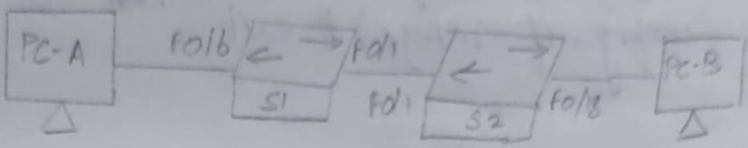
Ex.no: 8d) a) SIMULATE VIRTUAL LAN CONFIGURATION
Date: 9/9/24 USING CISCO PACKET TRACER

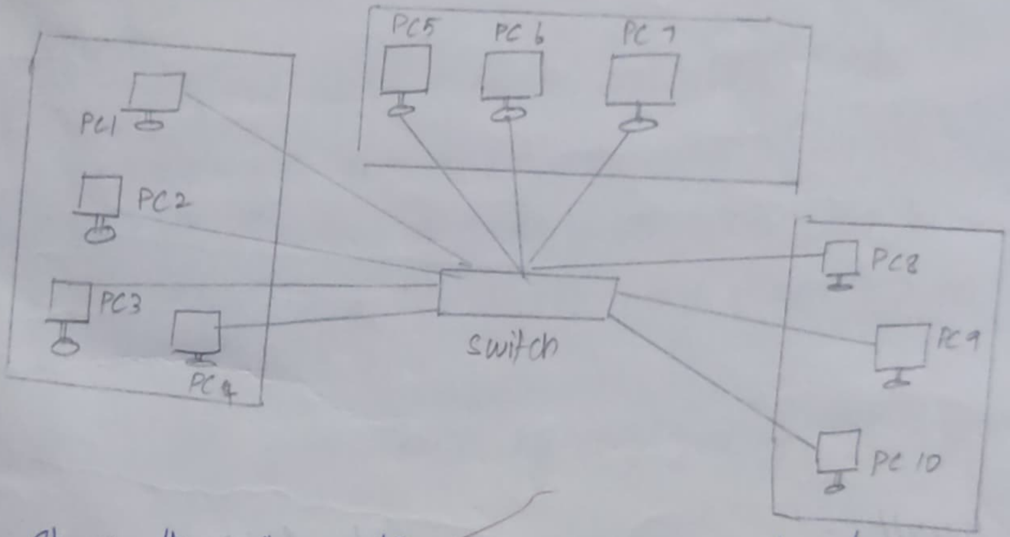
Aim :-
Simulate virtual LAN configuration using CISCO Packet Tracer simulation

Packet Tracer - Configure VLANs and Trunking - Physical Mode Topology



Student Observation

a) Draw and label the VLAN



b) Show the ip configuration for each device

Device	IP Address	Subnet Mask	Default Gateway
PC1	192.168.20.2	255.255.255.0	192.168.20.1
PC2	192.168.20.3		
PC3	192.168.20.4		
PC4	192.168.20.5		
PC5	192.168.20.6		
PC6	192.168.20.7		
PC7	192.168.20.8		
PC8	192.168.20.9		
PC9	192.168.20.10		
PC10	192.168.20.11		

c) Write the commands used for VLAN config in switch

```
switch > enable
switch # configure terminal
switch (config) # vlan 10
switch (config-vlan) # name Robotics
switch (config-vlan) # exit
switch (config) # interface range f 0/1 -10
switch (config-if-range) # switch port access vlan 10
switch (config-if-range) # exit
```

Result :-

Thus, the simulation of virtual LAN configuration using CISCO packet tracer has been performed and the output is verified

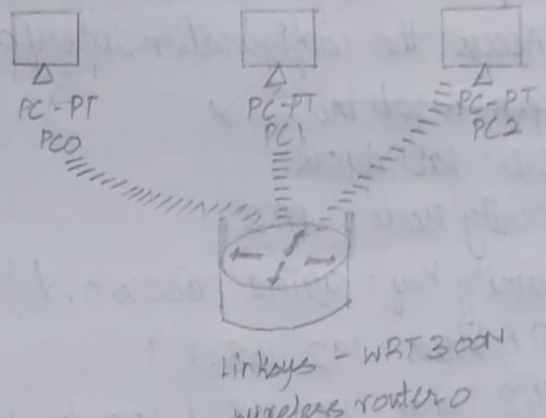
Ex. NO: 8b)
9/9/24

Practical - 8b)

AIM:-

b) configuration of Wireless LAN using CISCO Packet Tracer

Design a topology with three PCs connected from Linksys wireless routers



Procedure:-

- Configure static IP on PC and Wireless Router
- Set SSID to MotherNetwork
- set IP address of router to 192.168.0.1, PC0 to 192.168.0.2, PC1 to 192.168.0.3 and PC2 to 192.168.0.4
- Secure your network by configuring WAP key on Router
- Connect PC by using WAP key

Student Observation:

c) What is SSD of a wireless router?

