Manipal
Date 27-01-24
Page 1

LAB-1

1) Difference b/w hub and Switch

(i) Hubs:

→ Hubs are simple broadcasting all data to all devices which can cause congestion and collisions.

→ Data handling: Operates in half-duplex (send or recieve, not both)

→ Network Traffic: Creates more network traffic due to data broadcasting

→ Speed & performance: Slower, with frequent data collisions

→ Layer of operation: Operates at the Physical layer (layer 1)

→ Cost: generally cheaper and ~~so~~ simpler

(ii) Switches:

→ Switches are the connecting devices which sends data only to the specific intended recipient.

→ Data handling → operates in full duplex (send and recieve simultaneously)

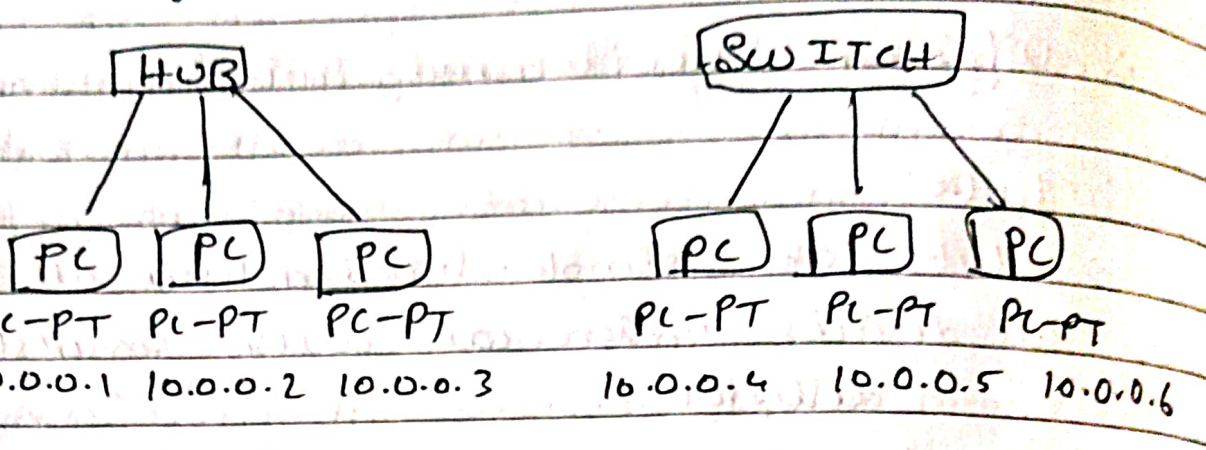
→ Network traffic → reduces data traffic by forwarding data only to the recipient

→ Speed & performance → faster, minimizes collision

→ Layer of operation → operates at data Link layer (layer 2) or Network layer (layer 3)

→ cost → More Expensive but offers better performance

Topology:-



→ The hub establishes a connection between all systems and only the recieving pc can relay a response message

→ Switch directly relays the message from the source pc to destination pc and executes a set of protocols

→ Each PC must be configured with the initial Ip address and must be connected to the respective Hub and Switch

Observation:-

The difference between Hub and Switch is that the hub relays the senders message to all the External devices connected to it and only the recieving device will accept packet

whereas a switch establishes a direct connection between the sending and recieving device

Manjal
Date 09-10-24
Page 3

Lab-2.

Create a topology involving multiple hubs and a switch connecting them to a simulator to simulate PDU [packet Data Unit].

Proceedure for creating the topology :-

Aim:- topology involving multiple hubs and a switch.

Step 1:- Select the hubs based on the requirement along with the required End devices [Here pc's] to be connected with the hub.

Step 2:- Select the type of connection [Here, automatically choosen connection type] that is used to connect between Hubs and the End devices.

Step 3:- Configure every End device by navigating through Config -> FastEthernet0, after this, provide the Ip address which is unique for each End device.

Step 4:- After Step 3, select a Switch [based on the requirement] and select the type of link [Here, Copper Crossover] that needs to be used to connect between Multiple hubs and the Switch.

(to a single hub)

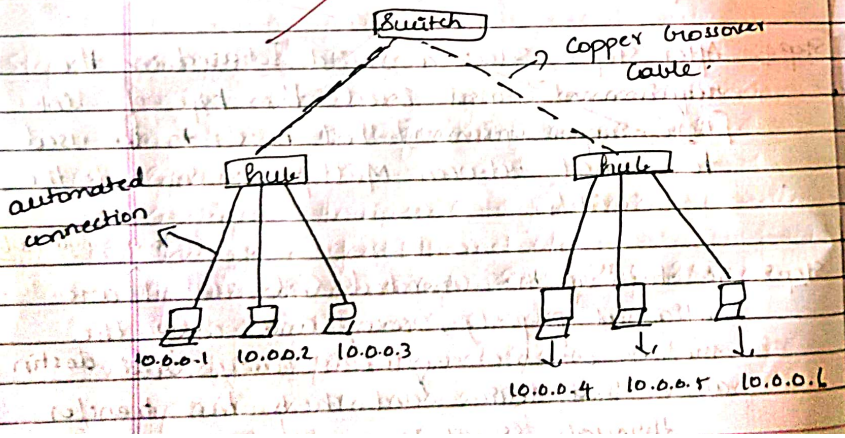
Step 5:- To check if the connected End devices are configured properly, drag and drop the message on the required Source and destination End device and check the transfer through Simulation mode

Step 6:- Similar to step 5, now drag and drop one the message b/w the source end device [connected to one hub] and the destination [connected to another hub] and check in simulation mode if packet transfer happens between the devices connected to two or more different hubs.

