

**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

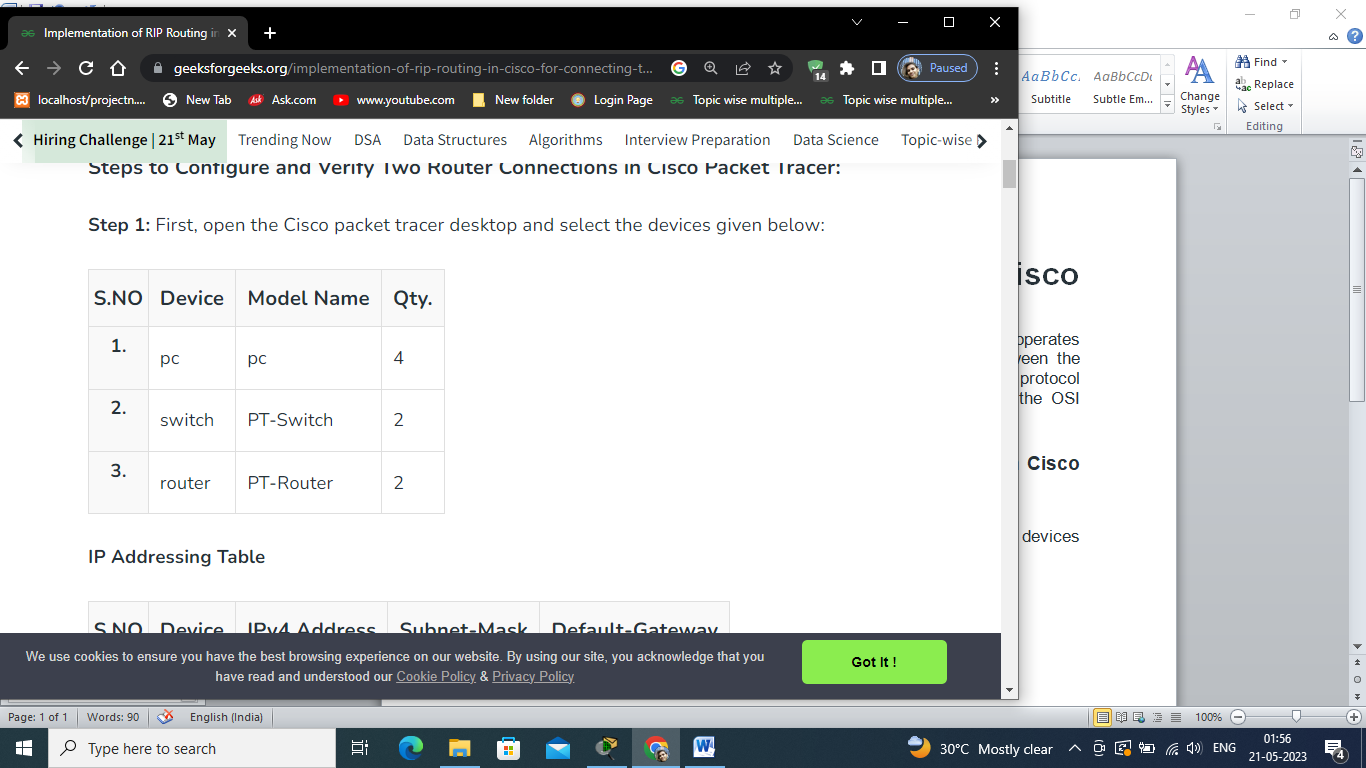
**Implementation of RIP Routing in Cisco for Connecting Two Routers**

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.

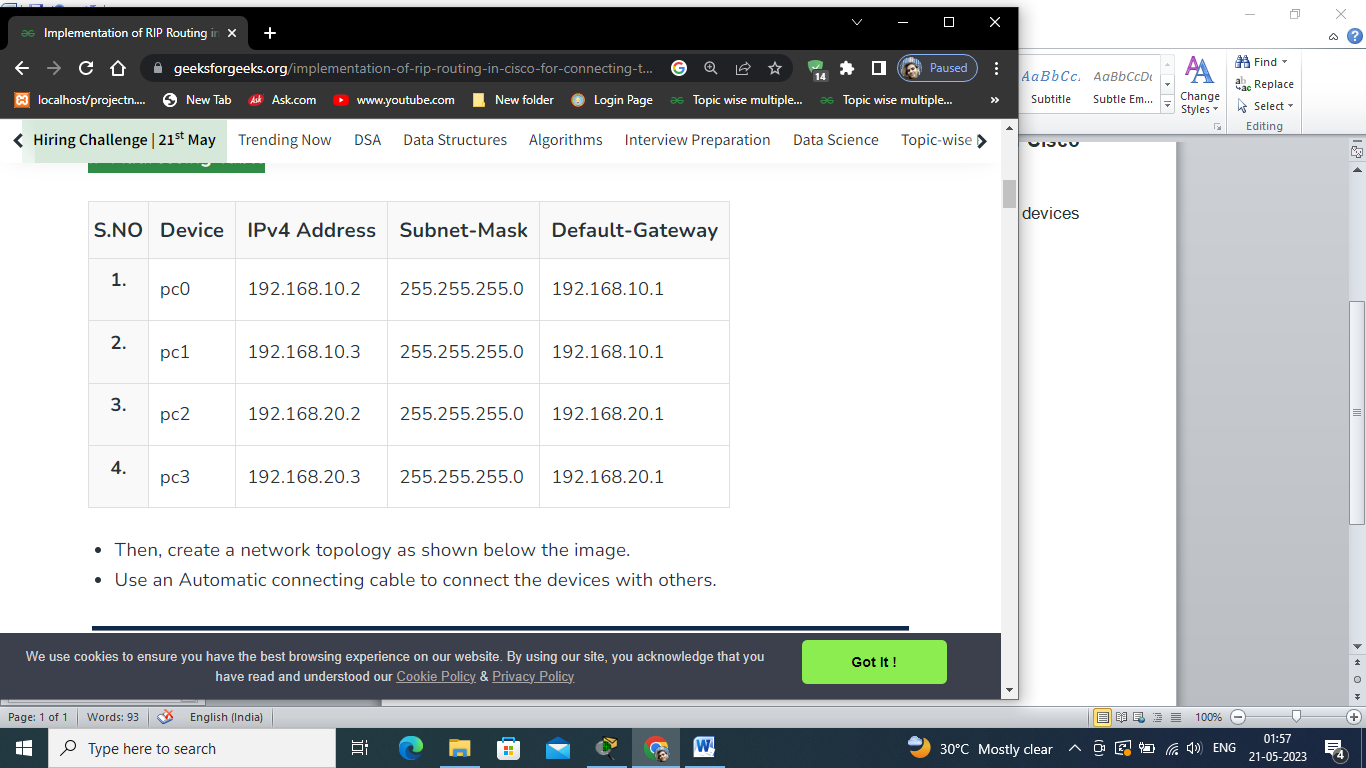
### Steps to Configure and Verify Two Router Connections in Cisco Packet Tracer:

**Step 1:**

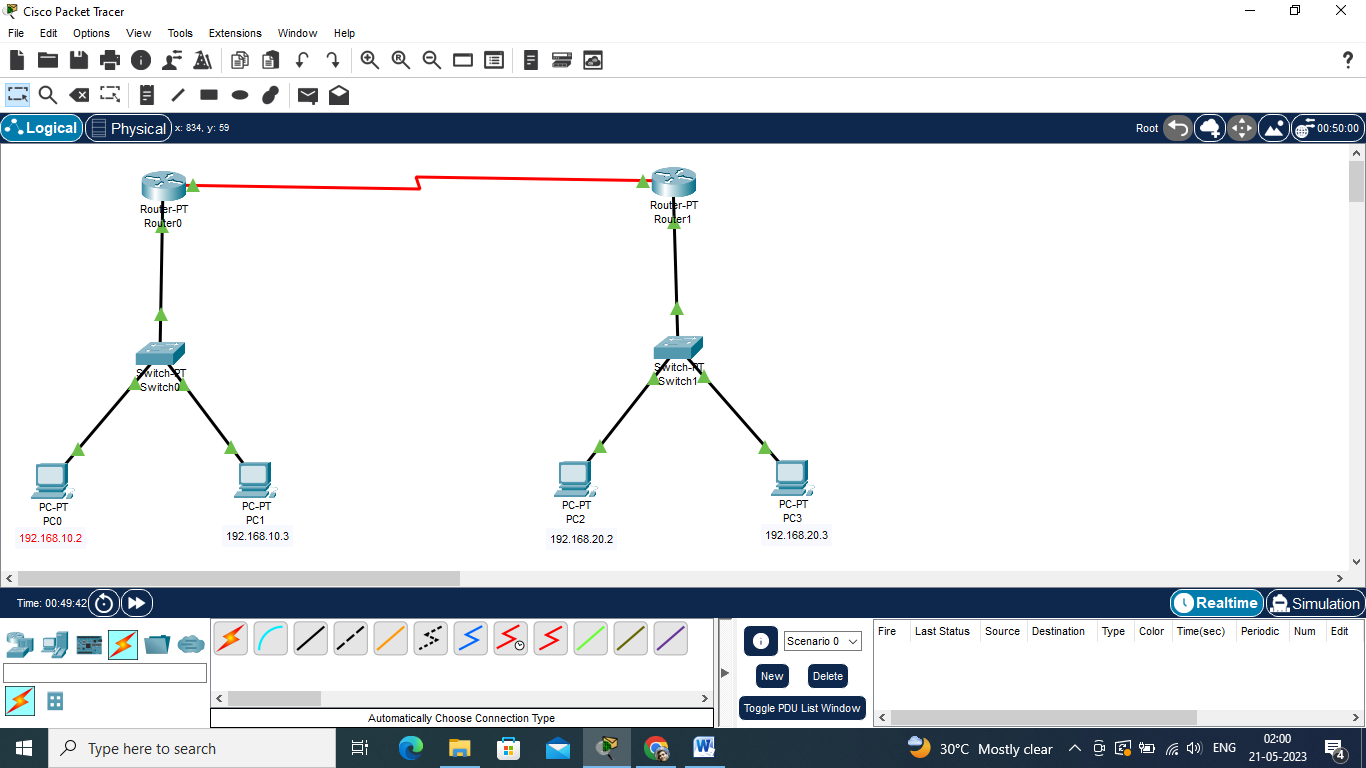
 First, open the Cisco packet tracer desktop and select the devices given below:



**IP Addressing Table :-**



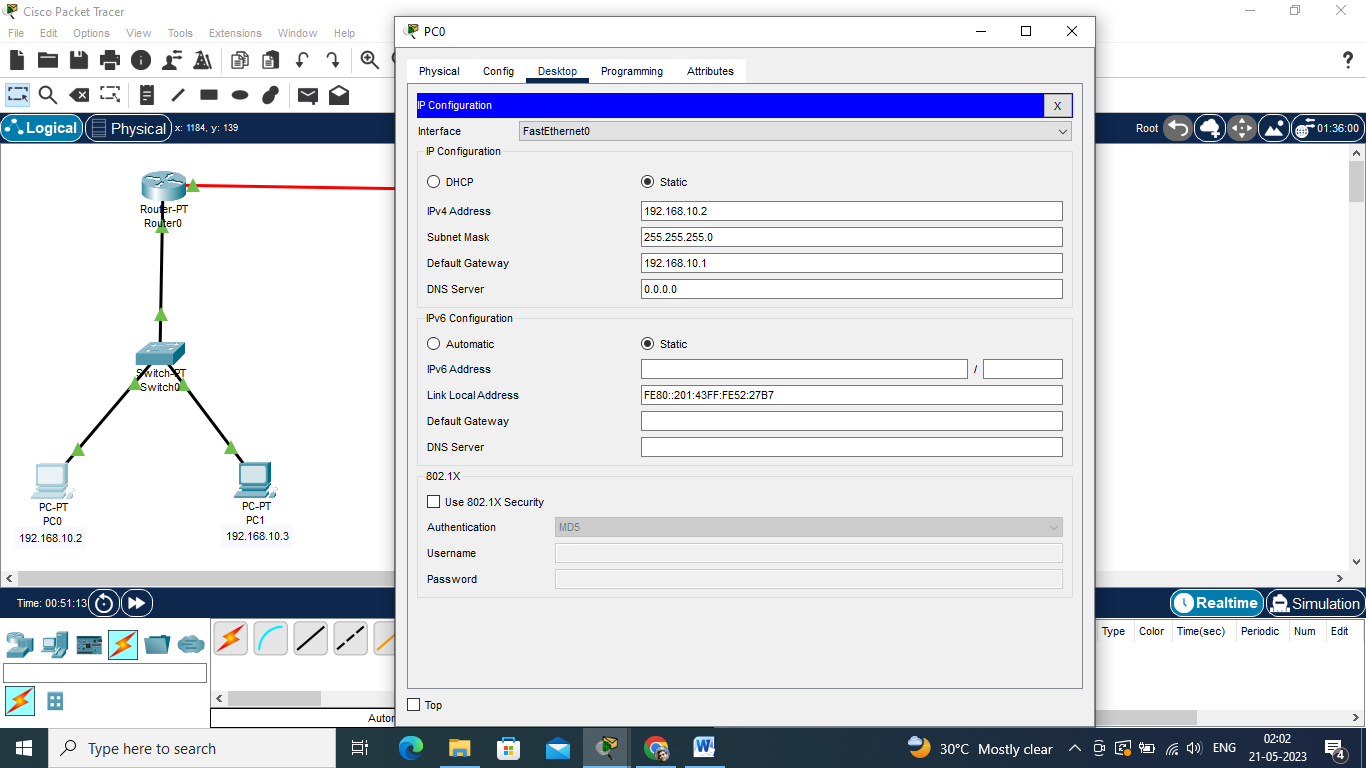
**Use an Automatic connecting cable to connect the devices with others:-**

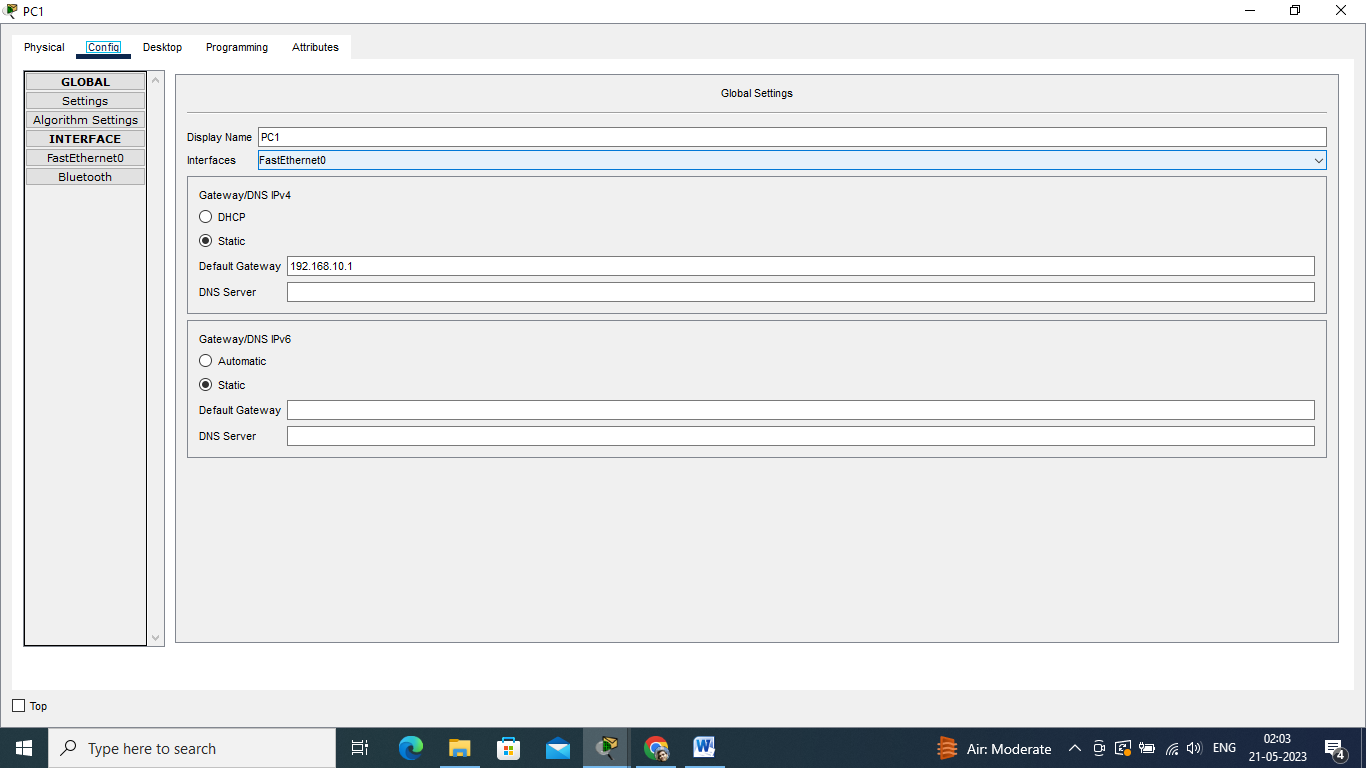


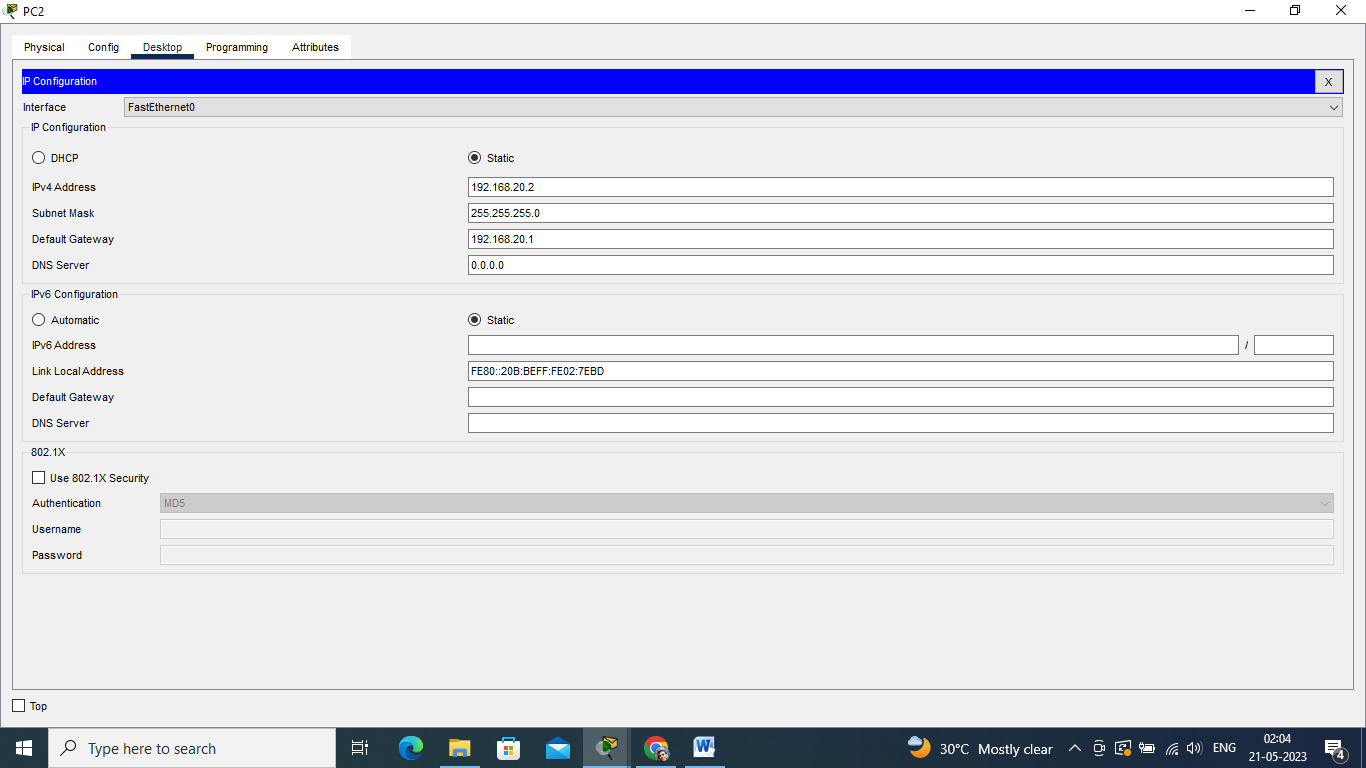
**Step 2:**

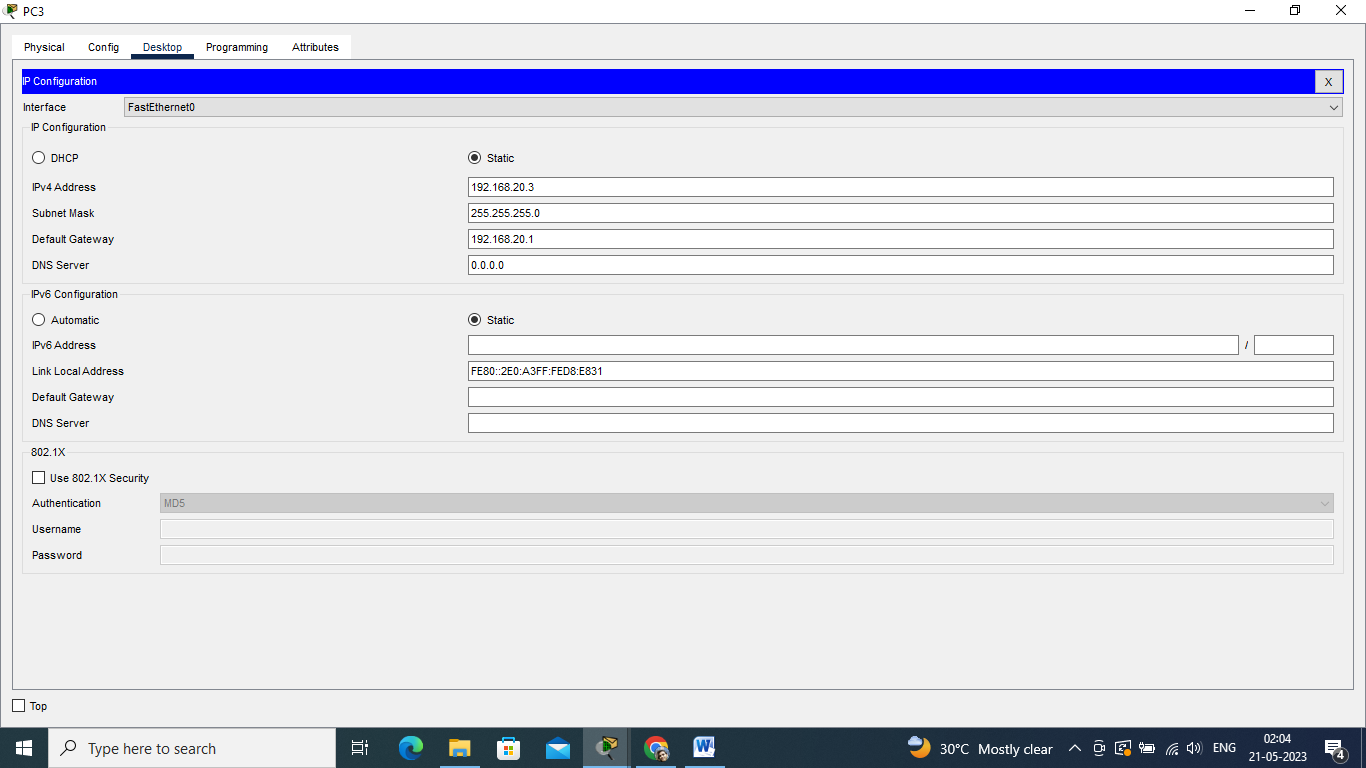
Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

* 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.