(Service Set Identifier) is the name of a wireless network. It helps user identify and connect network when multiple networks are available.

d) What is a security key in a wireless router?

A security key is a password used to secure a wireless network It prevents unauthorized access.

e) A simple wireless LAN

1. Connect to Access Point (AP)

Use ethernet to connect to the computer to the access point

2. Access the configuration Interface.

~~open a web bro~~

SSID: labNetwork

Security Mode: WPA2

Security Key: Secure P@ssword!

IP Address: 192.168.1.1

DHCP Range: 192.168.1.100 to 192.168.1.200

RESULT:

Thus the program to configure Wireless LAN using cisco packet tracer is successfully executed

9/9/24

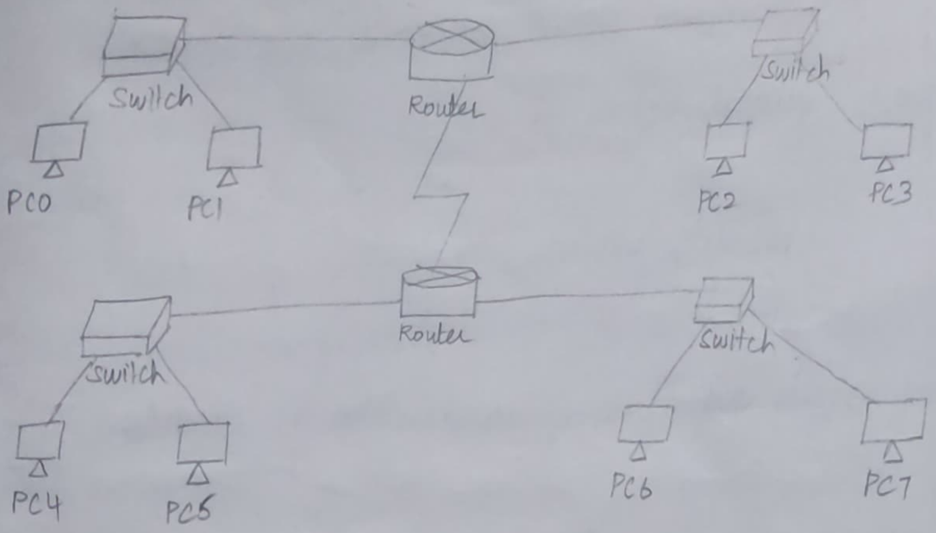
EX. NO: 9
21/9/24

SUBNETTING IN CISCO PACKET TRACER

AIM:
Implementation of subnetting in cisco packet tracer simulation

STUDENT OBSERVATION QUESTIONS.

a) Draw your implementation of subnetting.



b) Write down your understanding of subnetting.

Classless IP subnetting is a technique that allows for more efficient use of IP addresses by allowing for subnet masks that are not just the default masks for each IP class. This makes that we can divide our IP address space into smaller subnet, which can be useful when we have a limited no. of IP addresses but need to create multiple networks

Result:-

Thus the program to configure subnetting in CISCO PACKET TRACER is implemented successfully

29/9/4

Ex.No: 10
25/9/24

INTERNETWORK USING DHCP SERVER

AIM :-

Design and configure on internetwork using DHCP server

student observation :-

1) Write down the key features of configuring router & DHCP server

Router Configuration on:

-> Enabling DHCP server to automatically assign IP addresses to devices

-> Providing connection to the devices.

DHCP Server configuration.

-> Automatic IP address assignment for device joining the network

-> Setting up pool ranges for dynamic IP address

-> Configuring default gateway

2) What is the significance of DHCP server in internet working

* A dynamic host configuration protocol server simplifies IP address

* Automated IP Address allocation.

* No duplicate IP Addresses

* supports large-scale networks

* Provides a centralized point for managing IP address

Result Thus the program to configure DHCP server is implemented successfully

EX.NO:11a)
9/10/24

PRACTICAL - 11a)

AIM:

a) Simulate static Routing Configuration using CISCO Packet Tracee.

b) Simulate RIP using CISCO Packet tracer

(a) 1. Adding Static Routes: each muter knows only the networks directly connected to it add static route to reach a network not directly connected

Eg:- Router0, networks 10.0.0.0/8, 20.0.0.0/8 & 40.0.0.0/8 are directly connected, but 30.0.0.0/8 & 50.0.0.8

2) Creating Main & Backup Routes
Administrative Distance decides preference of routes: the lower the AD, the higher the preference.

