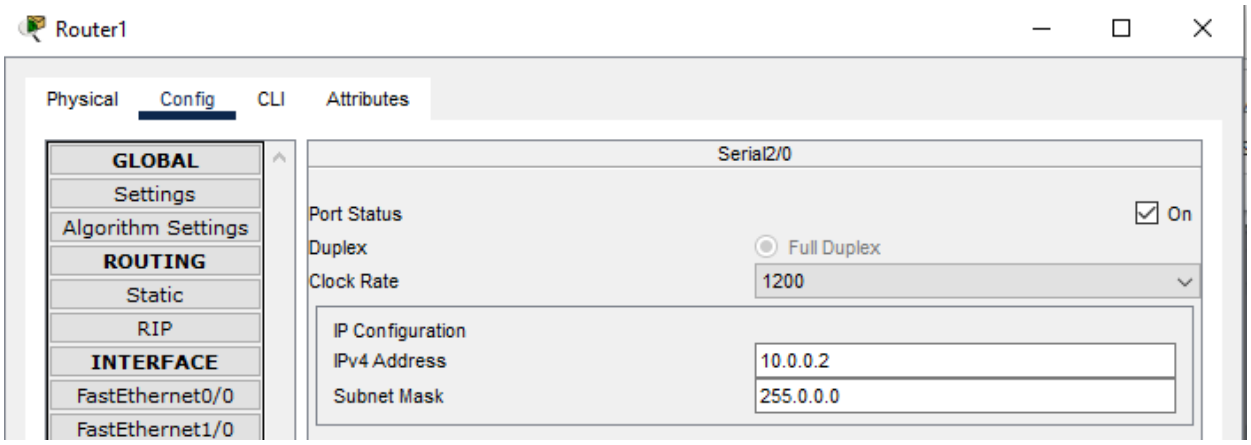
S.NO	Device	Interface	IPv4 Address	Subnet Mask
1.	router0	FastEthernet0/0	192.168.10.1	255.255.255.0
		Serial 2/0	10.0.0.1	255.0.0.0
2.	router1	FastEthernet0/0	192.168.20.1	255.255.255.0
		Serial 2/0	10.0.0.2	255.0.0.0

- To assign an IP address in router0, click on router0.
- Then, go to config and then Interfaces.
- Make sure to turn on the ports
- Then, configure the IP address in FastEthernet and serial ports according to IP addressing Table.
- Fill IPv4 address and subnet mask.







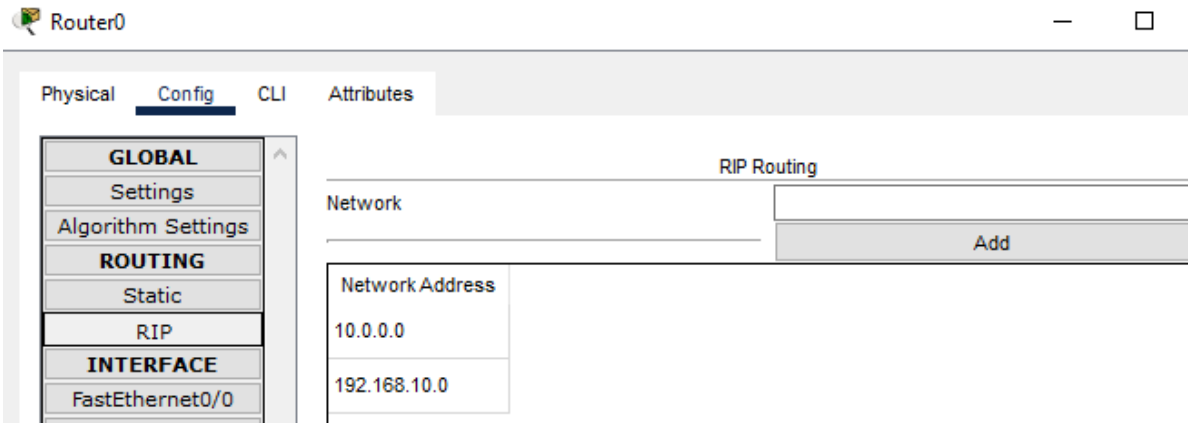


Step 4: After configuring all of the devices we need to assign the routes to the routers.
To assign RIP routes to the particular router:

- First, click on router0 then Go to CLI.
- Then type the commands and IP information given below.

