VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Computer Networks Lab

Submitted by

Sagar C Mannannavar(1BM18CS181)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
June-2023 to September-2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**Computer Networks**" carried out by **Sagar C Mannannavar(1BM18CS181),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the academic semester June-2023 to September-2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Analysis and Design of Algorithms (22CS4PCADA)** work prescribed for the said degree.

Dr. Shyamala G	**Dr. Jyothi S Nayak**
Assistant Professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

Index Sheet

Cycle 2

Cycle 2

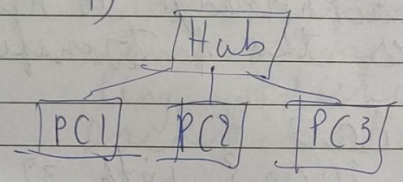
PAGE EDGE
DATE: 16/6/23

1) Create a topology consisting of 3 or more devices with a help of a 1) Hub 2) switch 3) Hub & switch simulate sending a simple PDU from source to destination & demonstrate a ping message

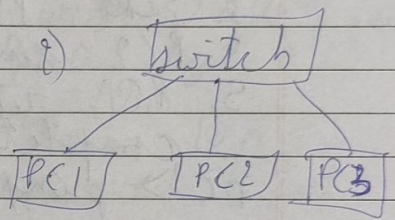
→ Aim:- To understand different ping message when they are caused in given topology

Topology:

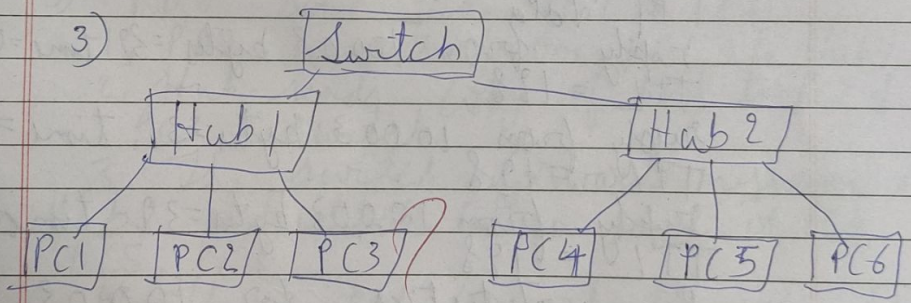
1)



2)



3)



Procedure:

1) Place the required no of devices (PC,s)
2) Add a generic Hub for 1) generic switch 3) both & connect them with the devices
3) In exp 3) 2 hubs & switch are connected
4) Send a simple PDU message from source PC to destination PC & observe the simulation. This is done in simulation mode

9) In real time mode, open the command prompt of PC1 & ping one of the end devices

Result

1) 1. The simple PDU is sent from PC1 to hub
2. Hub sends the PDU to all ports except PC1
3. The PDU is rejected by PC3
4. PC2 sends acknowledge to hub
5. PC1 recieves this & transfer is completed

PC > ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data

reply from 10.0.0.3 bytes=32 time=0ms TTL=128

reply from 10.0.0.3 bytes=32 time=0ms TTL=128

reply from 10.0.0.3 bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3

packets: sent = 4, recieved = 4, lost = 0 (0% loss)

Approximate Round trip time in milliseconds minimum = 0 ms

maximum = 0 ms Average = 0 ms

PAGE EDGE
DATE : / /

2) PDU sent from PC2 to PC3
1. The PDU is sent from PC1 to switch
2. Switch broadcasts the PDU to all outport except the inputport
3. PC3 acknowledge & sends the switch transfer this to PC1 without broadcast

PC > ping 10.0.0.3
ping 10.0.0.3 with 32 bytes of data
reply from 10.0.0.3 bytes = 32
time = 0 ms TTL = 128
ping statistics from 10.0.0.3
packets: sent = 4 recieved = 4 lost = 0

3) PDU sent from PC1 to PC6
1. PDU is sent from PC1 to Hub1
2. Hub sends copy of PDU to PC2 & PC3 & switch
3. The switch forwards message to Hub2 which inturn sends it PC4, PC5, PC6
4. The switch forwards the acknowledgement
5. The PC6 forwards the acknowledge to ~~hub~~
6. The switch forwards the acknowledge to hub 1
7. PC1 receives the acknowledgement & transfer is completed

PC > ping 10.0.0.6

pinging 10.0.0.6 with 32 bytes of data

reply from 10.0.0.6 bytes=32 time=0ms TTL=128

Ping statistics from 10.0.0.6

packets sent = 4 received = 4 lost = 0

Approximate round trip time in ms

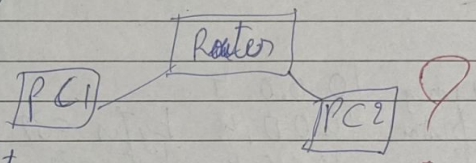
minimum = 0ms Max = 0ms Average = 0ms

PAGE EDGE
DATE : 23 / 6 / 23

Q Create a topology consists of 2 devices connected with help of a router

→ Aim :- * To create a topology using one router
* Using multiple routers

→ only one router



Result

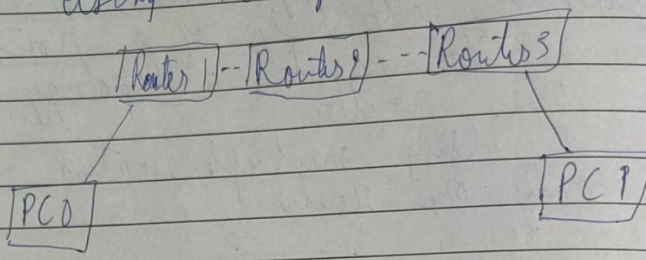
PC> ping 10.0.0.3

pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 255
Reply from 10.0.0.3 bytes = 32 time = 3ms TTL = 255
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 255
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 255

