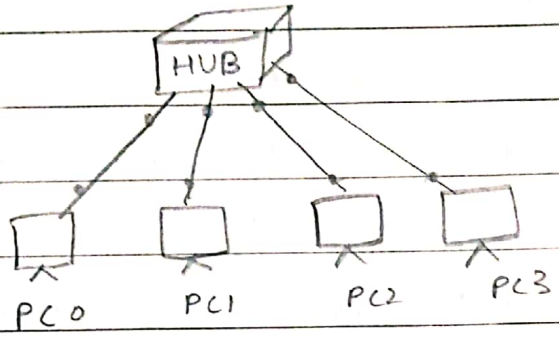
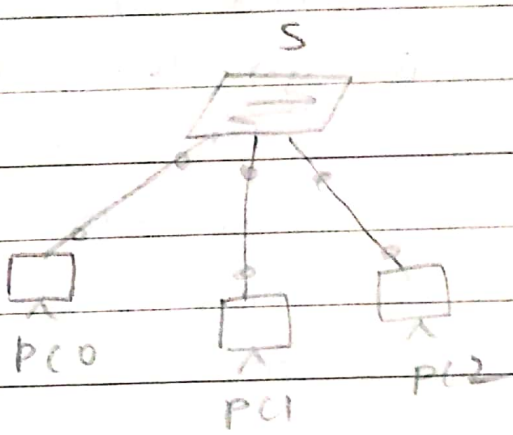
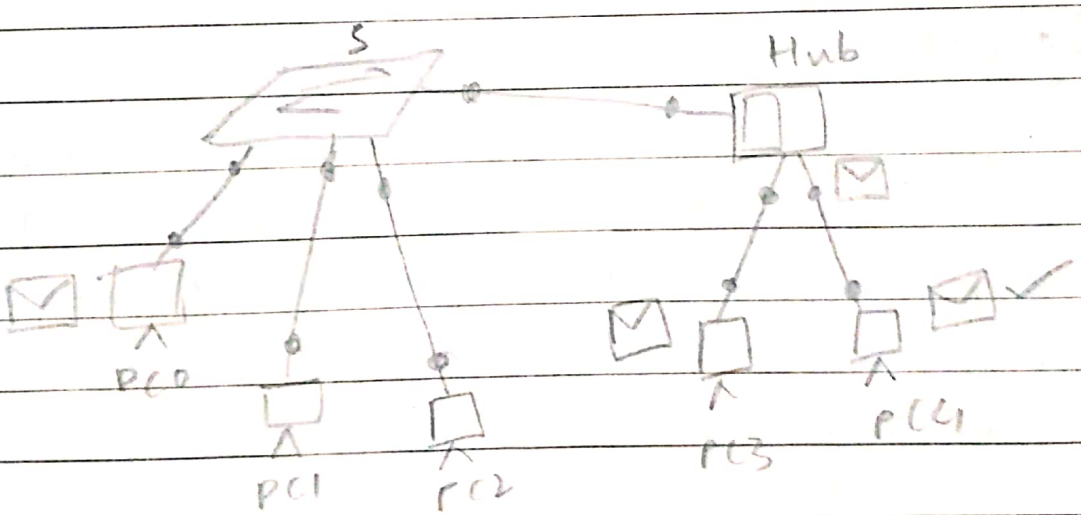
Exp : 1

16/6/23

1. Create a topology consisting of three or more devices connected with a hub







1. Insert end devices and Hub or router
2. assign IP address to the end devices (PC)
3. Connect the PC to Hub or switch
4. Send a simple PDU from one PC to another.
5. In command Prompt ping an ip address.

