Data Communication & Computer Networks Lab

Submitted by:

NAME : Rahul Gusain
SAP : 500084143
ROLL NO. : R214220900
BATCH: 1 DevOps

Submitted to:

Ved Prakash Bhardwaj

LAB FILE

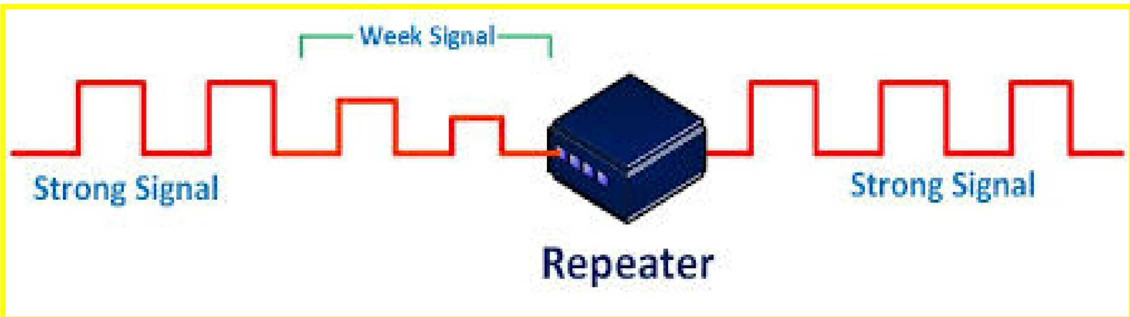
INDEX

EXPERIMENT 1

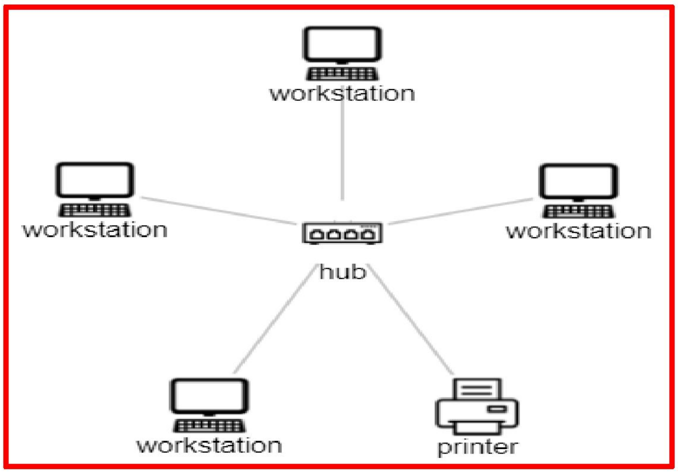
Familiarization of Network devices

REPEATER: A repeater is a network device that retransmits a received signal with more power and to an extended geographical or topological network boundary than what would be capable with the original signal.

A repeater is implemented in computer networks to expand the coverage area of the network, repropagate a weak or broken signal and or service remote nodes. Repeaters amplify the received/input signal to a higher frequency domain so that it is reusable, scalable and available.

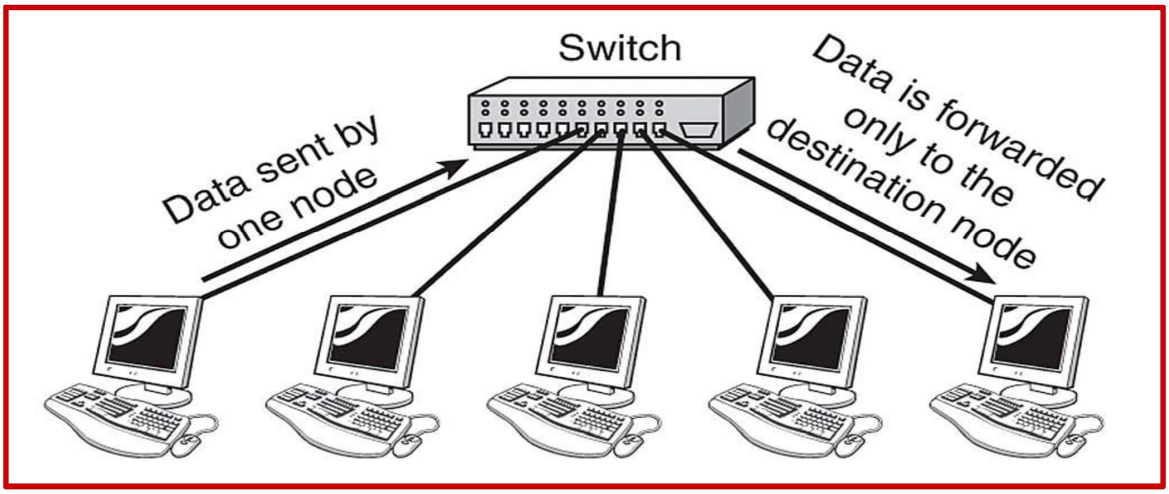


HUB: A hub is a common connection point, also known as a network hub, which is used for the connection of devices in a network. It works as a central connection for all the devices that are connected through a hub. The hub has numerous ports. If a packet reaches one port, it is able to see all the segments of the network due to a packet being copied to the other ports. A network hub has no routing tables or intelligence (unlike a network switch or router), which is used to send information and broadcast all network data across each and every connection.

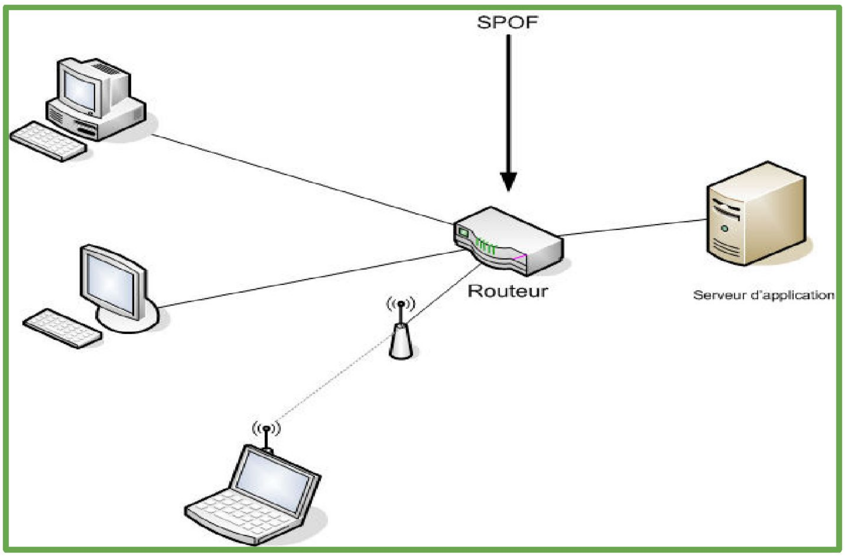


SWITCH: A switch is a multiport bridge with a buffer and a design that can help it function better (a large number of ports indicate less traffic). A switch is a data connection layer device. Before forwarding data, the switch may check for faults, which makes it extremely efficient because it doesn't transport packets with errors and only forwards good packets to the correct port. To put it another way, the switch separates hosts' collision domains while maintaining the broadcast domain

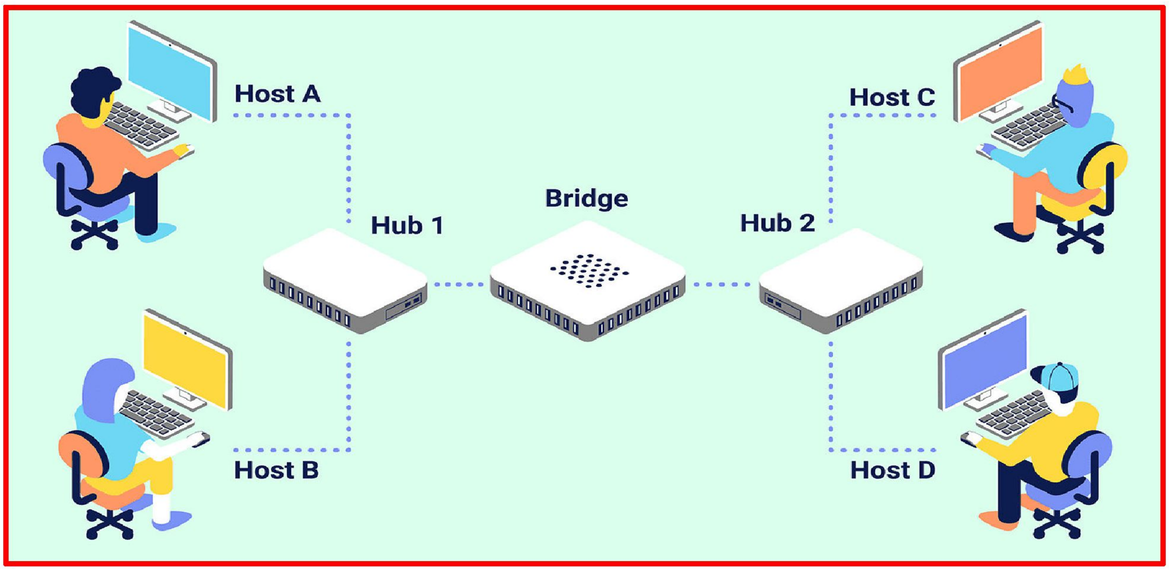
.



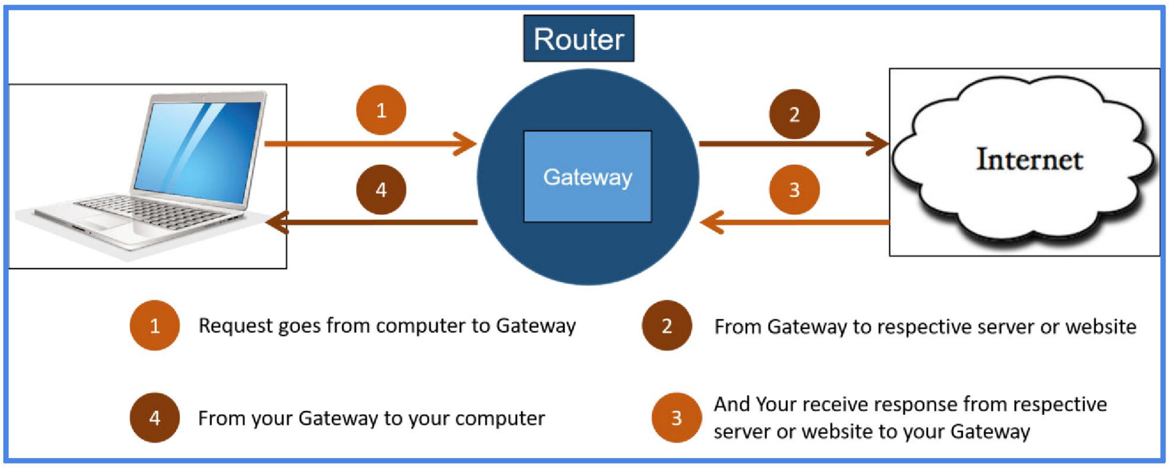
ROUTER: In the same way that switches route data packets based on their IP addresses, routers do the same. The router is basically a device that performs network layer functions. Routers connect LANs and WANs and decide how to route data packets using a dynamically updated routing table. The router divides the broadcast domains of hosts connected through it.



BRIDGE : A data connection layer device is referred to as a bridge. A bridge is a repeater that also has the capacity to filter data by reading the source and destination MAC addresses. It may also be used to connect two LANs using the same protocol. Because it only has one input and output port, it is a two-port device.



GATEWAY : A gateway is basically a device or hardware which acts as a "gate" among the networks. Thus it can also be defined as a node that acts as an entrance for the other nodes in the network. It is also responsible for enabling the traffic flow within the network. Gateway uses more than one protocol for communication thus its activities are much more complex than a switch or a router.



EXPERIMENT 2

Write a program for Bit stuffing and Destuffing in a bitstream.

Bit stuffing is the insertion of non-information bits into data. Note that stuffed bits should not be confused with overhead bits. Overhead bits are non-data bits that are necessary for transmission (usually as part of headers, checksums etc.).

Algorithm:

Stage 1-Create a 3 single cluster of adequate size

Stage 2-For Bit Stuffing: Create 2 single type pointer variable highlighting the principal component of information and stuff exhibit.

Stage 3-Traverse the information exhibit till last component.

Stage 4-Create an if condition to actually take a look at the component in string; in the event that the component is '0' store it in *y and increase the pointer variable.

Stage 5-If the component found isn't '0' then in else condition make some time circle which will run till *x is '1' and counter factor isn't 5 and afterward increase counter and store *x variable in *y variable and augmentation x and y.

Stage 6-is count=5 then store '0' at that record after which 5 sequential 1 happens and increase y. Store it in *y pointer variable.

Stage 7-For Bit Destuffing; x=stuff and y=destuff

Stage 8-All the means would be same as we did in stiffing other than the last advance rather than putting away '0' in the stuff exhibit just addition the pointer variable.