ping statistics for 10.0.0.3
packets : sent = 4 received = 4 lost = 0
Approximate round trip in ms
min = 0 max = 3ms average = 0ms

* using Multiple Router



Result

PC > ping 10.0.0.4
Reply from 10.0.0.4 bytes =32 time=0ms TTL=255
Reply from 10.0.0.4 bytes =32 time=0ms TTL=255
Reply from 10.0.0.4 bytes =32 time=0ms TTL=255
Reply from 10.0.0.4 bytes =32 time=0ms TTL=255

Ping statistics for 10.0.0.4
packets = sent=4 received=4 lost=0
Approximate round trip times in ms
min = 0ms max = 0ms Average = 0ms

Objective :-

* Managing traffic between these network by forwarding data to their intended IP address & allowing multiple devices to use the same internet connection.

* A Router in networking also enable high speed connectivity which is required for application such as streaming video & exchanging files over a network

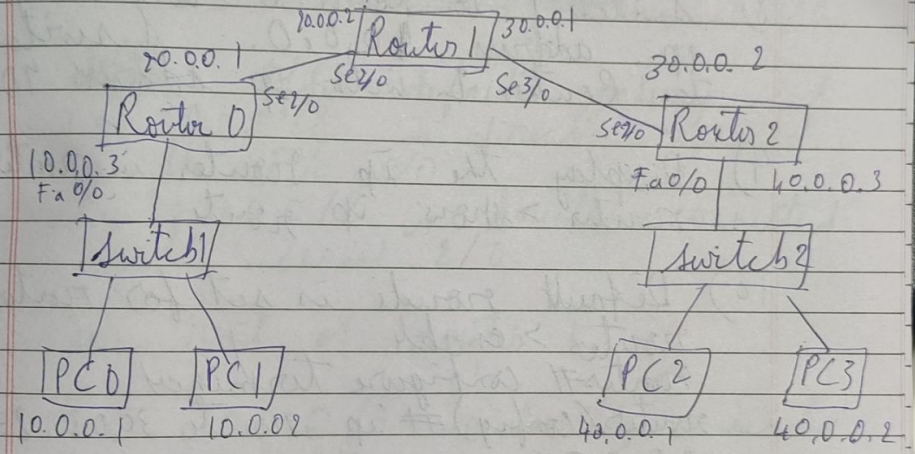
PAGE EDGE
DATE : 30/ 6/ 23

3 Configure default route, Static route to the Routers

~~Procedure~~

Aim :- To understand how to configure static routing & default routing

Topology



Procedure

1) Connect two end devices to a switch which is again connected to a routers and another same setup is connected in similar ~~fashion~~ type
2) Both routers/setup are connected with a router1
3) End ~~device~~ PC0 & PC1 are set with ip ~~address~~ 10.0.0.1 & 10.0.0.2 statically ~~like~~ by PC2 & PC3 with 40.0.0.1 & 40.0.0.2
4) PC0 & PC1 are connected to switch & then to router 0 ~~like~~ by PC2 & PC3 to router 2.

5) Switch 1 connected to router 0 with FA 0/1
& Switch 2 connected to router 2 with FA 0/1
6) Router 0 connected to Router 1 with
Se0/0 with Router 0 serial ip address 20.0.0.1
& Router 1 ip address 20.0.0.2
7) Router 1 connected to router 2 with Se0/0
& se0/0 with Router 1 ip address
30.0.0.1 & Router 2 ip address 30.0.0.2
8) Switch 1 to Router 0 connected with
ip address 10.0.0.0 & switch 2
to Router 2 with ip address 40.0.0

9) Display the ip routes in routes 0
router > show ip route

10) Default route is set for router 0
router > enable
router# configure terminal
router(config)# ip route 30.0.0.1 255.0.0.0
20.0.0.1
router(config)# no shutdown

11) Default route is set for router 1
router(config)# ip route 10.0.0.1 255.0.0.0
20.0.0.1
12) router(config)# ip route 40.0.0.1 255.0.0.0
30.0.0.1

12) Default route is set for router 1
router(config)# ip route 10.0.0.1 255.0.0.0
20.0.0.1

PAGE EDGE
DATE : / /

Results

ip route before default route is set in Routes 0
Gateway of last resort not set
C 10.0.8.0/8 is directly connected to Fast ethernet Fa0/0
C 20.0.0.0/8 is directly Connected to ~~Fast Ethernet Fa 0/0~~ serial 2/0