Observation

1. Hub broadcasts all received packets to all connected devices
2. In switch the connection status is initially red after around 30s it turns green
3. Switch initially broadcasts to all devices later updates the switching table and forwards only to the device
4. Ping sends and receives packets from source and destination device

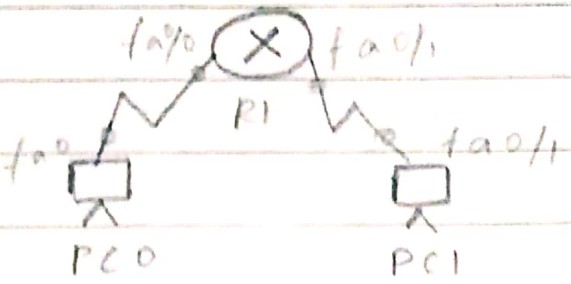
Result

Pinging 192.168.0.2 with 32 bytes of data
Reply from 192.168.0.2 : bytes = 32 time = 2ms TTL
Reply from 192.168.0.2 bytes = 32 time = 0ms TTL
Reply from 192.168.0.2 bytes = 32 time = 0ms TTL
Reply from 192.168.0.2 bytes = 32 time = 0ms TTL

Exp: 2

23/6/23

2. Create a topology consisting of two devices connected with the help of a router.



1. insert end devices (PC) and configure ip address (10.0.0.1 and 20.0.0.1)
   PC0 PC1
2. insert a router (Generic PT) and connect the PC with serial DCE cabel.
3. Configure the router IP address in the command terminal

```
Router >enable
Router #configure terminal
Router (config) #interface fa 0/0
Router (config-if) #ip address 10.0.0.1 255.0.0.0
Router (config-if) #no shutdown
exit.
Router (config) #interface fa1/0
Router (config-if) #ip address 20.0.0.1 255.0.0.0
Router (config-if) #no shutdown
```

4. Ping message from the command terminal.

Observation

1. If DNS gateway is not configured the

ping results in host unreachable

2. The network ID is the same for all the PC's / end devices connected to the same router.

Result

1. PC0 > ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1 : bytes = 32 time = 2 ms TTL: 255

Reply from 20.0.0.1 : bytes = 32 time = 2ms TTL: 255

Reply from 20.0.0.1 : bytes = 32 time = 2ms TTL: 255

Reply from 20.0.0.1 : bytes = 32 time = 2ms TTL: 255

Ping statistics for 20.0.0.1

Packets : Sent = 4 Recived = 4 , Lost = 0 (0% Loss)

Approximate round trip times in milli-seconds

minimum = 0ms, Maximum = 0ms, Average = 0ms.

2. //Before router config.

Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Request timed out

Request timed out

Request timed out

Request timed out

Ping statistics for 20.0.0.1

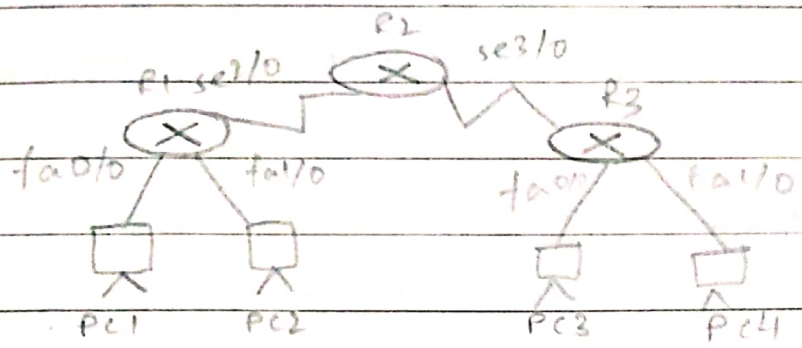
Packets : Sent = 4, Recived = 0, Lost = 4 (100% loss)

Exp: 2

Date 23/6/23

3. Explore ping responses using three routers.

Topology:



Procedure:

1. Insert 3 routers (Generic PT) and end devices, connect the routers using DCE cabel.
2. Configure IP address for the end devices
3. Configure router IP address, for each router

```
Router > enable
Router # configure terminal
Router (config) # interface se 2/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
Router (config-if) # no shutdown
```

4. Ping after each step to observe the different results.

Observation

1. "Destination host unreachable" is encountered when ping is performed without configuring the routers.

2. Request timed out is encountered when end device IP address is not set

3. Request timed out is encountered when DNS gateway is not configured

Result.

1. PC> ping 20.0.0.1

   pinging 20.0.0.1 with 32 bytes of data.

   Request timed out x 4

   Packets: sent=4, Received=0, Lost=4 (100% loss)

2. PC> ping 20.0.0.1

   pinging 20.0.0.1 with 32 bytes of data

   Reply from 20.0.0.1 bytes=32 time=2ms TTL=

   Packets: sent=4 Received=4, Lost=0 (0% loss)

3. PC> ping 20.0.0.1

   pinging 20.0.0.1 with 32 bytes of data

   Reply from 20.0.0.1 Destination host unreachable.