CLI command: network <network id>

RIP Routes for Router0 are given below:



Router(config)#router rip
Router(config-router)#network 192.168.10.0
Router(config-router)#network 10.0.0.0
Router(config-router)#
Router(config-router)#end
Router#configure terminal

```
Router(config)#router rip
Router(config-router)#network 192.168.10.0
Router(config-router)#network 10.0.0.0
Router(config-router)#
Router(config-router)#end
Router#configure terminal
```

RIP Routes for Router1 are given below:

```
                    ^
% Invalid input detected at '^' marker.

Router(config-if)#router rip
Router(config-router)#network 192.168.20.0
Router(config-router)#network 10.0.0.0
Router(config-router)#
```

Router(config-if)#router rip
Router(config-router)#network 192.168.20.0
Router(config-router)#network 10.0.0.0
Router(config-router)#
Router(config-router)#
Router(config-router)#end
Router#configure terminal

Step 5: Verifying the network by pinging the IP address of any PC.
we'll use the ping command to do so.

- First, click on PC0 then Go to the command prompt
- then type ping <IP address of targeted node>
- as we can see in the below image we are getting replies which means the connection is working very fine

Example : `ping 192.168.20.2`

PC0

Physical Config Desktop Programming Attributes

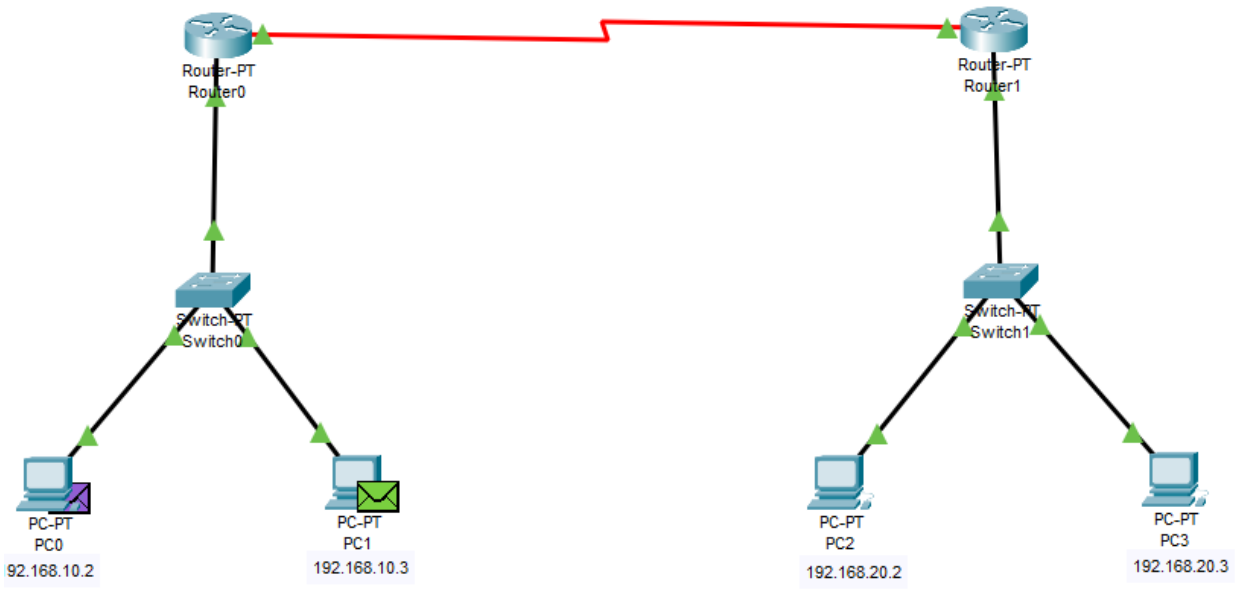
Command Prompt

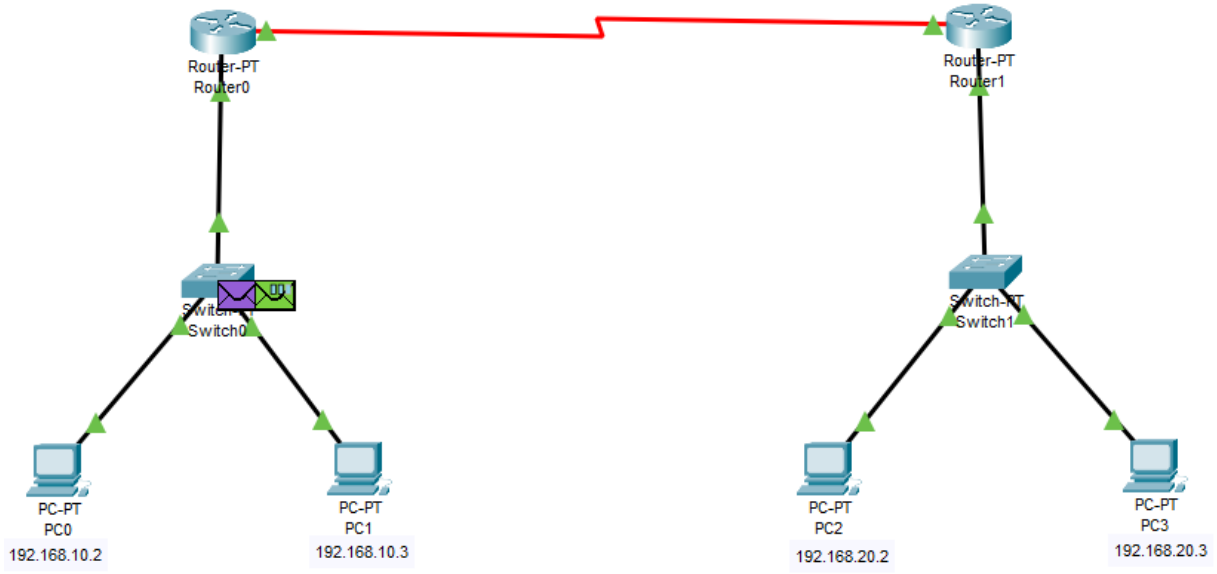
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.2: bytes=32 time=1ms TTL=126
Reply from 192.168.20.2: bytes=32 time=1ms TTL=126
Reply from 192.168.20.2: bytes=32 time=11ms TTL=126