3) Router Configurations
configure static routes on each router for networks not directly connected

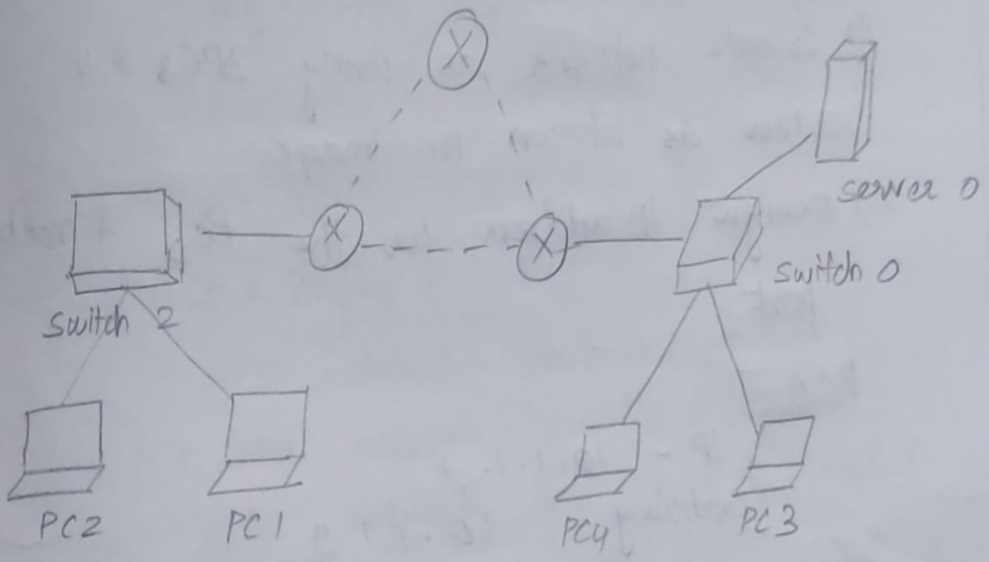
4) verifying Router:
verify routes by using commands.
Show ip route static

5) Testing Route Fail over:
→ Testing connectivity using tracert a ping from a device on a connected network.

-> Disconnect or "break" the link on the main route

6) Deleting a Static Route:

show ip route static



(b) (1) Initial IP configuration for devices

(2) Assign IP addresses to Devices for PCs and Routers

(3) Enable & Configure Interfaces on Routers

(4) Configure RIP on Routers

(5) verify and Test Redundancy

- use ping command on PC1

- use tracer to see RIP directory

Result:

Thus the static routing configuration using cisco packet tracer successfully

Ex. NO: 11b)
9/10/24

Practical - 11b)
RIP

AIM

To simulate RIP using cisco packet tracer

PROCEDURE

1) Create network as using 3PCs & 4 routers as shown in image
2) Assign IP address for the Pc, & router port

PC0

IP - 10.1.1.1
Gateway : 10.1.1.2

PC1

IP - 200.1.1.1
Gateway : 200.1.1.2

PC2

IP - 222.2.2.2
gateway - 222.2.2.12

Router 3

gig 0/0 - 20.1.1.1
0/1 - 192.168.1.1
0/2 - 10.1.1.1

Router 2

gig 0/0 - 20.1.1.2
0/1 - 172.1.1.1
0/2 - 200.1.1.2

Routee 1

gig 0/0 - 192.168.1.3
0/1 - 172.1.1.2
0/2 - 217.1.1.1

Routee 4

gig 0/0 - 217.1.1.2
0/1 - 222.2.2.12

3. Click on router 3

-> click config -> RIP
-> enter network 10.0.0.0 -> Add.
-> enter network 20.0.0.0 -> Add
-> enter network 192.168.1.0 -> Add
Thus step is done inorder to add the neighbouring network address for router 3