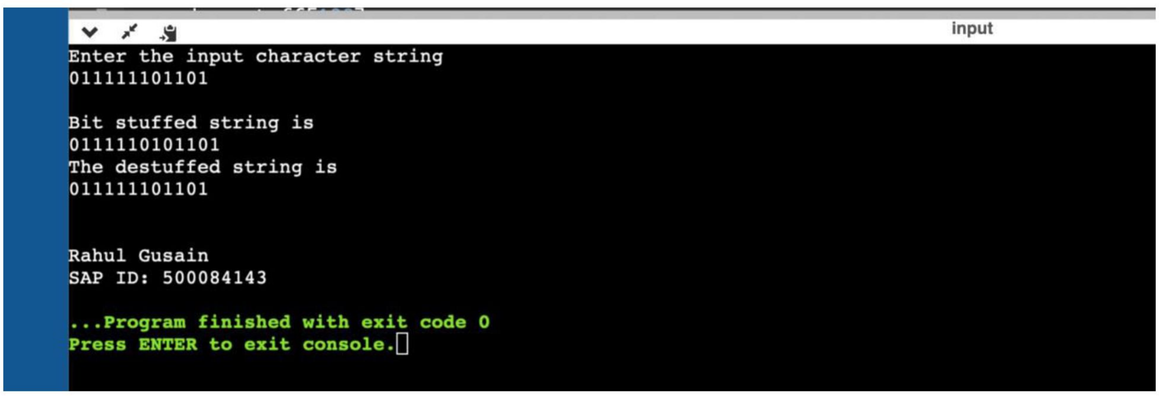
Code-

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
   char temp;
   char input[100];
   char stuff[100];
   char *x,*y;
   char destuff[100];
   int count=0;
   printf("enter the input character string \n");
   scanf("%s",input);
   // for Bit Stuffing
   x=input;
   y=stuff;
```

```
while(*x!='\0')
{
   if(*x=='0')
   {
      *y=*x;
      y++;
      x++;
   }
   else
   {
      while(*x=='1' && count!=5)
      {
         count++;
         *y=*x;
         y++;
         x++;
      }
      if(count==5)
      {
         *y='0';
         y++;
      }
      count=0;

   }
}
*y='\0';
printf("\nBit stuffed string is");
printf("\n%s",stuff);
// for Bit Destuffing and taking the value of x from y of stuffing part
x=stuff;
y=destuff;
while(*x!='\0')
{
   if(*x=='0')
   {
      *y=*x;
      y++;
```

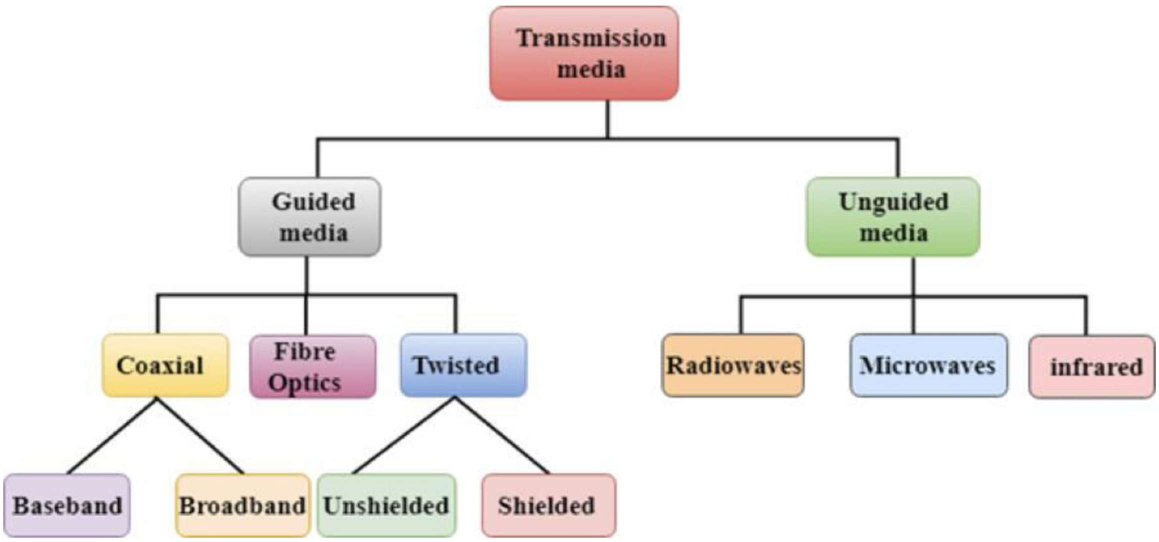
```
        x++;
      }
      else
      {
      while(*x=='1' && count!=5)
      {
        count++;
        *y=*x;
        y++;
        x++;
      }
      if(count==5)
      {
        x++;
      }
      count=0;
      }
  }
  *y='\0';
  printf("\nThe destuffed string is");
  printf("\n%s\n",destuff);
  printf("\n\nRahul Gusain\n");
  printf("SAP ID: 500084143");
  return 0;
}
```



EXPERIMENT 3

Familiarization of Transmission media

Aim: Study in detail the following transmission media in computer networks:



Apparatus (Software): No Software or hardware is needed.

Theory: Transmission media is a communication channel that transmits information from sender to receiver. Data is transmitted through electromagnetic signals. There are two types of transmission media, wired media and wireless media. In wired media, the medium characteristics are more important, whereas, in wireless media, the signal characteristics are more important. Different transmission media have different properties such as bandwidth, delay, cost and ease of installation and maintenance. The transmission media is available in the physical layer- the lowest layer of the OSI reference model.

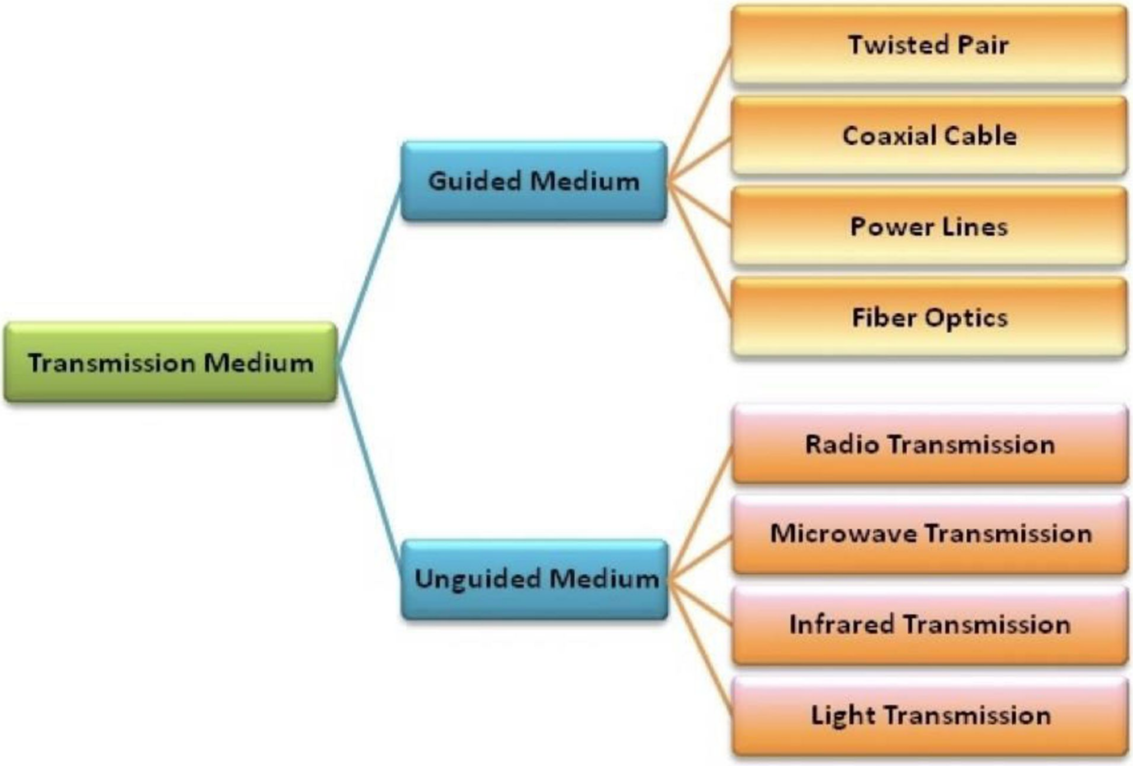
Procedure: Explain each transmission medium in detail with a suitable diagram. Also, write a short note on each transmission medium.

Conclusion: Gain knowledge about transmission media in computer networks.

The transmission medium can be defined as a pathway that can transmit information from a sender to a receiver. Transmission media are located below the physical layer and are controlled by the physical layer. Transmission media are also called communication channels.

Transmission media are of two types −

- Guided Transmission Medium
- Unguided Transmission Medium



Guided media	Unguided media
The signal requires a physical path for transmission.	The signal is broadcasted through air or sometimes water
It is called wired communication or bounded transmission media.	It is called wireless communication or unbounded transmission media.
It provides direction to signal for travelling. Twisted pair cable, coaxial cable and fibre optic cable are its types.	It does not provide any direction. Radio waves, microwave and infrared are its types.

Guided Transmission Medium

Guided transmission media are also called bounded media or wired media. They comprise cables or wires through which data is transmitted. They are called guided since they provide a physical conduit from the sender device to the receiver device. The signal travelling through these media are bounded by the physical limits of the medium.

The most popular guided media are −

- Twisted pair cable
- Coaxial cable
- Power lines
- Fibre optics

Twisted Pair Copper

It is the most used media across the world. All the local telephone exchanges are made of twisted-pair copper. These telephone lines are reused as last-mile DSL access links to access the internet from home.

Twisted pair copper wires are also used in Ethernet LAN cables within homes and offices.

It supports low to High Data Rates which is in the order of Gigabytes.

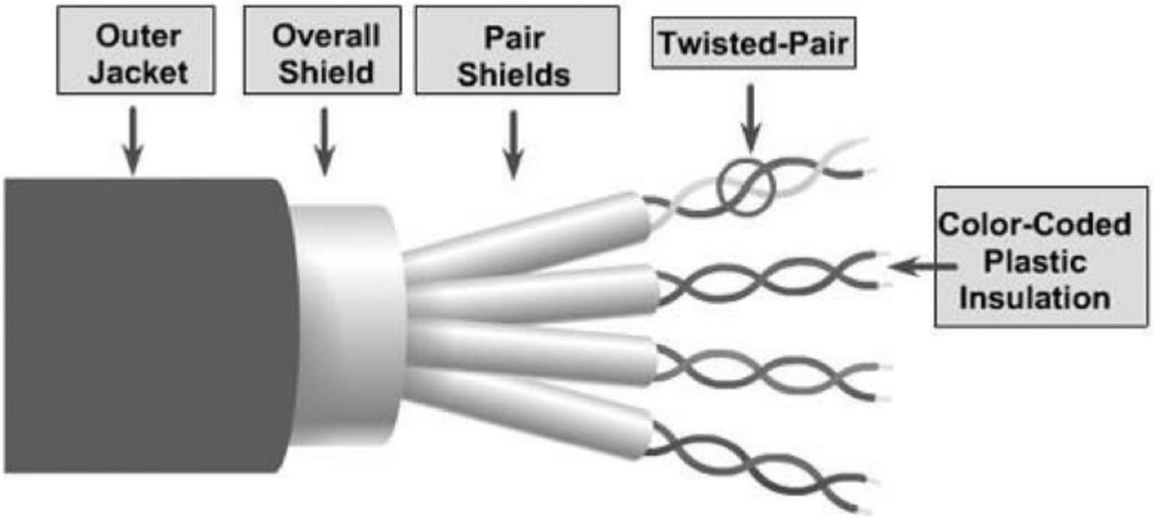
These wires are effective up to a maximum distance of a few kilometres/miles, because the signal strength is lost significantly beyond the distance.

Generally, they come in two variants as follows −

- UTP (unshielded twisted pair)
- STP (shielded twisted pair)

For every variant, there are multiple sub-variants, based on the thickness of the material (like UTP-3, UTP-5, UTP-7 etc.)

The twisted-pair copper is diagrammatically represented as follows −



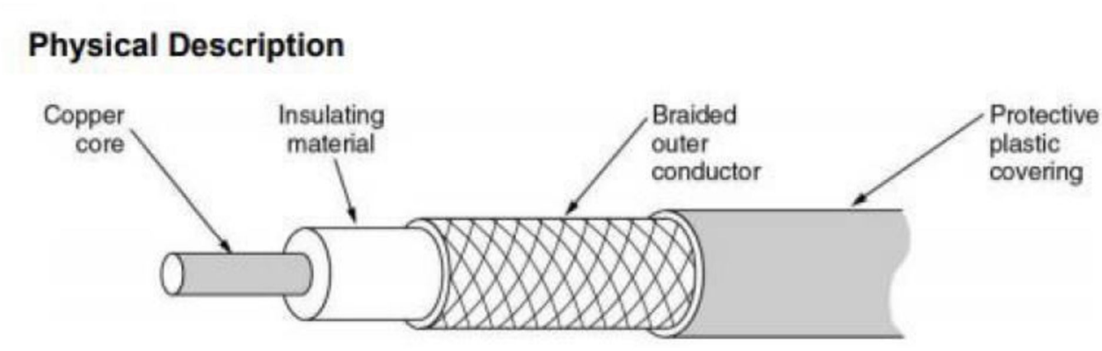
Copper Co-axial Cables

Co-axial copper cables consist of an inner copper conductor and an outer copper shield, which are separated by a dielectric insulating material, helpful in preventing signal losses.

Copper co-axial cables are used in cable TV networks and as trunk lines between telecommunication equipment.

It serves as an internet access line from the home and supports medium to high data rates.

The copper co-axial cable is diagrammatically represented as follows −



Fibre Optic Cables

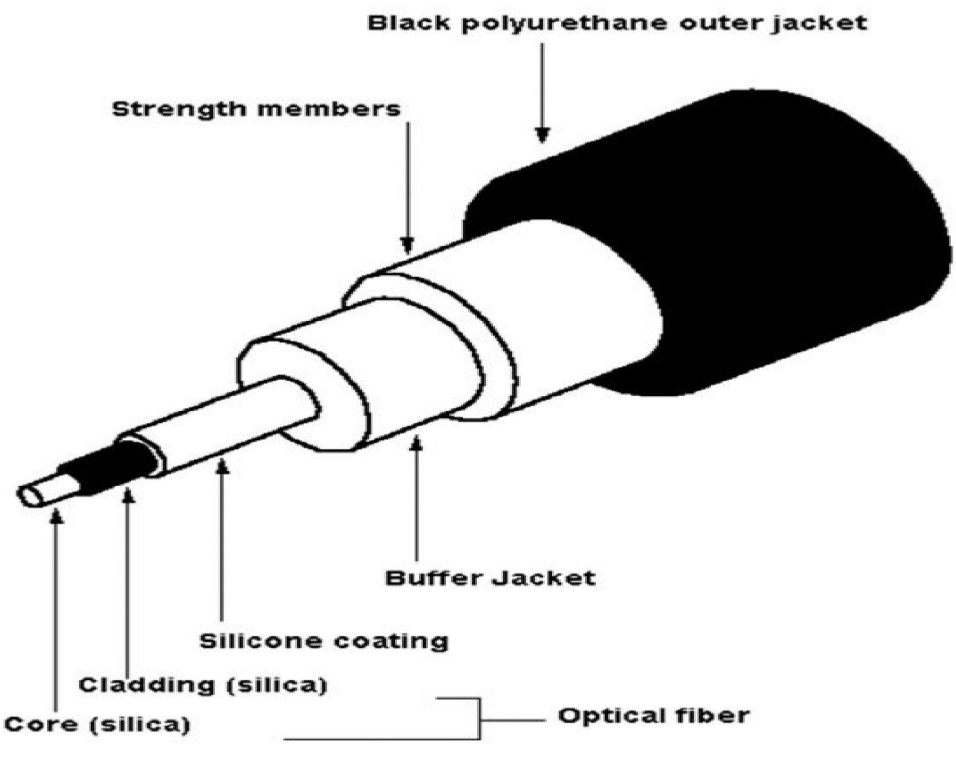
In fibre optic cables the information is transmitted by the propagation of optical signals (light) fibre optic cables and not through electrical/electromagnetic signals. Because of this, fibre optics communication supports longer distances as there is no electrical interference.

The fibre optic cables are made of very thin strands of glass (silica). It supports high data rates.

It is used for accessing the internet from home through FTTH (Fiber-To-The-Home) lines.

Examples − OC-48, OC-192, FTTC, HFC.

The fibre optic cable is diagrammatically represented as follows –



Unguided Transmission Medium

Unguided transmission media are also called wireless media. They transport data in the form of electromagnetic waves that do not require any cables for transmission. These media are bounded by geographical boundaries. This type of communication is commonly referred to as wireless communications.

Unguided signals can travel in three ways −

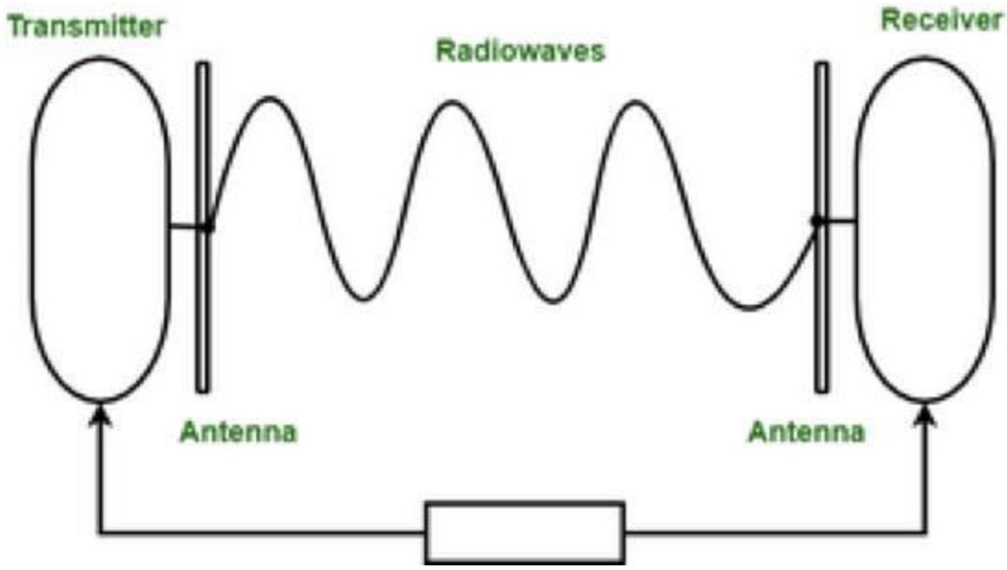
- Ground propagation
- Sky propagation
- Line-of-sight propagation

The commonly used unguided transmissions are −

- Radio transmission
- Microwave transmission
- Infrared transmission
- Light transmission

(i) Radio waves -

These are easy to generate and can penetrate through buildings. The sending and receiving antennas need not be aligned. Frequency Range: 3KHz – 1GHz. AM and FM radios and cordless phones use Radio waves for transmission.

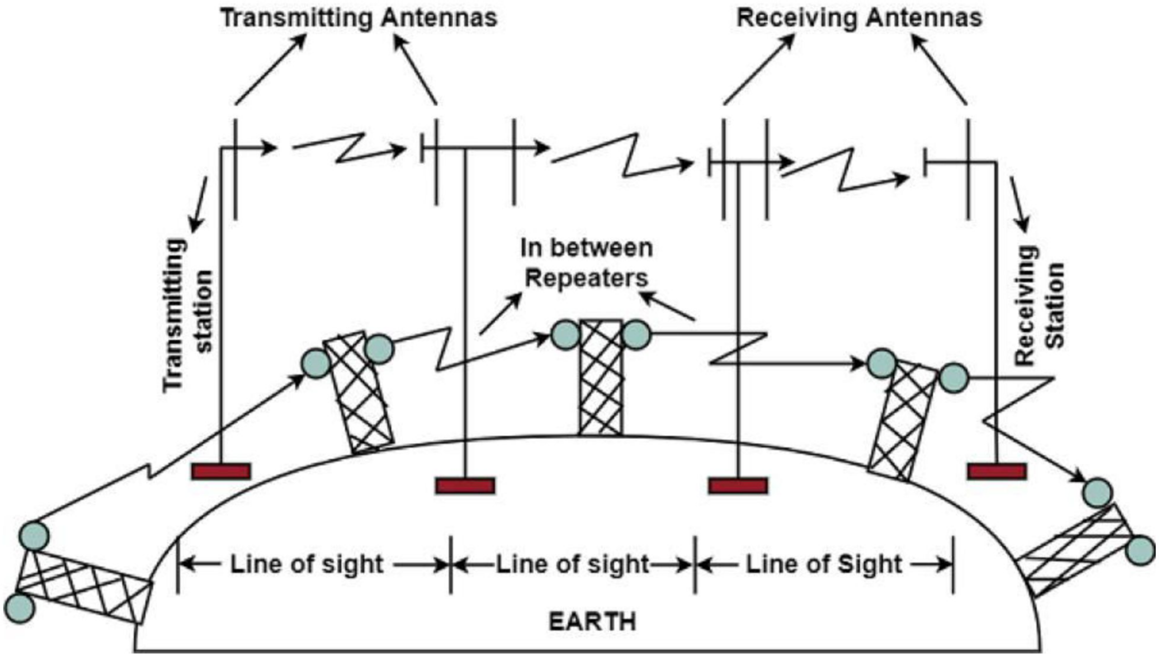


Further Categorized as (i) Terrestrial and (ii) Satellite.

(ii) Microwaves –

It is a line of sight transmission i.e. the sending and receiving antennas need to be properly aligned with each other. The distance covered by the signal is directly proportional to the height of the antenna. Frequency Range: 1GHz – 300GHz. These are majorly used for mobile phone communication and television distribution.

The electromagnetic waves having frequencies between 1 and 300 GHz are known as a microwave. Microwave systems use very high-frequency radio or television signals to transmit data through space. Therefore, the transmitter and receiver of a microwaves system, which is mounted on very high towers, should be invisible to each other, i.e., they both should be in a line of sight. Moreover, the signals become weak after travelling a certain distance and require power amplification.



(iii) Infrared –

Infrared waves are used for very short distance communication. They cannot penetrate through obstacles. This prevents interference between systems. Frequency Range: 300GHz – 400THz. It is used in TV remotes, wireless mouse, keyboards, printers, etc.

EXPERIMENT 4

Aim :- To implement hamming code and CRC

Theory:-

Hamming code is a set of error-correction codes that can be used to detect and correct the errors that can occur when the data is moved or stored from the sender to the receiver. It is a technique developed by R.W. Hamming for error correction

Algorithm:-

1.Number the bits starting from 1: bit 1, 2, 3, 4, 5, 6, 7, etc.

2.Write the bit numbers in binary: 1, 10, 11, 100, 101, 110, 111, etc.

3.All bit positions that are powers of two (have a single 1 bit in the binary form of their position) are parity bits: 1, 2, 4, 8, etc. (1, 10, 100, 1000)

4.All other bit positions, with two or more 1 bits in the binary form of their position, are data bits.

5.Each data bit is included in a unique set of 2 or more parity bits, as determined by the binary form of its bit position.

1. Parity bit 1 covers all bit positions which have the **least** significant bit set: bit 1 (the parity bit itself), 3, 5, 7, 9, etc.
2. Parity bit 2 covers all bit positions which have the **second** least significant bit set: bits 2-3, 6-7, 10-11, etc.
3. Parity bit 4 covers all bit positions which have the **third** least significant bit set: bits 4–7, 12–15, 20–23, etc.
4. Parity bit 8 covers all bit positions which have the **fourth** least significant bit set: bits 8–15, 24–31, 40–47, etc.

5. In general each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.

If a byte of data to be encoded is 10011010, then the data word (using _ to represent the parity bits) would be __1_001_1010, and the code word is 011100101010.

The choice of the parity, even or odd, is irrelevant but the same choice must be used for both encoding and decoding.

Code:-