After default gateway set
C 10.0.0.0/8 is directly connected with FA 0/0
C 20.0.0.0/8 is directly connected with serial 2/0

ip route of router 1

C 20.0.0.0/8 is directly connected serial 2/0
C 30.0.0.0/8 is directly connected serial 3/0

After statically setting ip route

C 20.0.0.0/8 is directly connected FA 0/0
C 30.0.0.0/8 is directly connected serial 2/0
S 40.0.0.0/8 [1/0] via 30.0.0.1
S 10.0.0.0/8 [1/0] via 10.0.0.3

Observation

static ip route can be configured on a router using CLI. show ip route output C - denote connected S - denotes static & S* denotes default gateway

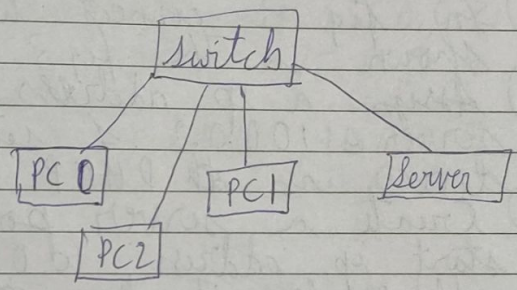
PAGE EDGE
DATE : 14/7/23

4 Configure DHCP within a LAN & outside LAN

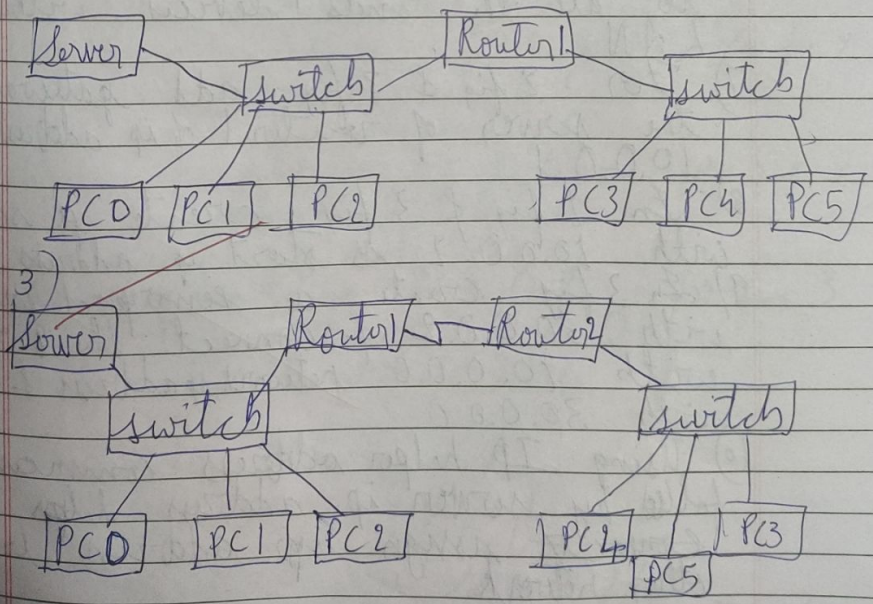
Aim :- To ~~configure~~ understand how to configure DHCP within LAN & outside LAN

Topology

i) within LAN



ii) outside LAN



Procedure

1) Connect the end devices to the switch and connect a server to the switch as show in diagram
2) Connect a router to a switch as shown in fig 2 & 3 & setup an another LAN network
3) In fig 3 connect two router as shown in the fig 3
4) Assign a ip address statically to server as 10.0.0.2 & set the end devices as DHCP
5) Create a server pool with the start ip address 10.0.0.3. for all 3 networks in fig
6) Automatically ip address is assigned to all the ends devices within the LAN
7) For 2 fig & 3 fig add gateway on the server of router 1 of ip address 10.0.0.1
8) In 2 fig & 3 fig create a serverpool with 10.0.0.2 as start ip address.
9) In 3 fig create a serverpool with 30.0.0.8 & Connect the 2 routers with 40.0.0.0 network ip address & LAN with 30.0.0.0
10) Using IP helper address command followed by server ip address helps to connect assign ip address to the other network

Procedure

1) Connect the end devices to the switch and connect a server to the switch as show in diagram
2) Connect a router to a switch as shown in fig 2 & 3 & setup an another LAN network
3) In fig 3 connect two router as shown in the fig 3
4) Assign a ip address statically to server as 10.0.0.2 & set the end devices as DHCP
5) Create a server pool with the start ip address 10.0.0.3. for all 3 networks in fig
6) Automatically ip address is assigned to all the ends devices within the LAN
7) For 2 fig & 3 fig add gateway on the server of router 1 of ip address 10.0.0.1
8) In 2 fig & 3 fig create a serverpool with 10.0.0.2 as start ip address.
9) In 3 fig create a serverpool with 30.0.0.8 & Connect the 2 routers with 40.0.0.0 network ip address & LAN with 30.0.0.0
10) Using IP helper address command followed by server ip address helps to connect assign ip address to the other network

