

Configure DHCP within a LAN and outside the LAN

Server:

D) DHCP

Desktop IP Conf  
Static

192.168.10.2

Gateway 192.168.10.1

Servers

DHCP	ON <input checked="" type="checkbox"/>	ON <input checked="" type="checkbox"/>
PoolName:	Switch1	Switch2
Gateway:	192.168.10.1	192.168.20.1
Start IP address:	192.168.10.3	192.168.20.2
Subnet Mask:	255.255.255.0	255.255.255.0

Router:

C21

no

Router# enable

# conf t

# int Fa 0/0

# ip address 192.168.10.1 255.255.255.0

# ip helper-address 192.168.10.2

# no shutdown  $\rightarrow$  also called as only veget  
do write memory

exit

# int Fa 0/1

# ip address 192.168.20.1 255.255.255.0

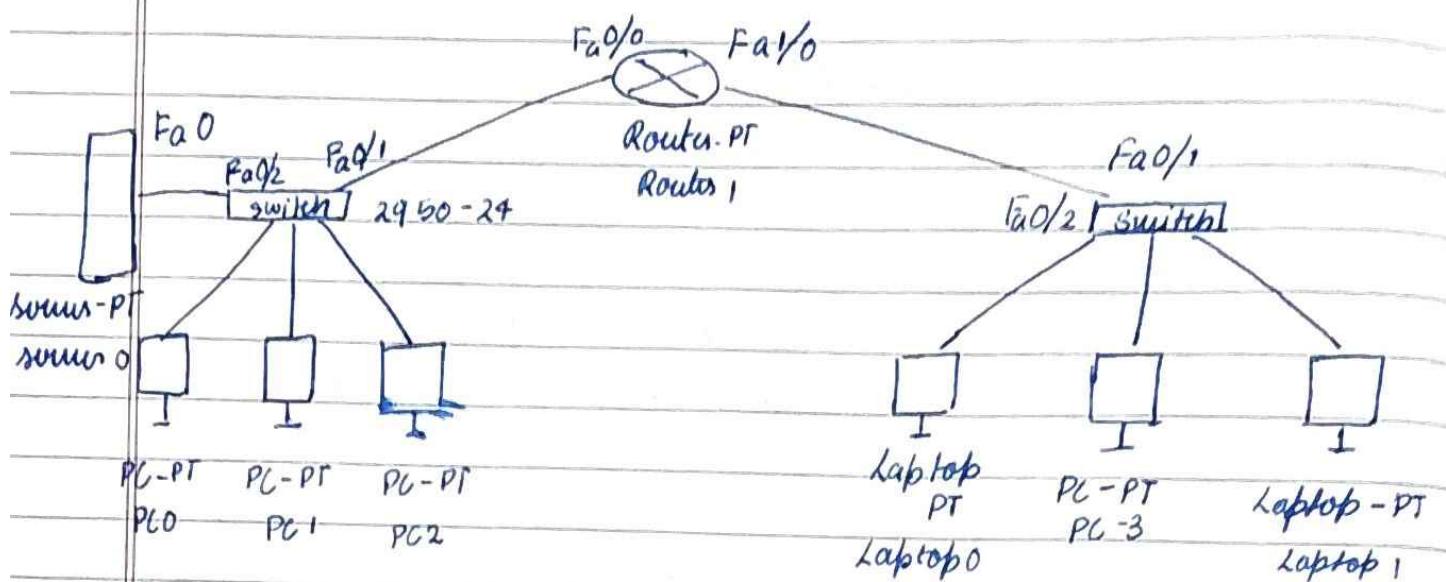
# ip address - address 192.168.10.2

# no shutdown

exit

exit

write memory



Q. Configure IP addresses to router in Packet Traces:

Explains the following message:

- (1) Ping response
- (2) Destination unreachable
- (3) Request timeout
- (4) Reply

(1) & (4) ping response and reply

pc > ping 192.168.1.5

Pinging 192.168.1.5 with 32 bytes of data.

Reply from 192.168.1.5 : bytes = 32 time = 0ms TTL = 128

Reply from 192.168.1.5 : bytes = 32 time = 0ms TTL = 128

Reply from 192.168.1.5 : bytes = 32 time = 0ms TTL = 128

Reply from 192.168.1.5 : bytes = 32 time = 0ms TTL = 128

ping statistics for 192.168.1.5

Packets: sent = 4, Received = 4 Lost = 0 (0% loss),  
Approximate round trip times in milliseconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms.