Step 7:- Now, select the desired end device, Right click on it, go to desktop and select command prompt there, we need to use "ping" command to send single datagrams per second to respective destination device.

Step 8:- For example, select a configured device go to command prompt & type "ping 10.0.0.3" where 10.0.0.3 is the ip address configured for another end device and check the packet transfer process.

Topology:-



Observations:-

-> From the topology, we understood how multiple hubs can be connected to a switch using type of connection as required.

-> We also understood how the configured devices connected to either a single or multiple hubs are able to communicate through pdu [packet data unit] and observed how datagrams are transferred between the configured devices.

-> We came to know about the ping command and how it works for the purpose of send the datagrams b/w the configured devices.

19/101

Page 6

Lab-3

Aim: Connecting and Configuring a Router

Step 1: Set up the devices:
Drag and drop :- → 1 Router
→ 2 Switches
→ 4 Computers.

Step 2: Connect Devices :-
→ Connect P1 and P2 to Switch 1 and P3 and P4 to Switch 2.
→ Connect Switch 1 to Router on fastEthernet 0/0
→ Connect Switch 2 to Router on fastEthernet 0/1

Step 3: Assign ip addresses to the respective computers
→ click on pc → Desktop

	Default gateway
PC1 → 10.0.0.1	10.0.0.1
PC2 → 10.0.0.2	10.0.0.1
PC3 → 10.0.0.3	10.0.0.2
PC4 → 10.0.0.2	10.0.0.2

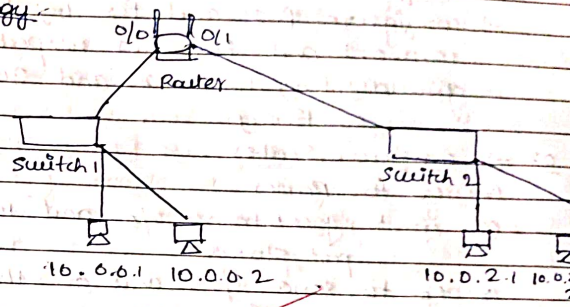
Step 4: Configure Router :-
click on Router, go to command line interface and enter the following commands :-
(i) Enable
(ii) Configure terminal
(iii) Interface fastEthernet 0/0
ip address 10.0.0.1 255.255.255.0
no shutdown
(iv) Interface fastEthernet 0/0
ip address 10.0.0.2 255.255.255.0
no shutdown
exit.

Manipal

Page 7

Step 5: Test Connectivity
→ go to PC1
→ go to cmd prompt in desktop and run
ping 10.0.0.2 (pc2)
ping 10.0.2.1 (pc3).

Topology:-



Observation :-
We need to configure Router and we need to execute certain commands and Router is used to connect multiple Networks and make it one single Network

FRIDAY

Date 18/10/24
Page 8

Configuring 2 Routers

AIM: configuring 2 Routers in a Network

Step 01: Drag and drop 2 pc's and 2 Routers configure each device

-> configure PC1 and PC2 by ip address 10.0.0.1 and 20.0.0.1 respectively and gateway has 10.0.0.2 and 20.0.0.2 respectively

02 -> configure Router

(*) up to Router

(*) up to fastEthernet 0/0 and turn on the port status

(*) up to Serial 2/0 and do the same for the 2nd router.

(*) Set ip address of router 0 as 30.0.0.1 and router 1 as 30.0.0.2

Step 03: Run the commands in Router CLI

-> Show ip route

-> config t

-> in router 0 :-

ip route 10.0.0.0 255.0.0.0 30.0.0.2

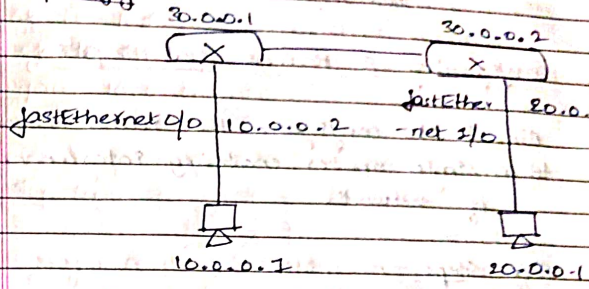
in router 1 :-

ip route 20.0.0.0 255.0.0.0 30.0.0.1

-> up to command prompt in PC and run ping 20.0.0.1

Manipal
Date
Page 9

Topology :-



Observation :

Configured routers and executed ping commands and observed outputs

Configure ip address of routers in Cisco packet tracer explore the following cmds, ping responses, request time, destination unreachable

25/10/01

Page 10

Lab-9

Configuring 3 Routers and 2 pc's

Aim: Aim is to configure the router network to a single router enabling interconnection of networks

Algo Steps:

Step 01:- Set up the devices:-
drag and drop :- 3 Routers, 2 pc's.