PC3
ip address 20.0.0.3 for fig 2
subnet 255.0.0.0
ip address 30.0.0.3 for fig 3

PC4
ip address 20.0.0.4 for fig 2
subnet 255.0.0.0
ip address 30.0.0.4 for fig 3

PC5
ip address 20.0.0.5 for fig 2
subnet 255.0.0.0
ip address 30.0.0.5 for fig 3

Observation

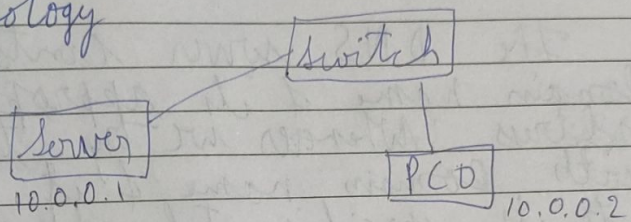
IP addresses for end devices in all the network are generated dynamically for the networks that doesn't have server. IP helper address is added which helps in generating IP address

PAGE EDGE
DATE: 21/7/23

5) Configure Web Server and RIP Routing protocol to routes

Aim :- Configure Web server, DNS within a LAN

Topology



Procedure

1) Connect end device to switch & connect a server to switch
2) Configure the server set ip address 10.0.0.1 & subnet mask 255.0.0.0
3) Turn on the DNS service in the server & give the domain name and address as ip address of server
4) Configure the end device set ip address 10.0.0.1 & subnet mask 255.0.0.0
5) Go to web browser in PC0 type domain name/ip address of the server
6) We can also change the index.html file in HTTP in services of server like as :- <h1> Sagar CMaranmavar </h1>
<h1> IBM21CS181 </h1>

Result

Sagar C Manharnavar
1BM21CS181

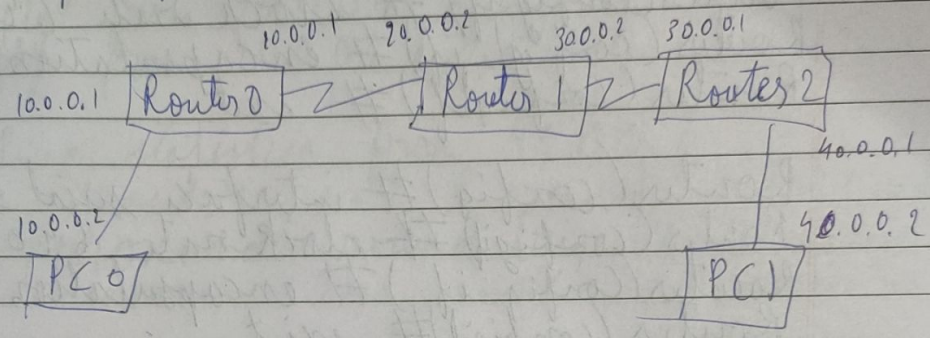
Observation

The DNS sever contains the Domain name & its appropriate IP address. Wherever we try to access with Domain name & it matched to its specific IP address by DNS sever. We can visit either by Domain name or by IP address assigned with it.

PAGE EDGE
DATE : / /

b) Aim :- Configuring RIP routing protocol

Topology



Procedure

1) Connect the Routers & end devices as shown above
2) Assign the IP addresses to the routers & the end devices as per the diagram & give the gateway to end device

Router 0 :-

Router(config)# interface serial 2/0
Router(config)# clock rate 64000
Router(config if)# encapsulation ppp
Router(config if)# exit

Router(config)# router rip
Router(config)# network 10.0.0.0
Router(config)# network 20.0.0.0

Router 1

Router(config)# interface serial 2/0
Router(config-if)# clock rate 64000
Router(config-if)# encapsulation ppp
Router(config-if)# exit

Router(config)# interface serial 3/0
Router(config-if)# clock rate 64000
Router(config-if)# encapsulation ppp
Router(config-if)# exit

Router(config)# router rip
Router(config-if)# network 20.0.0.0
Router(config)# network 30.0.0.0

Router 2

Router(config)# interface serial 2/0
Router(config-if)# clock rate 64000
Router(config-if)# encapsulation ppp
Router(config-if)# exit

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

Result
From PC0
> ping 40.0.0.2
ping 40.0.0.2 with 32 bytes of data

PAGE EDGE
DATE: / /

Reply from 40.0.0.2 bytes 32 Time 2ms TTL = 125
Reply from 40.0.0.2 bytes 32 time 4ms TTL=125
Reply from 40.0.0.2 bytes 32 time 4ms TTL = 125
Reply from 40.0.0.2 bytes 32 time 7ms TTL = 125

Ping statistics from 40.0.0.2
packets sent = 4 received = 4 lost = 0 (0% loss)
Approximate round trip time in ms
minimum 2ms max = 7ms Average 4ms

Observation
From the above experiment we come to know that RIP is a dynamic routing protocol that uses hop count as a routing metric to find the best path between the source the destination network