   Reply from 20.0.0.1 Destination host unreachable.

   Reply from 20.0.0.1 Destination host unreachable.

   Reply from 20.0.0.1 Destination host unreachable.

   Ping statistics for 20.0.0.1

   Packets: sent=4, Received=0, lost=4 (100% loss)

Exp 3.

Date 30/6/23
Page

4. Static Routing

- same topology.
- Static Routing commands for each router,

1. router> show ip route.
2. add the ip routes of the not connected ones.
3. router (config) # ip route 10.0.0.0 255.0.0.0 40.0.0.1

⇒ add network_address subnetmask interface

Observation

1. Routing algorithms are not implemented, hence each router needs static routing
2. With improper routing "request timed out" is displayed on ping command
3. Each router knows only the next router, thus that would be the via for others.

Result

```
PC> ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
  Reply from 10.0.0.1: bytes=32 time=7ms TTL=253
  Reply from 10.0.0.1: bytes=32 time=7ms TTL=253
  Reply from 10.0.0.1: bytes=32 time=7ms TTL=253
  Reply from 10.0.0.1: bytes=32 time=7ms TTL=253

Ping statistics for 10.0.0.1
  Packets: sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum=6ms, Maximum=10ms, Average=8ms
```

Date 30/1/23
Page

Default Routing

→ Can not be used when multiple routers are connected to a router., in such a case only static routing can be used.

→ Multiple routers connected = Backbone n/w

→ Routers with end devices can only be configured using default Routing.

Commands

default.

Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.1

Output

>show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, se 2/0
C 30.0.0.0/8 is directly connected, se 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

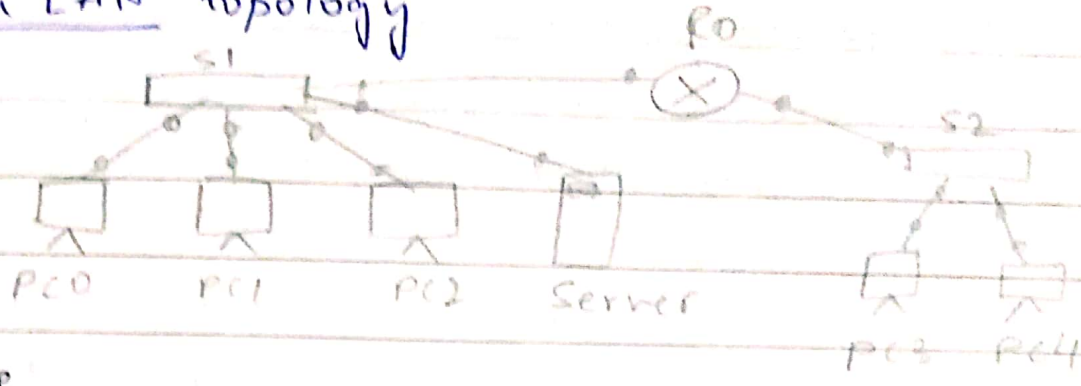
ping requests from PC4

→ pinging 10.0.0.10 with 32 bytes of data.
Reply from 10.0.0.10 : bytes = 32 time = 9ms TTL=12
Reply from 10.0.0.10 : bytes = 32 time = 5ms TTL=12
Reply from 10.0.0.10 : bytes = 32 time = 4ms TTL=12
Reply from 10.0.0.10 : bytes = 32 time = 5ms TTL=12
Ping statistics for 10.0.0.10
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss)

Tue 31/1/23

11/7/25 DHCP within a LAN and Outside a LAN

→ within a LAN topology



Procedure

1. Connect the devices as shown in the topology
2. In each of the end devices add DHCP in ip-config
3. Give static ip address for server and router.
4. In the router set "ip helper-address server#"