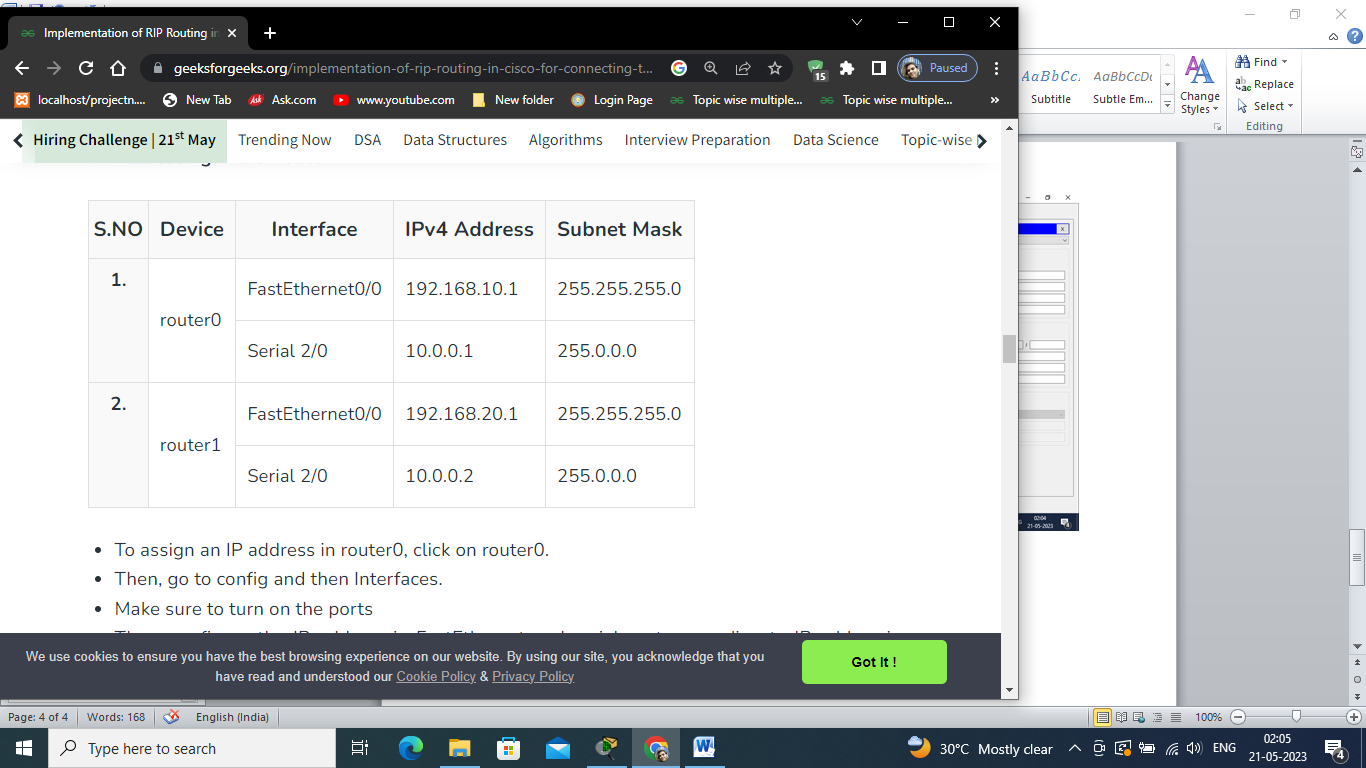




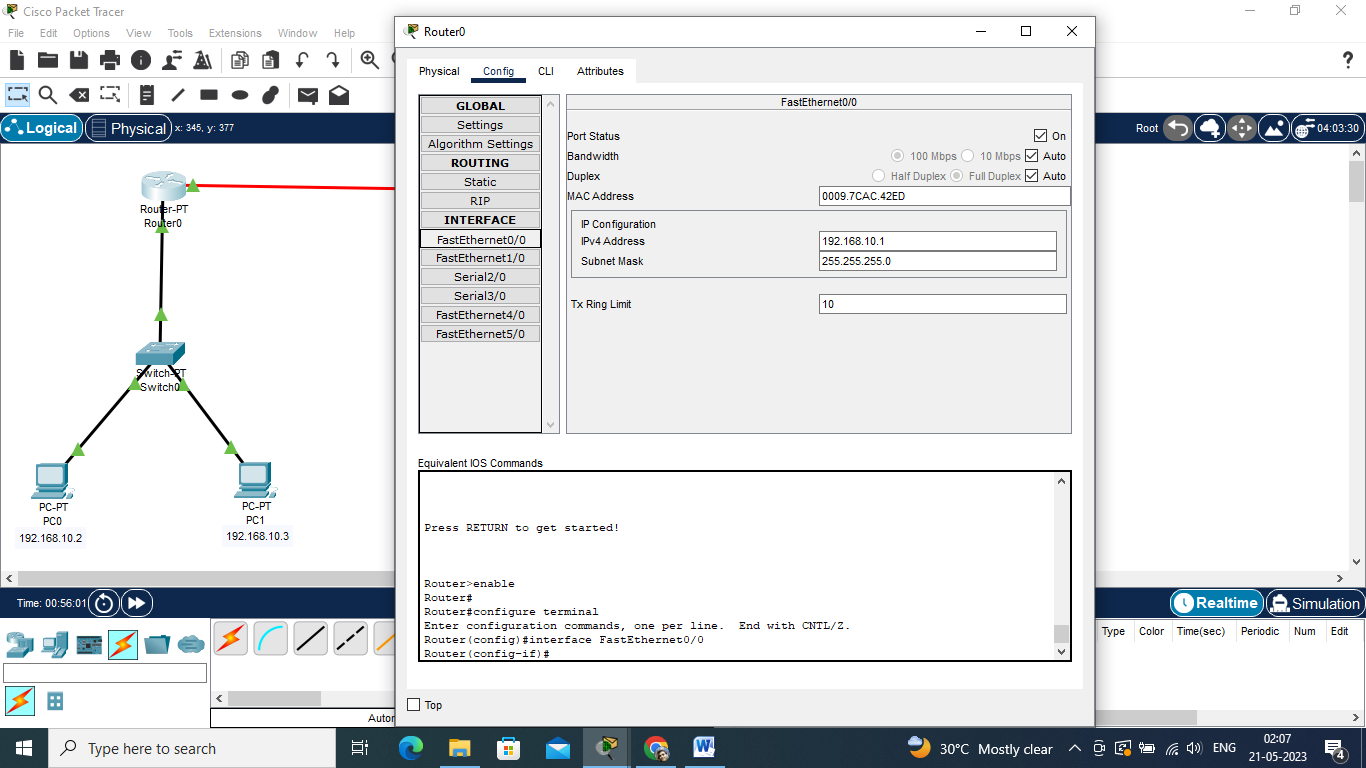


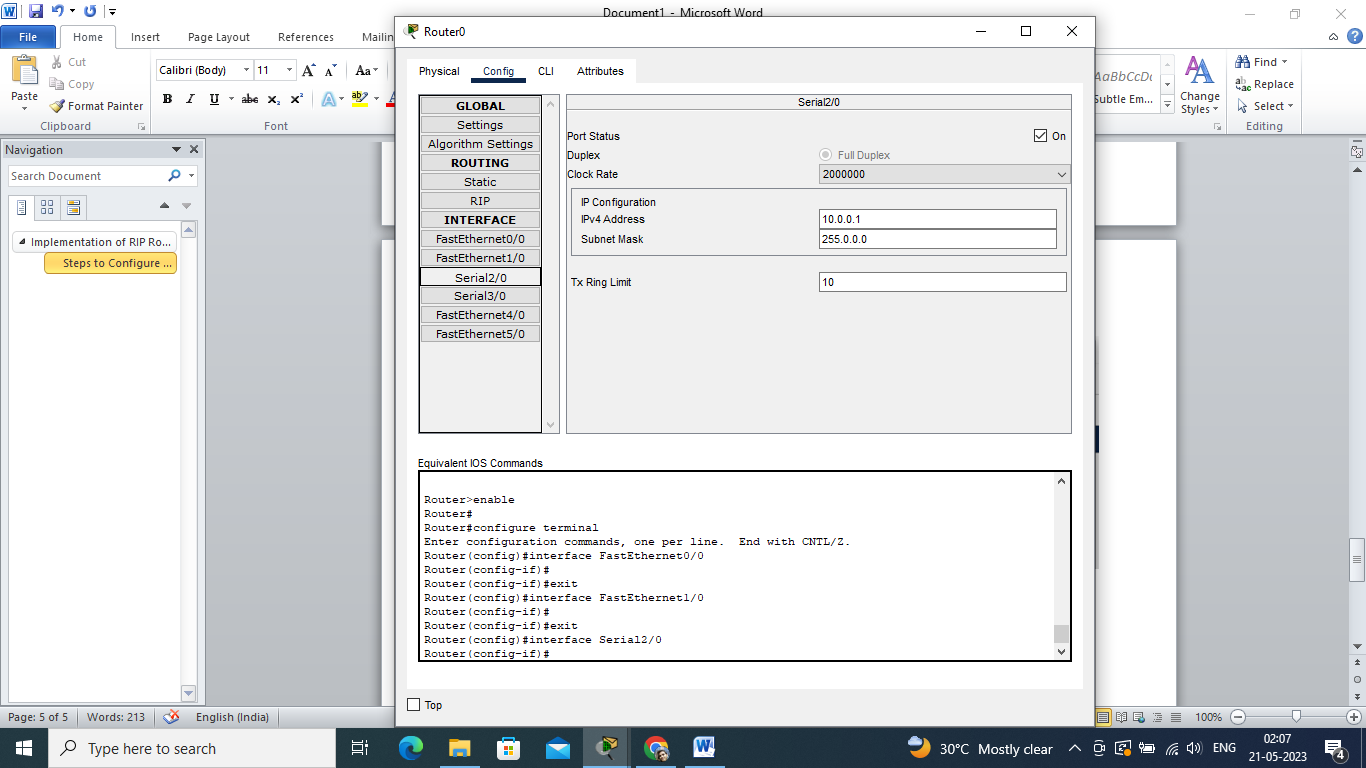
**Step 3:**Configure router with IP address and subnet mask.