10

Configure web servers, DNS within a LAN:

Steps:

(1) Do IP Config for all PC

IPV4 : 192.168.1.100

192.168.1.101

Subnet mask: 255.255.255.0

255.255.255.0

Default Gateway : 192.168.1.1

192.168.1.1

DNS Servers : 192.168.1.5

192.168.1.5

PC 1

PC 2

(2) Click on web servers

Select services

HTTP

Click on HTTP → Turn on  OFF

Select any message click & change

Ex: <html>

welcome to CN Lab

<br> <a href = "index.html" > Back </a>  
</html>

Save

(3) Desktop

IP Config → IPV4 : 192.168.1.6

Subnet mask: 255.255.255.0

Gateway : 192.168.1.1

DNS Servers: 192.168.1.5

(4) Click on DNS Servers:

Select services → DNS →  ON

Name: www.letslearn.com

Type: A Record → Record belong IPV4

Address: 192.168.1.6

DNS Server:

Desktop  $\rightarrow$  IPV4 : 192.168.1.5

Subnet Mask : 255.255.255.0

Default gateway : 192.168.1.1

DNS Server : 192.168.1.5

To check connectivity:

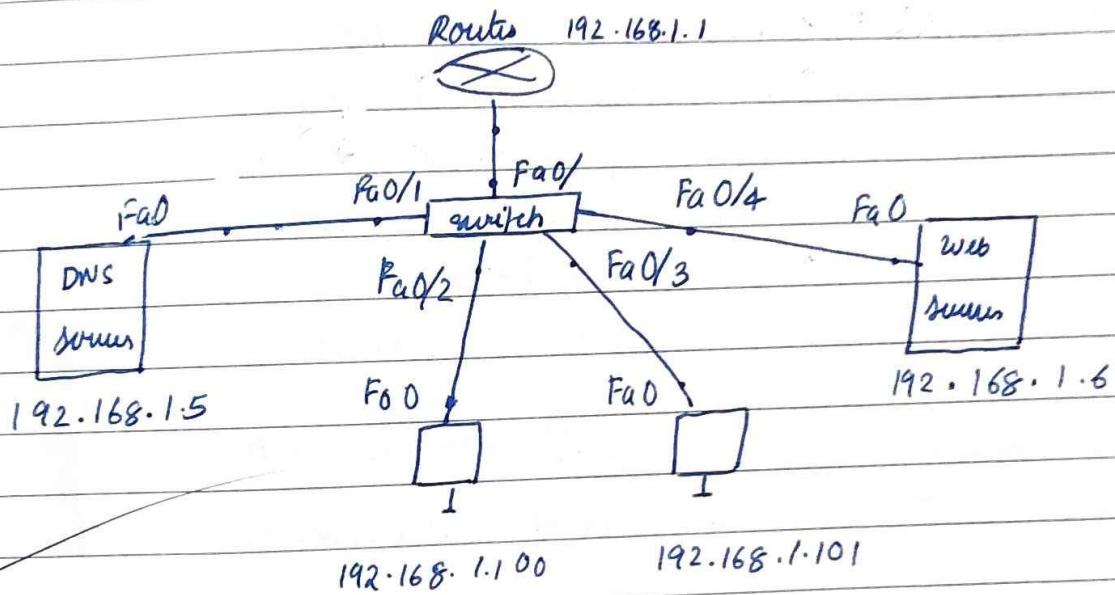
PC 1  $\rightarrow$  Go to command prompt

ping 192.168.1.5 (DNS Server)

ping 192.168.1.6 (Web Server)

Go to web browser

URL http://www.letslearn.com



## Static and Dynamic:

### STATIC

Routing :

Static  $\rightarrow$  unknown

Dynamic  $\rightarrow$  known

For Router 0:

- (i) config  $\rightarrow$  Fast Ethernet 0/0  ON  
 ↳ IP address : 192.168.1.1
- $\rightarrow$  Fast Ethernet 1/0  ON  
 ↳ IP address : 192.168.2.1
- $\rightarrow$  Serial 2/0  ON  
 ↳ IP address : 192.168.4.1

For Router 1:

- (ii) config  $\rightarrow$  Fast Ethernet 0/0  ON  
 ↳ IP address : 192.168.3.1
- $\rightarrow$  Serial 2/0  ON  
 ↳ IP address : 192.168.4.2
- $\rightarrow$  Serial 3/0  ON  
 ↳ IP address : 192.168.7.1

Router R0 (1.0, 2.0, 4.0) known network

$\hookrightarrow$  unknown network : 192.168.3.0

To reach 192.168.3.0

$\hookrightarrow$  1<sup>st</sup> hop : 192.168.4.2

Router R1 (3.0, 4.0)

$\hookrightarrow$  unknown network (2.0, 1.0)

To reach (192.168.2.0, 192.168.1.0)