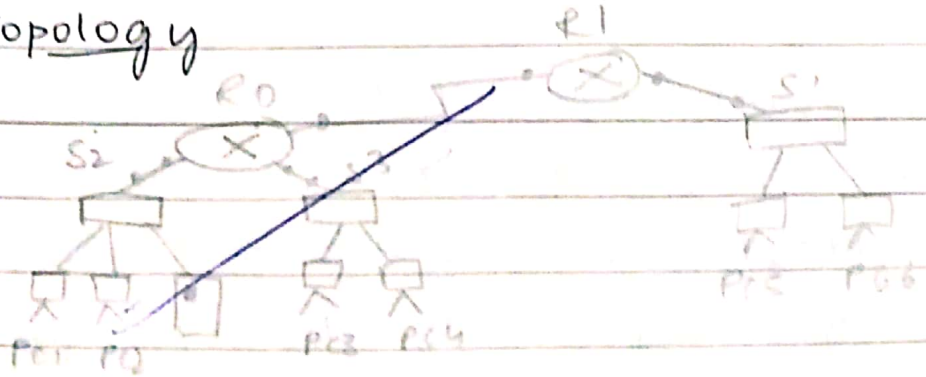
Output

DHCP request successful

IP address : 10.0.0.1

Aim: DHCP outside a LAN

Topology



Procedure

1. For the LAN topology add another router & PC
2. Configure the IP address for router & PC
3. Performe static routing on router 0 and 1

router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.2

4. Go to server in services create two more server pools with different name

	Default Gateway	DNS	Start IP	Subnet
Pool1	10.0.0.1	10.0.0.2	10.0.0.3	255
Pool2	10.0.0.1	10.0.0.2	20.0.0.2	255
Pool3	10.0.0.1	10.0.0.2	40.0.0.2	255

5. Go to PC5/PC6 switch IP config to DHCP.

Output

For PC5

DHCP request successful

IP address : 40.0.0.11

Subnet Mask : 255.0.0.0

Default Gateway : 10.0.0.1

DNS Server : 10.0.0.2

For PC6

DHCP request successful

IP address : 40.0.0.07

Subnet Mask : 255.0.0.0

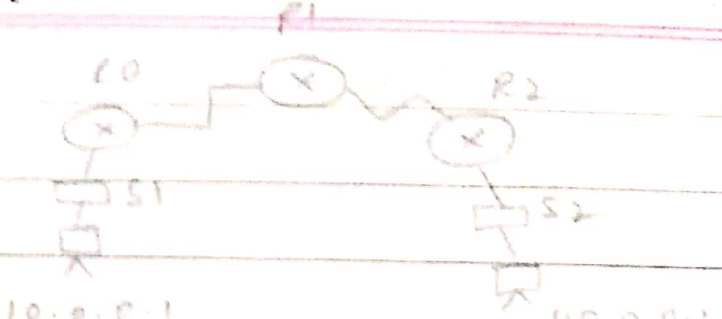
Default Gateway : 10.0.0.1

DNS Server : 10.0.0.2

Date 21/7/23
Page

Aim: Configure RIP

Topology:



Procedure

1. Configure IP addresses of the end devices
2. Configure Router interfaces statically.
3. Change the data rate to 64000 in all DCE
4. For each interface add #(router-if) encapsulation ppp

   PPP → point to point
5. Router (config-if) #clock rate 64000
   Router (config-if) #encapsulation ppp
   Router # show ip route

router# router rip
router #(config-router)#
network 30.0.0.0
router(config-router)# network 40.0.0.0

Output

C 10.0.0.0/8 directly connected, Fe 0/0
C 20.0.0.0/8 directlly connected, Se 2/0
R 30.0.0.0/8 via 20.0.0.2, 00:00:23, Se 2/0
R 40.0.0.0/8 via 20.0.0.2, 00:00:23, Se 2/0

Ping

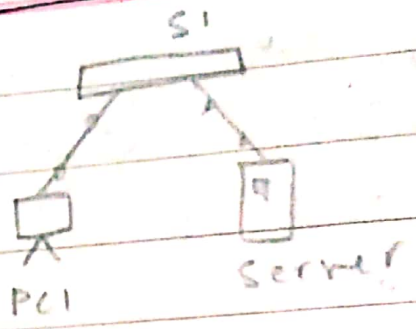
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2 : bytes = 32 time = 11ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 9ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 9ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 9ms TTL = 125

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

21/7/23

Aim: Configure DNS Server

Topology



Procedure

1. Configure IP addresses on PC and server
2. Configure DNS service on the generic server by adding name and address
3. Edit the HTTP html file
4. In PC open Web Browser and go to the URL
5. Make sure DNS is turned on.

