Hub / Switch.

Date 09/10/2024
Page 01

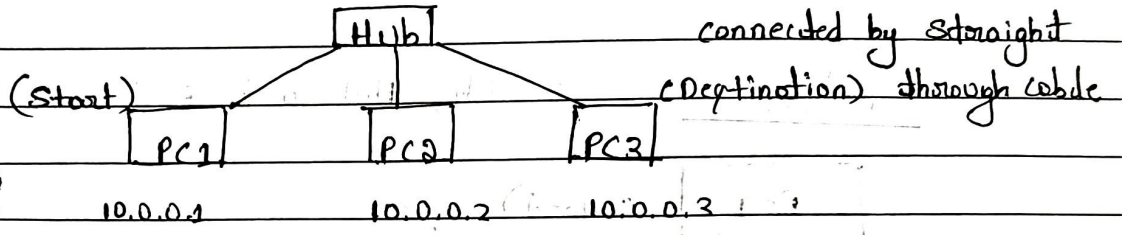
Aim: Simulate transmission of simple PDU using hub and switch as connecting devices using a Cisco Packet Tracer.

Devices used:

Hub, Switch, and End Devices.

Topology 1:

Hub and 3 End Devices.



Procedure and Observations.

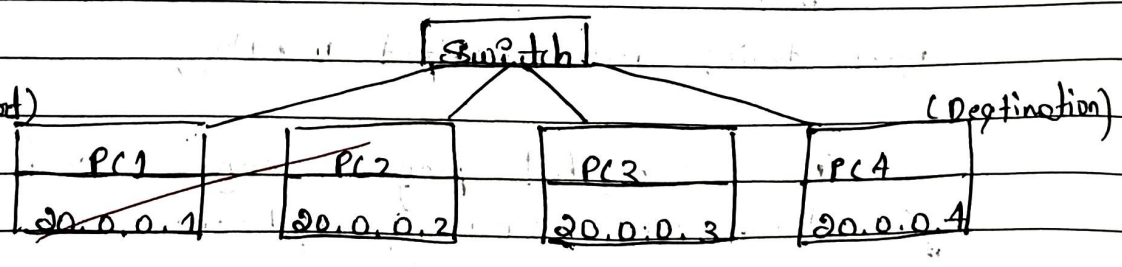
i. Connect end devices PC1, PC2 and PC3 to the hub through straight cable.

ii. Assign IP Address to each of the end devices.

iii. Select a simple PDU, select PC1 as start node and PC3 as destination.

During simulation, the message will be received by PC3 by PC1 and acknowledges the same.

Topology 2: Switch and End Devices.

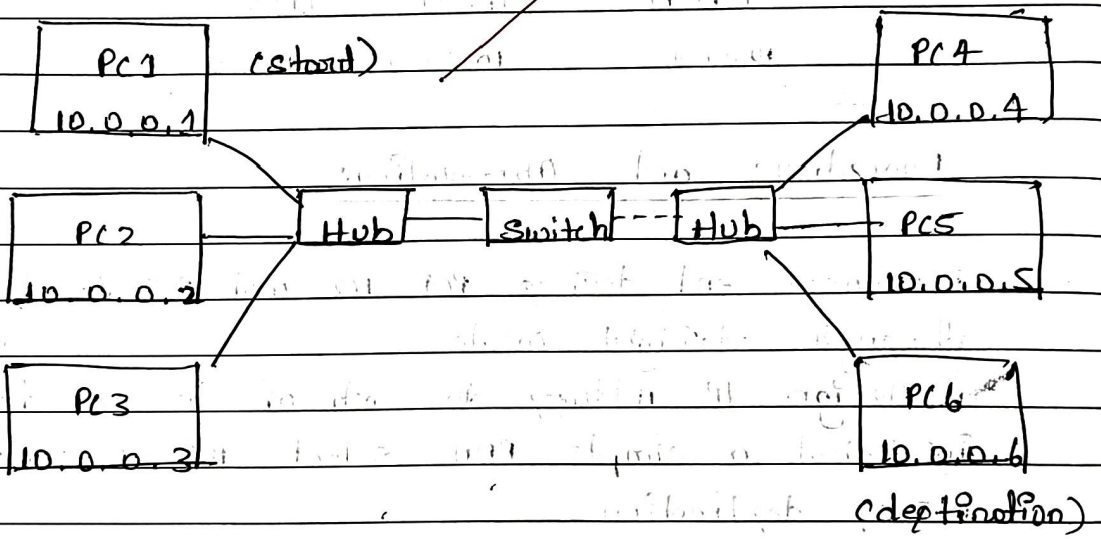


Connect 4 end devices PC1, PC2, PC3, PC4 to the switch with mentioned IP Address.

Select Simple PDC, PC1 as start and PC4 has destination and Simulate

Connection to be made through straight through cables. The message will be sent from PC1 to PC4 and in return the acknowledgement will be sent from PC4 to PC1

Topology 3: Switch, Hub and End devices



Connect the 3 end user devices PC1, PC2 and PC3 with mentioned IP address to a hub and futher connected to a switch.

The connection b/w the Hub and Switch is through a cross over cable.

Then connect switch to another hub with 3 end user device with menitod IP address

The succesfull ping message is confiring the connectivity b/w source & deptination

classmate
Date
Page 03

Difference between Hub / Switch and End device.

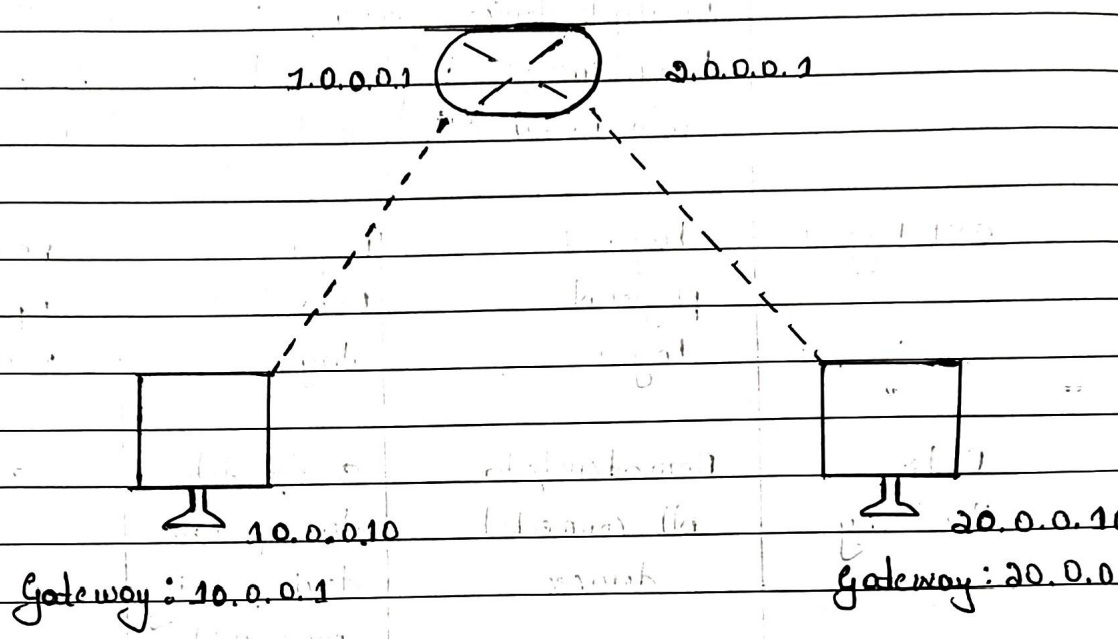
Component	Hub	Switch	End Device
Function	Connects multiple devices broadcasts data to all	Connects devices & forwards data selectively.	Source or destination of data in a network.
OSI Layer	Layer 1 physical layer.	Layer 2 Data link layer.	Varies. Application, Network.
Data Handling	Broadcasts to all connected devices.	Sends data to specific devices using MAC Address.	Sends or receives data.
Efficiency	Low causes network Congestion	High reduces congestion	Depends on the device functionality
Intelligence	No data filtering or decision-making	Filters & forwards data intelligently	No forwarding capability
Use case	Small or Simple network	Medium, larger or more complex	User device (PC's, phones)