Step 02:- Configuring pc's :-
-> Configure pc1 with the ip address 10.0.0.1 and for pc2, ip address -> 20.0.0.1
-> Enable the gateway for pc1 as 10.0.0.2 and for pc2, with gateway 20.0.0.2

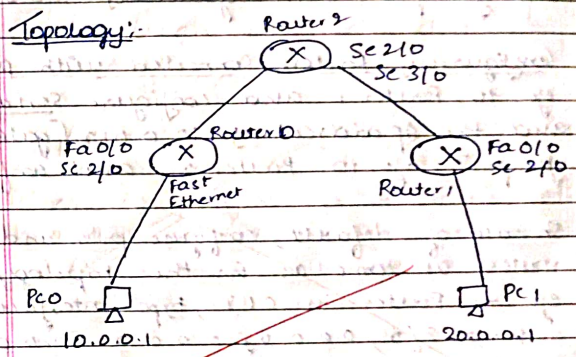
Step 03:- Configuring Routers connected with pc's
-> go to Router 1 and configure Fast Ethernet 0/0 with ip address 10.0.0.2 [gateway for pc1], go to port serial 2/0 and the ip address needs to be set as 30.0.0.1
-> go to RIP Routing and add the respective ip addresses which was set for the ports
-> IIIrd for Router 2, configure Fast Ethernet 0/0 with ip address 20.0.0.2 [gateway for pc2], go to port serial 2/0 and set ip address as ip 40.0.0.1
-> go to RIP Routing and add the respective ip addresses which was set for the ports

Page 11

Step 04:- Configuring Routers connected to 2 other Routers
-> go to port serial 2/0 and set the ip address as 30.0.0.4 with Subnet Mask 255.0.0.0
-> go to port serial 3/0 and set the ip address as 40.0.0.4 with Subnet mask 255.0.0.0
-> Add the respective ip addresses to the RIP Routing and setup the connection

Step 05:- go to CLI for pc0 and execute ping 20.0.0.1 IIIrd for pc1, execute ping 10.0.0.1 and check for the successful ping message

Topology:-



Observation:-
Executed ping command to test the successful connection of the Router Network connected to a single Router.

II Aim:- Configuring same topology with default Routing.

Steps:

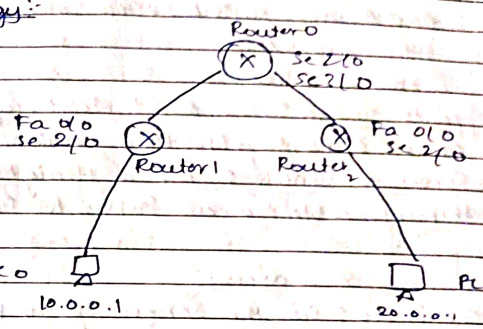
Step 01: Set up the devices:-
drag and drop :- 3 Routers, 2 pc's

Step 02:- Configure pc's:-
-> Configure pc1 with the ip address 10.0.0.1 and for pc2, ip address -> 20.0.0.1
-> Enable the gateway for pc1 as 10.0.0.2 and for pc2, with gateway 20.0.0.2

Step 03:- Configuring Routers connected with pc's
-> go to Router 1 and configure Serial 2/0 and the ip address as 10.0.0.2 [gateway for pc1], go to Router 2 and configure
-> In case of default, configure pc's and routers as same as in the topology
-> go to Router in CLI, type interface serial 2/0, 0.0.0.0, 0.0.0.0 30.0.0.2 in Router R1
-> In Router 2
in CLI, type interface serial 2/0,
ip Route 0.0.0.0 0.0.0.0 40.0.0.2
-> In Router R0
in CLI, type interface 2/0,
ip Route 0.0.0.0 0.0.0.0 30.0.0.1
Interface Serial 3/0
ip Route 0.0.0.0 0.0.0.0 40.0.0.1

Step 04:- go to CLI for pc0 and execute ping 20.0.0.1 "" for pc1, execute ping 10.0.0.1 and check for the successful ping message

Topology:-



Observation:-
Executed ping command to test the successful connection of the Router network connected to a single Router using default configuration

29/10/24

Lab - 5 :-

I. Demonstrating web Server & DNS :-

Step1:- Add a pc, switch (2960) and a server to the workspace

Step2:- Connect Devices
→ pc to switch : Use a Straight copper through cable, connect pc's to a switch
→ Server to Switch :- Use a Straight copper through cable, connect Server to Switch