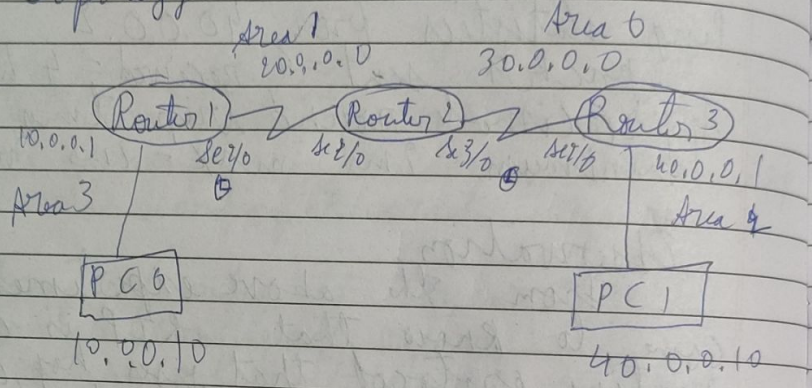
4/9/23
8

4-8-23

7 Configure OSPF routing Protocol

Aim :- To understand how to configure OSPF routing protocol

Topology



Procedure

1) Create a topology as shown above
2) Configure each Router with IP addresses for required interfaces & setup clock rate for interface having DCE symbol

Router 2

```
Router(config)# interface serial 3/0
Router(config-if)# ip address 30.0.0.1 255.0.0.0
Router(config-if)# encapsulation ppp
Router(config-if)# clock rate 64000
Router(config-if)# no shutdown

Router(config)# interface serial 2/0
Router(config-if)# ip address 20.0.0.2 255.0.0.0
Router(config-if)# encapsulation ppp
Router(config-if)# no shutdown
```

PAGE EDGE
DATE: / /

3) Configure ospf routing for each
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
R1(config-router)# no shutdown

4) There must be one interface up to keep ospf process up. So its better to configure loopback address to routers. It is a virtual interface that never goes down
R1(config)# interface se1/0
R1(config-if)# interface loopback 0
R1(config-if)# ip add 30.0.0.1 255.0.0.0
R1(config-if)# no shutdown

5) Virtual link must be created b/w R1 & R2 to connect area 0 & area 3
R1(config)# router ospf 1
R1(config-router)# area 1 virtual-link 2.2.2.2
R1(config-router)# exit
Same for R2

6) Ping PC0 to PC1 to check connectivity

Result

R3# show ip route
O IA 20.0.0.0/8 via 30.0.0.1 s1/0
O IA 10.0.0.0/8 via 30.0.0.1 se 1/0
C 40.0.0.0/8 directly connected Fa 0/0
C 30.0.0.0/8 directly connected se 2/0

R2# show ip route
C 20.0.0.0/8 directly connected se 2/0
C 30.0.0.0/8 directly connected se 3/0
C 172.16.1.0/16 directly connected, loop back 0

R1# show ip route
O IA 40.0.0.0/8 via 20.0.0.2, se 2/0
O IA 30.0.0.0/8 via 20.0.0.2 se 2/0
C 10.0.0.0/8 via directly connected Fa 0/0
C 20.0.0.0/8 directly connected se 2/0

Ping PC0 from PC1
PC1> ping 10.0.0.10
Reply from 10.0.0.10 bytes =32 time 5 ms TTL=125
Reply from 10.0.0.10 bytes =32 time 7 ms TTL=125
Reply from 10.0.0.10 bytes =32 time 4 ms TTL=125
Reply from 10.0.0.10 bytes =32 time 3 ms TTL=125

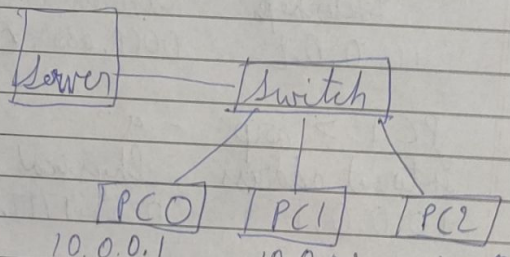
Ping statistics
Packets sent =4 received =4 lost=0
min =3 ms max =7 ms average =4 ms

PAGE EDGE
DATE : 11/8/23

8 To construct simple LAN & understand the concept & operation of Address Resolution Protocol (ARP)

Aim :- To understand the concept & operation of ARP

Topology



Procedure

1 Create topology as above shown
2 Configure IP address for each end devices
3 Use the "inspect tool" to see ARP table & MAC address table by
PC > arp -a
PC > show mac address-table
4 Go to simulation mode to send packets b/w end devices & observe the changes in ARP table, as & when new connection starts

Result

sending packets from PC0 to PC1

before sending packets

PC0> arp -a

No ARP entries found

After sending

PC0> arp -a

Internet address	Physical address	Type
10.0.0.2	00CD.0364.Ca2a	dynamic

PC1> arp -a

Internet address	Physical address	Type
10.0.0.1	0060.4797.e769	dynamic

Mac Address Table

Switch> show mac address-table

VLAN	MAC Address	Type	Ports
1	0060.4797.e769	dynamic	Fa0/0
1	00CD.0364.Ca2a	dynamic	Fa0/0