9/10/24

Router

classmate
Date 09/10/24
Page 1

Aim: Configure IP address to router in packet tracker explorer. the following message ping response destination unreachable request timeout replay.



Topology: Set 2 different IP address to two different PC's and connect with the router.
PC with 10.0.0.10 hop a gateway 10.0.0.0/1 with router
PC's with 20.0.0.10 hop a gateway 20.0.0.1 with router.

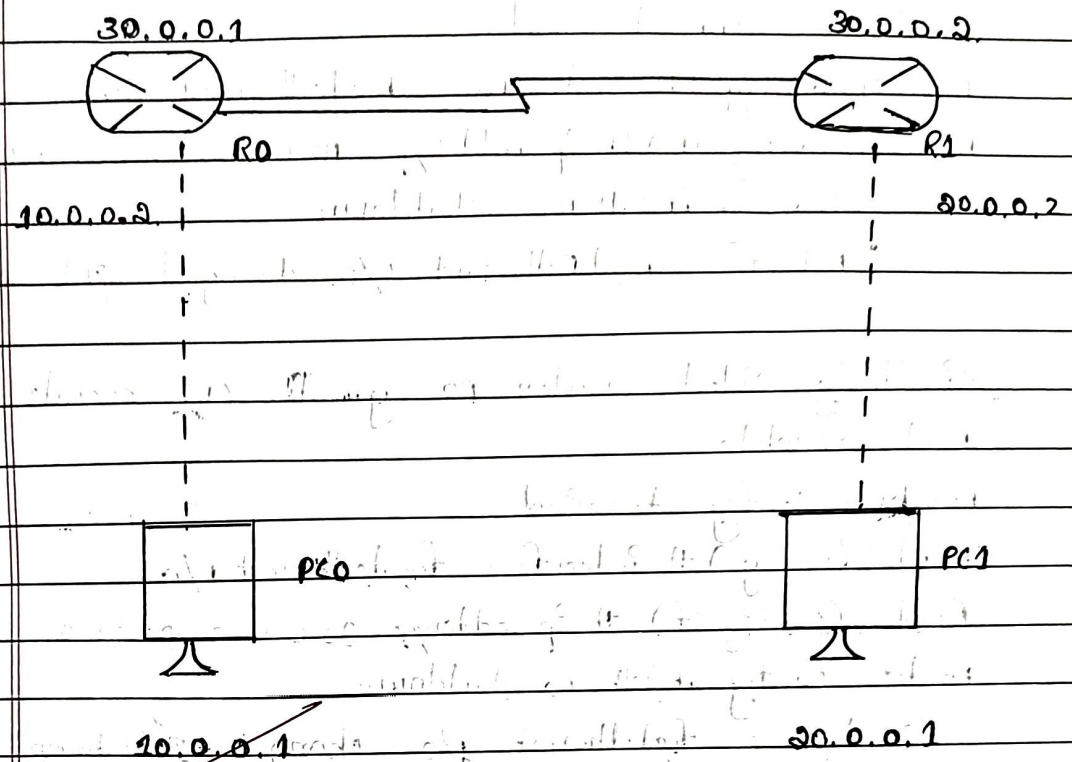
Procedure:

```
router -> enable
router # config Terminal.
router(config)# interface fastethernet 0/0
        ip address 10.0.0.1 255.0.0.0
        no shutdown
        #interface fastethernet 1/0
        #ip address 20.0.0.1 255.0.0.0
        #no shutdown
        exit.
show ip route.
```

10.0.0.0/8 is directly connected.
20.0.0.0/8 is directly connected.

Router.

Aim: Simulate using 2 Router and 2 end devices.



Procedure:

* Select a generic router R1
* Connect an end device PC1 to router R1 through parallel connection fastethernet 0/0
* Configure PC1 with Ip address 10.0.0.1 and gateway 10.0.0
* Similarly select another generic router R2 and connect an end device PC2 fastethernet 1/0.
* Configure PC2 with Ip address 20.0.0.1 and gateway 20.0.0.2.

Now Select router R1 go to CLI and execute the the following

Router >enable
Router # config terminal.
Router (config) # interface fastethernet 0/0
Router (config if) # ip address 10.0.0.2 255.0.0.0
Router (config if) # no shutdown
"Interface fastEthernet 0/0. changed state to up"

Similarly Select router R2 go to CLI execute the same

Router >enable
Router # config terminal.
Router (config) # interface fastethernet 1/0
Router (config if) # ip address 20.0.0.2 255.0.0.0.
Router (config if) # no shutdown.
"Interface fastethernet 1/0 changed state to up"

Hence the connection b/w Routers & end devices is established.

Now Connect router R1 with router R2 using Serial cable (Serially Connected)

To setup connection b/w routers again.

→ Select router R1 and go to CLI
Router (config) # interface serial 2/0
Router (config if) # ip address 30.0.0.1 255.0.0.0
Router (config if) # no shutdown.

→ Select router R2 and go to CLI
Router (config) # interface serial 3/0
Router (config if) # ip address 30.0.0.2 255.0.0.0
Router (config if) # no Shutdown
"Interface serial 2/0 changed state to up"
3/0.

Observations:

→ After setting up the mentioned topology Now try to ping PC2 with PC1.

Open command prompt for PC1 type Ping 20.0.0.1

→ Destination host unreachable

Packets send:4 received: 0 lost = 4 Loss = 100%.

It is also observed that the end system PC1 was only pinged with router R1 only.

ping 20.0.0.1 → Successful.

Packets sent:4 received: 4 lost:0 0%.

Hence although the routers were connected serially. the end devices were unreachable to ping other device.