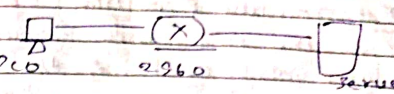
Step3:- Assign ip addresses:
Server ip : Set the Servers ip to 192.168.1.254
pc ip : Set the pc's ip to 192.168.1.100

Step4:- Configure devices :-
Server: go to the config tab, select Fast Ethernet, turn on the port, & set ip to 192.168.1.254.
Pc :- go to config tab, select 'Fast Ethernet', turn on the port, on set ip to 192.168.1.[illegible]

Step5:- Ping Test:
from the 'pc''s command prompt, ping the 192.168.1.254 to test the connection

Step6:- Web Server Test :-
Open the pc's web browser & type 192.168.1.254 & see the servers web page

Topology :-



II. Using Wireless Router & DNS setup :-

(1) Add Router
→ place a "Linksys wireless Router" in the work space and label it "192.168.1.1"

(2) Connect Router to Switch :
→ Add a Router "Wireless Router" in the workspace and label it 192.168.1.1.

→ Connect Router to Switch
Connect the router to the switch using "Straight-through cable" from Switch's Fast Ethernet to Routers Ethernet 0/3

(3) Connect Router to Ethernet :
→ use the cloud device to represent the internet and connect the routers wan port to it

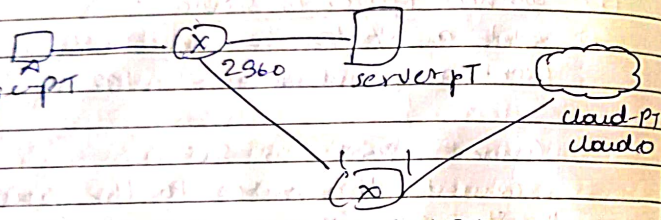
(4) Double-click the router, go to config > LAN and change the ip address to 192.168.1.1

(5) Ping Router from pc
on the pc, ping 192.168.1.1 to test connection to the router

(6) Set default gateway :-
pc's ip configuration, set the DNS Server to 192.168.1.254

(5) Test domain resolution :-
open the pc's web browser, type "type yahoo.com", & the web page should now load with the new domain name.

Aim :- To demonstrate the accessing of server page from any given device using using DNS & wireless Router.



O/p & observation :-

→ Created a basic LAN with a pc, switch and server, tested connectivity with pings, and added a router for internet access

III Basic DHCP configuration for a single N/w

Step 1 :- Set up LAN Topology (router, switch, server, PC)

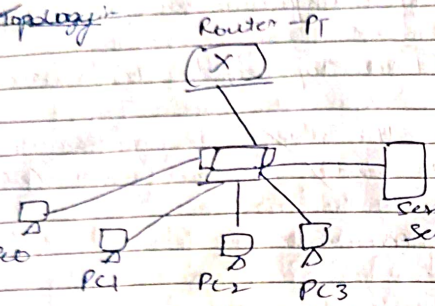
Step 2 :- Configure the router interface with an IP address (10.0.0.1) & subnet mask (255.0.0.0)

Step 3 :- Enter the commands
→ enable
→ config t
→ interface fastethernet 0/0
→ ip address 10.0.0.1 255.0.0.0
→ no shutdown
→ exit

Step 4 :- In DHCP tab, configure the following
→ Default gateway : 10.0.0.1
→ DNS Server : 10.0.0.2
→ Start IP Address : 10.0.0.10
→ Maximum users : set it to 500
→ TFTP Server : 10.0.0.2

Step 5 :- Configure pc to use DHCP
→ click on a pc, go to "Desk top" → "IP configuration",
→ Select DHCP
→ The pc will automatically recieve & IP from the DHCP server.

Topology:-



IV Using ip helper address to get DHCP from a Remote Network

Step 1:- Create Another N/w (with a different subnet eg 20.0.0.0/8 connected to a different router interface (fastethernet 0/1)

Step 2:- Configure router interfaces with following
Interface 0/0, 10.0.0.1 (for '10.0.0.0/8' network)
Interface 0/1, 20.0.0.1 (for the 20.0.0.0/8 network)

Step 3:- Enter the following commands:-
→ enable
→ config t
→ interface fastethernet 0/0
→ ip address 10.0.0.1 255.0.0.0
→ no shutdown
→ exit
→ interface fastethernet 0/1
→ ip address 20.0.0.1 255.0.0.0
→ no shutdown
→ exit