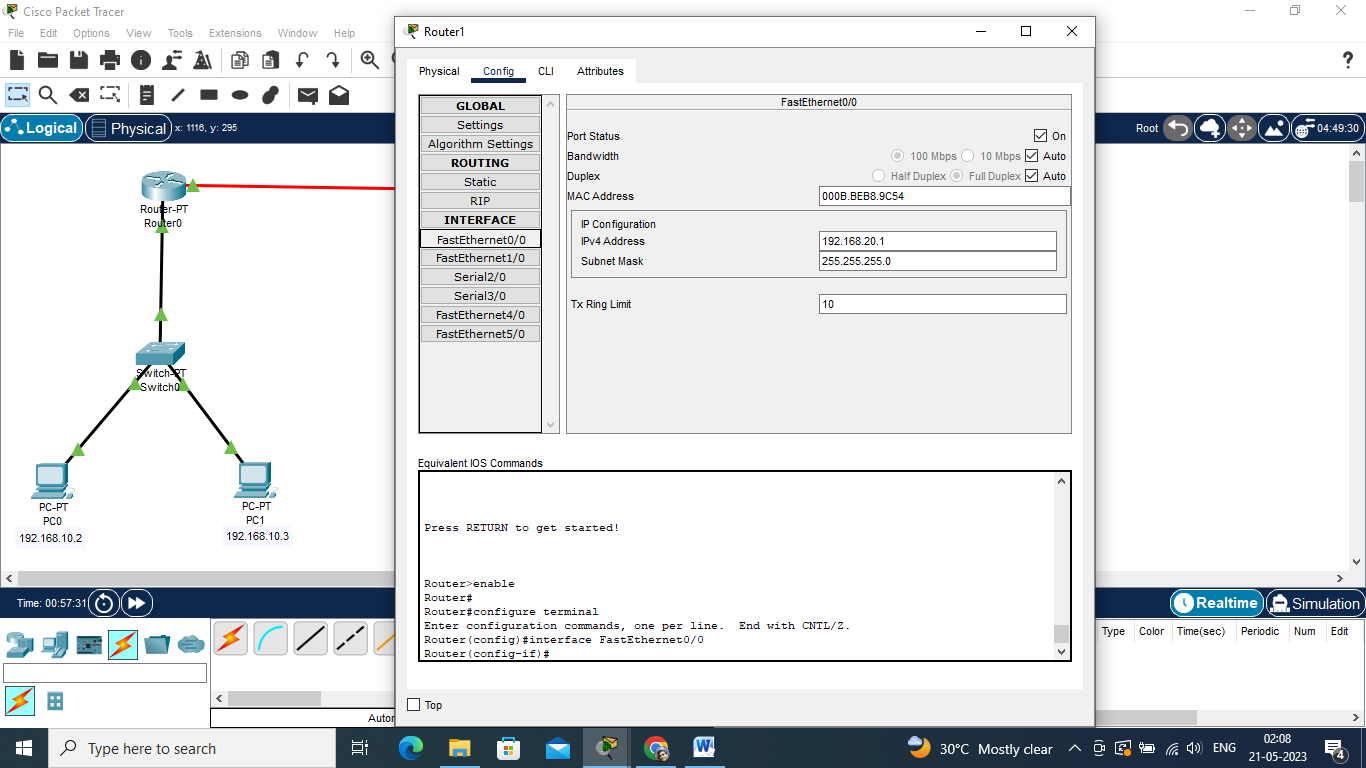
**IP Addressing Table Router:-**

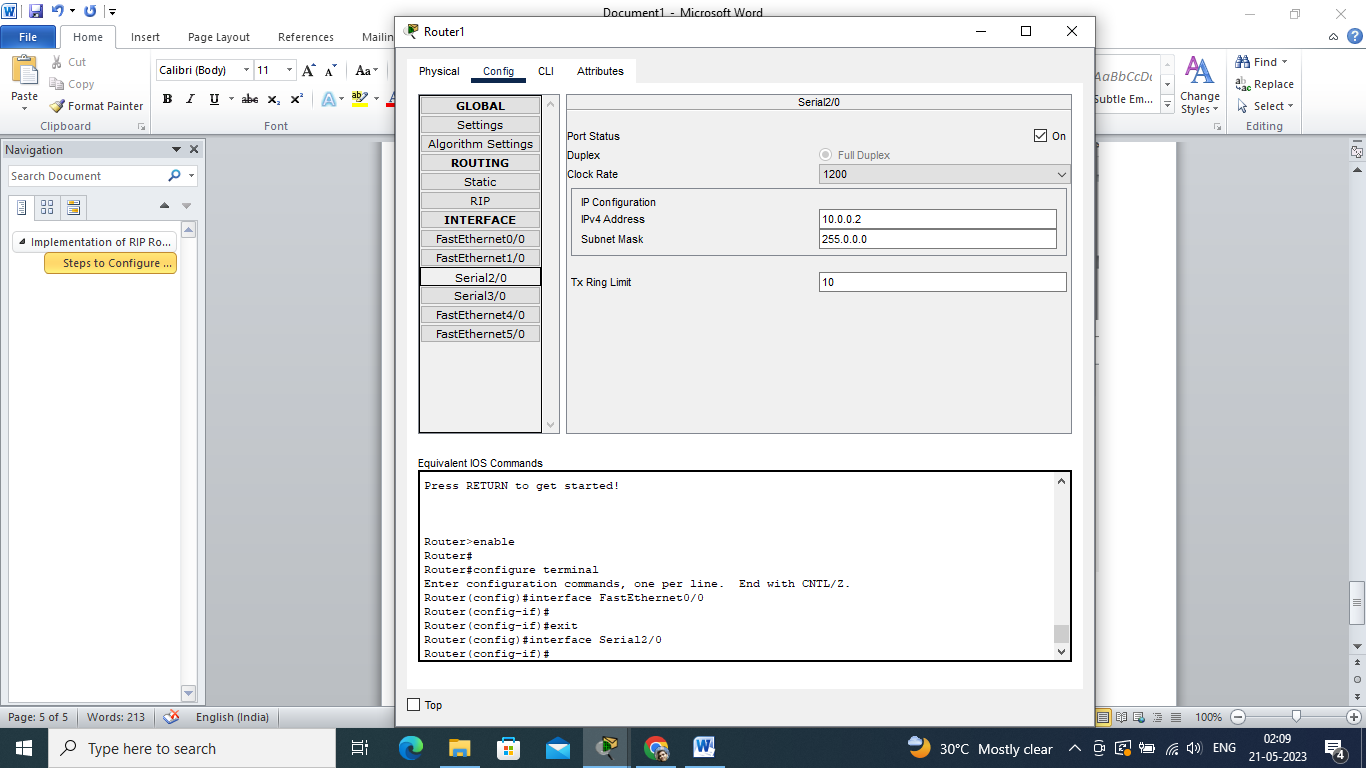


* To assign an IP address in router0, click on router0.
* Then, go to config and then Interfaces.
* Make sure to turn on the ports
* Then, configure the IP address in FastEthernet and serial ports according to IP addressing Table.
* Fill IPv4 address and subnet mask.









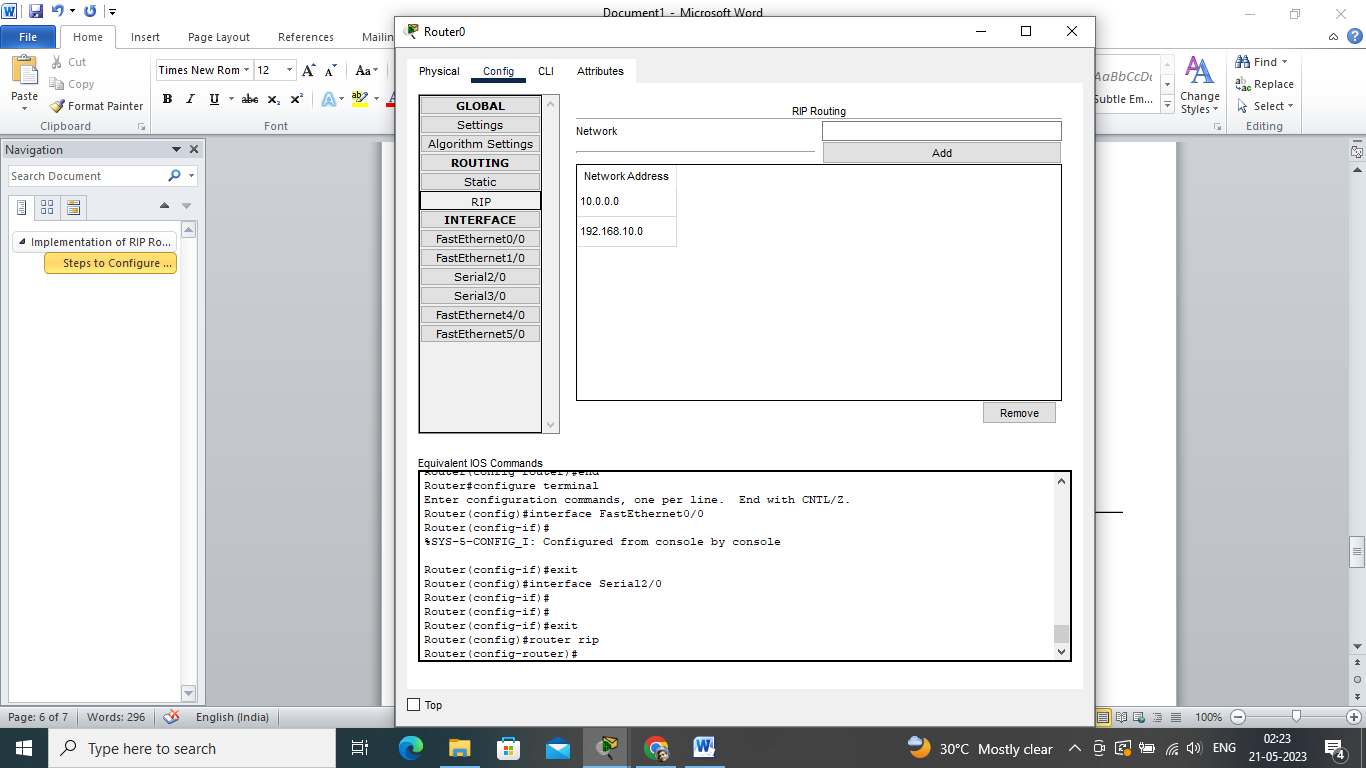
**Step 4:** After configuring all of the devices we need to assign the routes to the routers.

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

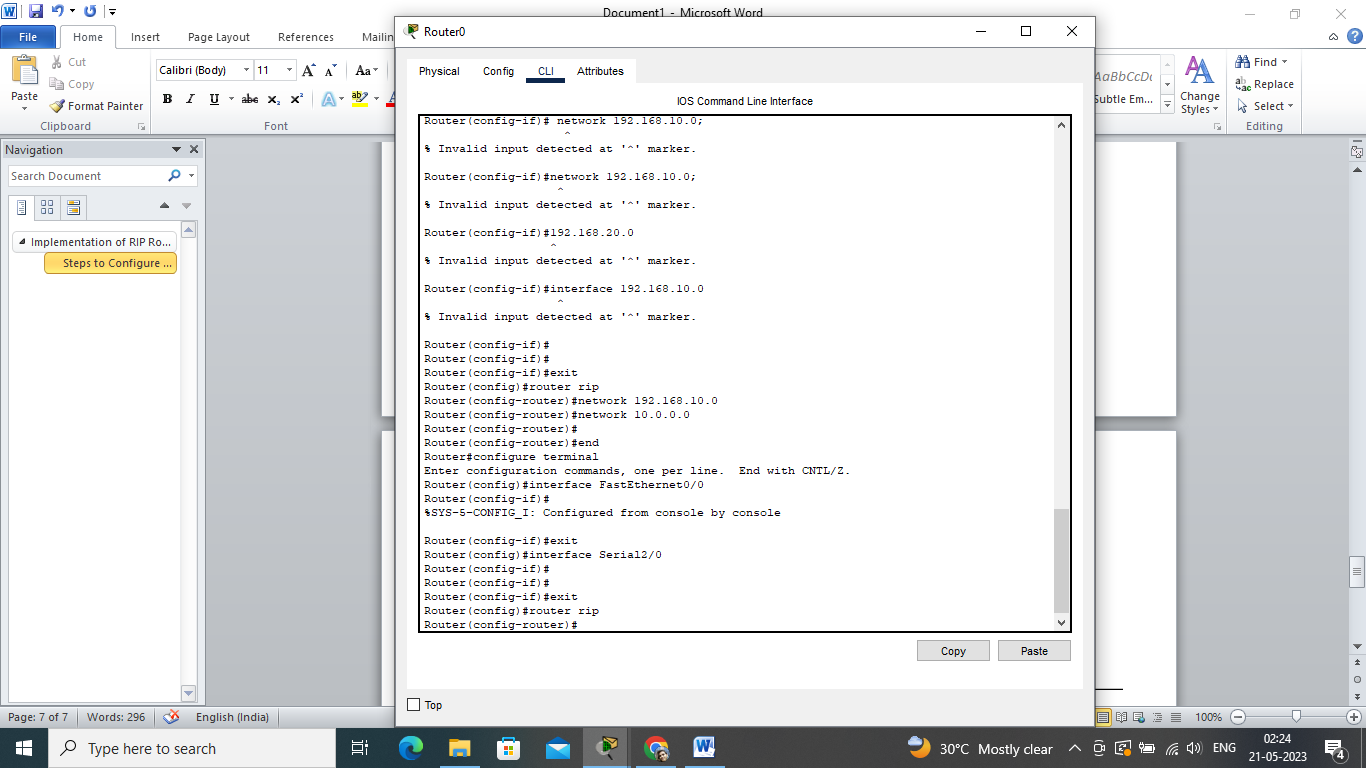
Router(config-router)#network 192.168.10.0