```
#include<stdio.h>
void main() {
```

```
int data[10];
int dataatrec[10],c,c1,c2,c3,i;
printf("Enter 4 bits of data one by one\n");
scanf("%d",&data[0]);
scanf("%d",&data[1]);
scanf("%d",&data[2]);
scanf("%d",&data[4]);
//Calculation of even parity
data[6]=data[0]^data[2]^data[4];
data[5]=data[0]^data[1]^data[4];
data[3]=data[0]^data[1]^data[2];
printf("\nEncoded data is\n");
for(i=0;i<7;i++)
printf("%d",data[i]);
printf("\n\nEnter received data bits one by one\n");
for(i=0;i<7;i++)
scanf("%d",&dataatrec[i]);
```

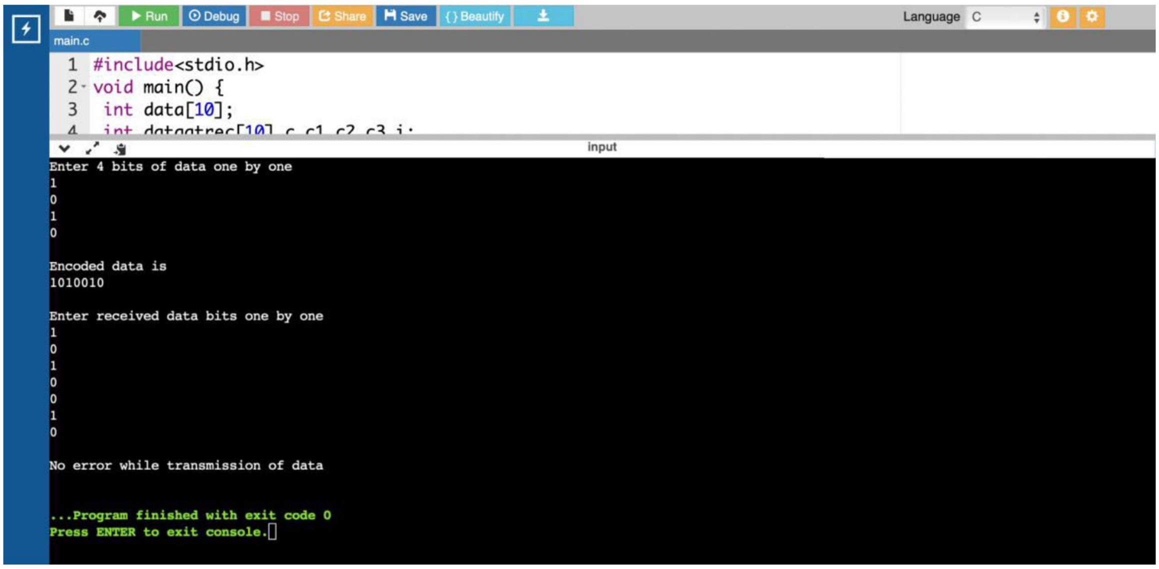
```
c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0
];
c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0
];
c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0
];
c=c3*4+c2*2+c1 ;
 if(c==0) {
printf("\nNo error while transmission of data\n");
 }
else {
printf("\nError on position %d",c);

printf("\nData sent : ");
 for(i=0;i<7;i++)
 printf("%d",data[i]);

printf("\nData received : ");
```

```
 for(i=0;i<7;i++)
 printf("%d",dataatrec[i]);
printf("\nCorrect message is\n");
//if errorneous bit is 0 we complement it else vice versa
if(dataatrec[7-c]==0)
dataatrec[7-c]=1;
 else
dataatrec[7-c]=0;
for (i=0;i<7;i++) {
printf("%d",dataatrec[i]);
}
}
}
```

Output:-



Code for CRC:-

```
#include<stdio.h>
char data[20],div[20],temp[4],total[100];
int i,j,datalen,divlen,len,flag=1;
void check();
int main()
{
    printf("Enter the total bit of data:");
    scanf("%d",&datalen);
    printf("\nEnter the total bit of divisor");
    scanf("%d",&divlen);
    len=datalen+divlen-1;
```

```
    printf("\nEnter the data:");
    scanf("%s",&data);
printf("\nEnter the divisor");
scanf("%s",div);

    for(i=0;i<datalen;i++)
    {
          total[i]=data[i];
          temp[i]=data[i];
    }
    for(i=datalen;i<len;i++)
    total[i]='0';
    check();
    for(i=0;i<divlen;i++)
          temp[i+datalen]=data[i];
    printf("\ntransmitted Code Word:%s",temp);
    printf("\n\nEnter the received code word:");
scanf("%s",total);
    check();
```

```
	for(i=0;i<divlen-1;i++)
		if(data[i]=='1')
		{
			flag=0;
			break;
		}
	if(flag==1)
	printf("\nsuccessful!!");
	else
	printf("\nreceived code word contains errors...\n");
}
void check()
{
  for(j=0;j<divlen;j++)
    data[j]=total[j];
  while(j<=len)
    {
    if(data[0]=='1')
      for(i = 1;i <divlen ; i++)
```

```
                    data[i] = (( data[i] == div[i])?'0':'1');
        for(i=0;i<divlen-1;i++)
data[i]=data[i+1];
        data[i]=total[j++];
    }
}
```

Output:-

input