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 11ms, Average = 4ms

C:\>
```

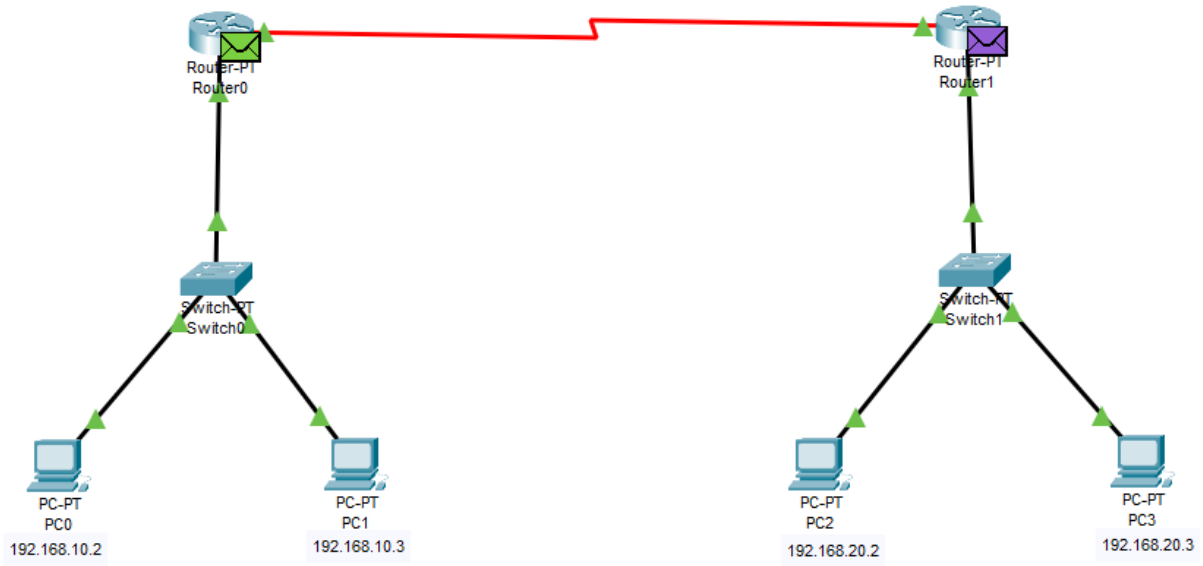
A simulation of the experiment is given below we are sending PDU from PC0 to PC2 and PC1 to PC3:

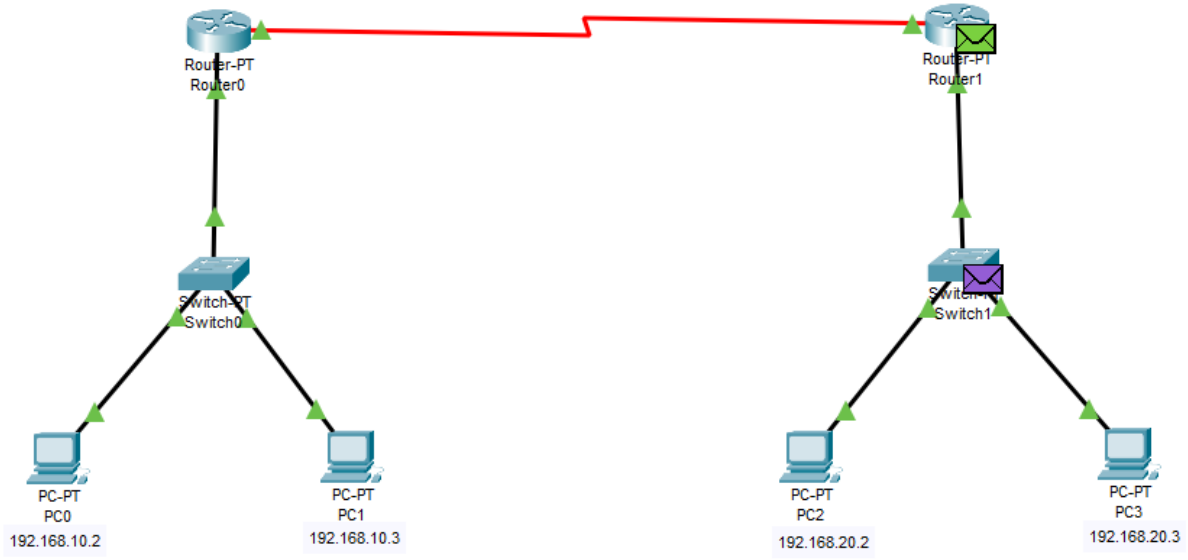
PDU List Window