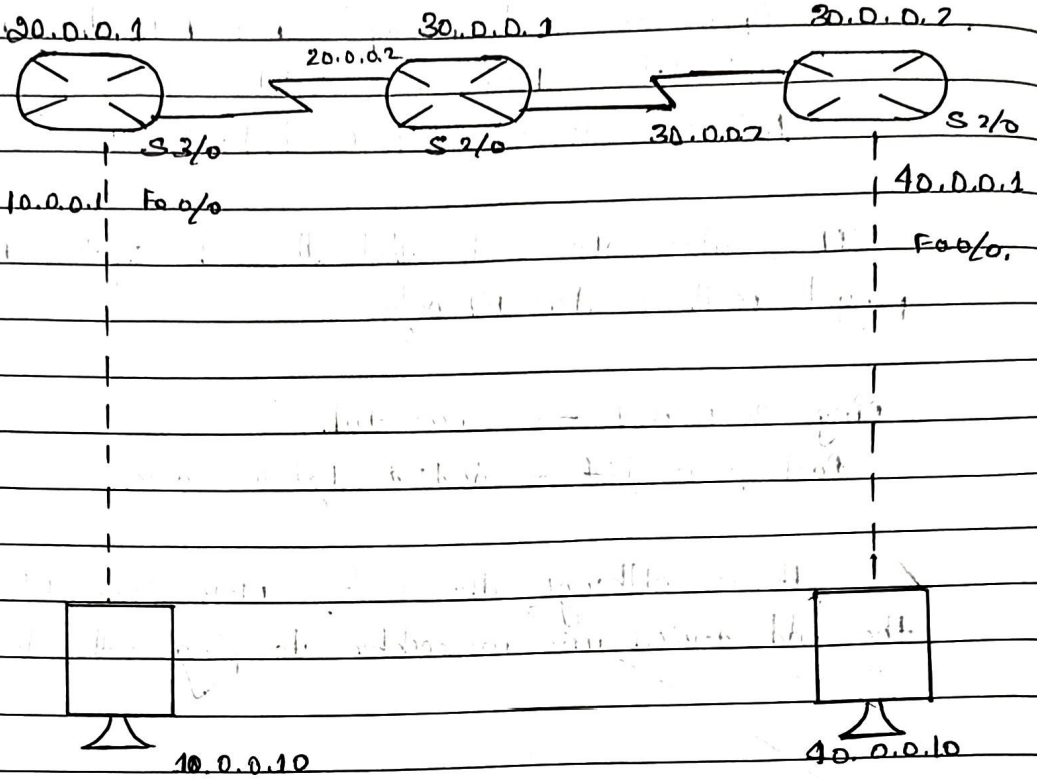
25/10/24

Lab-4

3 Routers and 2 PC

Date 23/10/21
Page 8

Aim: Configuration of default route, Static route to the Routers.



Procedure:

* Select the generic Router R1, R2, R3.
* Connect the End devices PC1 to R1 and PC2 to R3.

Now select router R1 go to CLI and execute the following.

```
Router > enable
Router # config terminal
Router (config) # Interface fastethernet 0/0
Router (config) # ip address 10.0.0.1 255.0.0.0
Router (config if) # no shutdown
```

classmate
Date
Page 9

Now connect router R1 with router R2 using serial cable (serially connected).

To setup connection b/w routers again.
-> select router R1 and go to CLI
Router (config) # interface serial 2/0.
Router (config if) # ip address 20.0.0.1 255.0.0.0.
Router (config if) # no shutdown.

Now connect the routers.

-> ip route 0.0.0.0 0.0.0.0 20.0.0.2

To check the connection between routers and other.

-> show ip route:

Observation: After setting up the mentioned topology now try to ping PC1 to PC2

open the command prompt for PC1 type ping 40.0.0.10 and observe the data transfer.

Reply from 40.0.0.10 : bytes = 32 time = 11ms TTL=125
Reply from 40.0.0.10 : byte = 32 time = 9ms TTL=125
Reply from 40.0.0.10 : byte = 32 time = 2ms TTL=125
Reply from 40.0.0.10 : byte = 32 time = 2ms TTL=125.

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

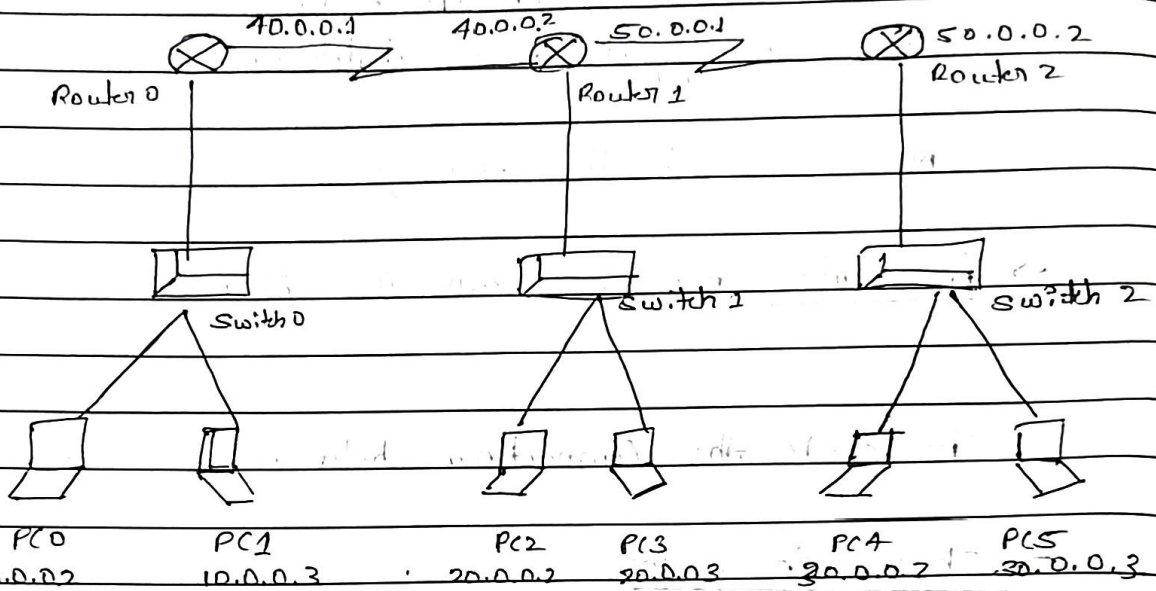
Lab-6

classmate
Date 20/11/24
Page 10

Aim: Configure RIP routing Protocol in Routers

Devices: 3 Router, 3 switch and 6 PC

Topology:



Procedure: Set up the topology as mentioned

Set the ip configuration for routers

Route R0 - 10.0.0.1, R1 - 20.0.0.1, R2 - 30.0.0.1

PC0 - 10.0.0.2
PC1 - 10.0.0.3 } Gateway 10.0.0.1

PC2 - 20.0.0.2
PC3 - 20.0.0.3 } Gateway 20.0.0.2

PC4 - 30.0.0.2
PC5 - 30.0.0.3 } Gateway 30.0.0.1

To establish connection between routers, serially goto CLI of routers and configure

R0 - Se 2/0 - 40.0.0.1

R1 - Se 2/0 - 40.0.0.2, Se 3/0 - 50.0.0.1

R2 - Se 2/0 - 50.0.0.2

To establish RIP in router, goto CLI of Router 0.

```
Router (config) # router rip
Router (config-router) # network 10.0.0.0
                       # network 40.0.0.0
```

Router 1

```
Router (config) # router rip
Router (config-router) # network 20.0.0.0
                       # network 40.0.0.0
                       # network 50.0.0.0
```

Router 2

```
Router (config) # router rip
Router (config-router) # network 30.0.0.0
                       # network 30.0.0.0
```

To overcome the rip connection, check show ip route in router CLI denoting R.