Router(config-router)#network 10.0.0.0

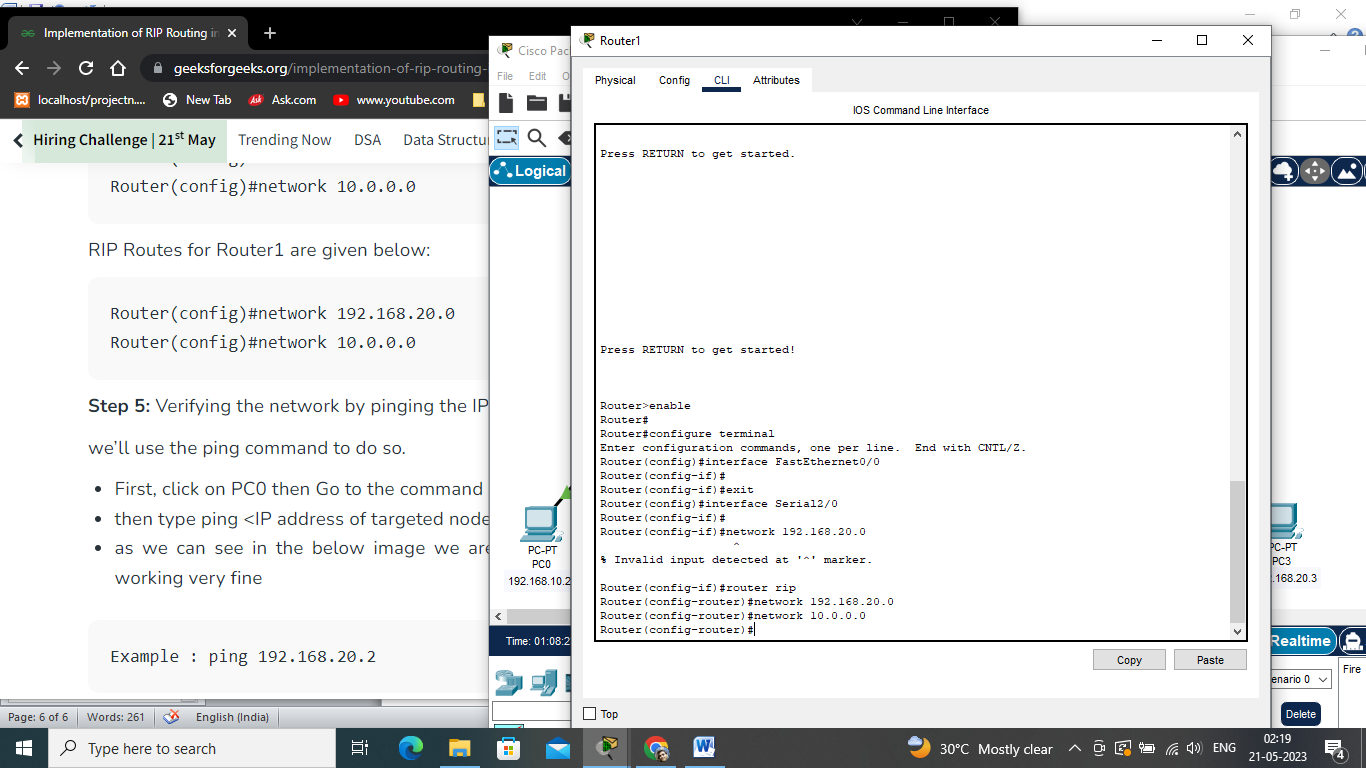
Router(config-router)#

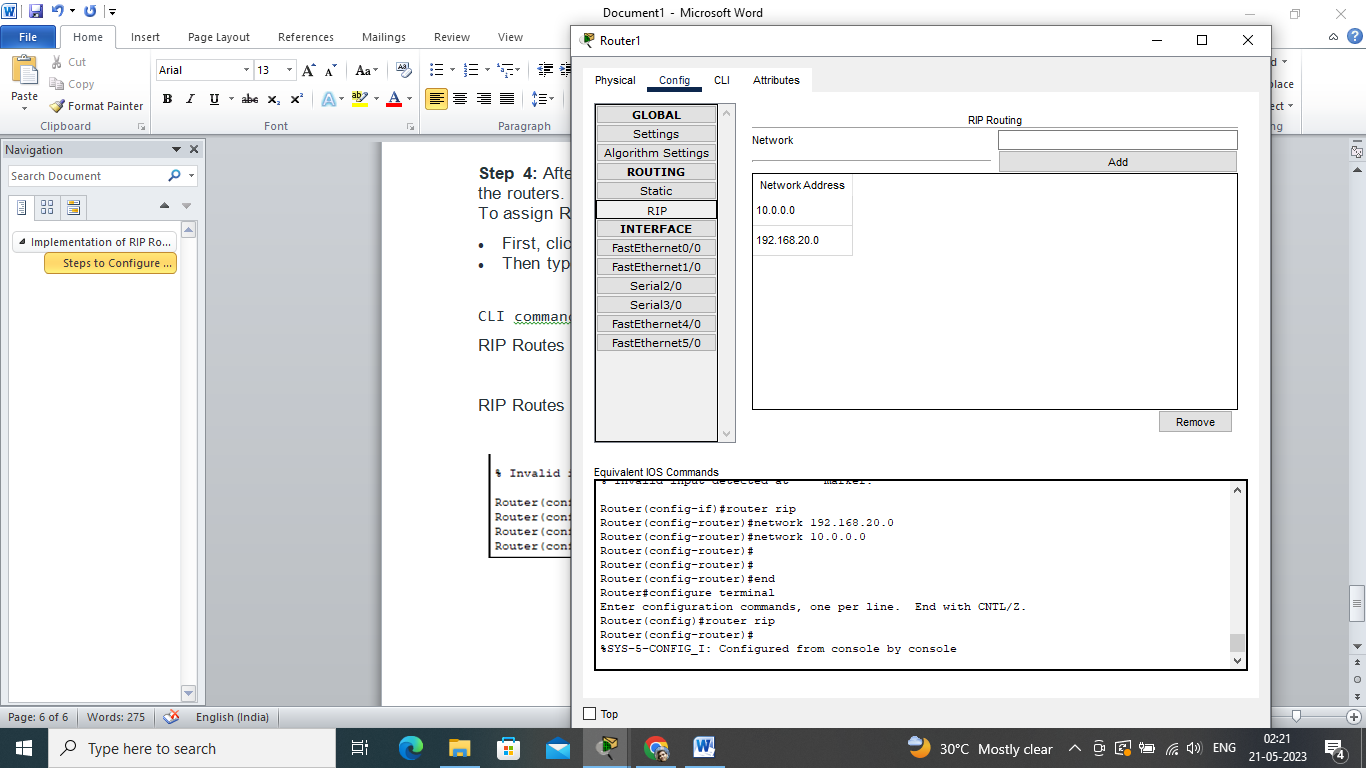
Router(config-router)#end

Router#configure terminal



**RIP Routes for Router1 are given below:**





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

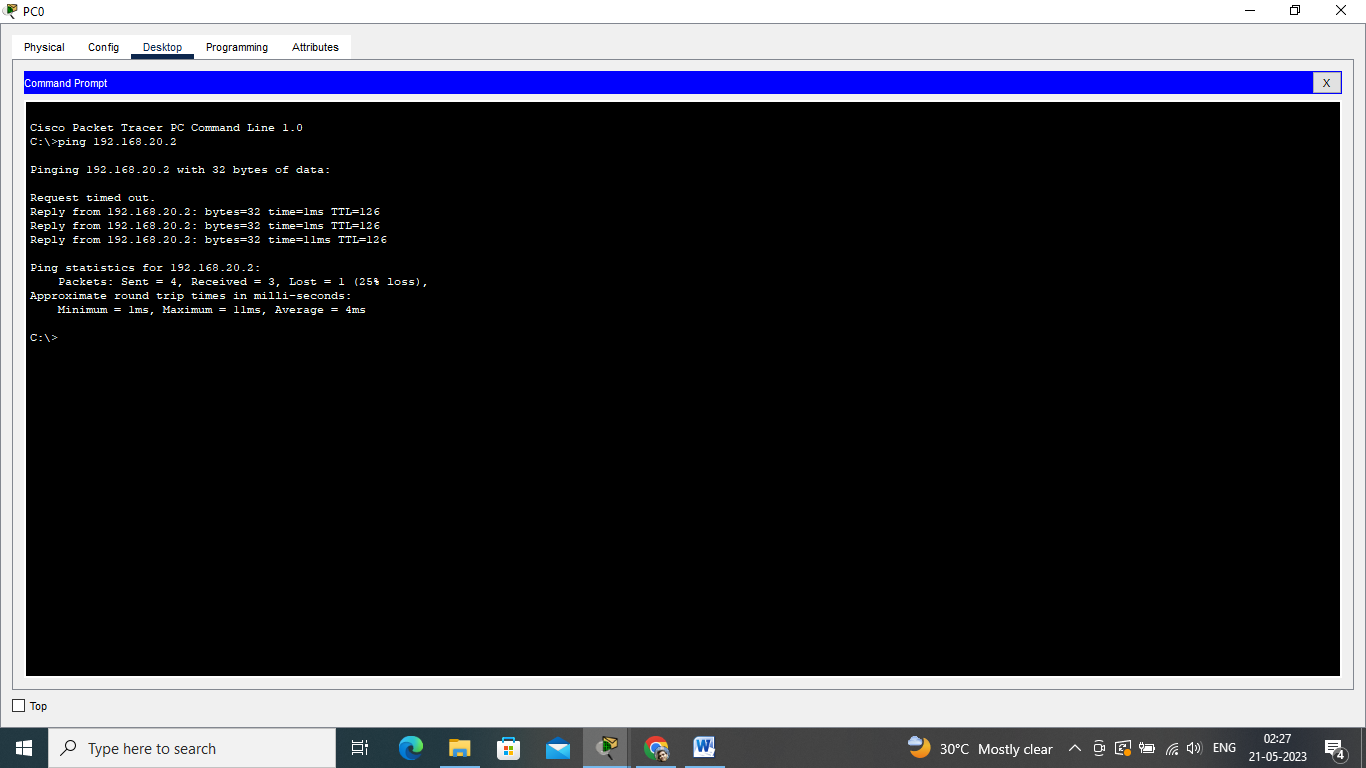
Router#configure terminal

**Step 5:** Verifying the network by pinging the IP address of any PC.

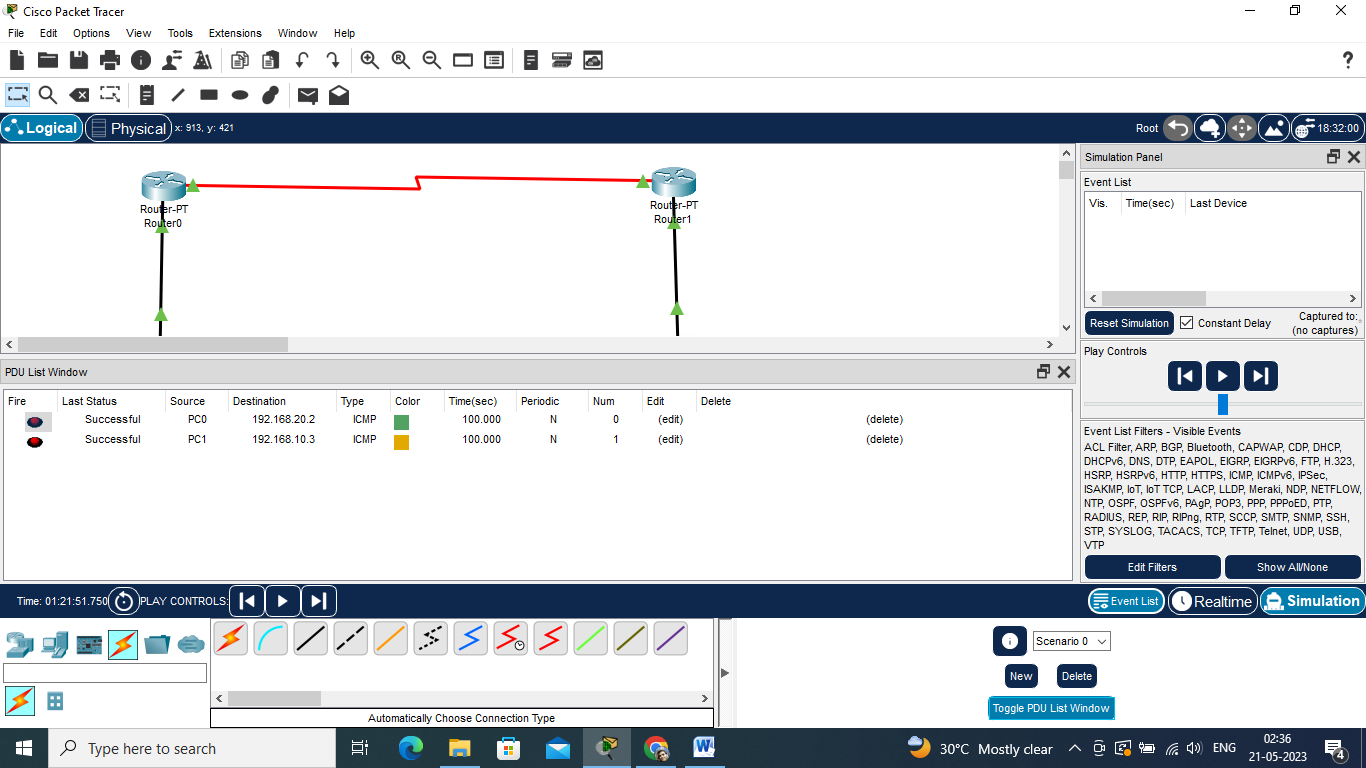
we’ll use the ping command to do so.