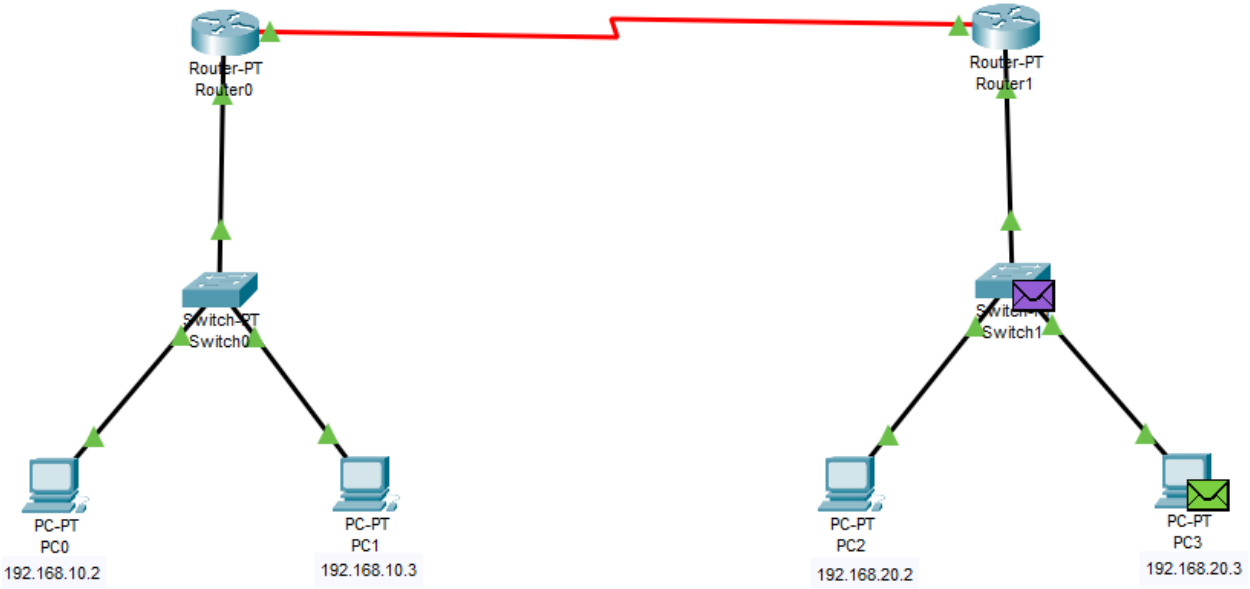
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	192.168.20.2	ICMP		100.000	N	0	(edit)	(delete)
	Successful	PC1	192.168.10.3	ICMP		100.000	N	1	(edit)	(delete)

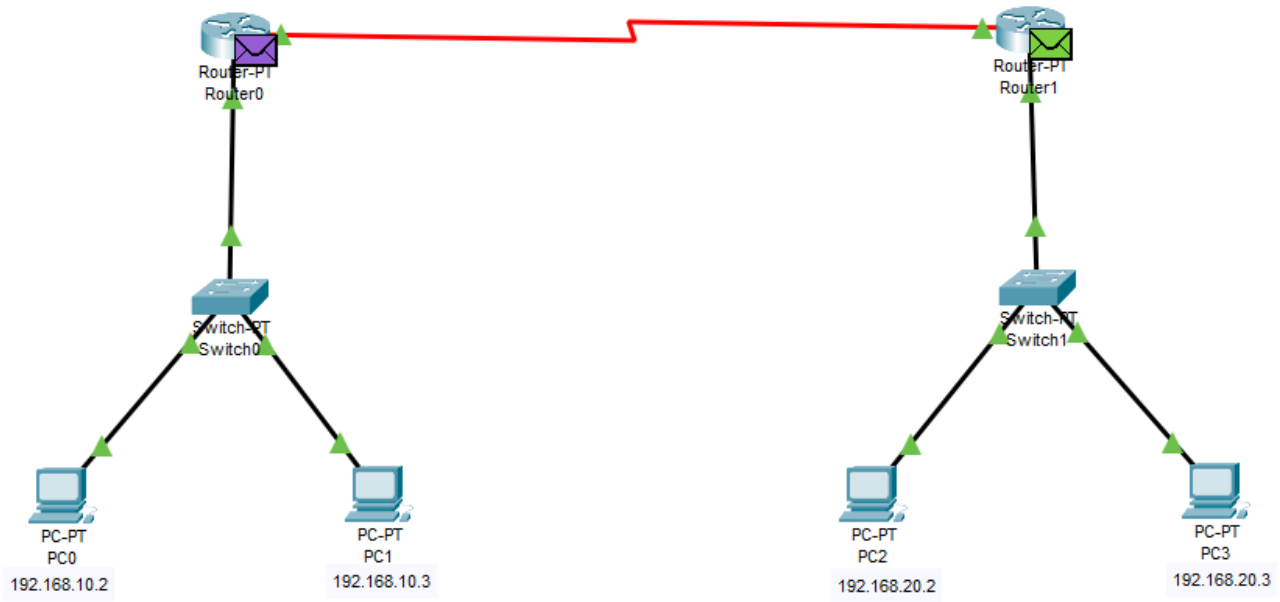

Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
92.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3

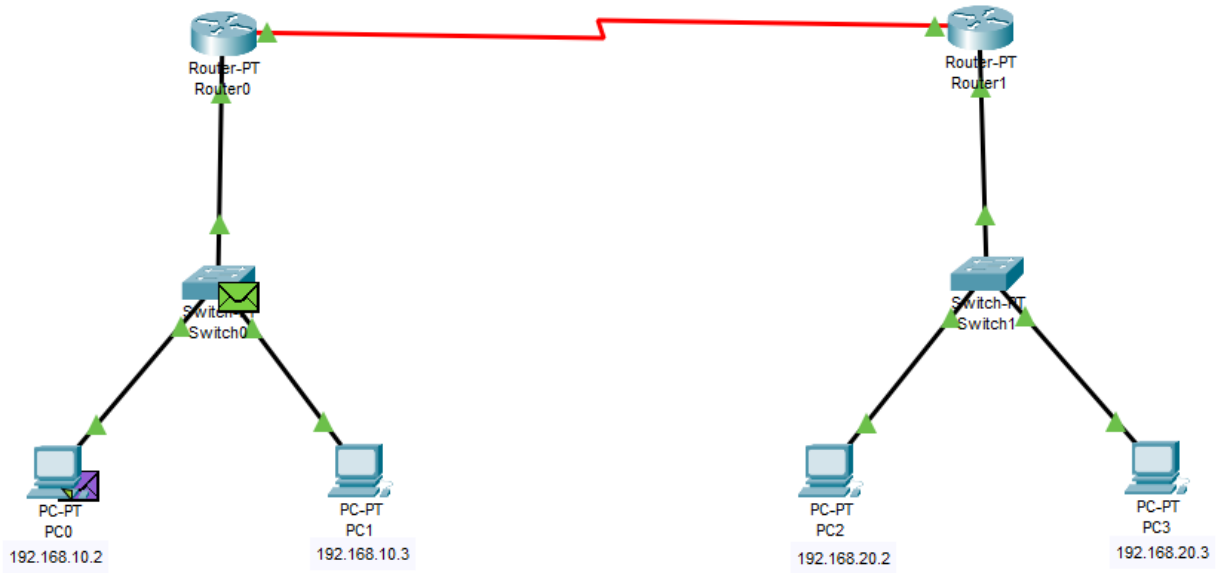

Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3

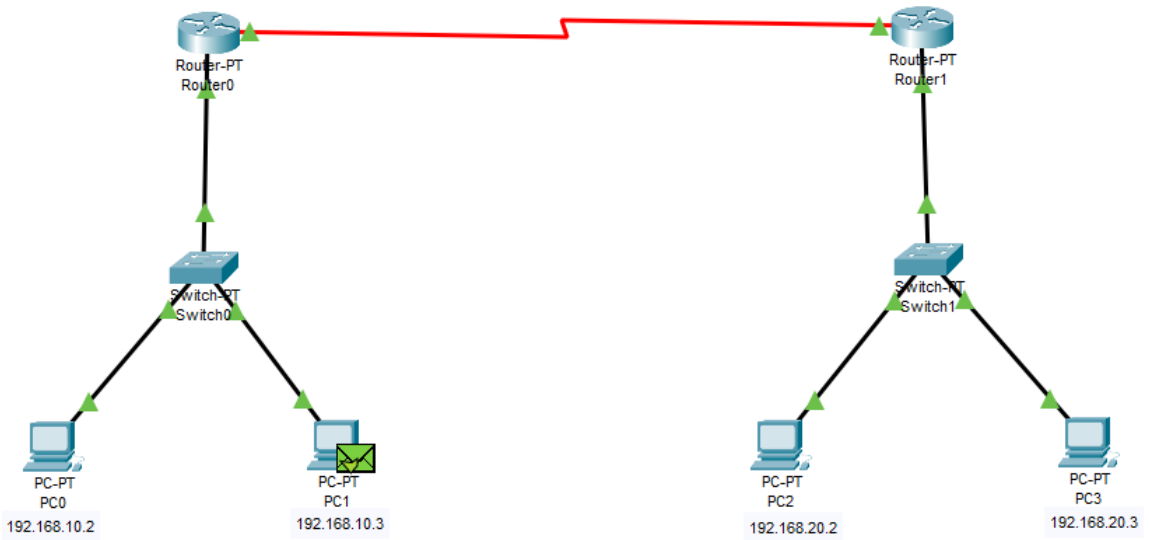

Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3


Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch1
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3


Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3


Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3


Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3

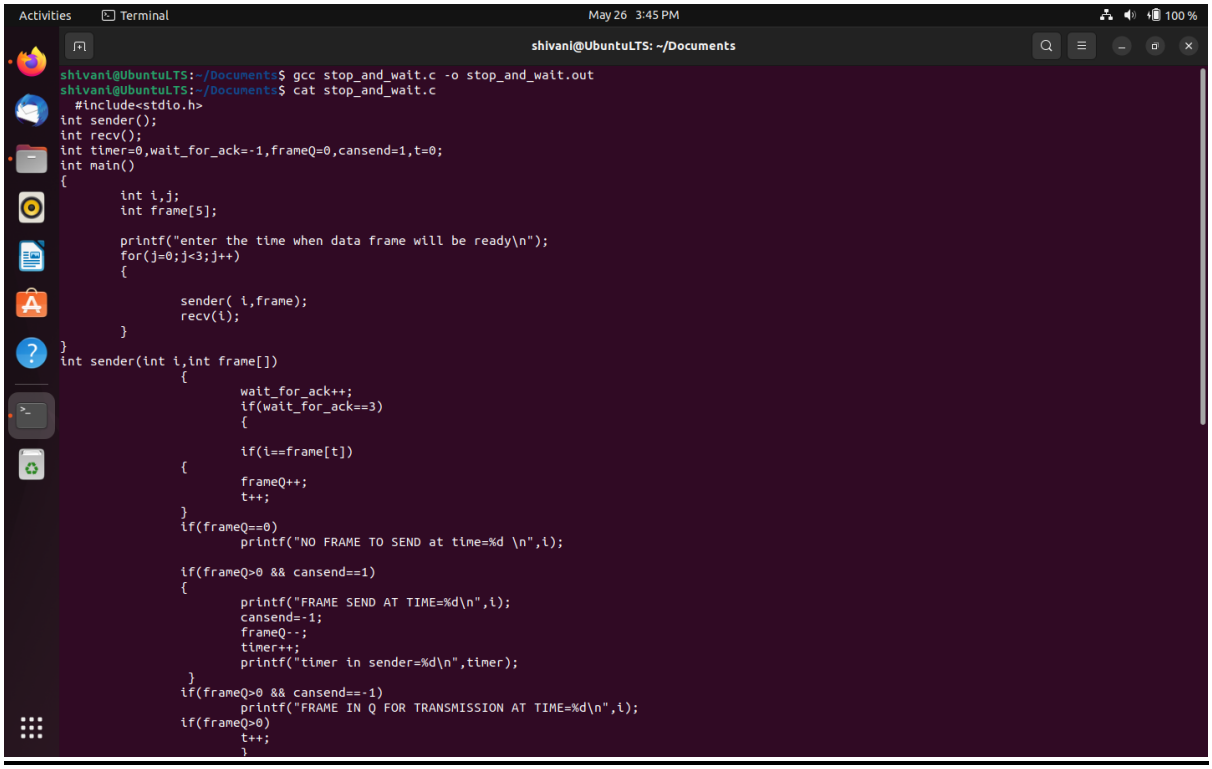

Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.10.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3

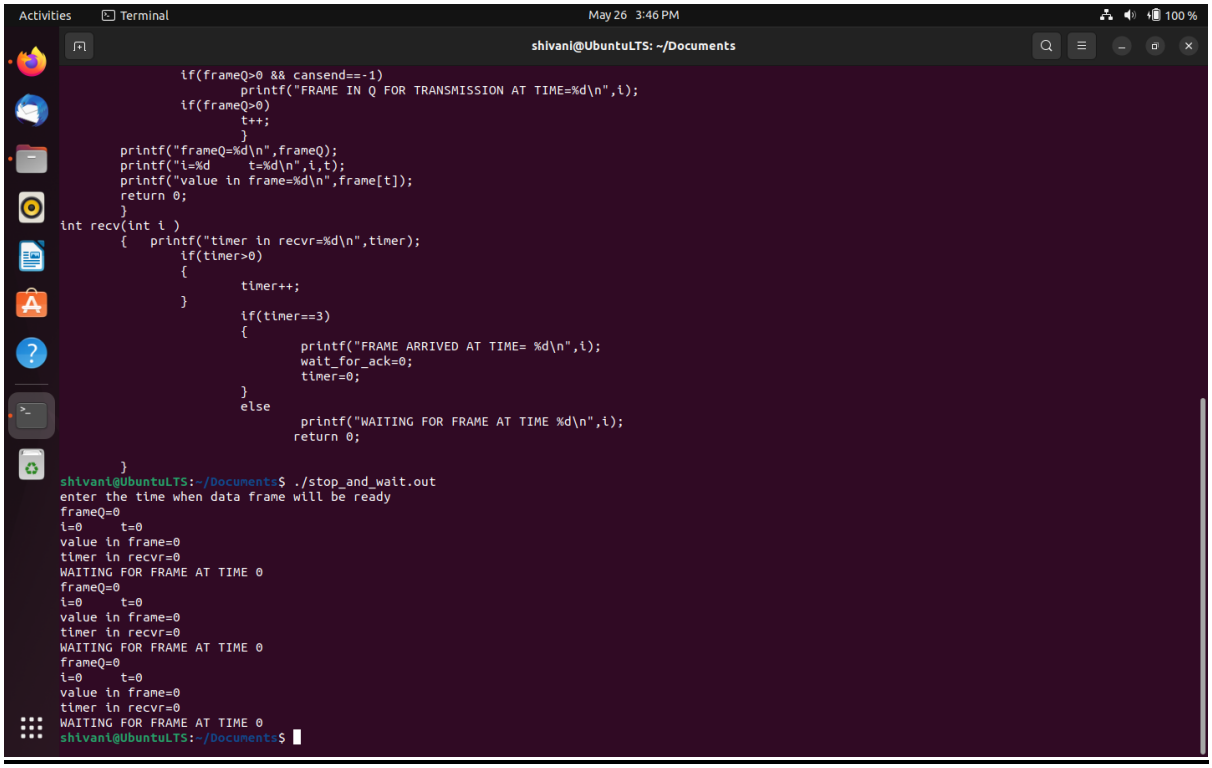
Simple stop and wait ARQ implementation in C language