```
Enter the size of the data array:
7
Enter data bits in the array one by one:
Enter bit 7:
1
Enter bit 6:
0
Enter bit 5:
0
Enter bit 4:
1
Enter bit 3:
1
Enter bit 2:
0
Enter bit 1:
1
Enter the size of the divisor array:
4
Enter divisor bits in the array one by one:
Enter bit 4:
1
Enter bit 3:
0
Enter bit 2:
1
Enter bit 1:
1
1.) First data bit is : 1
Remainder : 0101
2.) First data bit is : 0
Remainder : 1010
3.) First data bit is : 1
Remainder : 0011
4.) First data bit is : 0
Remainder : 0110
5.) First data bit is : 0
Remainder : 1100
6.) First data bit is : 1
Remainder : 1110
7.) First data bit is : 1
Remainder : 1010
101
Generated CRC code is:
1001101101
```

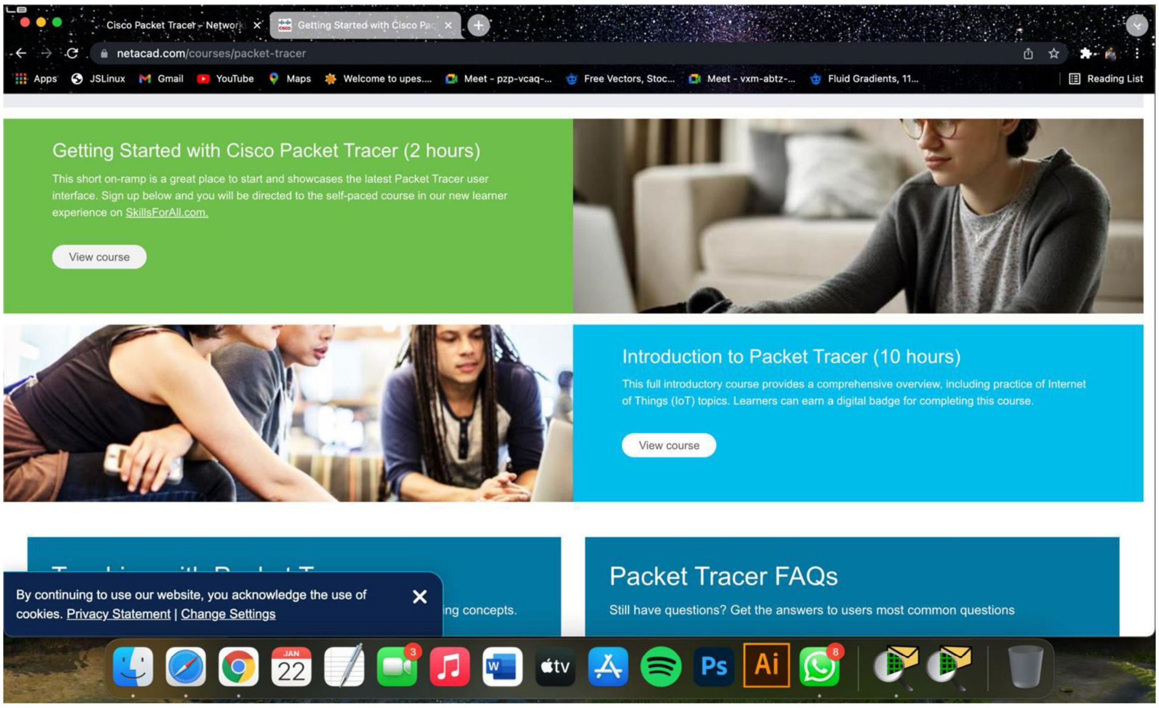
EXPERIMENT 5

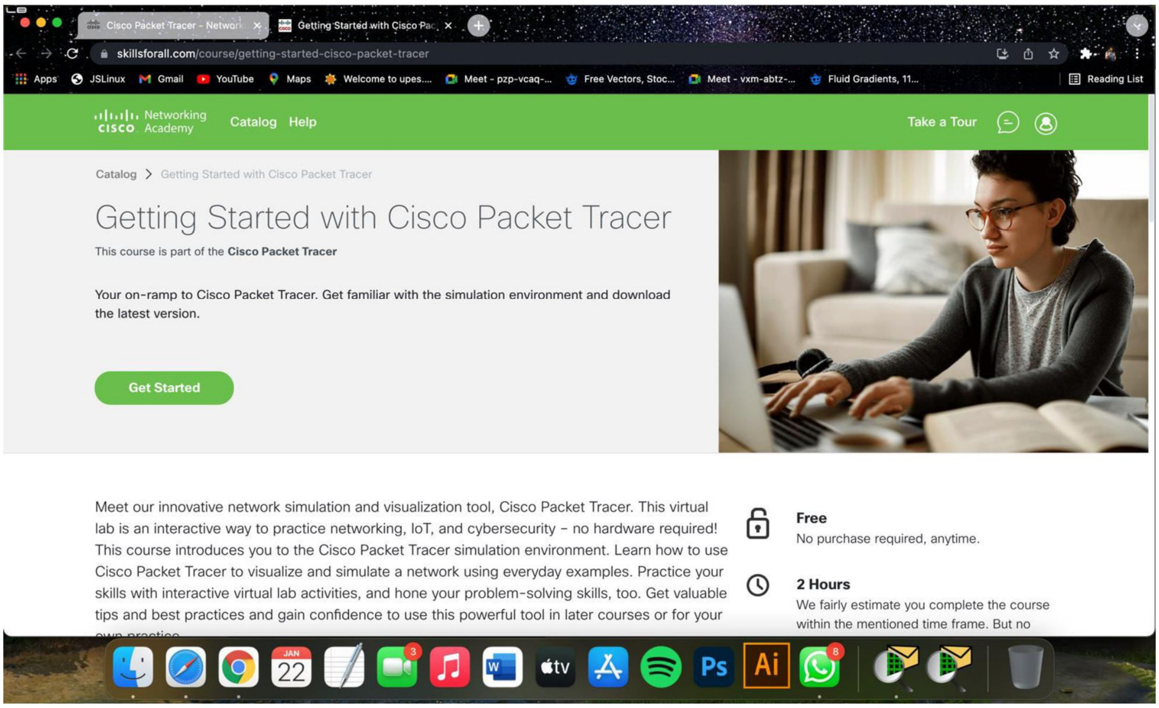
Simulation using Packet Tracer with help of Switch

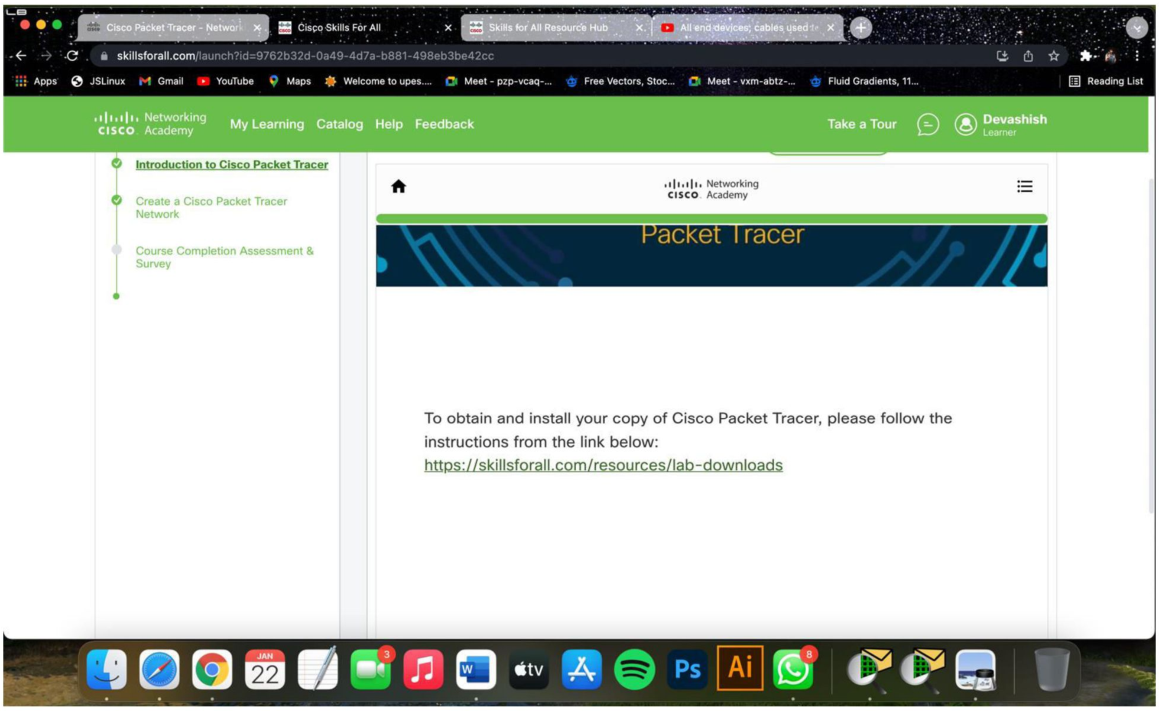
Download Cisco Packet Tracer

https://www.netacad.com/courses/packet-tracer

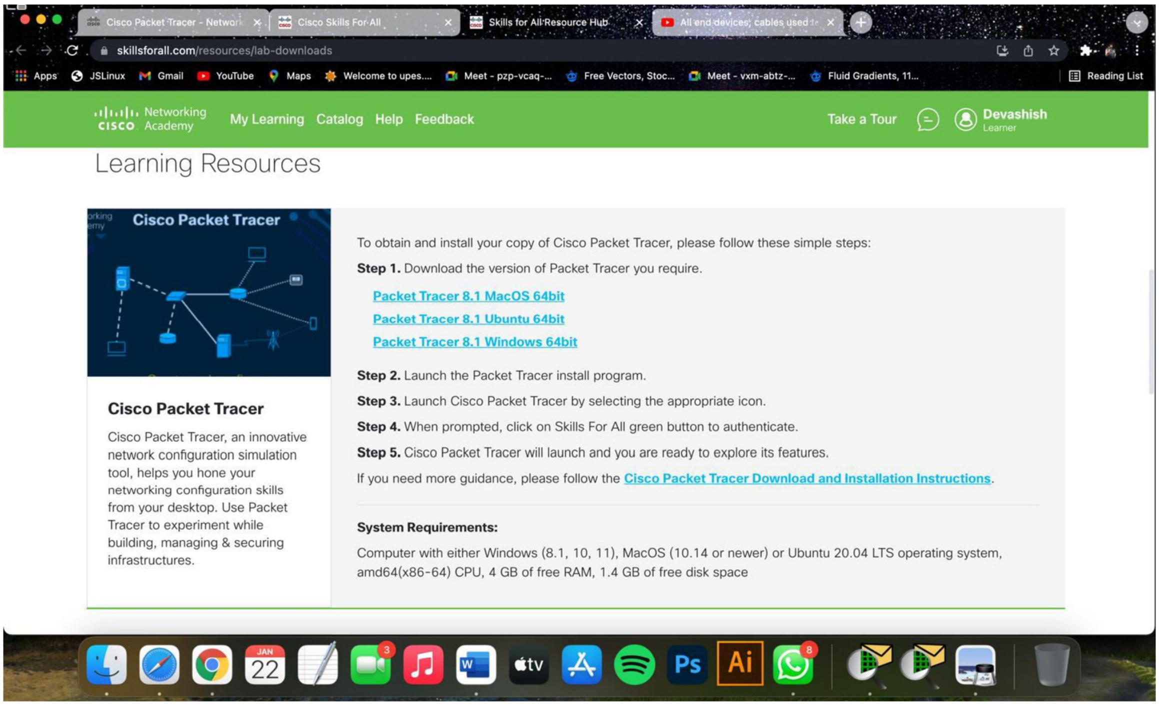
Screenshots :-

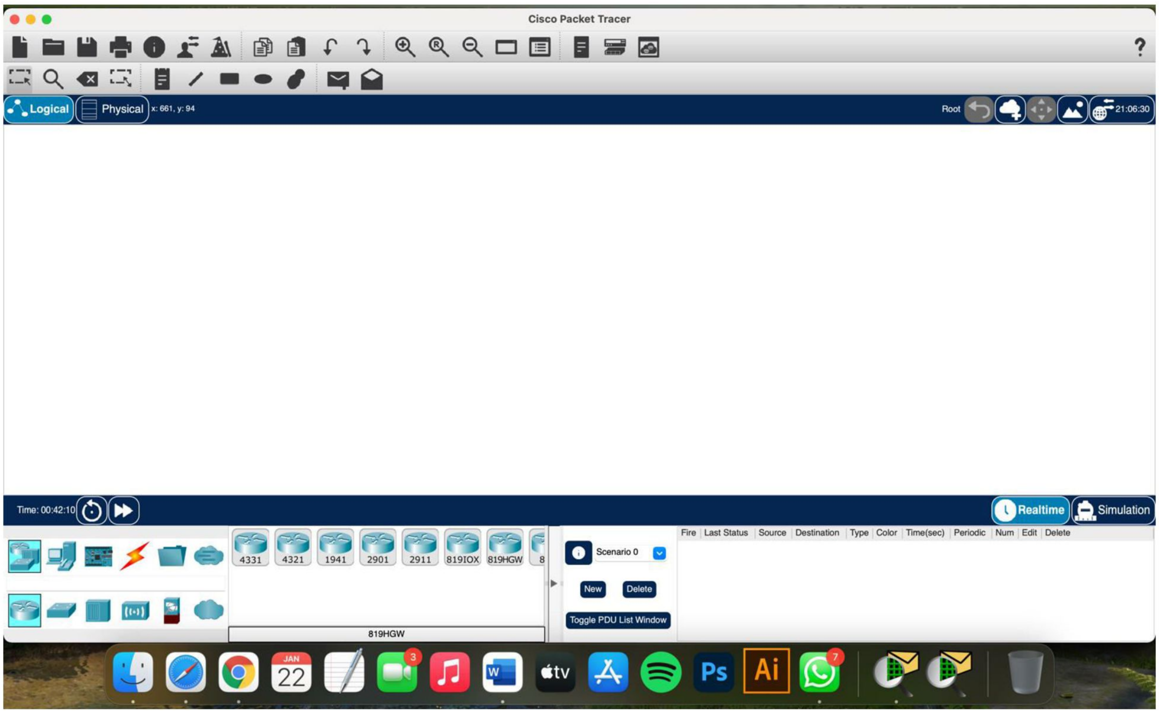






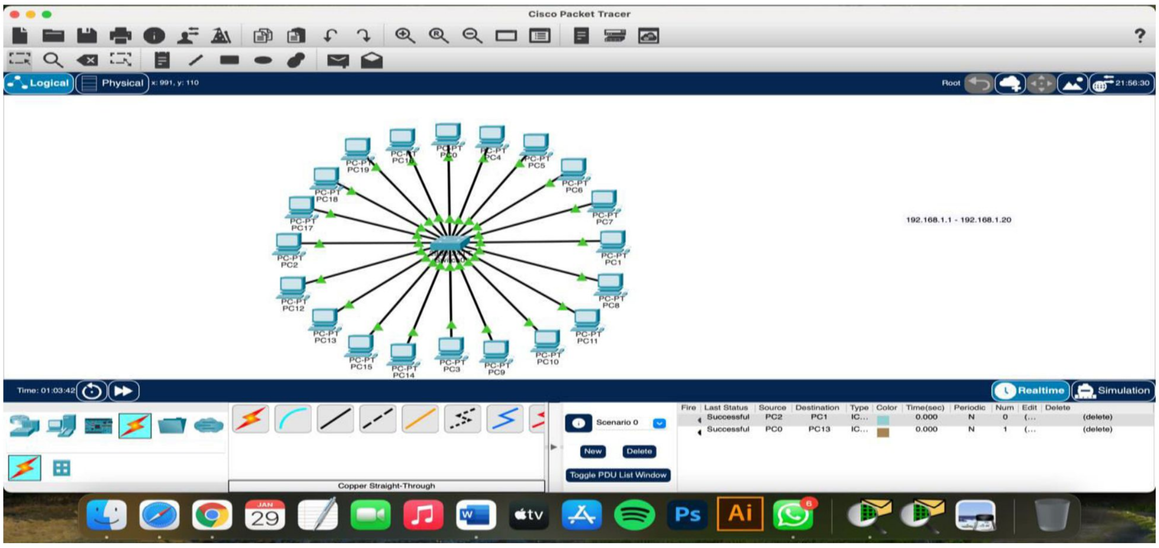
https://skillsforall.com/resources/lab-downloads


Cisco Packet Tracer - Network
Cisco Skills For All
Skills for All Resource Hub
All end devices, cables used
skillsforall.com/resources/lab-downloads
Apps
JSLinux
Gmail
YouTube
Maps
Welcome to upes....
Meet - pzp-vcaq-...
Free Vectors, Stoc...
Meet - vxm-abtz-...
Fluid Gradients, 11...
Reading List
Networking Academy
My Learning
Catalog
Help
Feedback
Take a Tour
Devashish
Learner
Learning Resources
Cisco Packet Tracer
Cisco Packet Tracer
Cisco Packet Tracer, an innovative network configuration simulation tool, helps you hone your networking configuration skills from your desktop. Use Packet Tracer to experiment while building, managing & securing infrastructures.
To obtain and install your copy of Cisco Packet Tracer, please follow these simple steps:
Step 1. Download the version of Packet Tracer you require.
Packet Tracer 8.1 MacOS 64bit
Packet Tracer 8.1 Ubuntu 64bit
Packet Tracer 8.1 Windows 64bit
Step 2. Launch the Packet Tracer install program.
Step 3. Launch Cisco Packet Tracer by selecting the appropriate icon.
Step 4. When prompted, click on Skills For All green button to authenticate.
Step 5. Cisco Packet Tracer will launch and you are ready to explore its features.
If you need more guidance, please follow the Cisco Packet Tracer Download and Installation Instructions.
System Requirements:
Computer with either Windows (8.1, 10, 11), MacOS (10.14 or newer) or Ubuntu 20.04 LTS operating system, amd64(x86-64) CPU, 4 GB of free RAM, 1.4 GB of free disk space


Cisco Packet Tracer
Logical
Physical
x: 661, y: 94
Root
21:06:30
Time: 00:42:10
Realtime
Simulation
4331
4321
1941
2901
2911
819IOX
819HGW
819HGW
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete

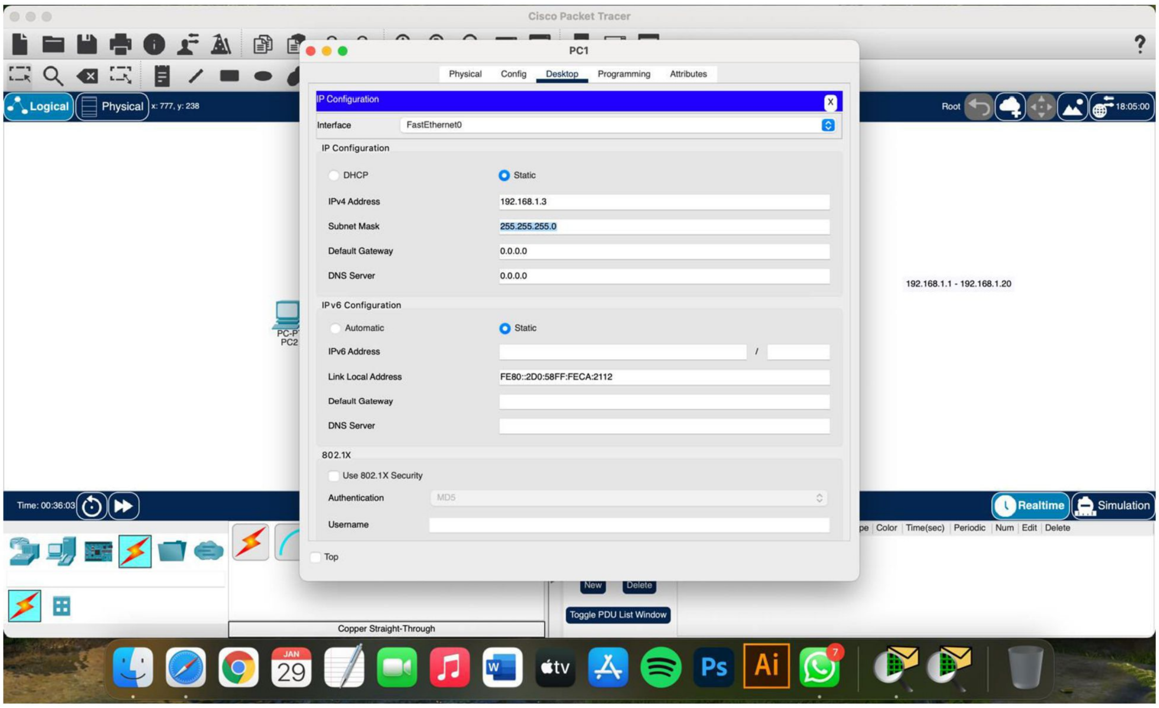
Task 1: Make a connection of 20 PCs using a switch and wires.

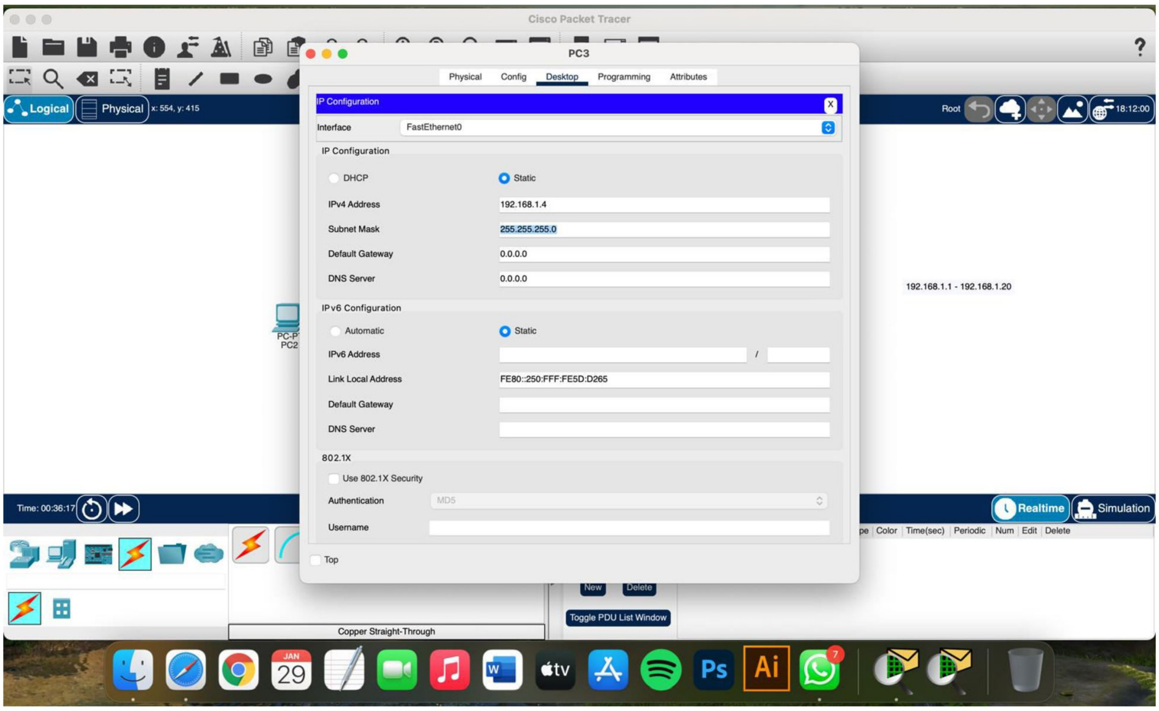
Screenshots :-



Task 2 : Assign every PC a IP address ranging from 192.168.1.1 to 192.168.1.20

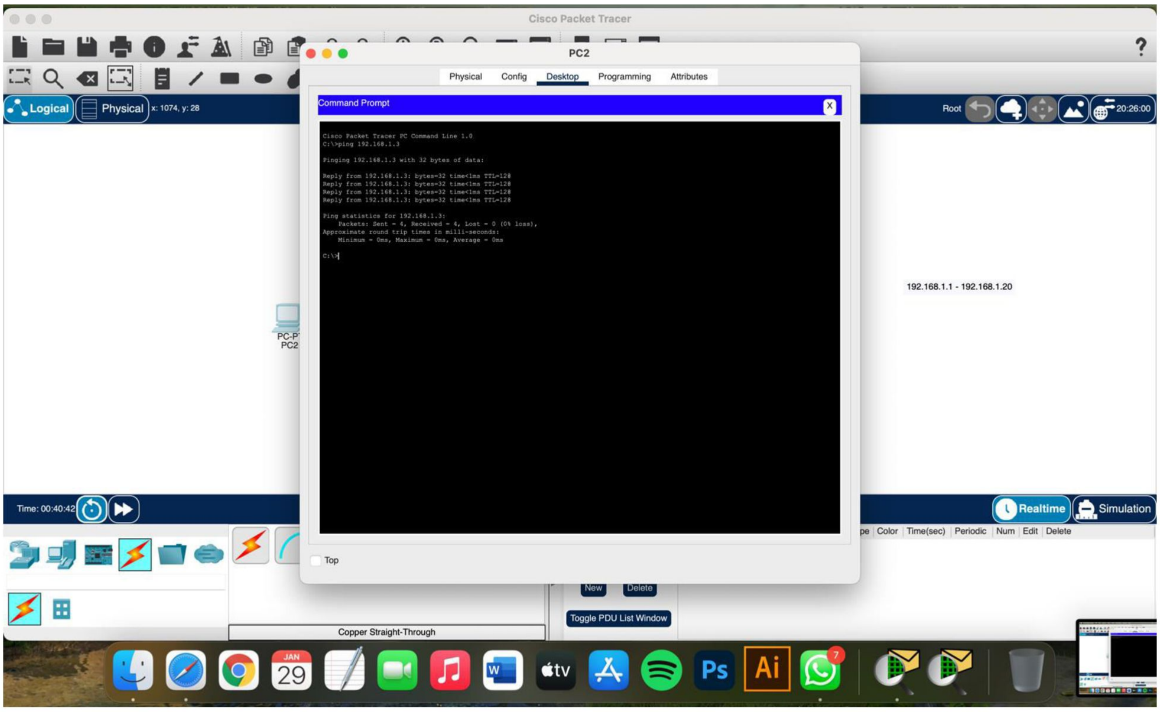
Screenshot:

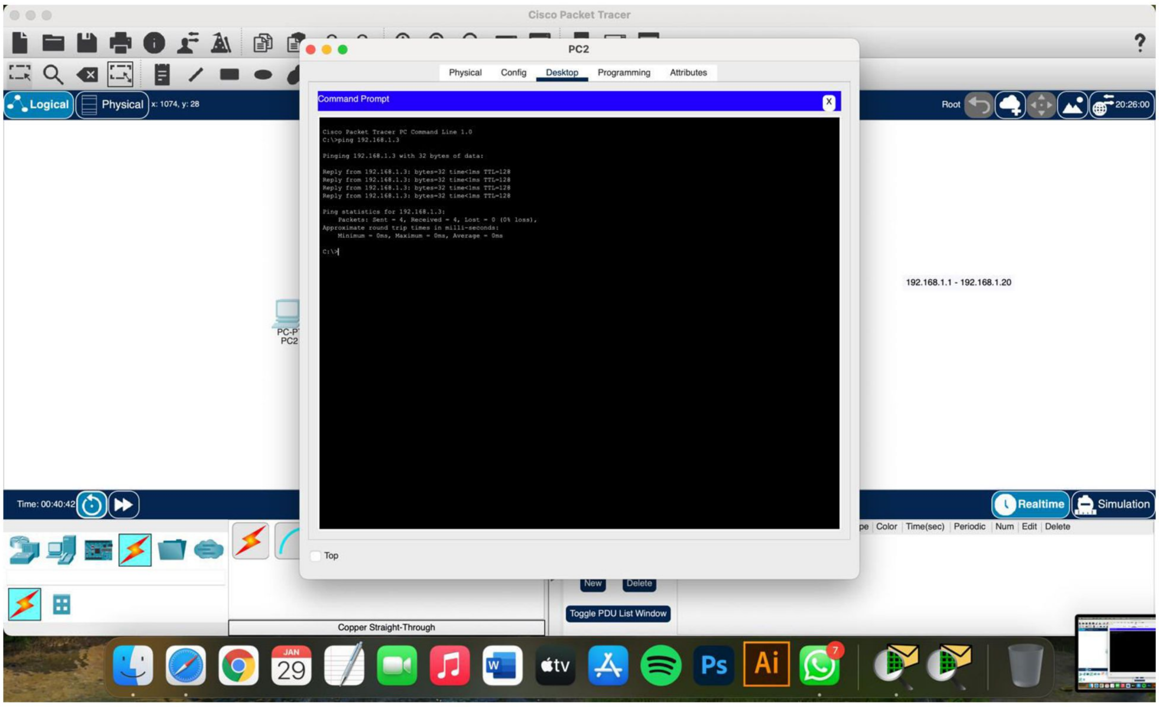

Cisco Packet Tracer
PC1
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
192.168.1.3
Subnet Mask
255.255.255.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::2D0:58FF:FECA:2112
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Top
192.168.1.1 - 192.168.1.20
Realtime
Simulation
Copper Straight-Through
Toggle PDU List Window


Cisco Packet Tracer
PC3
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
192.168.1.4
Subnet Mask
255.255.255.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::250:FFF:FE5D:D265
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Top
192.168.1.1 - 192.168.1.20
Realtime
Simulation
Copper Straight-Through
Toggle PDU List Window

Task 3: Checking the connection by using ping command in cmd prompt

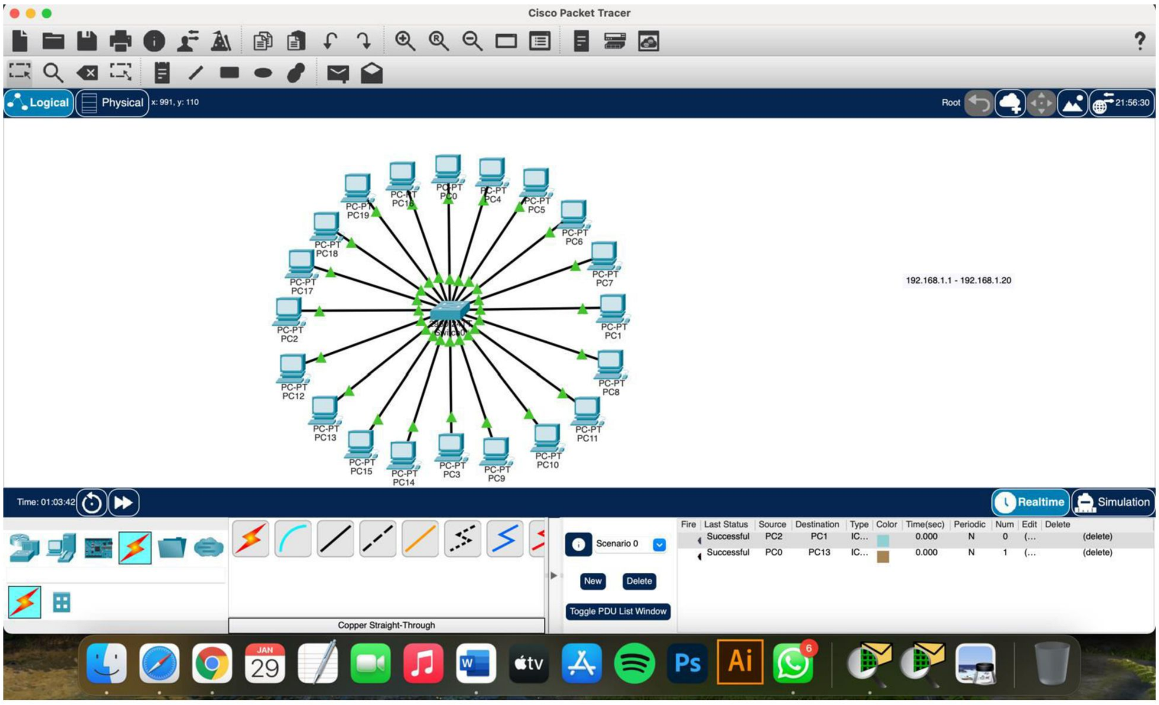
Screenshot:



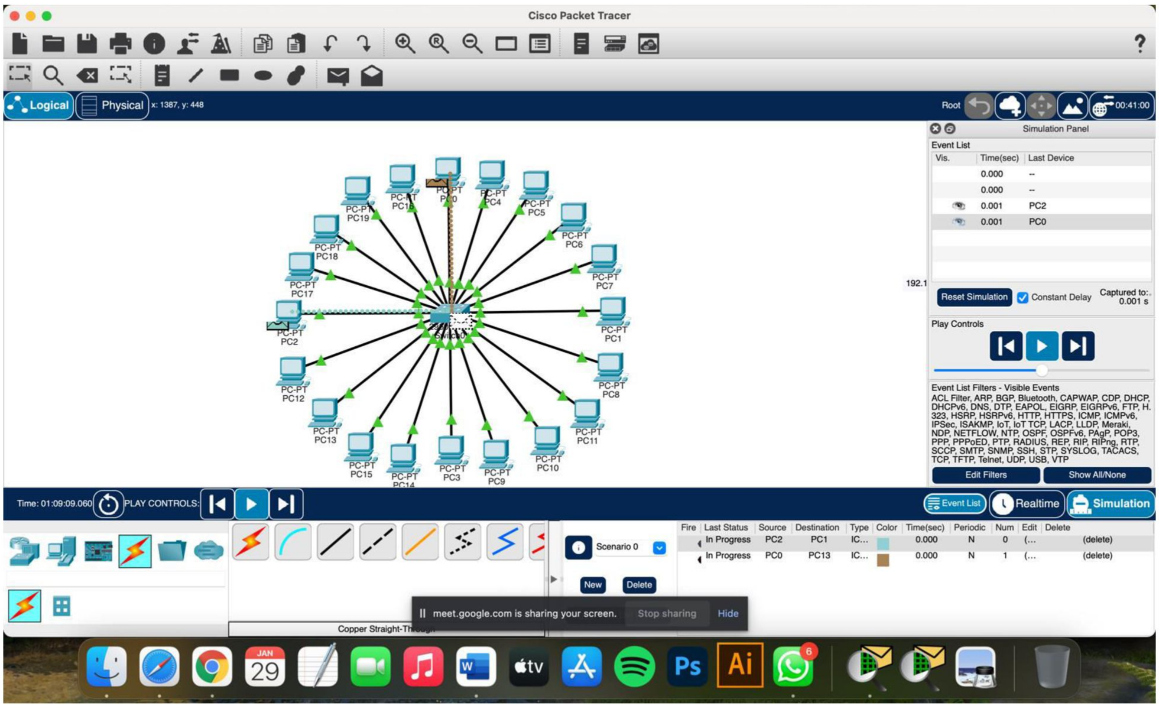


Task 4: Transferring PDU from one PC to another PC through the connection, in simulation mode

Screenshot:



Cisco Packet Tracer
Logical
Physical
x: 991, y: 110
Root
21:56:30
192.168.1.1 - 192.168.1.20
Time: 01:03:42
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Copper Straight-Through
Fire | Last Status | Source | Destination | Type | Color | Time(sec) | Periodic | Num | Edit | Delete
Successful | PC2 | PC1 | IC... | 0.000 | N | 0 | (... | (delete)
Successful | PC0 | PC13 | IC... | 0.000 | N | 1 | (... | (delete)

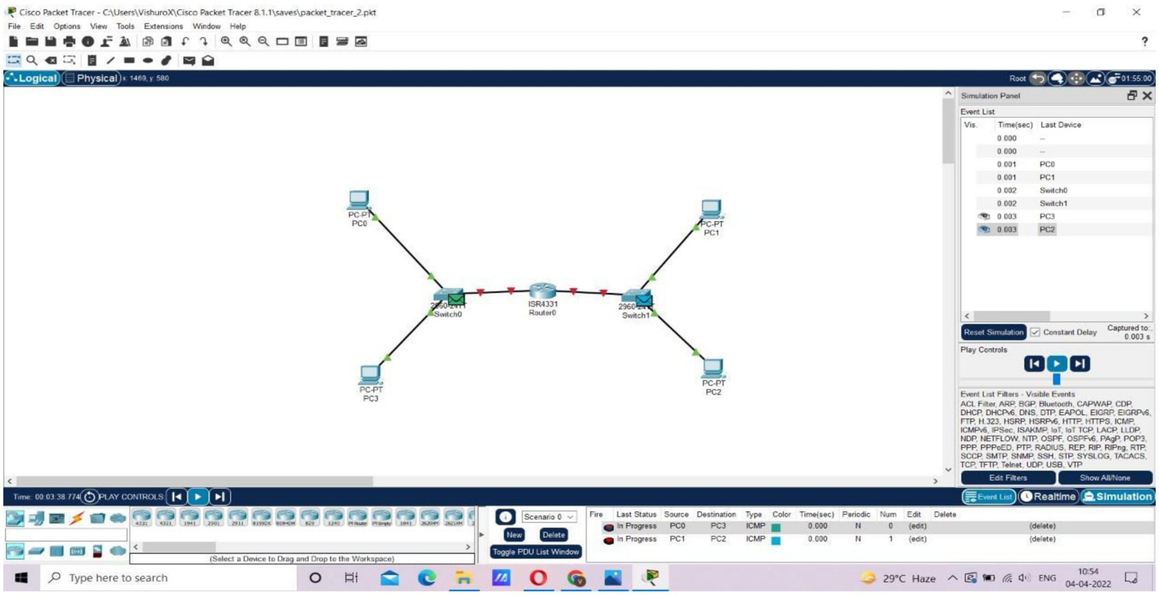


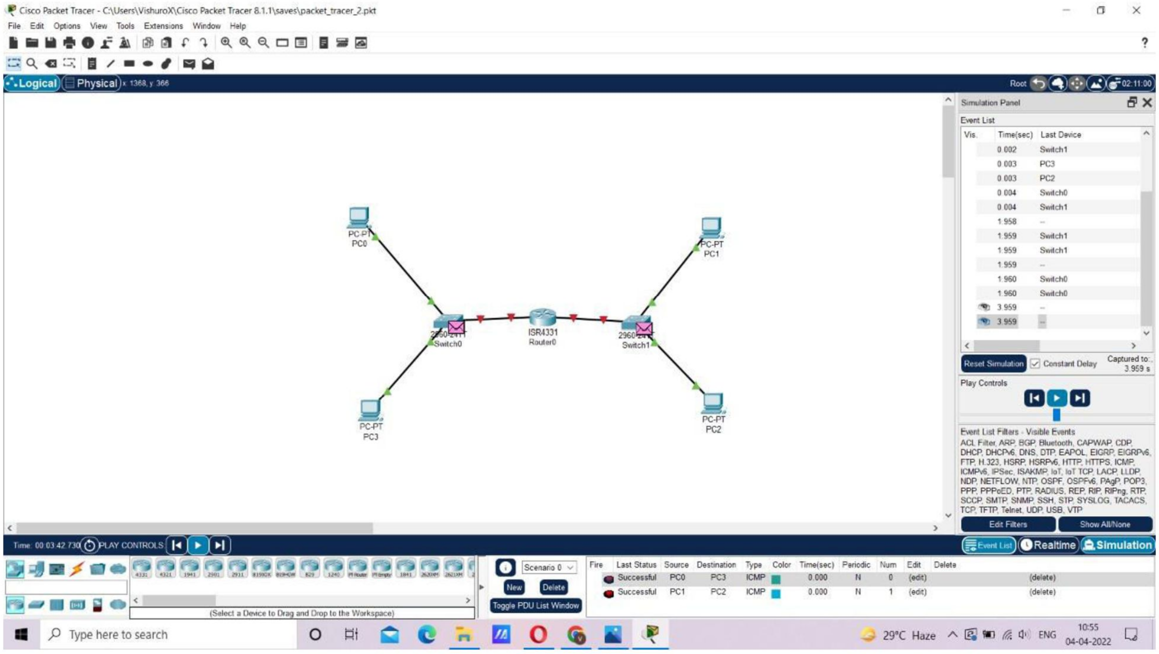
Cisco Packet Tracer
Logical
Physical
x: 1387, y: 448
Root
00:41:00
Simulation Panel
Event List
Vis. | Time(sec) | Last Device
0.000 | --
0.000 | --
0.001 | PC2
0.001 | PC0
Reset Simulation
Constant Delay
Captured to: 0.001 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, 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
Time: 01:09:09.060
PLAY CONTROLS:
Event List
Realtime
Simulation
Scenario 0
New
Delete
Fire | Last Status | Source | Destination | Type | Color | Time(sec) | Periodic | Num | Edit | Delete
In Progress | PC2 | PC1 | IC... | 0.000 | N | 0 | (... | (delete)
In Progress | PC0 | PC13 | IC... | 0.000 | N | 1 | (... | (delete)
meet.google.com is sharing your screen.
Stop sharing
Hide

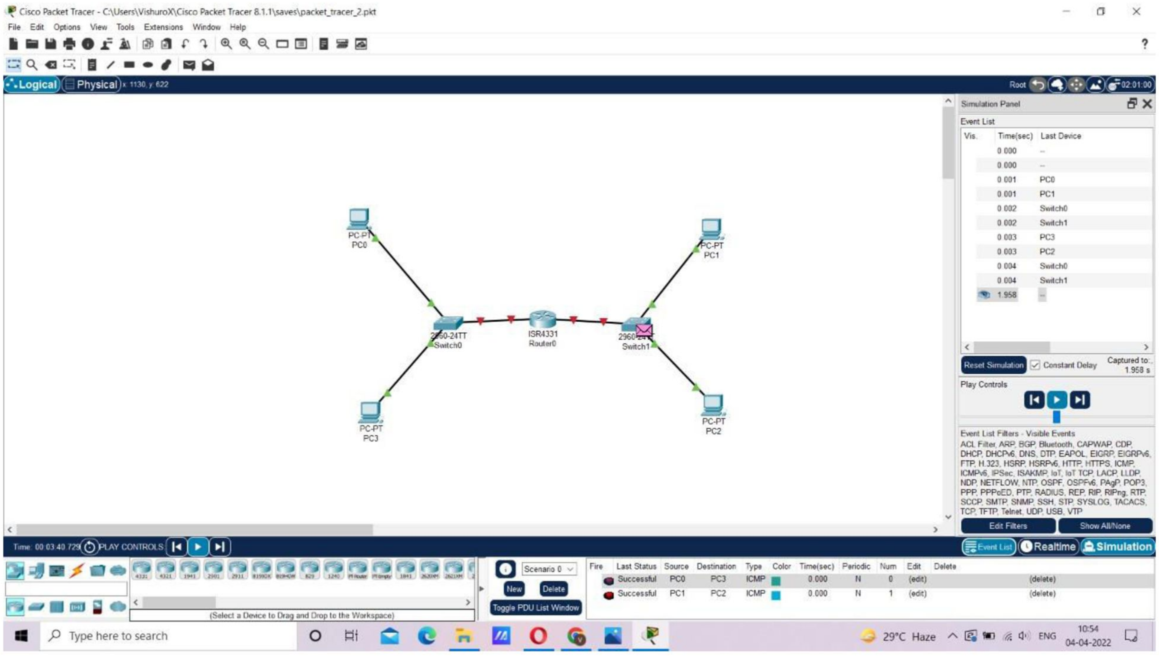
EXPERIMENT 6

Simulation using Packet Tracer with help of router

Screenshot:



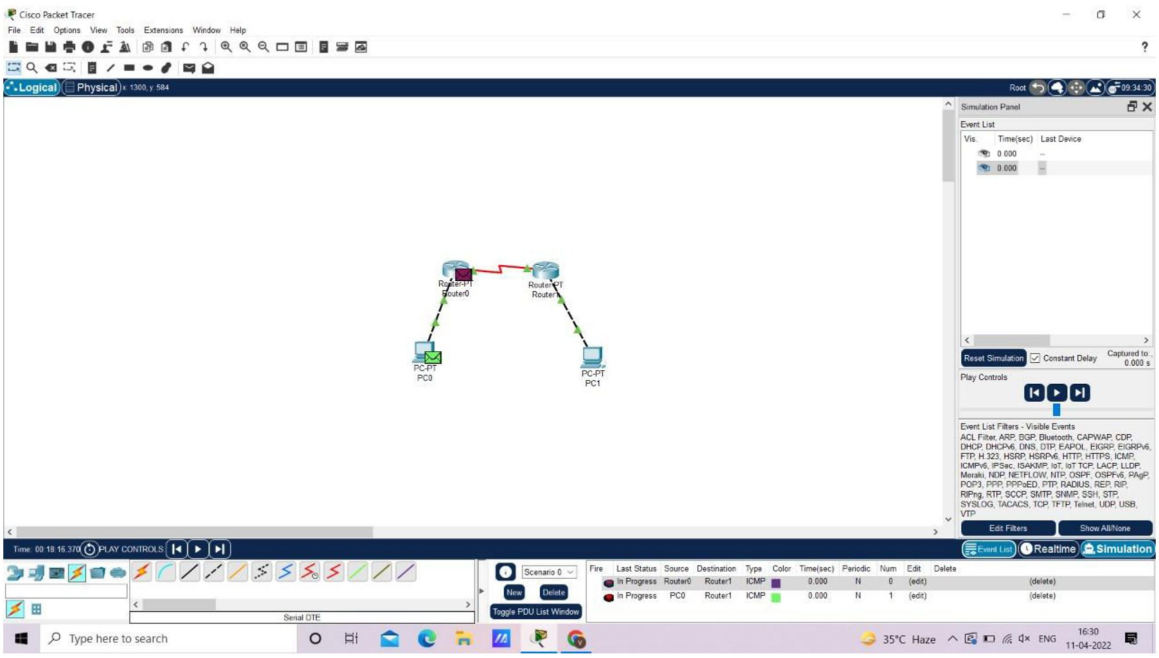

Cisco Packet Tracer - C:\Users\VishuroX\Cisco Packet Tracer 8.1.1\saves\packet_tracer_2.pkt
PC-PT
PC0
PC-PT
PC1
ISR4331