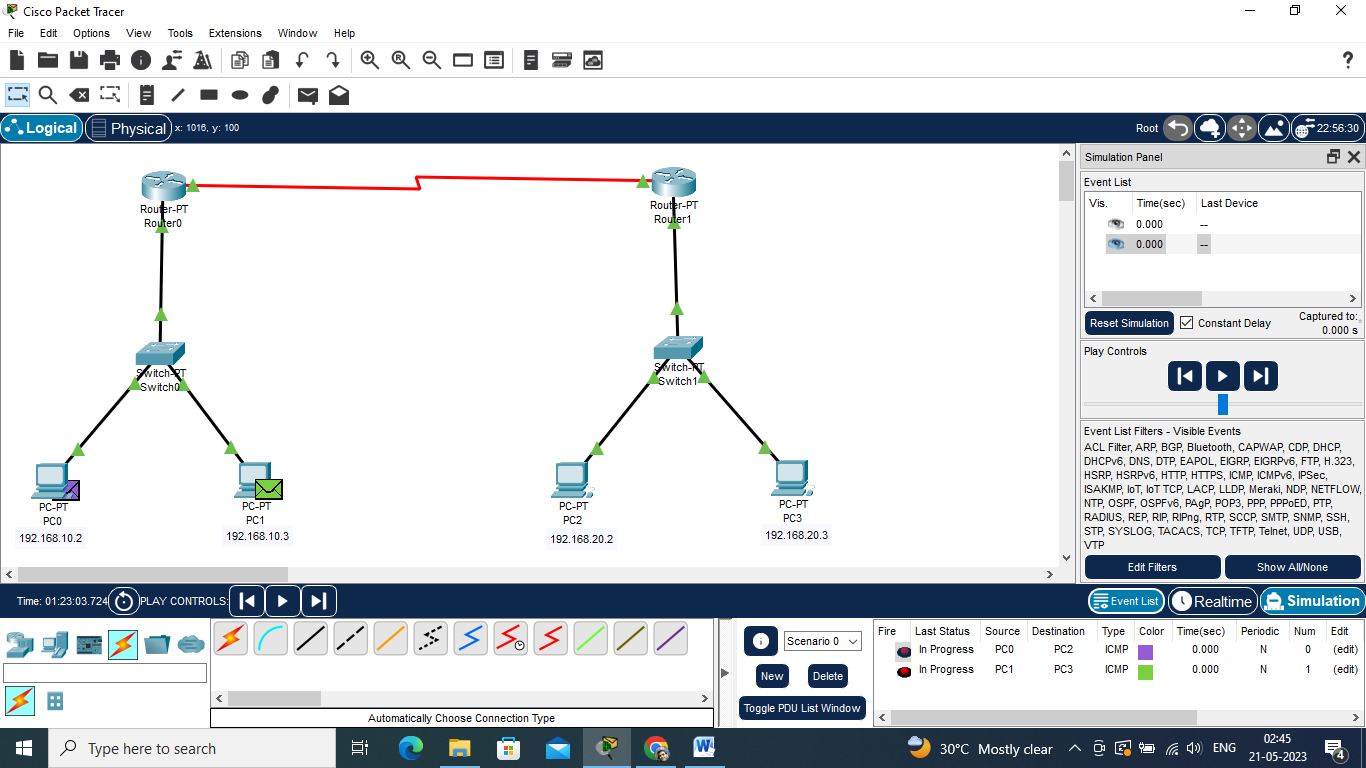
* First, click on PC0 then Go to the command prompt
* then type ping <IP address of targeted node>
* as we can see in the below image we are getting replies which means the connection is working very fine