Observation

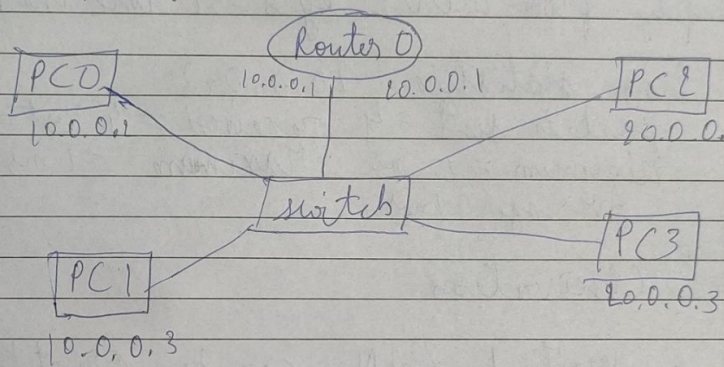
Initially the ARP table is empty after the last encapsulates a packet into a frame, if it is reflected in the MAC address table to determine the mapping of IP address to MAC address

PAGE EDGE
DATE: 11/8/23

9 To construct a VLAN & make PC's communicate among VLAN

Aim : Understand how to construct VLAN

Topology



Procedure

1) Create a topology as shown above
2) Assign IP address to the end devices
3) Add two VLAN Number & VLAN Name to the port connected to Router0 from switch like name V1, V13 & number 2 & 3 respectively
4) Configure the router by CLI commands

~~Router # int fa 0/0.1~~
Router (config)# int fa 0/0.1
Router (config-subif)# encapsulation dot1q 2
Router (config-subif)# ip add 10.0.0.1 255.0.0.0
Router (config-subif)# int fa 0/0.2
Router (config-subif)# ip add 20.0.0.1 255.0.0.0
Router (config-subif)# encapsulation dot1q 3
Router (config-subif)# no shutdown

Result

Pinging 20.0.0.2 to 20.0.0.2

PC0 > ping 20.0.0.2

Reply from 20.0.0.2 bytes=32 time=9ms TTL=127
Reply from 20.0.0.2 bytes=32 time=0ms TTL=127
Reply from 20.0.0.2 bytes=32 time=0ms TTL=127
Reply from 20.0.0.2 bytes=32 time=0ms TTL=127

Ping statistics 20.0.0.2
packets sent = 4 received = 4 lost = 0
Minimum = 0 ms Maximum = 9 ms
Average 1ms

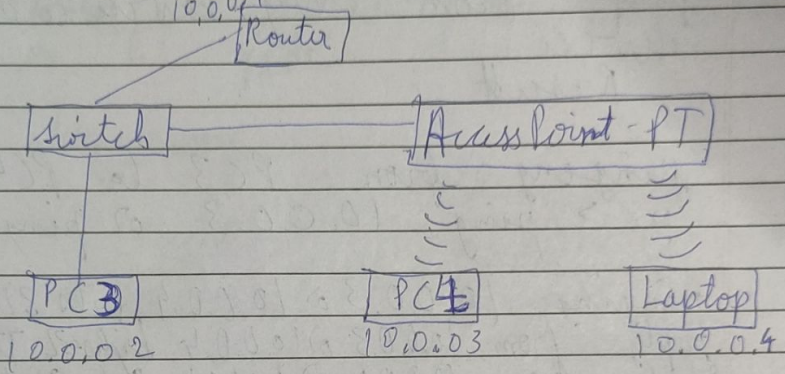
Observation

A virtual LAN can be created by specifying the VLAN no & VLAN name in the switch & the same should be added to the routes to make the routes identify the newly created virtual LAN. The physical & virtual LAN can communicate b/w each other even through both have different gateway

PAGE
DATE : / /

10 To construct a WLAN and make the nodes communicate

Aim :- To construct a WLAN and make the nodes communicate wirelessly



Procedure

* Construct the above topology
* Configure PC3 & Router 1 as is normally done.
* Configure access point 1 - port 1 -> SSID range - any name (WLAN here) select WEP and give any 10 digit key 0123456789 here
* Configure PC4 and laptop with wireless standards. switch off the devices. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS to the component listed in the LHS. Drag WMP300N wireless interface to the empty port switch on the device

* In the config tab a new wireless interface would have been added. Now configure interface would have been added. Now configure SSID, WEP, WEP key, IP address and gateway to device
ping from every device to every other device

Result

pinging from PC3 to PC4 Laptop
PC> ping 10.0.0.3 or ping 10.0.0.4

pinging 10.0.0.3 or 10.0.0.4 with 32 bytes of data
reply from 10.0.0.3 or 10.0.0.4 bytes=32 time 3ms TTL=
reply from 10.0.0.3 or 10.0.0.4 bytes=32 time 4ms TTL=
time 3ms
time 4ms

ping statistics
packets sent = 4 received = 4 lost = 0
Approximate round trip time in ms
min = 3 ms max = 4ms Average 3.5ms

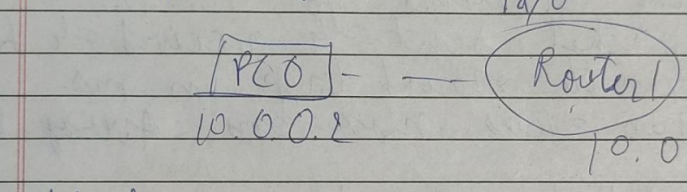
Observation
WLAN gives user the ability to move around within the area remains connected to the network through the gateway, a WLAN can also provide a connections to wider internet

PAGE EDGE
DATE : / /

To understand the operation of TELNET by accessing the router in server room from a PC in IT Office.