Now in PC0 goto command prompt and

```
PC > ping 30.0.0.2
pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2 : bytes = 32 time=7ms , TTL = 125
                      bytes = 32 time = 7ms, TTL = 125
                      bytes = 32 time = 9ms, TTL = 125
Reply from 30.0.0.2 : bytes = 32 time = 7ms . TTL = 125

Ping Statics for 30.0.0.2
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)
App round trip time in ms.
    Min = 7ms, max = 9ms, Avg = 7ms.
∴ ping was successfull.
```

⑦ Demonstrate the TTL or life of packet

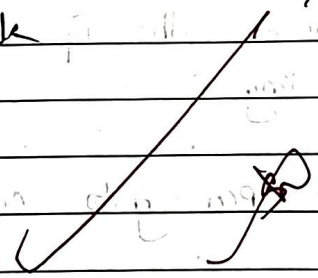
Procedure:

-> Setup the topology which is done in previous program
-> Goto Simulation, Select Simple PDU and select Source and destination PCs
-> Click on Auto capture/play then the packet will start to move and eventually reached destination.

Observation:

- Router is level 1, 2 & 3 device which contains the details about the message.

- TTL (Time to live) or life of a packet tells about how much time is required so that the message should stay in network

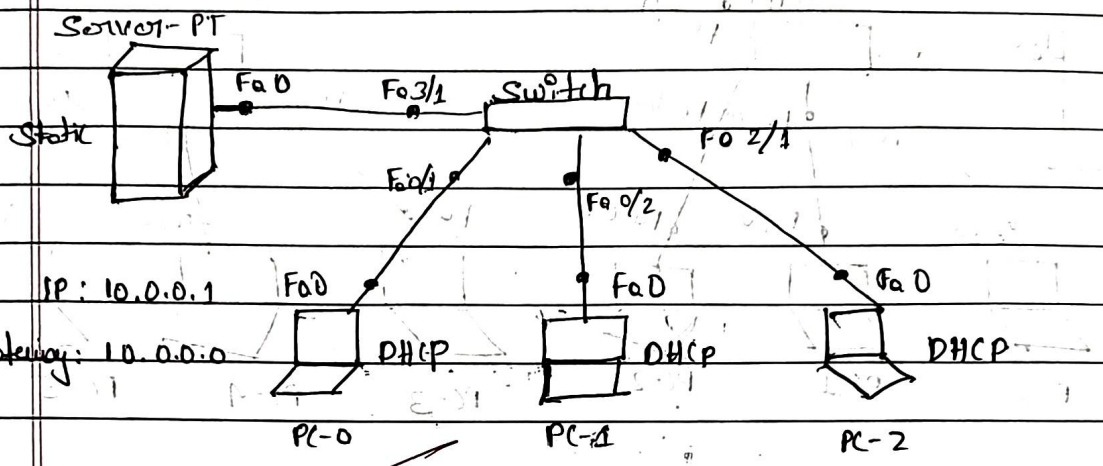


lab-5

5) Design a DHCP within LAN and outside LAN Dynamic Host Configuration Protocol.

Devices used: One switch, one server, 3 end devices.

Topology: (within LAN)

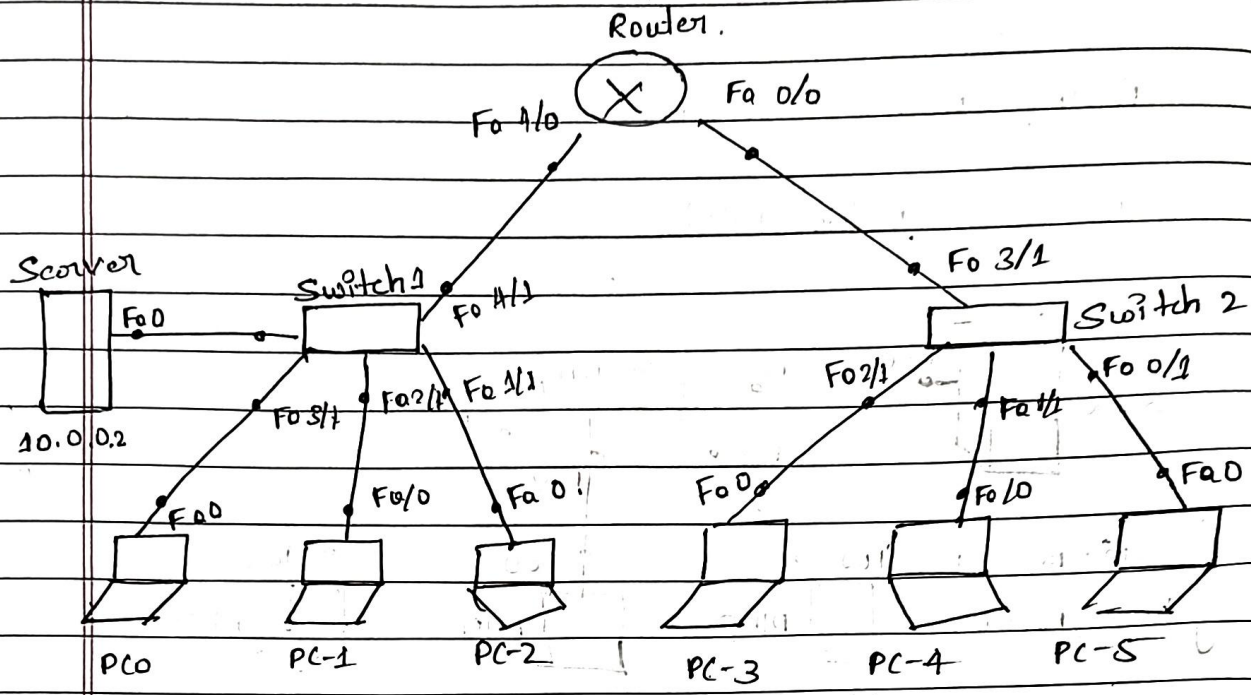


Procedure

Set up the topology as mentioned.
Goto server IP Configuration (Desktop)
Select IP address 10.0.0.1, 255.0.0.0, 10.0.0.0
Then to set up DHCP, goto config, services select DHCP.
Make DHCP to all three PC's
Dynamic IP addresses will be assigned.

Now ping PC-0 with PC-1 and PC-2
PC-0 had 10.0.0.2 so goto command prompt
-> ping was successful ping 10.0.0.3.

Outside LAN:



Set up the topology as mentioned

Go to Server - PT, Desktop, IP Configuration change default gateway to 10.0.0.1

Now goto Router CLI

Router > enable
Router # config terminal
Router (config) # interface fastethernet 4/0
Router (config if)# ip address 10.0.0.1 255.0.0.0
Router (config-if)# ip helper-address 10.0.0.2
Router (config-if)# no shut

Both networks have established connection with switch 1 switch 2 with router.