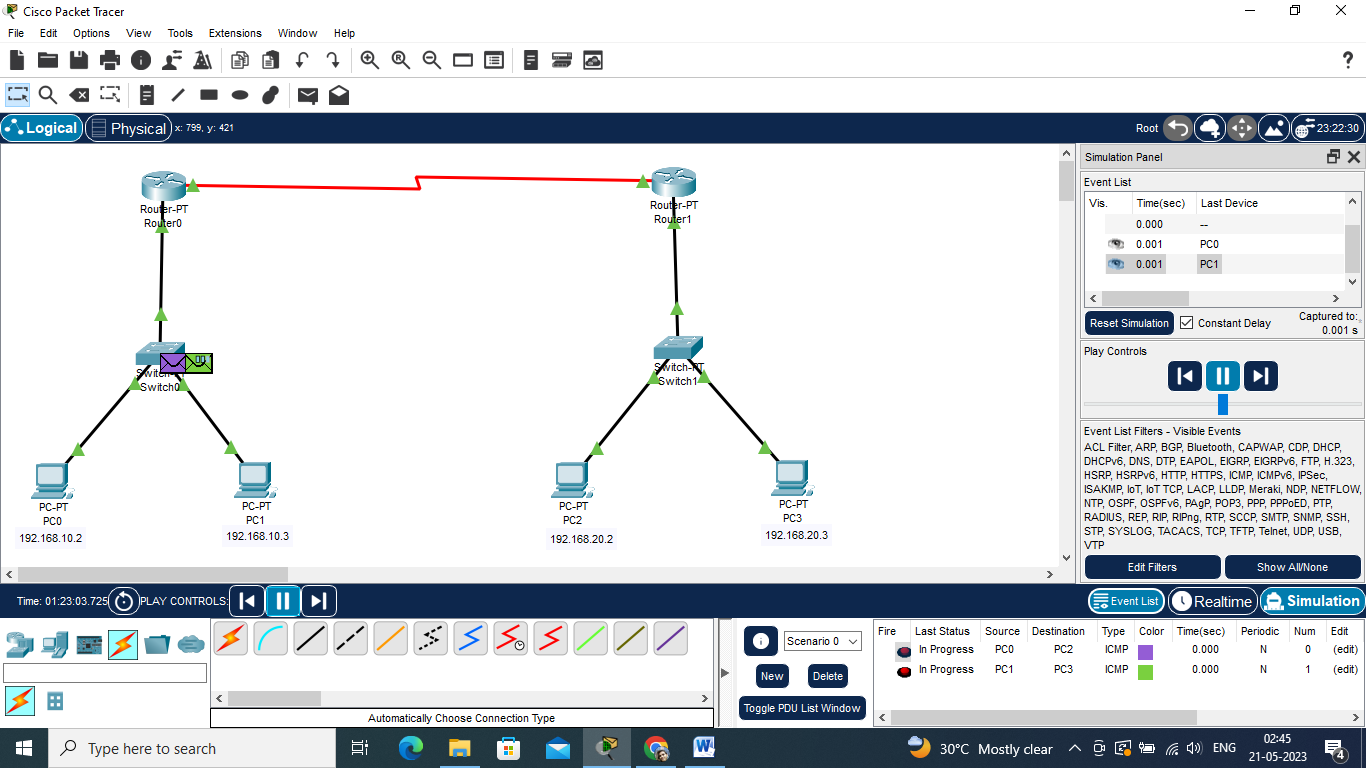
**Example :** ping 192.168.20.2

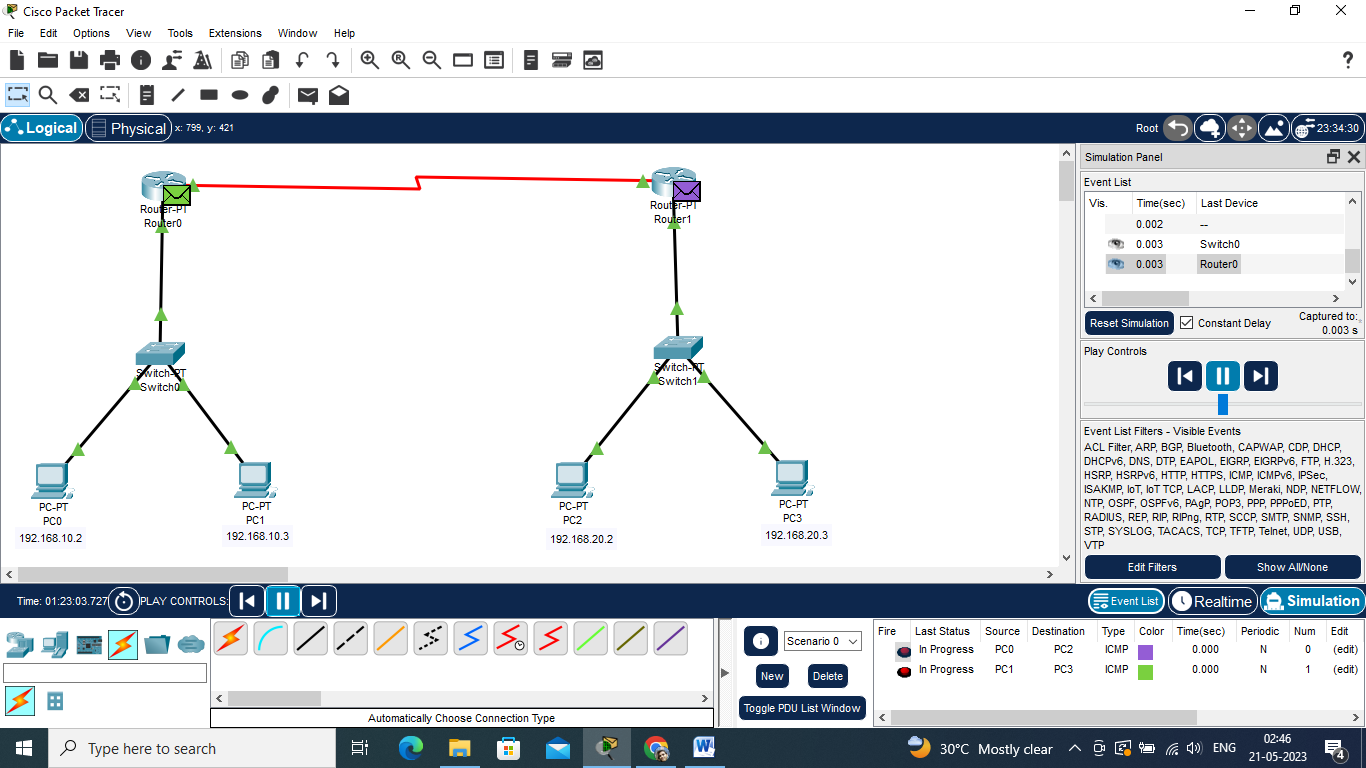


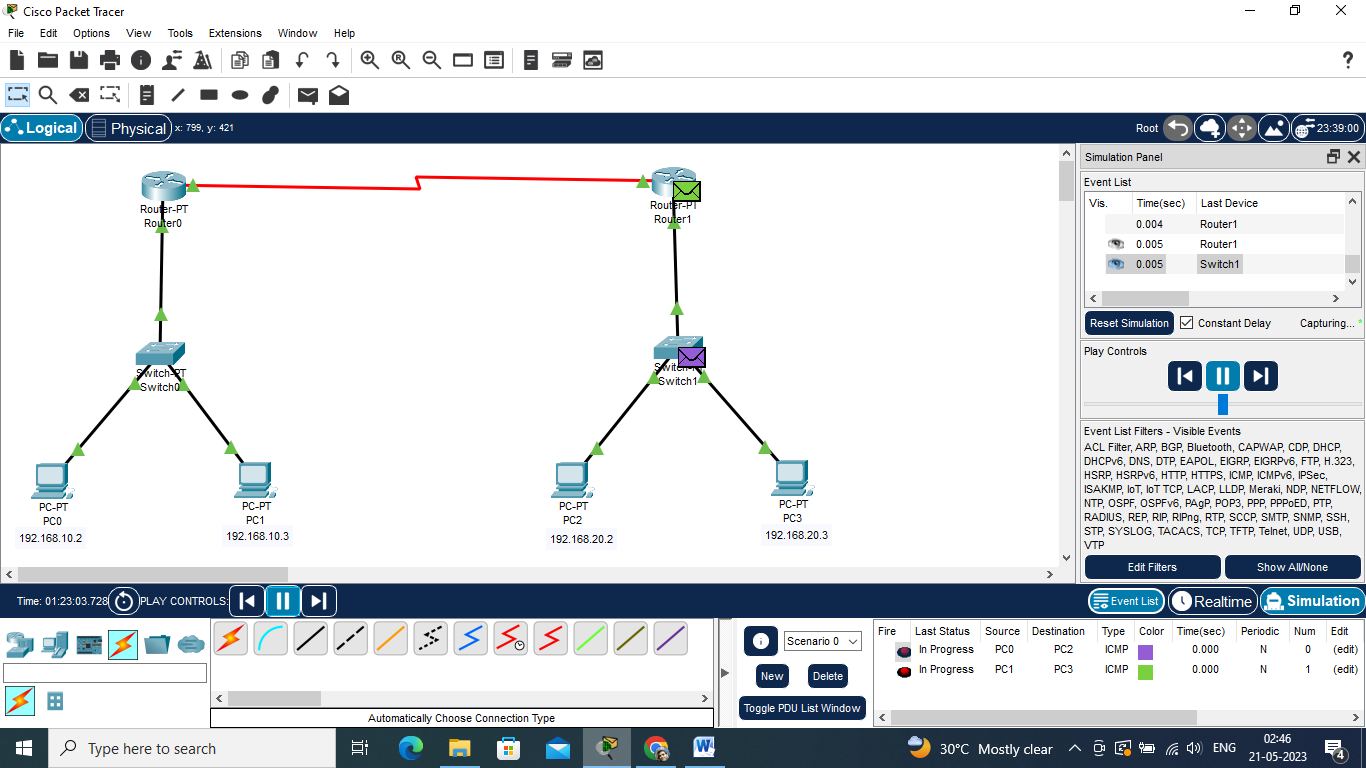
**A simulation of the experiment is given below we are sending PDU from  PC0 to PC2 and PC1 to PC3:**