Aim :- To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Topology



Working

1) Construct the above topology
2) Configure router with commands
ip add 10.0.0.1 255.0.0.0
no shut, Router(config)# enable secret p1
Router(config-if)# line vty 0 5
Router(config-line)# login
% login disabled on line 132, until password is set
% login disabled on line 133, until password is set
% login disabled on line 134, until password is set
% login disabled on line 135 until password is set
% login disabled on line 136 until password is set
Router(config-line)# password p1 exit
Router(config)# exit
Router# wr
Building configuration

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=0ms TTL=255
reply from 10.0.0.1 bytes=32 time=0ms TTL=255
reply from 10.0.0.1 bytes=32 time=0ms TTL=255
reply from 10.0.0.1 bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.1
packet: sent=4 received=4 lost=0
Approximate round trip in ms
min=0ms max=0ms Avrg 0ms

PC> telnet 10.0.0.1
Trying 10.0.0.1 ... open
User Access verification
Password: p0
r1> enable
Password: p1
r1# show ip route
10.0.0.0/0 is directly connected,
FastEthernet 0/0

PAGE EDGE
DATE : / /

Cycle -- 2

Write a program for congestion control using leaky bucket algorithm

```
#include <stdio.h>
#include <stdlib.h>
#define capacity 10
#define outputrate 3

void main() {
  int bucketcap=0, rate=3;

  while (1) {
    int pkts;
    printf("Enter no of packets ");
    scanf("%d", &pkts);
    if (bucketcap + pkts =< capacity) {
        bucketcap += pkts;
    printf("%d packets added to bucket");
    } else {
    int discard = bucketcap + pkt - capacity;
    printf("%d packets discared", discard);
    }
    if (bucketcap >0) {
      if (bucketcap >= rate) {
          printf("%d packets transmitted\n",
                rate);
          bucketcap -= rate;
      } else {
      printf("%d packet transmitted
        bucketcap =0;
      }
  }
```

```
        else {
            printf("Bucket is empty");
        }
    }
}
```

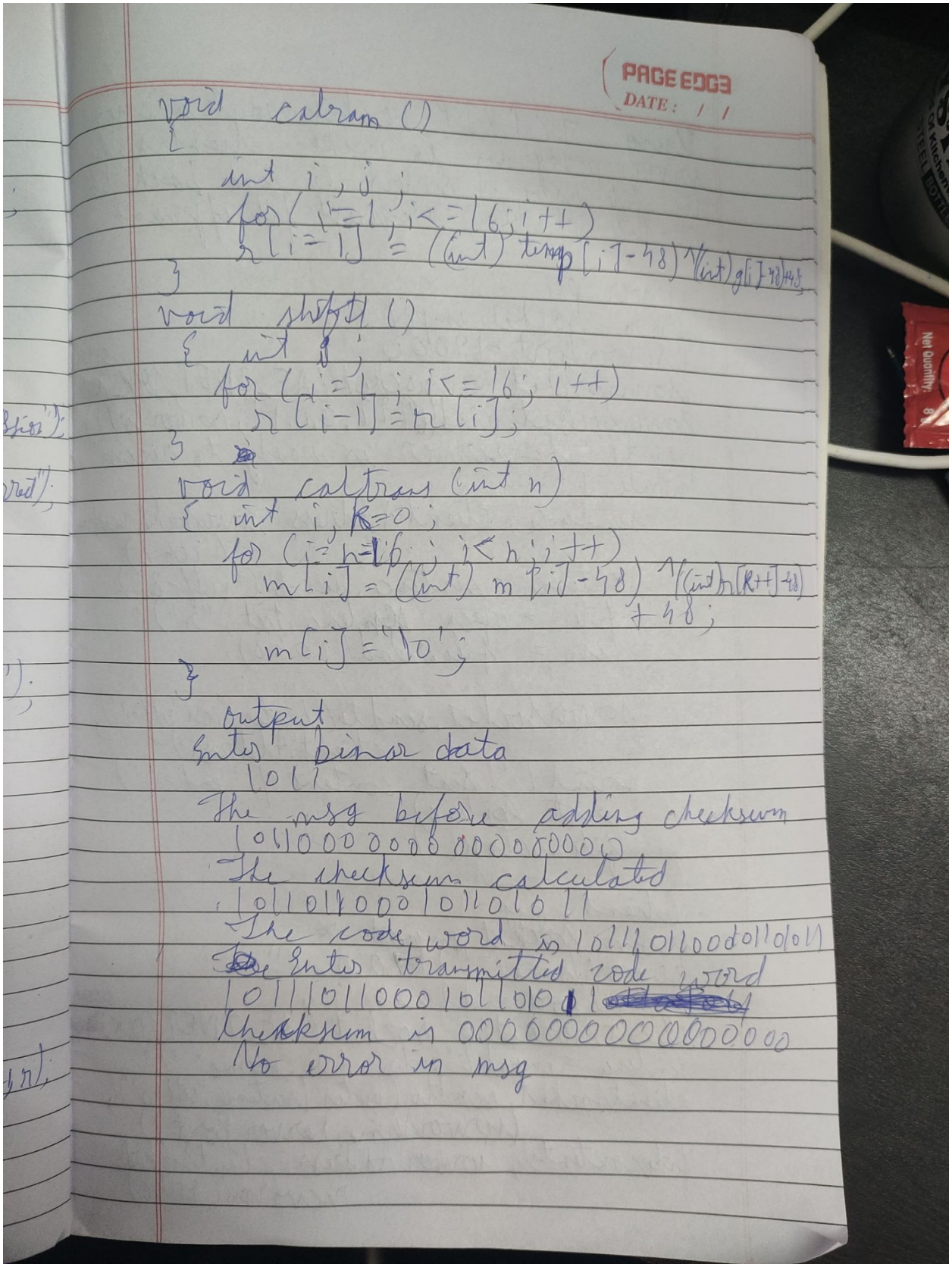
Output

```
Enter no of packets 3
3 packet added into bucket
3 packets transmitted
Enter no of packets 5
5 packet added into bucket
3 packets transmitted
Enter no of packets 11
1 packet discarded
3 packets transmitted
Enter no of packets 0
0 packet added into bucket
3 packets transmitted
Enter no of packets 0
0 packet added
3 packets transmitted
Enter no of packets 0
0 packet added
1 packet transmitted
```