Router0
Switch0
Switch1
PC-PT
PC3
PC-PT
PC2
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Edit Filters
Show All/None
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Type here to search
29°C Haze
ENG
10:55
04-04-2022

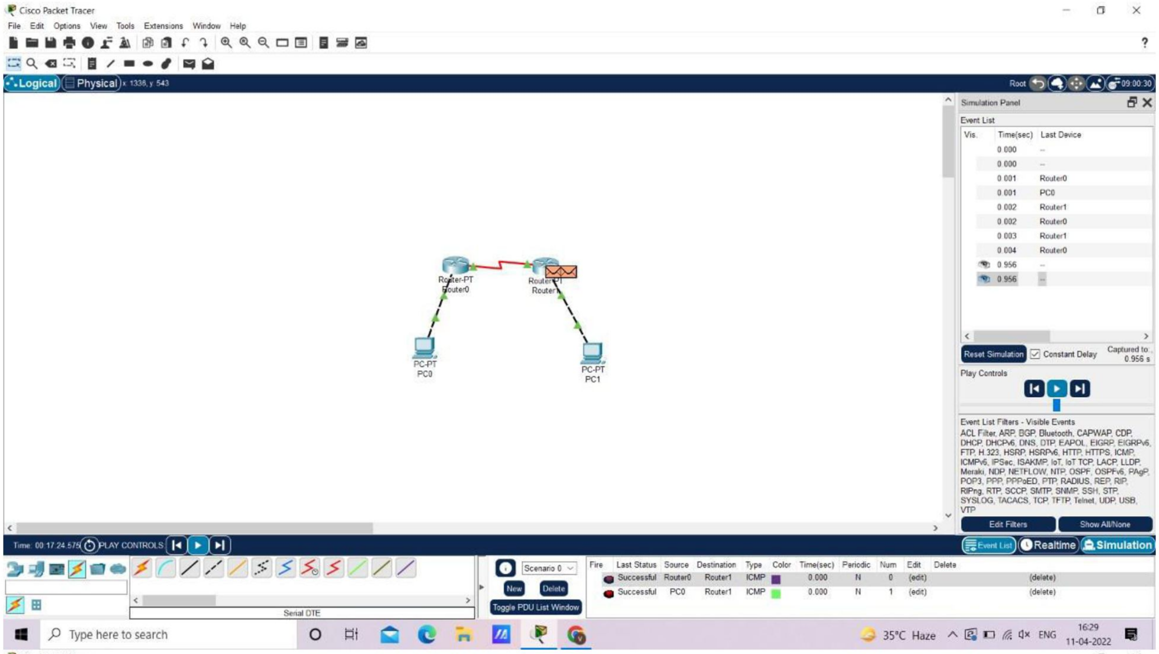

Cisco Packet Tracer - C:\Users\VishuroX\Cisco Packet Tracer 8.1.1\saves\packet_tracer_2.pkt
PC-PT
PC0
PC-PT
PC1
2960-24TT
Switch0
ISR4331
Router0
Switch1
PC-PT
PC3
PC-PT
PC2
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Edit Filters
Show All/None
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Type here to search
29°C Haze
ENG
10:54
04-04-2022

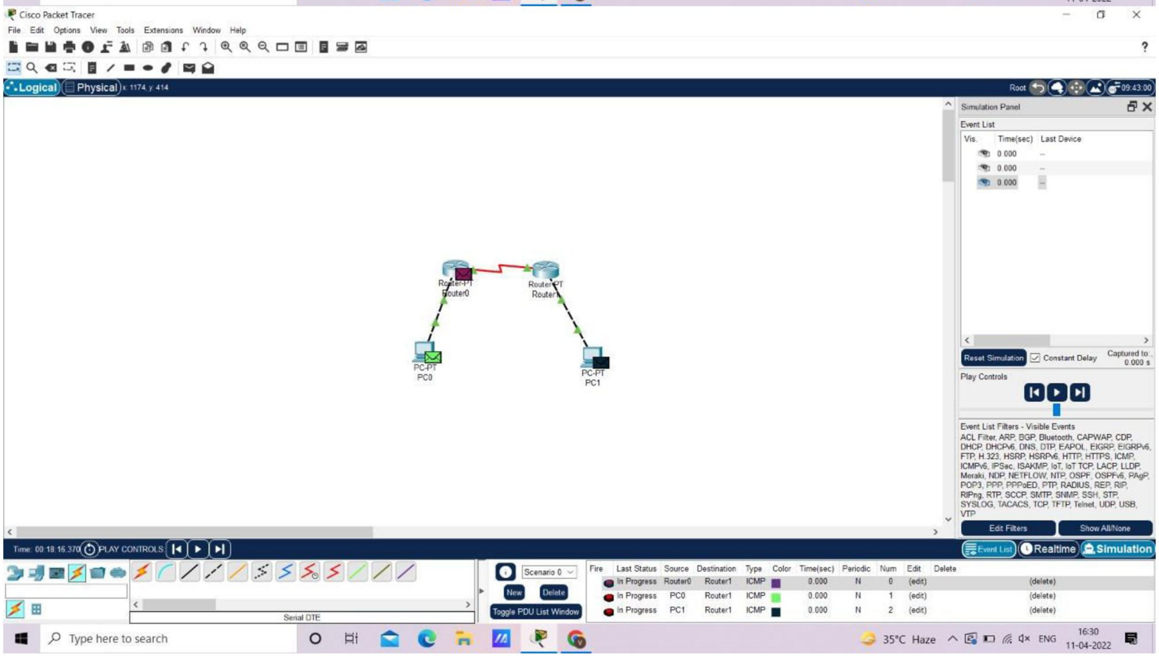
EXPERIMENT 7

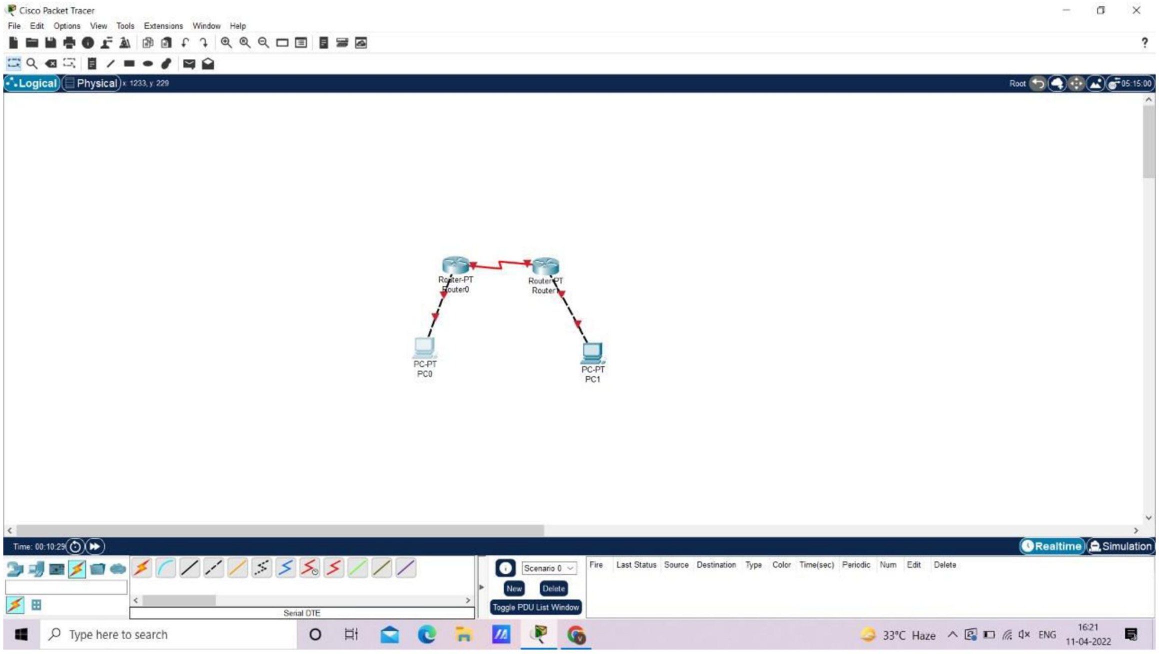
Set up network topology using two routers.

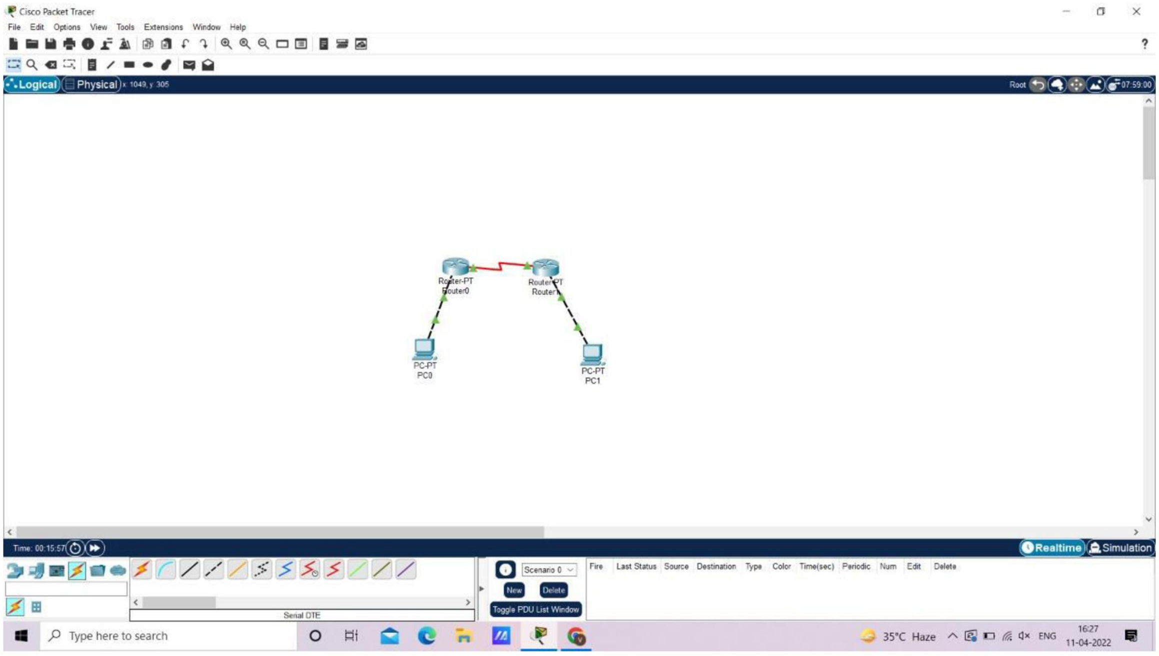
Screenshot:







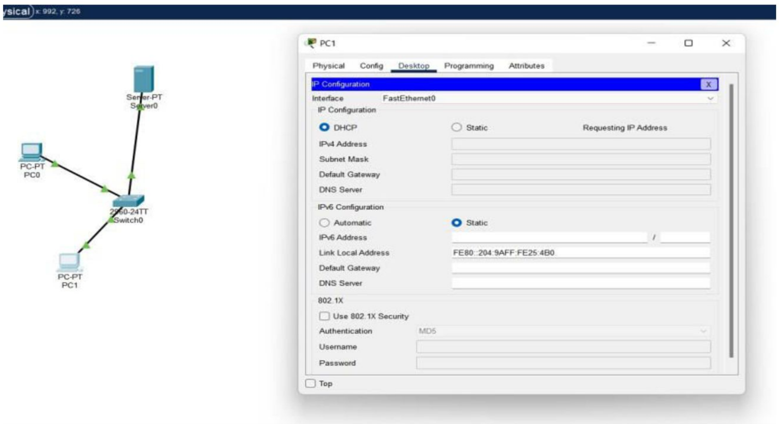


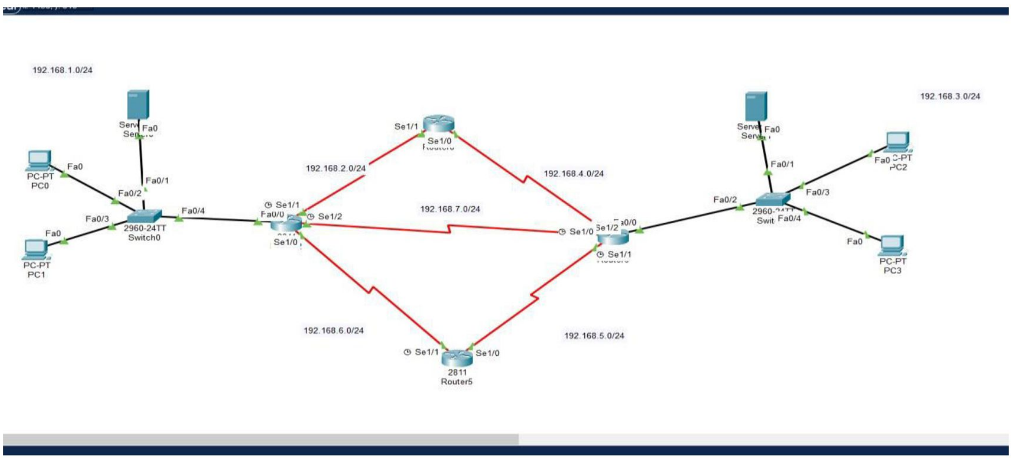


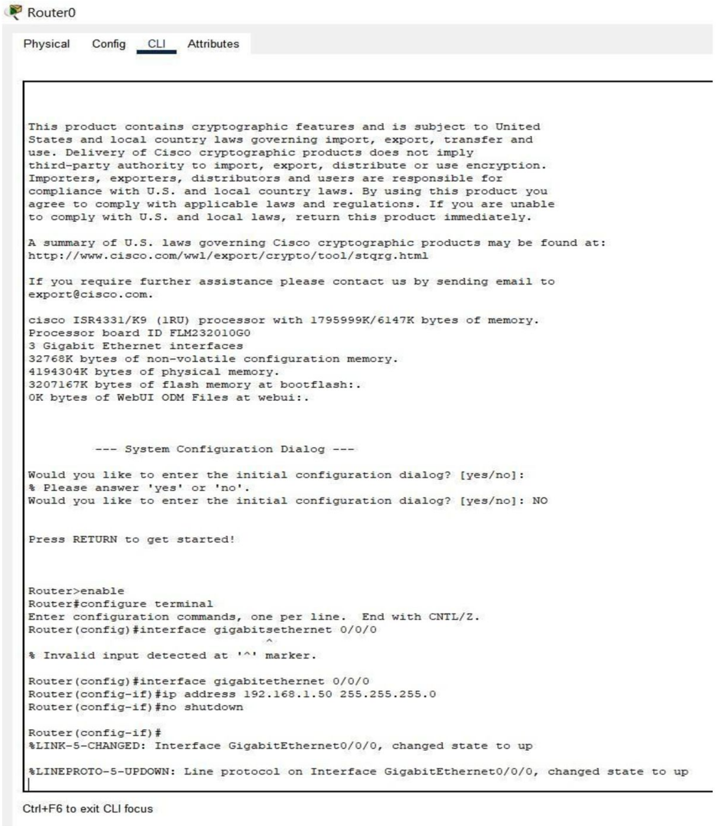
EXPERIMENT 8

Set up network topology using more than two routers

Screenshot:







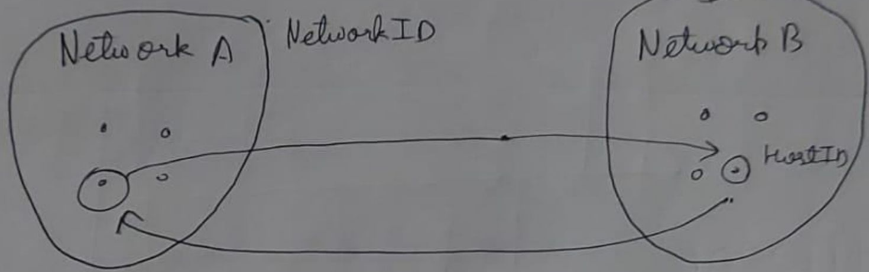
EXPERIMENT 9

IP ADDRESSING

IP Addressing

→ Network layer

↳ IP packets rather than data frames



→ Routing algo + Routing table

A-K-D-B

A-I-J-J-B

↓ they together serve line & effect

→ IP V4

|| 8 bit . 8 bit . 8 bit . 8 bit ||

(0-255) (0-255) (0-255) (0-255)

0, 1

Decimal form (0-9)

→ 32 bits ← length of IP Address

→ 2^{32} IP address / We can connect 2^{32} devices

Disadvantage: Storage of addresses 2^{32} is not enough for a network

→ IPV6

Hexadecimal (0-9) (A-F)

|| 16 bits . 16 bits . 16 bits . 16 bits . 16 bits . 16 bits 16 bits 16

length of IP's in V6 = 128 capacity $= 2^{128}$

IP Version Classes [V_4]

Class A → 32 bits
Class B → 32 bits
Class C → 32 bits
Class D → 32 bits
Class E → 32 bits

2^{32} devices can connect.

Every particular class consists of 32 bits

Maximum length of IP Address is 32 bits.

Class A →

N. ID	Host ID
← 8 bits →	← 24 bits →

Class B →

N. ID	Host ID
← 16 bits →	← 16 bits →

Class C →

N. ID	Host ID
← 24 bits →	← 8 bits →

Class D → Multicasting.
Class E → Research & Experimental work } Reserve

How to Identify the class if address is given:

Class A leading bit will be having 0 (single zero)

$\underbrace{0000\ 0000}_{\text{octet}} . \underbrace{0000\ 0000}_{\text{to}} . \underbrace{0000\ 0000}_{} . 0000-$

$\underbrace{0111\ 1111} . \phi\phi\phi\phi\phi\phi\phi\phi . \phi\phi\phi\phi\phi\phi\phi\phi . \phi\phi\phi\phi\ \phi\phi\phi\phi$

one condition*

$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$64+32+16+8+4+2+1$

$=127.$

0.0.0.0

to

127.000

Start Address	End Address
0.0.0.0 →	127.255.255.255

Total no. of subnets $\Rightarrow 2^7$

No. of hosts $\Rightarrow 2^{24}$

0.0.0.0

127.255.255.255

Class B

For class B, we can identify it as the leading bits are → 10