Output

URL http://ibm21cs257.html

Name: Amrutha Muralidhar

USN : 1BM21CS257

Observation

→ DNS (Domain Name System)
It stores the names and ip addresses to that name.
If the DNS server is not added in the PC ip configuration then the page will not open i.e. blank screen is seen.

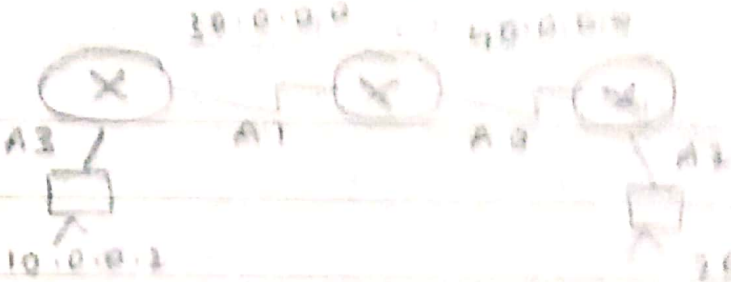
→ HTTP (Hypertext Transfer Protocol)

→ RIP, HTTP are in Application layer.

4/5/23

Aim: Configure OSPF routing protocol

Topology



Procedure

1. Launch Cisco Packet Tracer and create topology
2. Configure IP addresses for the PCs and routers
3. Configure OSPF routing using, #configure terminal

```
#router ospf 1
# router-id 1.1.1.1
# network 10.0.0.0 0.255.255.255 area 3
# network 30.0.0.0 0.255.255.255 area 1
```

directly connected

4. add loopback for all routers

```
#int loopback 0
#ip add 11.11.1.1
255.255.0.0
#no shutdown
```

5. create a virtual link b/w area 0 & area 3

```
R1#router ospf 1
R1#area 1 virtual-link 2.2.2.2
```

on router 2

```
R2#router ospf 1
R2#area 1 virtual-link 1.1.1.1
```

6. Now, show ip route b/w routers and ping

Observation

1. Virtual-link is provided to connect area 0 router to router area.
2. Network address is masked host address is shown.

3. Area 0 is the backbone network [router-no

Result

1. O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:01:56, serial 1/0

C 40.0.0.0/8 is directly connected, Fa 2/0

O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:01:5

C 30.0.0.0/8 is directly connected, se 1/0

2. Pinging from PC1 to PC2

ping 20.0.0.2

Reply from 20.0.0.2 ; bytes = 32, Time = 2ms, TTL=127

Reply from 20.0.0.2 : bytes = 32, Time = 2ms, TTL=127

Reply from 20.0.0.2 : bytes = 32, Time = 2ms, TTL=127

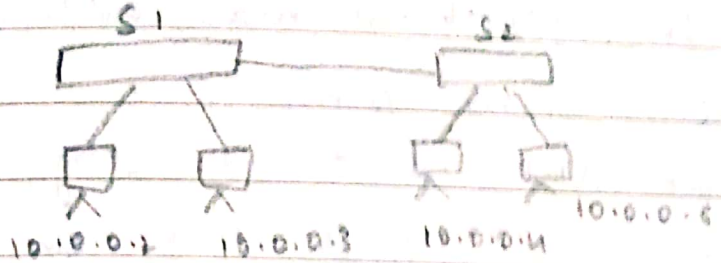
Reply from 20.0.0.2 : bytes = 32, Time = 2ms, TTL=127

Ping statistics for 20.0.0.1

Packets: sent = 4 received = 4 lost = 0 (0% loss)

Date 11/02/23
Page

8. Operation of ARP in a simple LAN Topology.



Procedure

1. Create a simple LAN topology
2. Configure IP addresses for all PC's.
3. In command prompt for any PC enter, PC> arp -a . It would not return any entry.
4. Ping any devices and then check for arp table.