We can notice ip address of

PC0 - 10.0.0.5	PC3 - 20.0.0.4
PC1 - 10.0.0.6	PC4 - 20.0.0.3
PC2 - 10.0.0.3	PC5 - 20.0.0.5

Now try to ping PC0 with PC5
Goto Desktop of PC0, command prompt

ping was Succesfull.

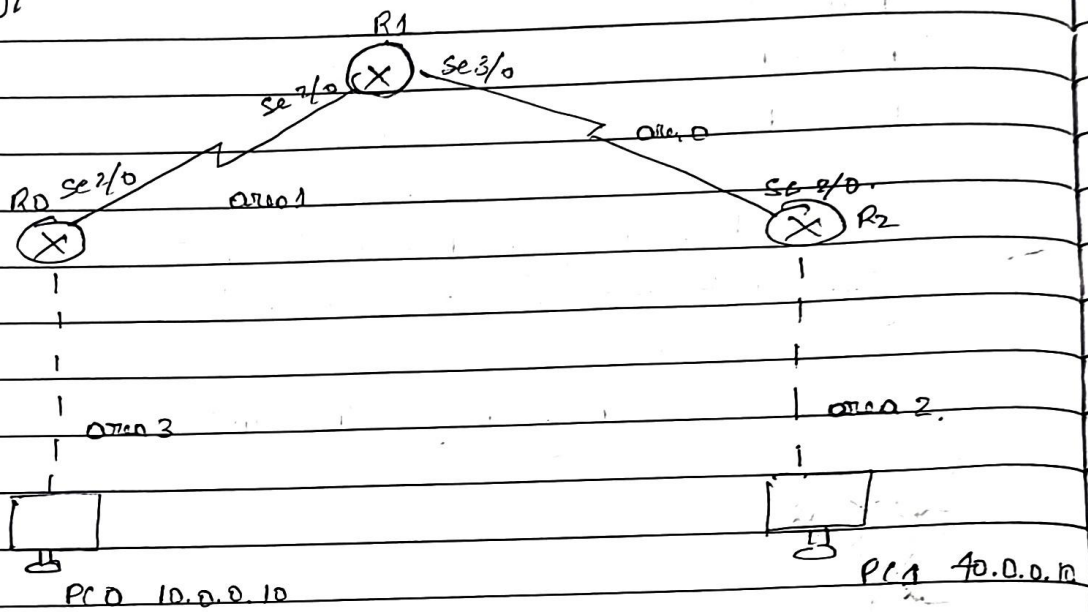
Lab-7

classmate
Date 27/11/24
Page 16

6→ Aim: Configure OSPF routing protocol

Devices: 3 Router, 2 PC,

Topology:



Step 2: Configure ip address to all interface.

In Router R1

R1 (config)# interface fogit ethernet 2/0
R1 (config-if)# ip address 10.0.0.1 255.0.0.0
R1 (config-if)# no shutdown
exit.

R1 (config)# interface serial 1/0
R1 (config-if)# ip address 20.0.0.1 255.0.0.0
R1 (config-if)# encapsulation ppp
R1 (config-if)# clock rate 64000
R1 (config-if)# exit.

In Router R2

R2 (config)# interface serial 1/0
R2 (config-if)# ip address 20.0.0.2 255.0.0.0
encapsulation ppp
no shutdown
exit

R2 (config)# interface serial 1/1
R2 (config-if)# ip address 30.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shutdown
exit

In Router R3

R3 (config)# interface serial 1/0
R3 (config-if)# ip address 30.0.0.2 255.0.0.0
R3 (config-if)# encapsulation ppp
no shutdown
exit

R3 (config)# interface fastethernet 2/0
R3 (config-if)# ip address 40.0.0.1 255.0.0.0
no shutdown
exit

Step 3: Now, Enable ip routing by configuring OSPF routing protocol in all routers.

In Router R1

R1 (config)# router OSPF 1
R1 (config-router)# router-id 1.1.1.1
R1 (config-router)# network 10.0.0.0 0.255.255.255 area 3
R1 (config-router)# network 20.0.0.0 0.255.255.255 area 1
exit

In Router R2

R2 (config) # router OSPF 1

R2 (config-router) # 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.

In Router R3

R3 (config) # router OSPF 1

R3 (config-router) # 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

You have to configure router id when we configure ospf .. It is used to identify the router.

Step 4: Now check routing table of R1.

Router # show ip route.

C. 10.0.0.0/8 is directly connected FastEthernet 2/0

C. 20.0.0.0/8 is directly connected Serial 1/0

O IA 40.0.0.0/8 [110/129] via 20.0.0.2 00:04:29 Serial 1/0

O IA 30.0.0.0/8 [110/128] via 20.0.0.2 00:07:29. Serial 1/0

Here, R2 knows Area 0. Network 20.0.0.0 connected to R2 from R1 learns network through this networks.

R3 (config) # router OSPF 1 , here 1 is process ID

There must be one interface up to keep OSPF process up. So its better to configure loopback address to router. If it is a virtual interface never goes down once we config.

R1 (config)# interface loopback 0
R1 (config-if)# ip address 172.16.1252 255.255.0.0
no shutdown

R2 (config)# interface loopback 0
R2 (config-if)# ip address 172.16.1.253 255.255.0.0
no shutdown

R3 (config)# interface loopback 0
R3 (config-if)# ip address 172.16.1.254 255.255.0.0
no shutdown

step 5: Now check Routing table of R3.

R3 # show ip route
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:15:38 serial 1/0
C 40.0.0.0/8 is directly connected, FastEthernet 2/0
C 30.0.0.0/8 is directly connected, serial 1/0.

Here, R3 doesn't know about the area 3, so we have to create virtual link b/w R1 and R2.

step 6: Create virtual link b/w R1, R2 by this we create a virtual links to connect area 3 to area 0.

In Router R1,
R1 (config)# router OSPF 1
R1 (config)# area 1 virtual link 2.2.2.2

In Router R2
R2 (config)# router OSPF 1
R2 (config-router)# area 1 virtual link 1.1.1.1
exit

Step 7: R2 and R3 get updated about Area 3, Now, check routing table of R3.

R3 # show ip route

O IA 20.0.0.0/8 [110/721] via 30.0.0.1, 00:01:51, Serial 1/0
C 30.0.0.0/8 is directly connected, FastEthernet 2/0
O IA 10.0.0.0/8 [110/729] via 30.0.0.1, 00:01:51, Serial 1/0
C 80.0.0.0/8 is directly connected, Serial 1/0.

Step 8: Check connectivity b/w host 10.0.0.10 to 40.0.0.10.