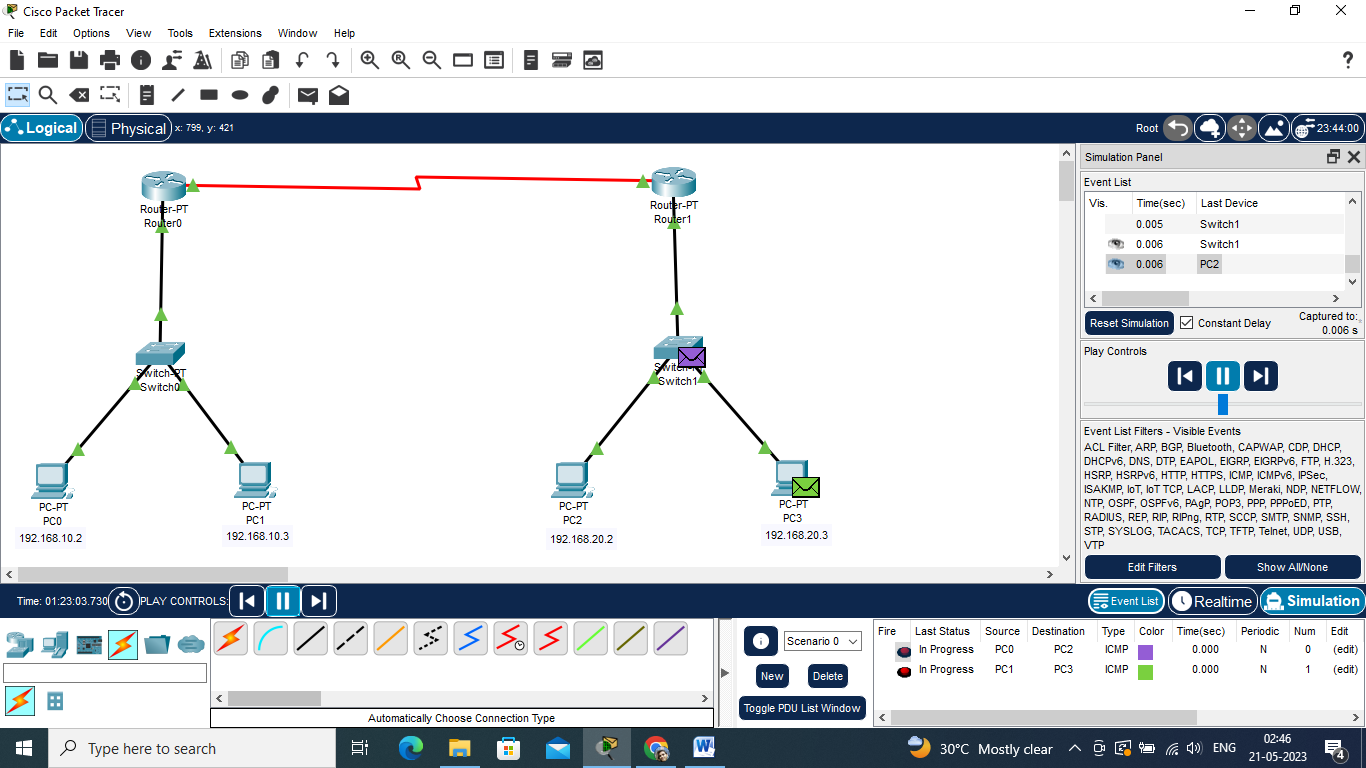


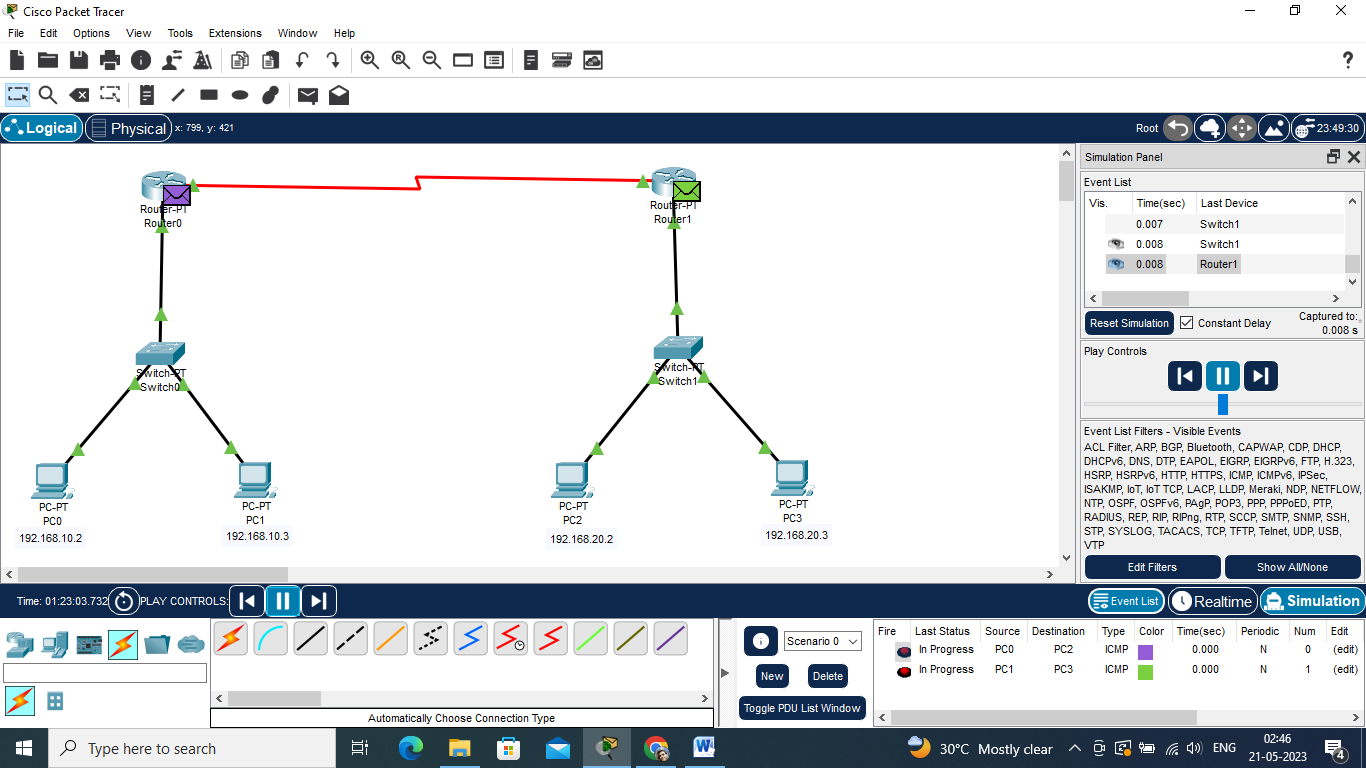


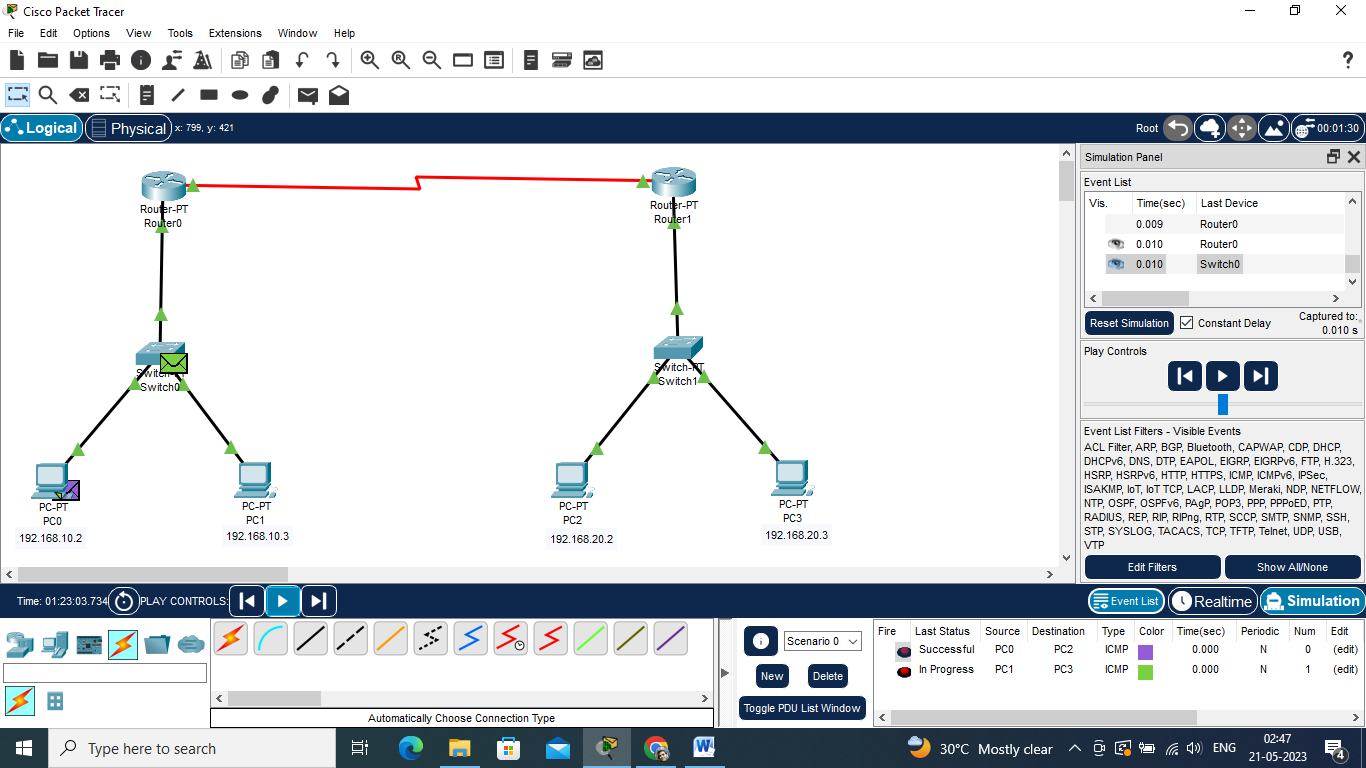


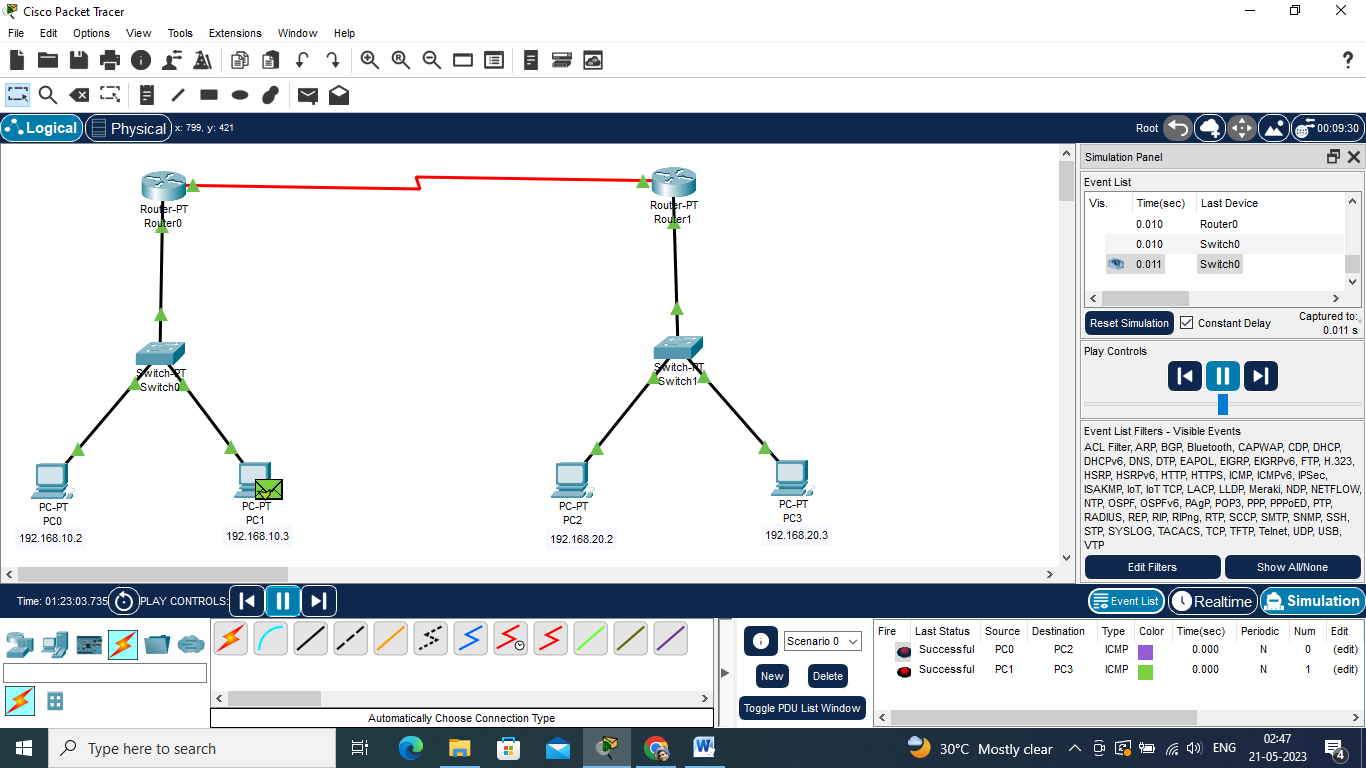








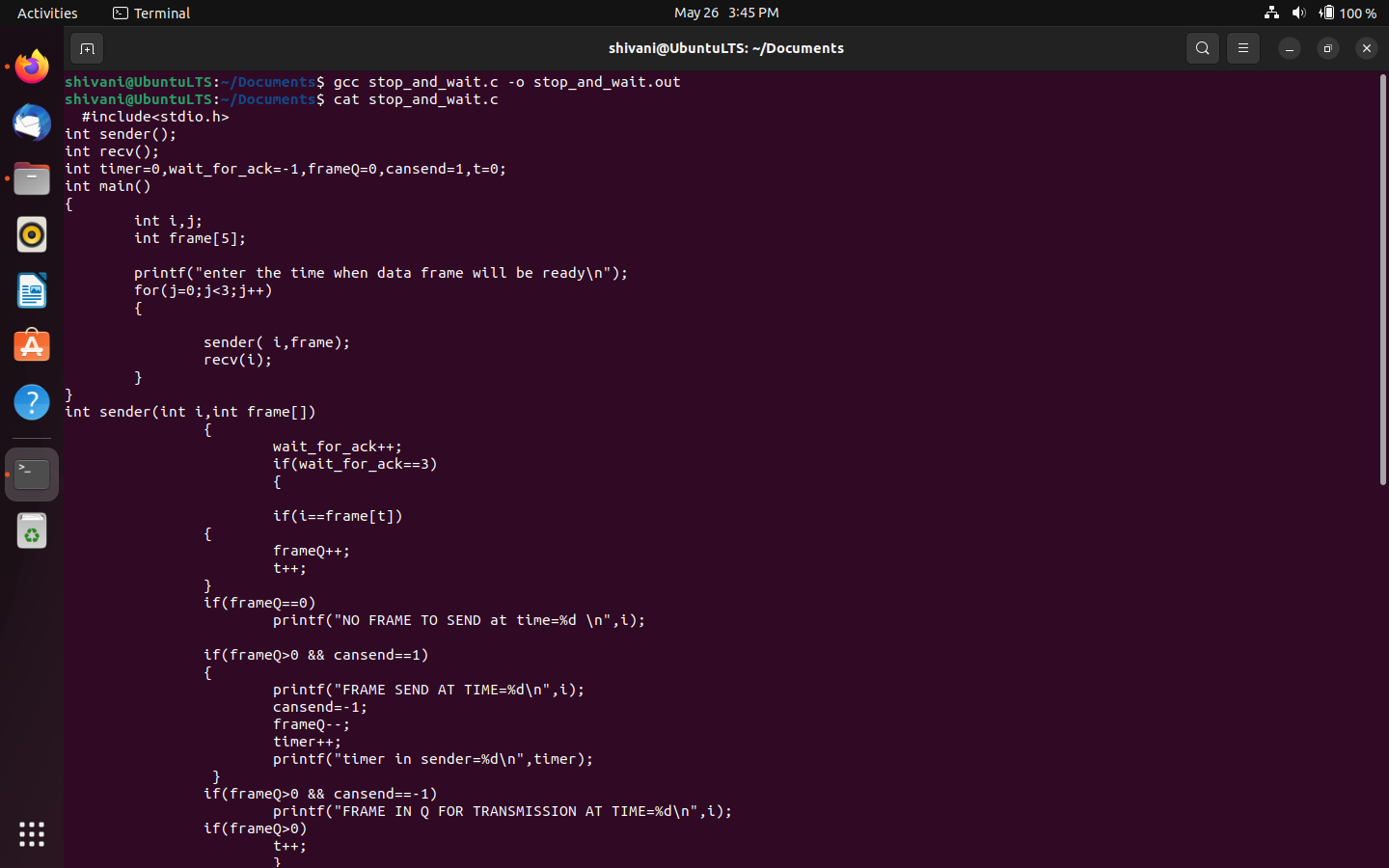


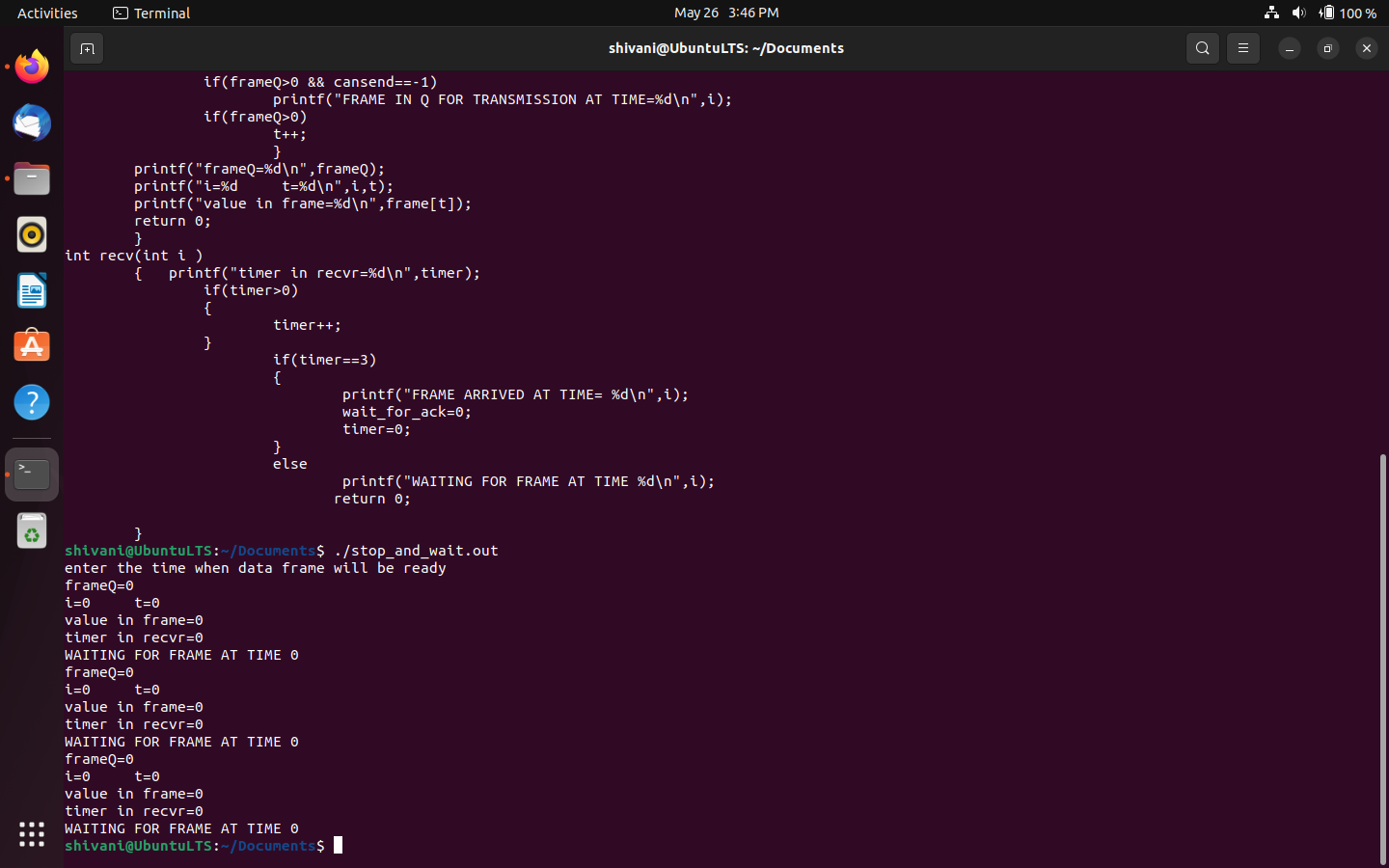


**Simple stop and wait ARQ implementation in C language**

|  |
| --- |
| #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:-**

****

****