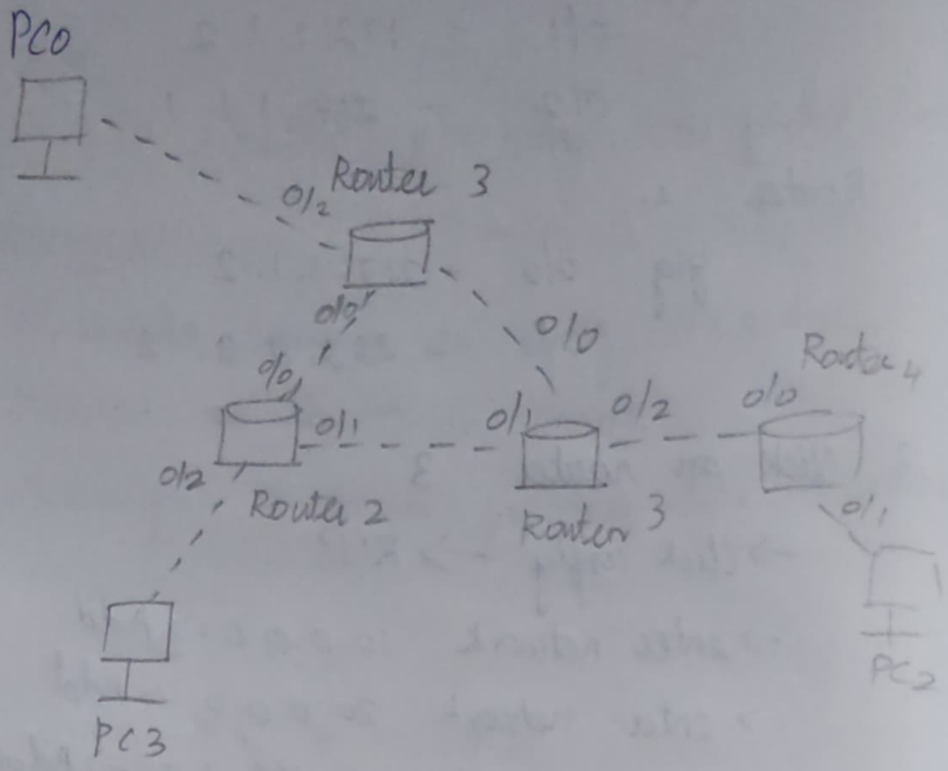
4. Do same for Router 2, 1, 4

5. Now to display the routing table click on router (say router 1)

-> then on CLI & type the command

```
#exit
#exit
#show ip route
```

Diagrammatic representation



Output

R - 10.0.0.0/8 via 192.168.1.1 gig 0/0

R - 20.0.0.0/8 via 192.168.1.1 gig 0/0

172.1.0.0/16 is variable connected 2 subnet mask

C - 172.1.0.0/16 is directly connected gig 0/1

L - 172.1.1.2/32 is directly connected gig 0/1

19/9/24

Result:
Thus RIP is simulated using cisco packet trace successfully

EX.NO: 12a)
16/10/24

PRACTICAL - 12a) ECHO CLIENT SERVER

AIM

a) Implement echo client server using TCP/UDP sockets

client

```
import socket
import time
def ping-server (host = '127.0.0.1', port=12345):
        with socket.socket (socket.AF_INET,
                    Socket.SOCK_DGRAM) as S;
        try:
            S.sendto (b "Hello", (host, port))
        except s.timeout:
            print ("Request timed out ")

        If  --name-= 'main ':
                ping-server ()
```

Server:

```
    import socket
    def start_server (host = '127.0.0.1', port=12345 )
        with socket.socket (socket.AF_INET, socket_SOCK
                                        DGRAM) as S:
            s.bind ((host, port))
            print ( f "UDP server running on
                        {host}:{port}
        while True :
            data, addr : s.recvform (1024 )
```

```
print(f" Received message from {add.
                                {data.decode()}")

if __name__ == "__main__":
    start_server()
```

O/P:-

```
python server.py
UDP server running on 127.0.0.1:1259
Received message from ('127.0.1', 5720)
                                     Hello

python client.py
Received reply from server: Hello, client
```

Result:

Thus the program for echo client server is successfully executed & output is verified

19/9

EX.NO: 12b) 16/10/24

12b) Implement chat client server using TCP/UDP Sockets.

```
chat serv.py
import socket
def recvr():
        port = 12345
        host = '127.0.0.1'
        with socket.socket(socket.AF_INET, socket.
                                SOCK_DGRAM) as S:
          s.bind((host, port))
          while(True)
          d; add - s.recvfrom(1024)
          print("client ", {d.decode()})
          a = input("Enter reply")
          s.sendto(a.encoded(), add)
          if(a == "end"):
                break
                exit

recvr1()
```

```
recvr2.py

    import socket
    import time
    def recvr2(a):
        host = '127.0.0.1'
        port = 12345
```

```
with socket. socket (socket . AF_INET, Socket
                     SOCK-DGRAM) as s:

    s. sendto (a. encode(), (host, port))
    d, addr = s. recvfrom (1024)
    print ({d. decode()})
while(True):
    a = input ("Enter Message ")
    if (a=='end"):
        recvr2(a)
         break
    else:
        recvr2(a)
```