Result

PC> arp -a

No ARP Entries found

PC> ping 10.0.0.3

PC> ping 10.0.0.4

PC> arp -a

Internet Address	Physical Address	Type
10.0.0.3	0090.0cbc.a17c	dynamic
10.0.0.4	00d0.baaa.558d	dynamic
10.0.0.5	0002.17c7.d2e2	dynamic
10.0.0.6	00d0.bc31.344a	dynamic

→ ARP table maintains IP to MAC mapping.

→ In switch CLI to show arp table cmd → Switch> show mac address-table

→ it learns after every ping

→ To view the ARP Table directly use inspect tool (🔍) and click on the device. and select ARP Table.

2) ARP Table for 10.0.0.2

IP Address	Hardware Address	Interface
10.0.0.3	0009.7c9B.70DC	Fa0
10.0.0.7	0001.638D.409B	Fa0
10.0.0.5	00D0.976F.E51B	Fa0

3) Switch> show mac address-table

MAC Address Table

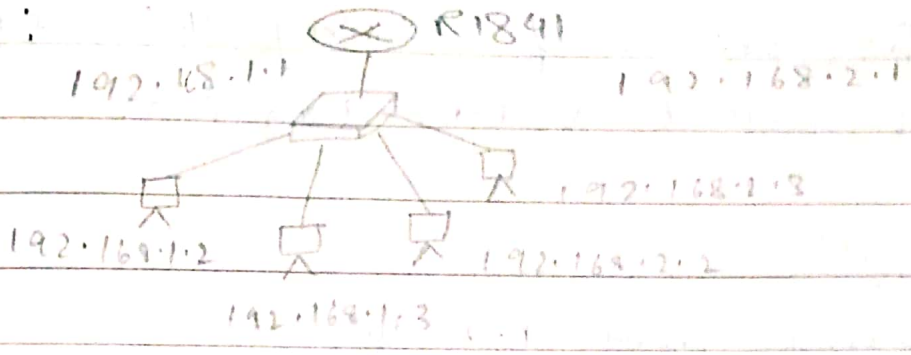
VLAN	Mac Address	Type	Ports
1	0001.423a.e612	Dynamic	Fa 4/1
1	0001.638d.409b	Dynamic	Fa 4/1
1	00d0.baaa.558d	Dynamic	Fa 3/1
1	0090.0cbc.ad7c	Dynamic	Fa 1/1

→ ARP request is a broadcast, response is unicast.

11/8/23

Aim: To create VLAN - Virtual LAN

Topology:



Procedure

1. Create the topology using 1841 router
2. In switch, go to config tab and select VLAN database and give any number and name
3. Select the interface fa 4/1 and make it trunk
4. make the interface have access to new VLAN database

   switch (config) # switchport access vlan 2
5. Config router and select VLAN 2, enter

   router (vlan) # exit

   router (config) # int fa 0/0.1

   router (config-subif) # encapsulation dot1q 2

   # ip add 192.168.2.1 255.255.255.0

   # no shut

Observation

1. VLAN trunking allows switches to forward frames from different VLANS over a single link called trunk.
2. This is done by adding an additional header called tag to the Ethernet frame.

The process of adding this extra header is called VLAN tagging.

Result

Pinging from PC1 to PC4

ping 192.168.2.3

Reply from 192.168.2.3 ; bytes = 32 ; Time = 2ms ; TTL = 127

Reply from 192.168.2.3 ; bytes = 32 ; Time = 2ms ; TTL = 127

Reply from 192.168.2.3 ; bytes = 32 ; Time = 2ms ; TTL = 127

Reply from 192.168.2.3 ; bytes = 32 ; Time = 2ms ; TTL = 127

Ping statistics for 192.168.2.3

Packets : sent = 4 received = 4 lost = 0 (0% loss)

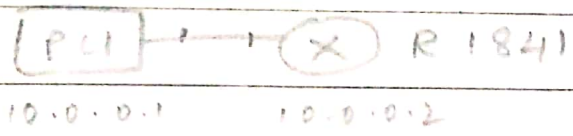
VLAN database of switch

VLAN No.	VLAN Name
1	default
2	VLAN2NEW
1002	Fddi-default
1003	token-ring-default
1004	Fddinet-default

11/8/23

Operation of TELNET by accessing the router in server room from a PC in IT office.