#ping 40.0.0.10

Now, if we get the reply without drops then the connectivity is established

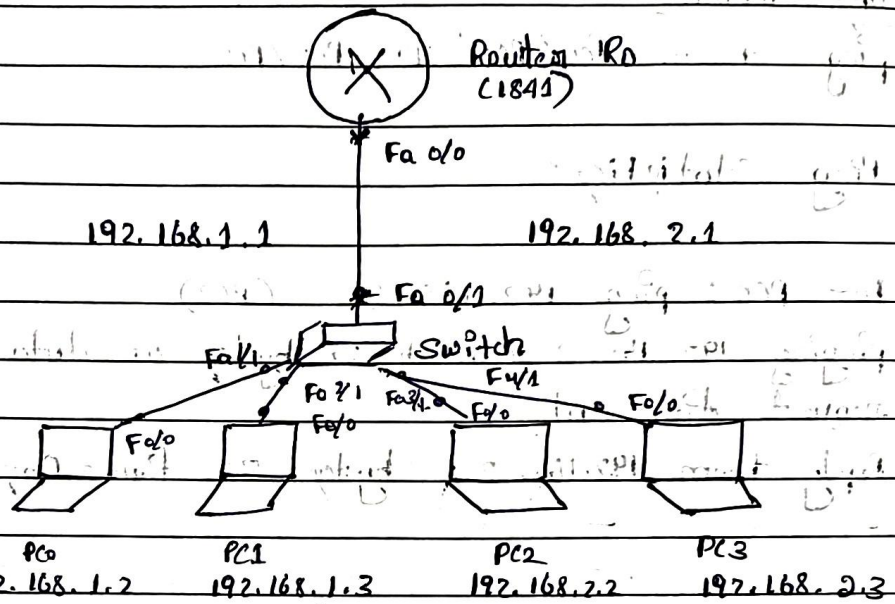
30/12/24

Lab - 8

9) Construct a VLAN and make the PC's communicate among a VLAN

Devices Used: 1 router, 1 switch, 4 PC's

Topology:



Procedure: Set up the topology as seen above.
Use specific router 1841

→ Assign IP addresses to the PCs as shown in topology

→ Go to switch, config and select VLAN database
Give VLAN number 2
VLAN Name: CSE click add.

The port to switch which is connected to the router Fa 0/1 Select trunk

For the PC's, PC0 and PC3. Select the respective ports in the switch and change the VLAN to 2

click on the router and goto CLI

Execute the commands and also add the VLAN number (2) and VLAN Name (CSE) in the config of Router.

exit
config terminal
Interface fastethernet 0/0.1
encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0
no shut
exit

Observation:

ping from PC0/PC1 to PC2/PC3.

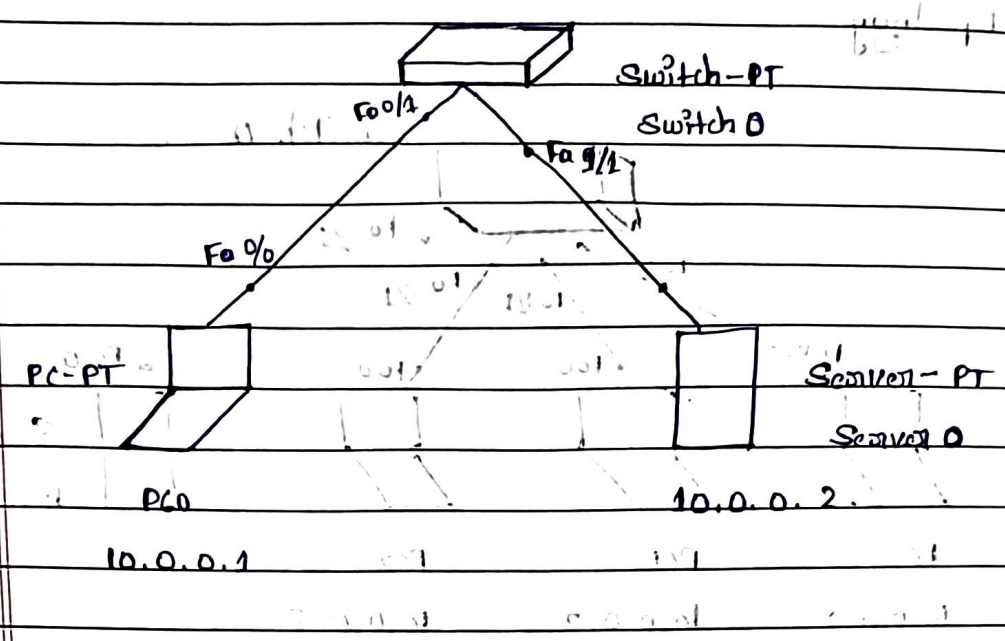
Ping Statistics:

For PC0: ping 192.168.2.2 (PC3)
pinging 192.168.2.2 with 32 bytes of data
request time out
Reply from 192.168.2.2 bytes = 32 time = 0ms TTL = 127.

Pinging Statistics for 192.168.2.2:
Packets Sent = 4 received = 3 lost = 1 loss = 25%.

30/12/24

10) Configure Web Server, DNS within a LAN



Procedure:

- Select 1 switch, 1 PC and 1 server, setup the topology as shown.
- Configure the PC and server with ip addresses as mentioned.
- click on server, goto config click on DNS and ensure DNS service is ON, add the name & the IP addresses.
- Click on PC, then web browser in Desktop.
  enter URL 10.0.0.2 and name given.
- click on server then HTTP, change the text and click '+'
- Goto web browser of PC, type URL and observe the changes to the text.

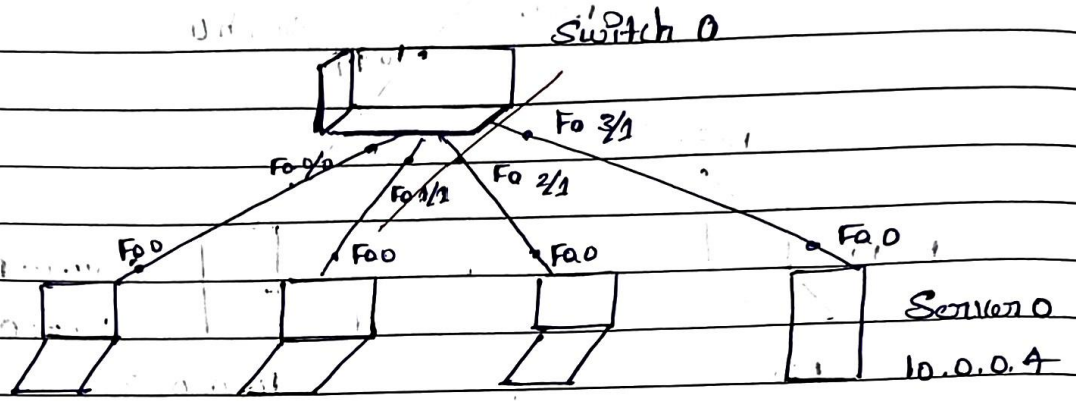
Observation:

DNS resolves domain names to IP addresses, allowing clients to access the web server using domain names of IP addresses with successful communication verified by HTTP requests.

- The server web page could be successfully accessed from the PC by entering the resource URL
- The DNS server could hence be configured within a LAN by enabling the DNS.

11) To Construct a Simple LAN and Understand the concept and Operation of address Resolution Protocol (ARP)

Topology



Procedure:

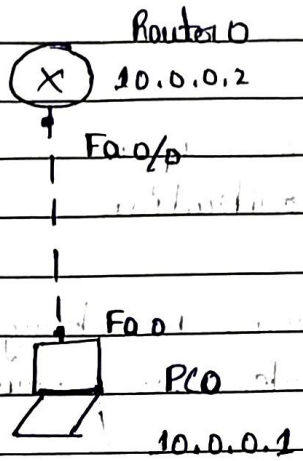
Select 1 switch, 1 server and 3 end devices / PC's
Configure the devices as shown in the topology
Select Inspect tool and click on a PC say PC0 then click on ARP table. An empty ARP table appears.
Do the same for other PC's, Server and switch.
Select Simple PDU and choose source and destination, click on Simulation and keep checking the ARP table after every click on Capture / forward.
click on Switch - CLI and type : show mac address - table.