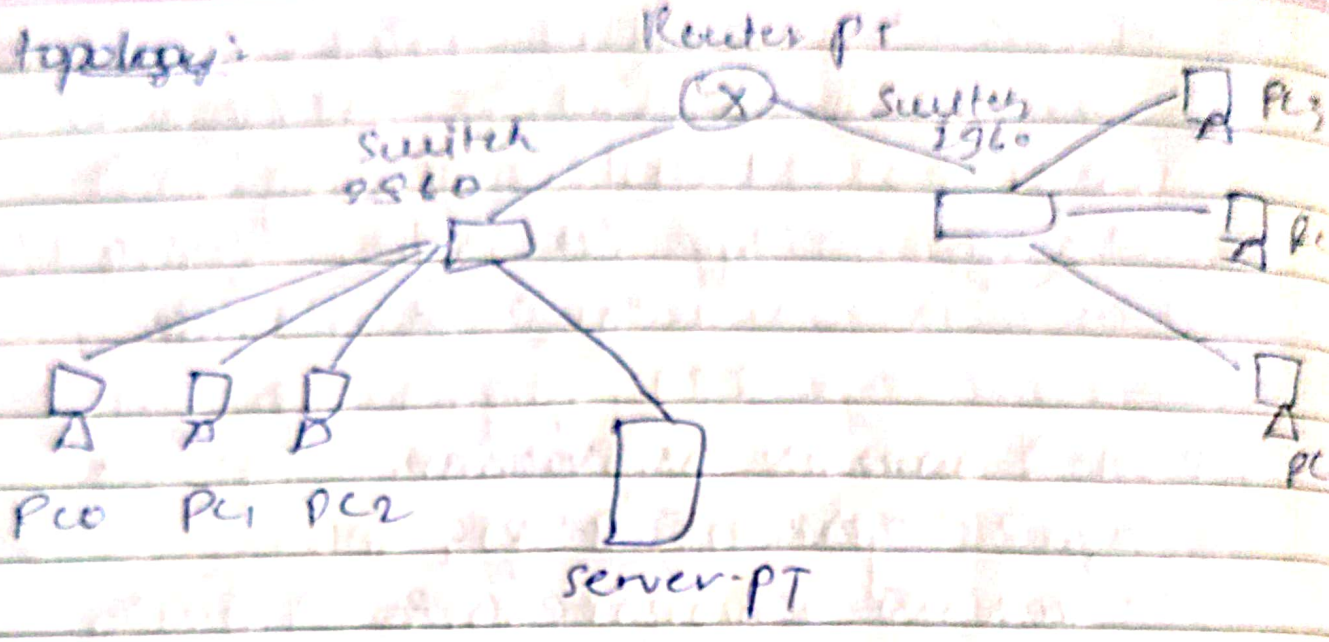
Step 4:- Configure DHCP server for 20.0.0.0/8 N/w:-
→ On server, create a new DHCP pool for the 20.0.0.0 N/w
→ pool name: 20 Network
→ Default gateway: 20.0.0.1
→ DNS server: 10.0.0.2
→ Start ip address: 20.0.0.16
→ Subnet Mask: 255.0.0.0
→ Maximum users: 100
→ TFTP server: 10.0.0.2
Click add & Save

Step 4:- To allow 20.0.0.0 N/w to access DHCP server on 10.0.0.2 configure the ip-helper address on the router interface connected to the 20.0.0.0 N/w

Interface fastethernet 1/0
ip helper address 10.0.0.2
exit

Step 5:- On any pc in the 20.0.0.0 N/w go to desktop → ip configuration, Select DHCP
→ The pc will recieve an ip from the DHCP server on the 10.0.0.0 N/w

Topology:



Aim:- To create basic DHCP configuration for a single N/w using ip-helper address

Observation:- DHCP assigns ip address automatically within a N/w

→ Use ip helper address on the router to allow DHCP clients in a different n/w to get ip address from DHCP server in another N/w

LAB-6

Cyclic Redundancy Check [CRC] Algorithm:

```
#include <stdio.h>
#include <string.h>

int crc(char *ip, char *op, char *poly,
                    int mode){
    strcpy(op, ip);
    if(mode){
        for(int i=1; i<strlen(poly); i++){
            strcat(op,"0");
    }}
    for(int i=0; i<strlen(ip); i++){
        if(op[i]=='1'){
            for(int j=0; j<strlen(poly); j++){
                if(op[i+j]==poly[j]){
                    op[i+j]='0';
                }
                else{
                    op[i+j]='1';}}}}

    for(int i=0; i<strlen(op); i++){
        if(op[i]=='1'){
            return 0;
        }
    }
    return 1;
}
```

```
int main() {
    char ip[50], op[50], recv[50];
    char poly[] = "10001000000100001";
    printf("Enter the ip message : ");
    scanf("%s", ip);
    crc(ip, op, poly, 1);
    printf("Transmitted message : %s%s\n",
            ip, op + strlen(ip));
    printf("Enter recieved message : ");
    scanf("%s", recv);

    if (crc(recv, op, poly, 0)) {
        printf("No error in data\n");
    }
    else {
        printf("Error in data transmissio\n");
    }
    return 0;
}
```

Opp:

Enter the ip message : 11111
transmitted message : 11111 0001111 0
1111 0

Enter Recieved message : 111
Error in data transmissio occurred

Leaky Bucket Algorithm :

```
#include <stdio.h>

#define MAX_CAPACITY 5
#define LEAK_RATE 1

void leakyBucket(int incomingPackets[], int
                 numPackets) {
    int bucket = 0;
    int time = 0;

    for (int i = 0; i < numPackets ; i++) {
        printf("Time %d: Incoming packet = %d
               \n", time, incomingPackets[i]);

        if (bucket + incomingPackets[i] <= MAX_CAPA
                                          CITY) {
            bucket += incomingPackets[i];
            printf("Bucket : %d Packets Accepted
                   \n", bucket);
        }
        else {
            int droppedPackets = incomingPackets[i] -
                (MAX_CAPACITY - bucket);
            bucket = MAX_CAPACITY;
            printf("Bucket : %d packets Overflow!
                   %d packets dropped \n", bucket,
                   droppedPackets);
        }
```

```
    bucket -= LEAK_RATE;
    if (bucket < 0) bucket = 0;
    printf("Bucket leaks %d", bucket);

    time++;
    printf("\n");
  }
}

int main() {
  int incomingPackets[] = {2, 3, 4, 1, 2, 3, 1, 5, 2, 1, 3};
  int numPackets = sizeof(incomingPackets) / sizeof(incomingPackets[0]);

  printf("Leaky Bucket Algorithm");

  leakyBucket(incomingPackets, numPackets);
  return 0;
}
```

O/P:-

Leaky Bucket Algorithm

Time 0: Incoming packet = 2
Bucket : 2 packets (accepted)
Bucket leaks : 1 packet left in the bucket

Time 1 : Incoming packet = 3
Bucket : 4 packets (Accepted)
Bucket leaks : 3 packets left in bucket.

Time 2 : Incoming packet = 4
Bucket : 5 packets (overflow ! 2 packets are dropped)
Bucket Leaks : 4 packets left in the bucket

Time 3 : Incoming packet = 1
Bucket : 5 packets (Accepted)
Bucket leaks : 4 packets left in the bucket

Time 4 : Incoming packet = 2
Bucket : 5 packets (overflow ! 1 packet dropped)
Bucket leaks : 4 packets left in bucket

Time 5 : Incoming packet = 3
Bucket : 5 packets (accepted)
Bucket leaks : 4 packets left in the bucket

LAB - 7

22-11-24

Implementation of Routing Information Protocol

Aim: In order to implement RIP protocol through router topology.

Step 01: Add 6 PC's, 3 switches and 3 routers to the workspace.

Step 02: Configure PC's

→ PC0 → IP address: 10.0.0.2, subnet mask: 255.0.0.0
gateway: 10.0.0.1

→ PC1 → IP address: 10.0.0.3, subnet mask: 255.0.0.0
gateway: 10.0.0.1

→ PC2 → IP address: 20.0.0.2, Subnet: 255.0.0.0
gateway: 20.0.0.1

→ PC3 → IP address: 20.0.0.3, Subnet: 255.0.0.0
Gateway: 20.0.0.1

→ PC4 → IP address: 30.0.0.2, gateway: 30.0.0.1

→ PC5 → IP address: 30.0.0.3, gateway: 30.0.0.1

Step 03: Connect PC0 & PC1 to Switch 0 → Router 0
Connect PC2 & PC3 to Switch 1 → Router 1
Connect PC4 & PC5 to Switch 2 → Router 2 etc
all of them connected through auto connection

Step 04: Configure Routers

→ Router 0: FastEthernet 0/0 → IP address: 10.0.0.1
Serial 2/0 → IP address: 40.0.0.1

→ Router 1: FastEthernet 0/0 → IP address: 20.0.0.1
Serial 2/0 → IP address: 40.0.0.2
Serial 3/0 → IP address: 50.0.0.1

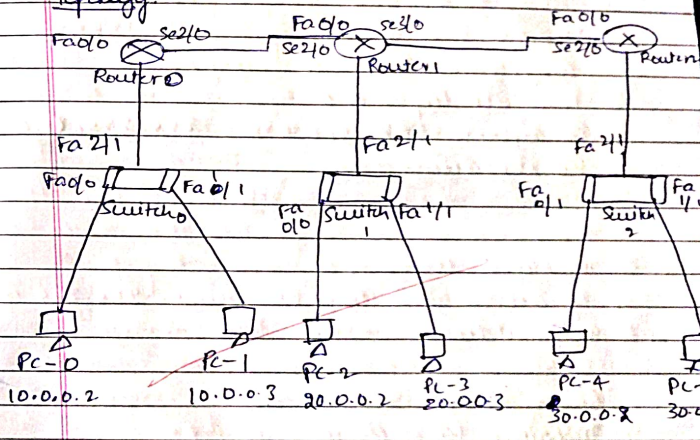
→ Router 2: FastEthernet 0/0 → IP address: 30.0.0.1
Serial 2/0 → IP address: 50.0.0.2

Step 05: Connect Router 0 → Router 1, Router 1 → Router 2 using autoConnection cable

Step 06: Go to Router RIP & add the n/w. In case of R0, add 40.0.0.0, carry out the same 10.0.0.0 for other routers