After that it has ranges like

10000000 · 00000000 · 00000000 · 00000000

to

10111111 · 11111111 · 11111111 · 11111111

128 · ~~to~~ 0 · 0 · 0 to 191 · 255 · 255 · 255

Class C

leading bit → 110

11000000 · 00000000 · 00000000 · 00000000

~~11001111~~ to

11011111 · 11111111 · 11111111 · 11111111

192 to 225

192 · 0 · 0 · 0 to 223 · 255 · 255 · 255

Class D

leading bit → 1110

$\underbrace{11100000}_{224}$ · 00000000 · 00000000 · 00000000

to

$\underbrace{11101111}_{239}$ · 11111111 · 11111111 · 11111111

224 · 0 · 0 · 0 to 239 · 255 · 255 · 255

Class E.

leading bit => 1111

11110000 . 00000000 . 00000000 . 00000000

to

11111111 . 11111111 . 11111111 . 11111111

240.0.0.0 to 255.255.255.255

Class A

No. of sub nets => 2^7

No. of hosts -> 2^{24}

Class B

No. of subnets -> 2^{14}

hosts => 2^{16}

Class C

No. of sub nets -> 2^{21}

hosts = 2^8.

Subnet Mask

Class A — 8 bits NID, 24 bits HID

11111111.00000000.00000000.00000000

255.0.0.0

Class B — 16 bits NID, 16 bits HID

11111111.11111111.00000000.00000000

255.255.0.0

Class C — 24 bits NID, 8 bits HID

11111111.11111111.11111111.00000000

255.255.255.0

Note: Two addresses are always reserved for network and broadcasting

0 → for network

255 → for broadcasting.

How to obtain Network ID-

Q 192.10.15.5

A = 0 to 127

B - 128 to 191

C - 192 to 223

Class is C

Subnet mask is

→ 255.255.255.0

Binary con. 1100.0000.0000010.000 0101

11111111 111111 0000000

AND operation gives NID 11000000.0001010.0001111.00000

decimal 192.10.15.0

Classless Addressing

Concept says there are no classes, we only have network ID & host ID

Representation ⇒ x.y.z.w/n — Network ID/block ID.

200.10.20.40/28

28	4
N.ID	H.ID

⟵ 32 ⟶

2^4 No. of hosts.

11111111.11111111.11111111.11110000

This will give the network ID.

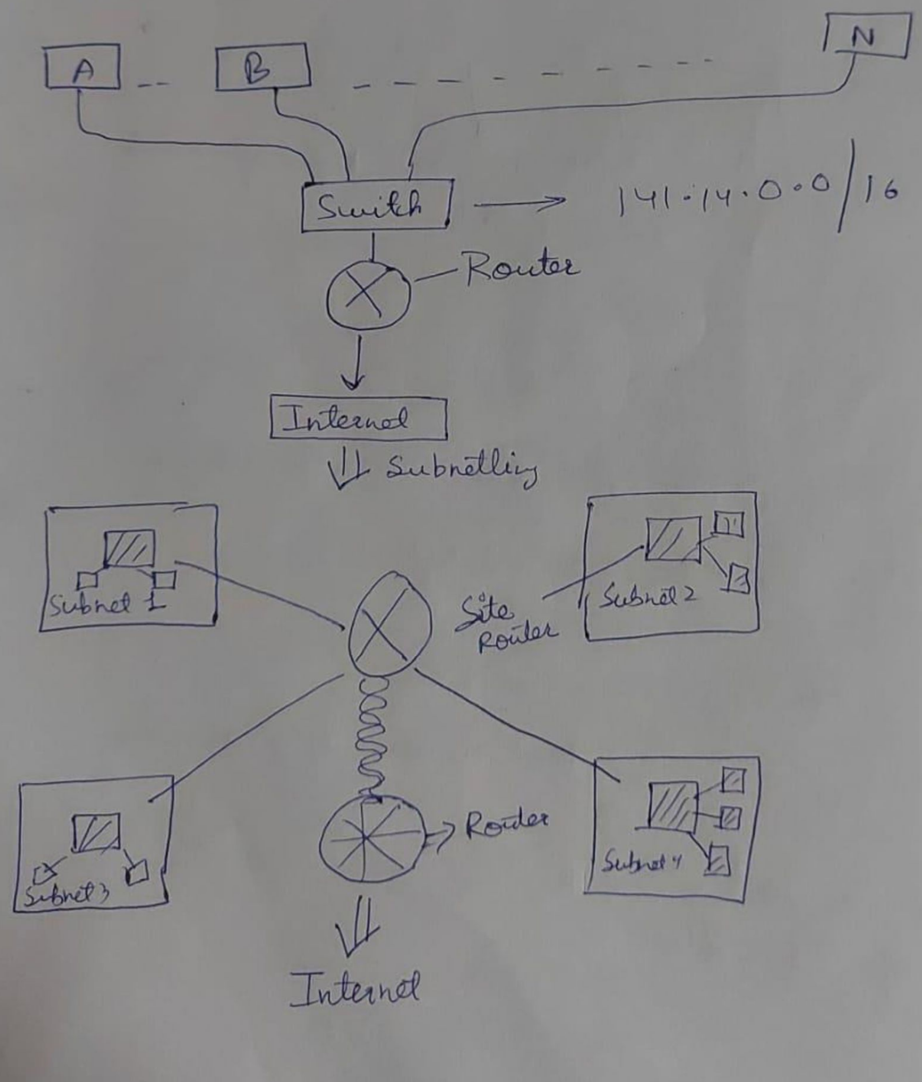
255. 255.255.240

Classless addressing has flexibility
→ Not wasting IP addresses

Division is done at run time so there is no wastage of IP address

Subnetting

It is the division of networks into subnetwork. It is the concept of splitting a block to the smaller blocks. Here, we are dividing the N/w into smaller subnetworks and every subnetwork has its own ID



Length of subnetwork = $\underbrace{n}_{\text{Length of host ID}} + \log_2(n)$ → No. of subnets

For previous example

$= 16 + \log_2 4$

$= 16 + 2$

$= 18$

$\underbrace{200 \cdot 10 \cdot 20} \cdot 0$

will be fixed

Rule is to never change the network bits.

class C

Finding Range in subnetwork.

200.10.20. _ _ _ _ _ _ _ _

↓ fix

S_1

1 bit → 2 subnet

2 bit → 4 subnet

3 bits → 8 subnet

200.10.20.0 _ _ _ _ _ _ _

0 0 0 0 0 0 0 1

↓

0 1 1 1 1 1 1 1

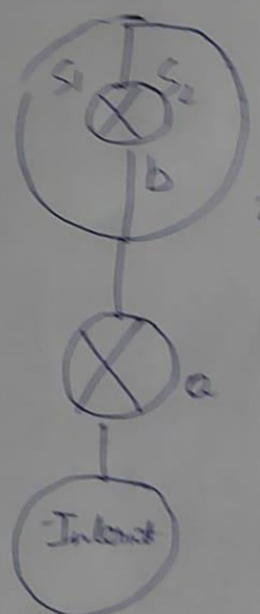
200.10.20.1 _ _ _ _ _ _ _

S2

↓

1 1 1 1 1 1 1 1

Subnet S1	Subnet S2