Observation: Initially ARP of all devices observed empty. After every click on Capture / forward, in the ARP table update of ARP requests and replies are exchanged showing the mapping b/w IP address & MAC address of device involved in the communication.

30/12/24

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

Topology:



Procedure:

Select 1 router and 1 PC. Configure/Setup the topology as shown goto CLI of router and type the following commands.

```
enable
config terminal
hostname CSE
enable secret password
interface fastethernet 0/0
ip address 10.0.0.2 255.0.0.0
no shut
line vty 0 3.
login
password psw
exit
exit.
```

Now goto Command prompt PC0, enquire ping 10.0.0.2 ip Successful.

Adab in the cmd of PC, type telnet 10.0.0.2 give password which was set and observe the changes

Observation

telnet 10.0.0.2
Trying 10.0.0.2 ... Open
User Access Verification
Password psw

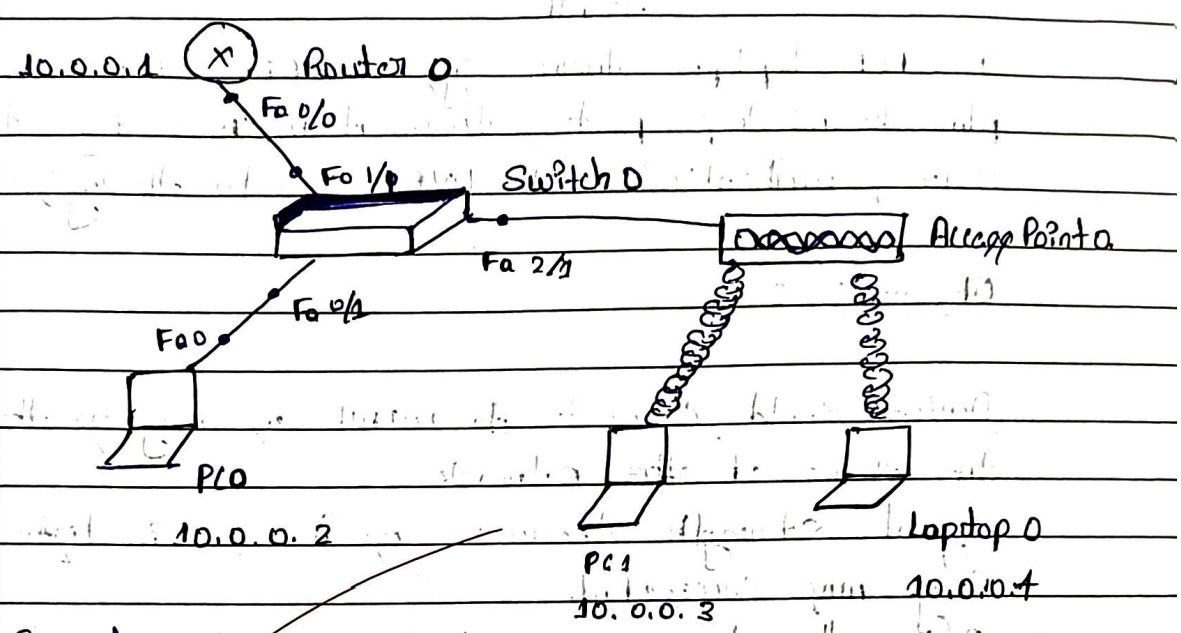
Here the name PC changed to CSE which was the hostname execute the commands.

enable
password psw1
Show ip route
10.0.0.8 is directly connected. Gigethernet 0/0.

30/12/24

13) To construct a WLAN and make the nodes communication wirelessly.

Topology:



Procedure:

Select 1 router, 2 PCs, 1 laptop, 0 switch and on access point.
Set up the topology as shown (Initially no connection bln end point & Access point)
Configure the devices with their ip addresses as shown.
To Configure access point, goto port 1.
give a name to SSID as CSE
Select WEP and give 10 digit WEP key as 1234567890
ensure the port status is ON
Configure PC1 and laptop with wireless standards.
For PC0, switch off, drag existing PT-HOST-NM-1AM to unpaired slotted in LHS.
Drag WMP300N wireless interface to the empty port
Switch on PC of Access Point 0.

In the config tab a new wireless interface would have been added configure SSID, WEP, WEP key, IP address & gateway for the device [Port status should be set 'on' set SSID name & Do similar process for laptop ping from every device to all other device and observe the result.

In PC1, laptop, turn systems OFF, remove the ports, place the wireless point & turn it on. In config, set some SSID & authentication to WEP & enter the key).

Observation:

- Device could connect to WLAN as long as they are in the range of the network.
- Signal strength decrease as increase in distance.
- Ping was successful.
- After the setup, PC1 & laptop0 wireless connection are observed with Access point as indicating successful wireless connection.

5/5
30/1/24

Part B

1. Write a program for error detecting code using CRC-CCITT (16 bits)

Solution:

```
#include <iostream>
#include <string.h>
using namespace std;

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";

 cout << "Enter the input message in binary" << endl;
 cin >> ip;
 crc(ip, op, poly, 1);
 cout << "The transmitted message is: " << ip << op << endl;
 cout << "Enter the received message in binary" << endl;
 cin >> recv;
 if (!crc(recv, op, poly, 0))
    cout << "No error in data" << endl;
 else
    cout << "Error in data transmission has occurred" <<
endl;
 return 0;
}
```

Output 1

Enter the input message in binary: 111101
The transmitted message is: 11110111010111100111010
Enter the received message in binary 111101
No error in data.

Output 2.

Enter the input message in binary: 111101
The transmitted message is: 11110111010111100111010
Enter the received message in binary 11110
Error in data transmission has occurred.