O/P:

```
python \chat-serv.py
client {'hi'}
Enter reply hello
Client {"How are you"}
Enter Reply Im fine

python .\recvr.py
   Enter Message hi
   {'hello'}
   Enter Message How are you
   {'Im fine'}
   Enter Message
```

Result:

Thus the program for chat client server successfully executed & the output is verified

EX. NO: 13
19/10/24

PRACTICAL 13 PING PROGRAM

AIM

Implement your own ping program

Server.py

```
import socket
def start_server ( host = '127.0.0.1', port =12345):
        with socket. socket (socket.AF_INET,
                Socket.SOCK_DGRAM) as S.
          s.bind ((host, port))
          print ("UDP Server running on {host}:
                                   {port}")
        while True:
          data, addr = s.recvfrom (1024)
          print (f" Received message from {addr}:
                            {data.decode()}")
          s.sendto (b'Pong', addr)

  if __name__ == "__main__":
        start_server ()
```

client.py

```
import socket
import time
def ping-server(host='127.0.0.1', port=12345).
    with socket.socket(socket.AF_INET, socket.
                                SOCK_DGRAM) as s:

     try:
        s.settimeout(2)
        start = time.time()
        sendto(b 'Ping', (host, port))
        except socket.timeout:
            print("Request timed out")

if __name__ == "__main__":
        ping-server()
```

output

```
python server.py
UDP server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 53009): ping

python client.py
Received pong from ('127.0.0.1', 12345) in
                0.00 seconds.
```

Result:-

Thus the ping program is successfully executed & the output is verified.

Ex.No: 14
23/10/24
AIM:

14) Packet Sniffing

```
from scapy.all import sniff
from scapy.layers.inet import IP,TCP,UDP,ICMP

def packet_callback(packet):
    if IP in packet:
        ip_layer = Packet[IP]
        protocol = ip_layer.proto
        src_ip = ip_layer.src
        dst_ip = ip_layer.dst

        #Determine the protocol
        protocol_name = ""
        if protocol == 1:
            protocol_name = "ICMP"
        elif protocol == 6:
            protocol_name = "TCP"
        elif protocol == 17:
            protocol_name = "UDP"

        else:
            protocol_name = "Unknown Protocol"

        print(f"Protocol: {protocol_name}")
        print(f'Source IP: {src_ip}")
        print(f"Destinati on IP: {dst_IP}")
        print("-"*50)
```

```
def main():
    sniff (iface =" Wi-Fi", prn=packet_callback, filter=' ip',
                                                     Store=0)

if _name_ = = "_main_":
    main()
```

O/P :

```
Protocol : TCP
Source IP: 20.247.164.142
Destination IP: 172.20.10.2

Protocol : TCP
Source IP: 20.247.164.142
Destination IP: 172.20.10.2
```

Result:

Thus the packet sniffing program is successfully executed & the output is verified.

19/8/24

Ex: NO: 15
23/10/24

Practical - 15

Aim :-

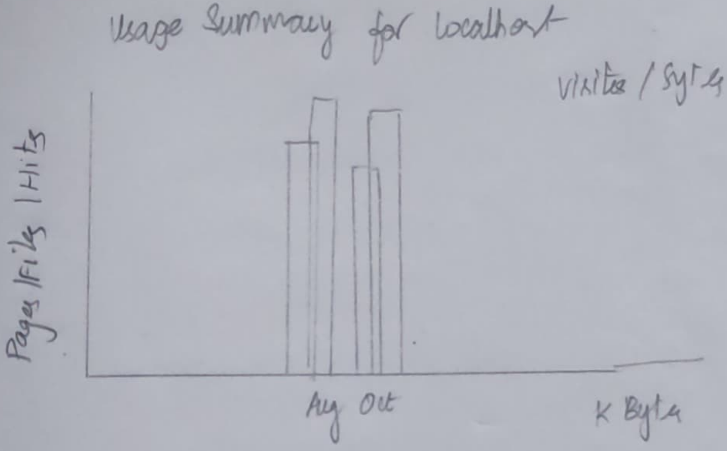
To analyze the different types of weblogs using webalizer tool.

Procedure :

Step 1 : Run webalizer windows version

Step 2 : Input Web log file

Step 3 :- Press run webalizer



Result :- Thus the different types of weblogs is analyzed

19/11/24