Step 07: Execute Ping command to check connectivity

Topology:-



Observation:-

Observed Routing info protocol plays important role

Exercise 02 :- Inbound & outbound details

Steps :-

01:- Do simulation in the network

02:- In Event List
click on (Router to Router) info
click on PDU and further check
after other info it will be decremented

03:- observe that TTL decrements
based on routing decision

LAB - 8

Step by step implementation of OSPF configuration

Step 01:- create the topology

design a topology in Cisco Packet Tracer with:

→ Three routers R_1, R_2, R_3

→ host connected to R_1, R_3 (Host 1 in n/w 10.0.0.0/8 & host 2 in n/w 40.0.0.0/8)

Step 02:- Assign Ip Adress to Interfaces

Router R_1

Interface fastethernet 2/0

ip address 10.0.0.1 255.0.0.0

no shutdown

exit

interface serial 1/0

ip address 20.0.0.1 255.0.0.0

encapsulation ppp

clock rate 64000

no shutdown exit

In the same way assign port no to R_2 & R_3

Step 03:- Configure OSPF on all Routers R.

Router OSPf 1

router-id 3.3.3.3

network 10.0.0.0 0.255.255.255 area 3

network 20.0.0.0 0.255.255.255 area 1

exit

R2

```
# router ospf 1
# router-id 2.2.2.2
# network 20.0.0.0 0.255.255.255 area 1
# network 30.0.0.0 0.255.255.255 area 0
# exit
```

R3

```
# router ospf 1
# router-id 3.3.3.3
# network 30.0.0.0 0.255.255.255 area 0
# network 40.0.0.0 0.255.255.255 area 2
# exit
```

Step 04 :-

Configure loopback interfaces R1

```
# interface loopback 0
# ip address 172.16.1.252 255.255.255.0
# no shutdown
```

R2

```
# interface loopback 0
# ip address 172.16.1.253 255.255.255.0
# no shutdown
```

R3

```
# interface loopback 0
# ip address 172.16.1.254 255.255.0.0
# no shutdown
```

Step 5 : test

```
# show ip route
```

Step 6 :- Configure virtual link

R1

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

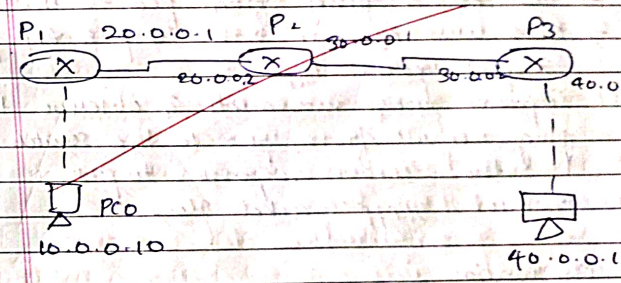
R2

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

Step 7 :- Test connectivity

→ in pc1, ping 40.0.0.10

Topology :-



LAB - 9

Aim:-

(1) To construct simple LAN and understand the concept and operation of ARP

Procedure:-

(1) Assign ip address uniquely to all pc's & servers

(2) Go to simulation panel and inspect ARP table
   → click on pc0, right-click, and inspect its ARP table
   → do the same for server

(3) Check initial ARP entries
   → open command prompt on pc0
   → Try arp -a to confirm ARP entries

(4) Ping the server from pc0
   → initiate a ping from pc0 to the server
   → This will generate both ICMP & ARP packets

(5) Inspect packet types

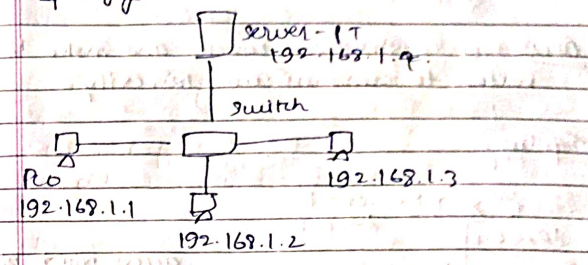
(6) Capture packet movements & observe the ping request from pc0 to the switch & the servers response to pc0

(7) Monitor ICMP packet flow.

Observation:-

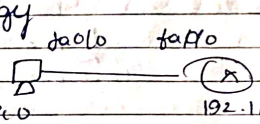
→ Constructing simple LAN for observing the behaviour of ARP.

Topology:-



(2) Telnet

Topology



Procedure

① Create the topology
② Configure ip address & gateway for pc0
③ Configure the router in cli
④ go to pc - cmd
   type ping 192.168.1.1
   telnet 192.168.1.1

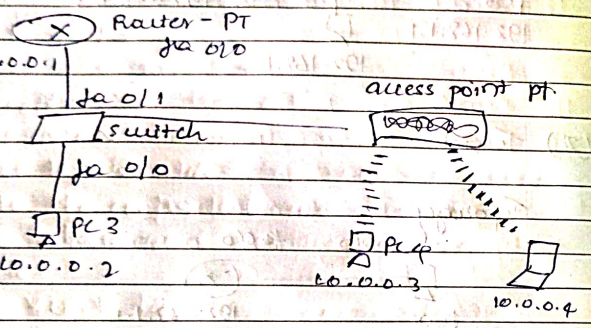
Observation:-

1. Ping:- Successful pings indicate that router is forwarding ICMP packets correctly between diff subnets. ARP helps to map ip addresses
2. Telnet:- Successful Telnet connections indicate that PC0 can establish a remote connection to PC1 / servers through the router

WLAN :

AIM : To construct a WLAN and make the nodes to communicate wirelly.