200.10.20.0 — Subnet ID	200.10.20.1 — Subnet ID
200.10.20.127 — broadcast address	200.10.20.255 — Direct broadcast address



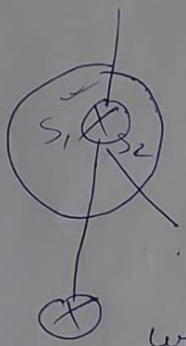
200.10.20.0

lets take IP address 200.10.20.10

Internal router is fixed to solve the confusion of where to send data when more subnetworks are available

Do the AND operation of Subnet mask and given IP address and then check in which range the new network ID falls and than decide in which subnet the data packet will move.

Q IP Address of data = 200.10.20.15



Find subnet mask of internal router

when no. of subnets are 2

∴ bit fixed = 1

255.255.255.1 0 0 0 0 0 0 0

255.255.255.128

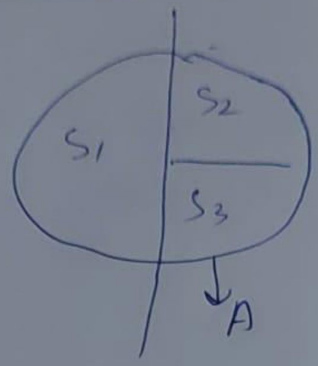
Given IP Address = 200.10.20.15

$\boxed{200.10.20.0}$

↓

S1

Variable Length Subnet Masking



200.10.20.0 _ _ _ _ _ _ _

↓

0 1 1 1 1 1 1 1

$\underline{\underline{S_1}}$

200.10.20.0
to
200.10.20.127

Already fixed

$\underline{\underline{S_2}}$

200.10.20.$\boxed{1}$ _ _ _ _ _ _ _ _

↑

Fix this bit

1 $\boxed{0}$ 1 1 1 1 1 1

200.10.20.128
to
200.10.20.191

] S_2 Subnet

For Subnet S_3 Already fixed

200.10.20. $\boxed{1}$ $\boxed{1}$ 0 0 0 0 0 0

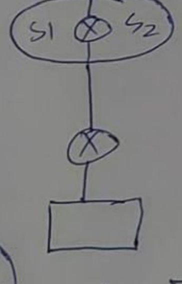
6

$\boxed{4}$

1 1 1 1 1 1 1 1

200.10.20.192
to
200.10.20.255 } S_3

Subnetting in CIDR (Classless Interdomain routing)



195.10.20.128/26

10 000000 → Host ID

fix the first host bit

S_1

10 $\boxed{0}$ 00000

10 0 11111

195.10.20.128/27

195.10.20.159/27

Changes from 26 to 27

S_2

10 $\boxed{1}$ 00 000

10 1 11111

195.10.20.160/27

to

195.10.20.191/27

Changes from 26 to 27

because we have converted the entire IP network 195.10.20.128/26 into two ~~subdomain~~ subnetworks by fixing 1 more bit.

$$\underbrace{26}_{\text{Earlier number of fixed bits}} + 1 = 27$$

Supernetting

→ opposite of subnetting.
→ Get all IP addresses in binary.
→ Find matching bits from left to right
→ Rewrite matching bits and add remaining zeroes.
→ Find new subnet mask for getting

$\boxed{\text{Network ID + Host ID}}$

Ex A

1) 192.168.0.0/24
2) 192.168.1.0/24
3) 192.168.2.0/24
4) 192.168.3.0/24

1) 11000000 · 10101000 · 00000000 · 00000000
2) 11000000 · 10101000 · 00000001 · 00000000
3) 11000000 · 10101000 · 00000010 · 00000000
4) 11000000 · 10101000 · 00000011 · 00000000

8 | 8 | 6 bits

Common bits

22 bits ⇒ 11000000 · 10101000 · 000000 00 · 0000

$\boxed{8+8+6}$

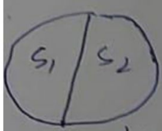
192.168.0.0

Get the subnet mask ? →
N/w bits = 1
host bits = 0

11111111 · 11111111 · 11111100 · 00000000

$\boxed{255 \cdot 255 \cdot 252 \cdot 0}$

Example B



200·10.10·0

Class C

2 subnet

200·10·10·0
to
200·10·10·127

On fixing 0

200·10·10·128
to
200·10·10·255

On fixing 1

Subnet S1

Subnet ID

200·10·10·0

Direct Broadcast Address

200·10·10·127

Subnet S2

~~200to~~
200·10·10·127

200·10·10·255