Topology



Procedure

1. Create a topology and assign IP addresses.
2. click on router - CLI

```
(router-config) # hostname router1
# enable secret password 1
# int fa0/0
# ip add 10.0.0.2 255.0.0.0
#no shut
# line vty 0 4
# login
# password p0
```

3. In PC cmd enter telnet 10.0.0.2
4. provide access details and type show ip route

Observation

Router's command line can be accessed from PC's CLI using the given password details.

Result

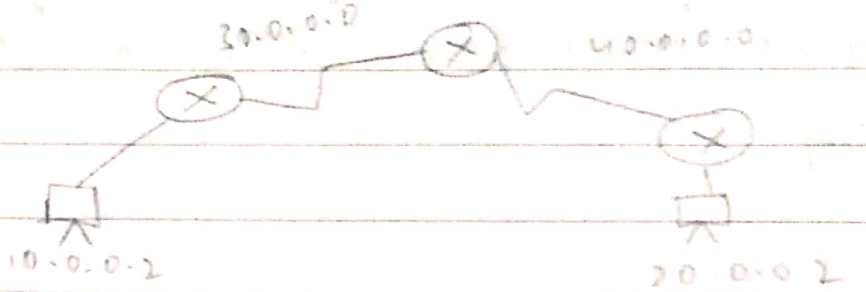
```
PC> telnet 10.0.0.1
Trying 10.0.0.1 ... open
User Access Verification
r1>enable
r1 # show ip route.
C   10.0.0.0/0 is directly connected, fa0/0
```

12

11/8/25

Aim: Demonstrate the TTL / Life of a Packet

Topology



Procedure

1. Create the topology and assign IP addresses
2. In simulation mode send PDU from PC to 2nd
3. Use capture button to capture every transfer
4. Click on PDU during every transfer to see the inbound and outbound details

Observation.

There is a difference of 1 in TTL when it crosses every router.

Result

IP

4	IHL	DSCP:	TL: 28
ID: 0x6		0x	0x0
TTL: 252	PRO: 0x1		CHKSUM
SEC IP: 10.0.0.2			
DST IP: 40.0.0.2			
OPT: 0x0			0x0
Data (Variable length			

12/8/25

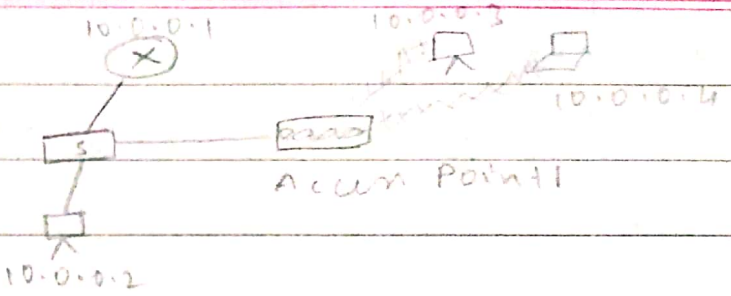
ICMP

8	16	31
TYPE:	CODE:	CHECKSUM
ID: 0x7		SEQN: 6

18/8/23

Aim: To construct WLAN and make nodes communi

Topology



Procedure

1. Construct the above topology and assign IPadd.
2. Configure Access Point - Port 1 → SSID Name - any name (WLAN)
3. Select WEP and give any 10 digit hex key
4. Configure PC2 and Laptop with wireless In physical switch off device and remove PT-HOST-NM-1AM and insert WMP300N
5. In config. → wireless 0 → SSID → WEP → WEP key.

Observation

SSID (service set Identifier) is used to differentiate b/w multiple wi-fi networks. incorrect SSID name will not connect.

Result

PC> ping 10.0.0.4

pinging 10.0.0.4 with 32 bytes of data
pinging 10.0.0.4 with 32 bytes of data
pinging 10.0.0.4 with 32 bytes of data
pinging 10.0.0.4 with 32 bytes of data