Write a programs for error detecting code using CRC-CCITT (16 bits)

```
#include<stdio.h>
char m[50],g[50],r[50],q[50],temp[50];
void caltrans(int);
void crc(int);
void calcrc();
void shiftl();
void main()
{
    int n,i=0
    char ch,flag=0;
    printf("Enter the frame bits:");
    //getch(m);
    while((ch=getche(stdin))!='\n'){
        m[i++]=ch;
    } n=i;}
    for(i=0;i<16;i++){
        m[i++]='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';
        g[17]='\0';
    printf("generator %s\n",g);
    crc(n)
    printf("\n quotient %s",q);
    caltrans(n);
    printf("transmitted frame %s");
    printf("enter transmitted frame");
    scanf("%d",&m);
```

```
printf("\n CRC checking \n");
crc(n);
printf("\n Last remainder %s", r);
for(i=0; i<16; i++)
    if(r[i] != '0')
        flag = 1;
    else
        continue;
if(flag == 1)
printf("\n Error during transmission");
else
    printf("\n Received frame is correct");
}
void crc(int n) {
int i, j;
for(i=0; i<n; i++)
    r[i] = m[i];
printf("\n Intermediate remainder \n");
for(i=0; i<n-16; i++)
{
    if(r[0] == '1')
    { q[i] = '1';
    calcrem();
    } else {
        q[i] = '0';
        shiftl();
    }
    r[16] = m[i+17];
    r[17] = '\0';
    printf("\n remainder %d : %s", i+1, r);
    for(j=0; j<=17; j++)
        temp[j] = r[j];
}
q[n-16] = '\0';
```



```
void caltrans ()
{
    int i, j;
    for (i=1; i<=16; i++)
      r[i-1] = ((int) temp[i]-48) ^ ((int) g[i]-48)+48;
}
void shift ()
{ int i;
  for (i=1; i<=16; i++)
    r[i-1] = r[i];
}
void caltrans (int n)
{ int i, k=0;
  for (i=n-16; i<n; i++)
    m[i] = ((int) m[i]-48) ^ ((int) r[k++]-48)
           +48;
  m[i] = '\0';
}
```

output

Enter binar data
1011

The msg before adding checksum
1011000000000000

The checksum calculated
1011010001011010

The code word is 1011101100001101011

Enter trasmitted code word
1011101100010110101

checksum is 0000000000000000

No error in msg

Using TCP/IP sockets write a client server program to make client sending the file name & server to send back the content of the request file if present

server.py

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

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

Client.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
sentence = input("Enter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),
                    (serverName, serverPort))
fileContents, serverAddress = clientSocket.
                    recvfrom(2048)
```

clientSocket.close()

Sentence.txt
Hi, this is sagar

Output
server.py
The server is ready to receive

Client.py
Enter file name : sentence.txt
reply from server:
Hi, this is sagar

Using UDP socket, write a client server program to make client sending the file name & server sends back content of file if present

server.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.
                              recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
    serverSocket.sendto(bytes(con, "utf-8"),
                        clientAddress)
    print("sent content of ", end = '')
    file.close();
```

client.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.
                              recvfrom(2048)
print(fileContents.decode("utf-8"))
```

client_socket.close()
client_socket.close()

sentence.txt
Hi, this is sagar (M
output
Server.py
The Server is ready to receive
the content of sentence.txt
Client.py
Enter file name : sentence.py
reply from server
Hi, this is sagar (M