30/12/24

2 Write a program for congestion control using Leaky bucket algorithm.

Algorithm: 1. Start
2. Set the bucket size or the buffer size
3. Set the output rate
4. Transmit the packets such that there is no overflo
5. Repeat the process of transmission until all packets are transmitted.
6. Stop.

```
#include <iostream>
#include <string.h>
using namespace std;
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define NOF_PACKETS 10
int rand(int a) {
    int rn = (random() % 10) % a;
    return rn == 0 ? 1 : rn;
}
int main() {
    int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate,
p_sz_rm = 0, p_sz, p_time, op;
    for (i=0; i<NOF_PACKETS; ++i)
        packet_sz[i] = rand(6) * 10;
    for (i=0; i<NOF_PACKETS; ++i)
        printf("\n packet [%d]: %d bytes \t", i, packet_sz[i]);
        printf("\n Enter the Output rate:");
        scanf("%d", &o_rate);
        printf("Enter the Bucket Size:");
        scanf("%d", &b_size);
        for (i=0; i<NOF_PACKETS; ++i)
```

```
{
    if((packet_sz[i] + p_sz_rm) > b_size)
    if (packet_sz[i] > b_size)
        printf("\n\n Incoming packet size (%d bytes) is greater than bucket capacity (%d bytes) - PACKET REJECTED", packet_sz[i], b_size);
    else
        printf("\n\n Bucket capacity exceeded - PACKETS REJECTED!!");
    else {
        p_sz_rm += packet_sz[i];
        printf("\n\n Incoming Packet size: %d", packet_sz[i]);
        printf("\n Bytes remaining to transmit: %d", p_sz_rm);
        p_time = rand(4)*10;
        printf("\n Time left for transmission: %d units", p_time);
        for (clk = 10; clk <= p_time; clk += 10)
        {
            sleep(1);
            if (p_sz_rm)
            {
                if (p_sz_rm <= 10_rate)
                    p_sz_rm = 0;
                else
                    op = 0_rate; p_sz_rm -= 0_rate;
                printf("\n Packet of size %d Transmitted", op);
                printf("\n Bytes Remaining to Transmit: %d", p_sz_rm);
            }
            else {
                printf("\n Time left for transmission: %d units", ptime - clk);
                printf("\n No packets to transmit");
            }
        }
    }
}
```

packet[0]: 30 bytes
packet[1]: 10 bytes
packet[2]: 10 bytes
packet[3]: 50 bytes.
packet[4]: 30 bytes
packet[5]: 50 bytes.
packet[6]: 10 bytes.
packet[7]: 20 bytes.
packet[8]: 30 bytes.
packet[9]: 10 bytes.

Enter the Output rate: 100
Enter the Bucket Size: 50
Incoming Packet size: 30.
Bytes remaining to Transmit: 30.
Time left for transmission: 20 units.
Packet of size 30 Transmitted --- Bytes Remaining to Transmit: 0
Time left for transmission: 0 units.
No packets to transmit!!

Incoming Packet size: 10
Bytes remaining to Transmit: 10
Time left for transmission: 10
Packet of size 10 Transmitted --- Bytes Remaining to Transmit: 0
Time left for transmission: 10 units.
no packets to transmit!!
Time left for transmission: 0 unit.
No packets to transmit!!

Incoming Packet size: 50
Bytes remaining to Transmit: 50
Time left for transmission: 10 units
Packet of size 50 Thousand --- Bytes Remaining to Transmit 0

Incoming Packet size: 30
Bytes remaining to Transmit: 30
Time left for transmission: 30 units
Packet of size 30 transmitted -- Bytes Remaining to Transmit: 0
Time left for transmission: 10 units
No packets to transmit!!
Time left for transmission: 0 units
No packets to transmit!!

Incoming Packet size: 50
Bytes remaining to Transmit: 50
Time left for transmission: 20 units
Packet of size 50 Transmitted -- Bytes Remaining to Transmit: 0
Time left for transmission: 0 units
No packets to transmit!!

Incoming Packet size: 10
Bytes remaining to Transmit: 10
Time left for transmission: 10 unit
Packet of size 10 Transmitted -- Bytes Remaining to Transmit: 0

30/12/24

3.) Using TCP/IP sockets, write a client-server program. to make client sending the file name and the server to send back the contents of the requested file if present.

Client side

```
#include <unistd.h>
int main() {
    int soc, n;
    char buffer[1024], fname[50];
    struct sockaddr_in addr;
    soc = socket(PF_INET, SOCK_STREAM, 0);
    addr.sin_family = AF_INET;
    addr.sin_port = htons(7891);
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    while(connect(soc, (struct sockaddr *) &addr, sizeof(addr))
    printf("\n Client is connected to Server");
    printf("\n Enter file name:");
    scanf("%s", fname);
    send(soc, fname, sizeof(fname), 0);
    printf("\n Received response\n");
    while((n = recv(soc, buffer, sizeof(buffer), 0)) > 0)
    printf("%s", buffer);
    return 0;
}
```

Server Side

```
#include <stdio.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <unistd.h>
int main() {
    int welcome, new_soc, fd, n;
```

```
char buffer[1024], fname[50];
struct sockaddr_in addr;
welcome = socket(PF_INET, SOCK_STREAM, 0);
addr.sin_family = AF_INET;
addr.sin_port = htons(7890);
addr.sin_addr.s_addr = inet_addr("127.0.0.1");
bind(welcome, (struct sockaddr *) &addr, sizeof(addr));
printf("\n Server is Online");
listen(welcome, 5);
new_soc = accept(welcome, NULL, NULL);
recv(new_soc, fname, 50, 0);
printf("\n Requesting for file: %s\n", fname);
fd = open(fname, O_RDONLY);
if(fd<0)
send(new_soc, "\n File not found\n", 15, 0);
else
while((n = read(fd, buffer, sizeof(buffer)))>0)
send(new_soc, buffer, n, 0);
printf("\n Request Sent\n");
close(fd);
return 0;
}
```

Output:

```
Server is Online
Requesting for file: dept.txt
Request Sent

Client is connected to server.
Enter file name: dept.txt
Received response.
Hello world.
```

30/12/24

4) Using UDP

make client

send back the

```
// Server
#include <stdi
#include <sto
#include <sy
#include <ar
#include <sy
#include <ne
#define POR
#define MA
int main() {
  char buff
  char *mes
  int listen
  struct soc
  bzero(&s
  listenfd =
  servaddr.
  servaddr.
  servaddr.
  bind(list
  sizeof(serv
  dem = size
  int n = r
(struct so
  buffer[n]
  puts(buff
  send to C
&cliaddr, size
}
```

1) Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

// Server program for udp connection.

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define PORT 5000
#define MAXLINE 1000
int main() {
    char buffer[100];
    char *message = "Hello Client";
    int listenfd, len;
    struct sockaddr_in servaddr, cliaddr;
    bzero(&servaddr, sizeof(servaddr));
    listenfd = socket(AF_INET, SOCK_DGRAM, 0);
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;
    bind(listenfd, (struct sockaddr*) &servaddr,
sizeof(servaddr));
    len = sizeof(cliaddr);
    int n = recvfrom(listenfd, buffer, sizeof(buffer),
0, (struct sockaddr*) &cliaddr, &len);
    buffer[n] = '\0';
    puts(buffer);
    sendto(listenfd, message, MAXLINE, 0, (struct sockaddr
*) &cliaddr, sizeof(cliaddr));
}
```

Udp client driver program,

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdlib.h>
#define PORT 5000
#define MAXLINE 1000
int main(){
    char buffer[100];
    char *meggage = "Hello Sever";
    int sockfd n;
    struct sockaddr_in servaddr;
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if(connect(sockfd, (struct sockaddr*) &servaddr,
sizeof(servaddr)) < 0)
    {
        printf("\n Error: Connect Failed \n");
        exit(0);
    }
    send to(sockfd, meggage, MAXLINE, 0, (struct
sockaddr*) NULL, sizeof(servaddr));
    recvfrom(sockfd, buffer, sizeof(buffer), 0, (struct
sockaddr*) NULL, NULL);
    puts(buffer);
    close(sockfd);
}
```

Output

Server o/p.
Server is online.
Hello server.

client o/p.
Hello client

30/12/24