18/8/23

Write a program for error detection crc

```
#include<stdio.h>
char m[50], g[50], r[50], q[50], temp[50];
void crc(int){
  int i, j;
  for(i=0; i<n; i++)
    temp[i]=m[i];
    for(i=0; i<16; i++)  r[i]=m[i];
for(i=0; i<n-16; i++){
    if(r[0]=='1'){  q[i]='1';
                    calram(); }
    else{  q[i]='0';
           shift(); }
  r[16]=m[17+i];
  r[17]='\0';
  for(j=0, j<=17; j++)
      temp[j]=r[j]; }
  q[n-16]='\0';}
void calram(){
  int i, j;
  for(i=1, i<=16, i++)
      r[i-1]=(int)temp[i]-48)^(g[i]-48);}
void shift(){ int i;
  for(i=1, i<16, i++)
      r[i-1]=r[i];}
void main(){ int i, n;
```

```
char ch, flag = 0;
printf("Enter the frame bits \n");
while (ch = getc(stdin) != '\n') {
    m[i++] = ch;    n = i;
for( i=0 ; i<16 ; i++)
        m[n++] = '0';    m[n] = '\0';
printf("Message after appending 16 zeros
        %s ", m);
for(i=0; i<=16; i++)
    g[i] = '0';
    g[0] = g[4] = g[11] = g[16] = '1';
printf("\n Generator : %s ", g);
 crc(n);
printf("Quotient : %s", q); caltrans(n);
printf("\n Transmitted Frame %s \n
  Enter received frame ");
  scanf("%s", m);   crc(n);
  printf("Last remainder : %s", r); }
```

Output

```
Enter frame bits : 1011
Generator : 10001000000100001
Quotient : 1011
Transmitted frame : 10111011000101101011
Enter received frame : 10111011000101101011
Last remainder : 00000000000000000
```

|23| Leaky bucket algorithm implementation

```
#include <stdio.h>
void main() {
 int in, out, buck, n, store = 0;
 printf("Enter bucket size, outgoing rate
    and no. of inputs \n");
 scanf("%d %d %d", &buck, &out, &n);
 while(n!=0){
  printf("Enter incoming packet size\n");
  scanf("%d", &in);
  printf("Incoming packet size %d ", in);
  if(in <= (buck-store)) {
      store += in;
    printf("Bucket buffer size %d out of
      %d\n", store, buck); }
  else {
    printf("Wrapped %d no of packets ", in-buck);
    printf("Bucket buffer size %d out of %d",
        store, buck); }}
  store = store - out;
  printf("After outgoing %d packets left
    out of %d in buffer ", store, buck);
  n--; }}
```

Output

Enter bucket size, outing rate & no. of inputs : 20 10 2

Enter the incoming packet size : 30
Incoming packet size : 30
Dropped 10 no of packets
Buffer size 0 out of 20
After outgoing 10 packets left out of 20 is buffer
Enter the incoming packet size : 10
Incoming packet size : 10
Buffer size 20 out of 20
After outgoing 10 packets out of 20 in buffer.

1/9/23 Using TCP/IP sockets, write a client-server program to make client sending file name and the server to send back if present.

ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_NET, SOCK.SIREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\nEnter file name :")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print('\nFrom Server:\n')
print(fileContents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.connect((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.
sentence = connectionSocket.accept()
                           .recv(1024)
```

connectionSocket.send(l.encode())
print('in Sent contents of '+sentence)
fle.close()
connectionSocket.close()

Output

The server is ready to receive
Enter file name : serverTCP.py
From server :

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.
    accept()
    fle = open(sentence, "r")
    l = fle.read(1024)
    connectionSocket.close()
```

The server is ready to receive
Sent contents of serverTCP.py
The server is ready to receive

1/23 Using UDP sockets, write client-server program to make client-server send files

clientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"),
(serverName, serverPort))
fileContents, serverAddress = clientSocket.recv(2048)
print("\nReply from Server\n");
print(fileContents.decode("utf-8"))
# for i in file
clientSocket.close()
```

serverUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recv(204
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
```

```
serverSocket.sendto(bytes(con,"utf-8"),
    clientAddress)
print("\n Sent contents of ", end=" ")
print(sentence)
file.close()
```

Output

```
Enter file name: ServerUDP.py
Reply from server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
    while 1:
        print("The server is ready to receive")
        sentence = sentence.decode("utf-8")
        file = open(sentence, "r")
        l = file.read(2048)
        serverSocket.sendto(bytes(l, "utf-8"),
                clientAddress)
        print("\n Sent contents of ", end=" ")
        file.close()
```

The server is ready to receive
Sent contents of serverUDP.py
The server is ready to receive