```
#include<stdio.h>
int sender();
int recv();
int timer=0,wait_for_ack=-1,frameQ=0,cansend=1,t=0;
int main()
{
        int i,j;
        int frame[5];

        printf("enter the time when data frame will be ready\n");
        for(j=0;j<3;j++)
        {

                sender( i,frame);
                recv(i);
        }
}
int sender(int i,int frame[])
                {
                        wait_for_ack++;
                        if(wait_for_ack==3)
                        {

                        if(i==frame[t])
                {
                        frameQ++;
                        t++;
                }
                if(frameQ==0)
                        printf("NO FRAME TO SEND at time=%d \n",i);

                if(frameQ>0 && cansend==1)
                {
                        printf("FRAME SEND AT TIME=%d\n",i);
                        cansend=-1;
                        frameQ--;
                        timer++;
                        printf("timer in sender=%d\n",timer);
                 }
                if(frameQ>0 && cansend==-1)
                        printf("FRAME IN Q FOR TRANSMISSION AT TIME=%d\n",i);
                if(frameQ>0)
                        t++;
                        }
        printf("frameQ=%d\n",frameQ);
```

```
        printf("i=%d     t=%d\n",i,t);
        printf("value in frame=%d\n",frame[t]);
        return 0;
        }
int recv(int i )
        {   printf("timer in recvr=%d\n",timer);
                if(timer>0)
                {
                        timer++;
                }
                        if(timer==3)
                        {
                                printf("FRAME ARRIVED AT TIME= %d\n",i);
                                wait_for_ack=0;
                                timer=0;
                        }
                        else
                                printf("WAITING FOR FRAME AT TIME %d\n",i);
                               return 0;

        }
```

Output:-




Activities
Terminal
May 26 3:46 PM
100 %
shivani@UbuntuLTS: ~/Documents
if(frameQ>0 && cansend==-1)
printf("FRAME IN Q FOR TRANSMISSION AT TIME=%d\n",i);
if(frameQ>0)
t++;
}
printf("frameQ=%d\n",frameQ);
printf("i=%d t=%d\n",i,t);
printf("value in frame=%d\n",frame[t]);
return 0;
}
int recv(int i)
{ printf("timer in recvr=%d\n",timer);
if(timer>0)
{
timer++;
}
if(timer==3)
{
printf("FRAME ARRIVED AT TIME= %d\n",i);
wait_for_ack=0;
timer=0;
}
else
printf("WAITING FOR FRAME AT TIME %d\n",i);
return 0;
}
shivani@UbuntuLTS:~/Documents$ ./stop_and_wait.out
enter the time when data frame will be ready
frameQ=0
i=0 t=0
value in frame=0
timer in recvr=0
WAITING FOR FRAME AT TIME 0
frameQ=0
i=0 t=0
value in frame=0
timer in recvr=0
WAITING FOR FRAME AT TIME 0
frameQ=0
i=0 t=0
value in frame=0
timer in recvr=0
WAITING FOR FRAME AT TIME 0
shivani@UbuntuLTS:~/Documents$