Topology :



Procedure :

(•) Router
→ go to Config Tab
→ Assign an ip address 10.0.0.1

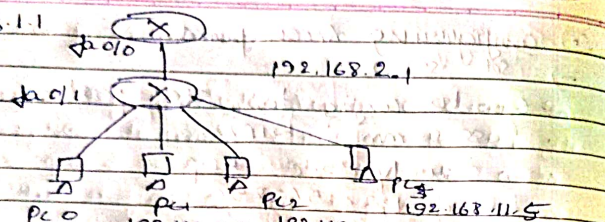
(•) PC 3
→ go to Desktop tab
→ open ip configuration tab & assign
ip address 10.0.0.2
subnet mask 255.0.0.0
gateway : 10.0.0.1

(•) Configuring Access point :
→ set SSID
→ go to config tab, select port
SSID nm : WLAN
Security Mode : WEP
Key : 1234 567890

(•) Configuring PC 4 & laptop for wireless
→ install Wireless NIC
→ switch off PC 4 & laptop
→ Drag the WMP300N wireless interface and to their empty ports
→ switch on the devices
→ Configure wireless settings :
→ go to config tab
→ Select the new Wireless interface
→ set
SSID : WLAN
Security mode : WEP
ip address : eg. PC 4 : 10.0.0.3
laptop : 10.0.0.4

Ping
open cmd
ping every other device

VLAN:



Procedure:

Create VLAN

→ select the switch

→ go to the config tab and choose VLAN database

→ Enter

VLAN: 12

VLAN name: NEWLAN

(•) set trunck port

→ go to interface settings under config

→ Select the interface connected to router

→ change the mode to TRUNK.

(•) Configuring the router

Add VLAN on the router VLAN database

→ Select the router

→ go to the config tab & select VLAN

→ Add

VLAN NO: 2

VLAN name: NEW LAN

(•) Configure sub interfaces for VLAN:

→ open CLI

→ Enter cmds

Router > en

Router # config t

Router (config) # interface fastethernet 0/0.1

Router (config-subif) # ip address 192.168.2.1 255.255.255.0

Router (config-subif) # no shut

exit.

Ping:-

ping every device

Socket programming

(1) Using Tcp/Ip Sockets, write a client-server program to make client sending the file name and server to send back the contents of the requested file if present

client Tcp py
from socket import *
ServerName = '127.0.0.1'
Server port = 12000
client socket = Socket (AF_INF, SOCK STREAM
client socket . connect ((Server Name, Server port))
sentence = input ("\enter file name:")

client Socket. Send (Sentence. encode())
file contents = client Socket. recv (1024. decode())
print ("\n From server \n")
print (file contents)
clientSocket. close()

Server Tcp py
from Socket import *
Server name = '127.0.0.1'
Server port = 1200 0
ServerSocket = SOCKET(AF_INET, SOCK-STREAM)
server Socket. bind ((server name, server port))
server Socket. Listen (1)
while 1:
print("server is ready to recieve")
connection Socket, addr = Server Socket
sentence = connection Seocket. recv (2024) . decode

I = file. read (1024)
connection Socket. send (1. encode())
print ("\n sent contents of " + sentence)
file. close()
connectionSocket. close()

2. using udp sockets, write a client server program to make client sending the file name and the server to sent the back the contents of the requested file if present.

client - UDP. py
from socket import *
Servername = "127. 0.0.1"
server port = 12000
client Socket = Socket (AF-INET, SOCK-DGRAM)
sentence = input ("\n Enter file name:")

client Socket. send to (bytes (Sentence, "utf-8"
(Server name, Server port))
file Contents, ServerAdress = client Socket.
recv from ( ))
print ("\n Reply from server: \n")
print (file contents. decode ("utf-8"))
clientSocket. close()
client Socket. close()

Server UDP.py

```
from socket import *
serverport = 12000
Serversocket = socket (AF_INET, SOCK_DGRAM)
Server socket.bind ("127.0.0.1", server_port)
print ("The server is ready to recive")
while 1:
    sentence, client Address = server Socket. recv
                                from (2048)
    sentence = sentence.decode ("utf-8")
    file = open(sentence, "r")
    con = file. read (2048)
    Server Socket. send to (bytes (con, "utf-8"),
                            client Address)
    print ('\n Sentence contents of', end = " ")
    print (Sentence)
    file. close()
```