$\hookrightarrow$  1<sup>st</sup> hop : 192.168.4.1

Router 0 → config → Routing

static Network : 192.168.3.0

Next hop : 192.168.4.2

Router 1 → config → Routing

static Network : 192.168.1.0

Next hop : 192.168.4.1

Network : 192.168.2.0

Next hop : 192.168.4.1

Network : 192.168.3.0

Next hop : 192.168.7.2

Network : 192.168.6.0

Next hop : 192.168.7.2

Router 0 → config → Routing → static

Network : 192.168.3.0

Next hop : 192.168.4.2

Network : 192.168.5.0

Next hop : 192.168.4.2

Network : 192.168.6.0

Next hop : 192.168.4.2

Router 2 :

config → Fast Ethernet 0/0  ON

↳ IP address : 192.168.5.1

→ Fast Ethernet 1/0  ON

↳ IP address : 192.168.6.1

→ Serial 2/0  ON

↳ IP address : 192.168.7.2

Config  $\rightarrow$  Routing  $\rightarrow$  static

Network : 192.168.2.0

Net hop : 192.168.7.1

Network : 192.168.1.0

Net hop : 192.168.7.1

Network : 192.168.3.0

Net hop : 192.168.7.1

Changes only his part for all the Router

DYNAMIC

address only "known network"

I For Router 0 :

Config  $\rightarrow$  Routing  $\rightarrow$  RIP

$\hookrightarrow$  Network : Add 192.168.1.0  
Add 192.168.2.0  
Add 192.168.4.0

II For Router 1 :

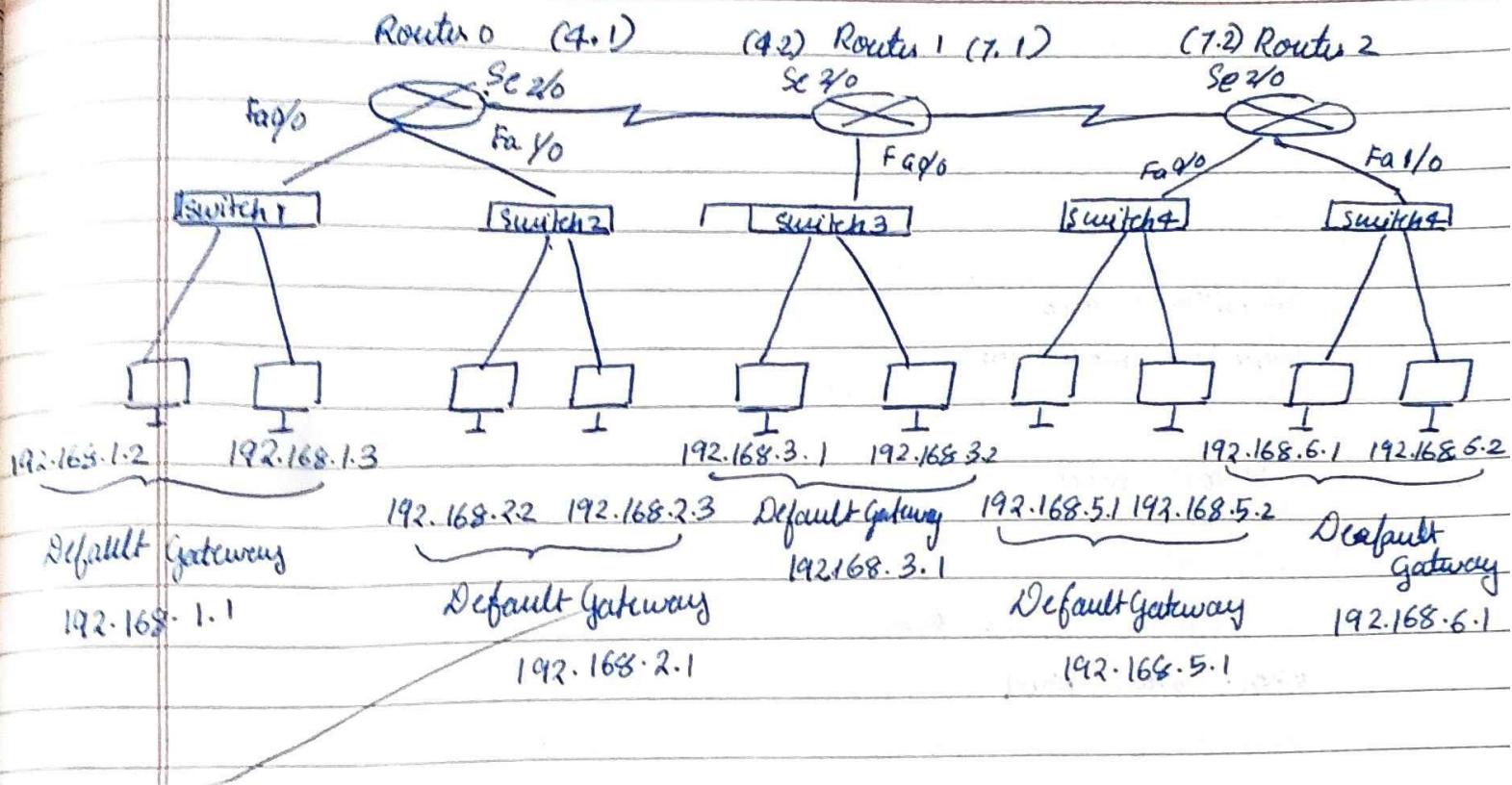
Config  $\rightarrow$  Routing  $\rightarrow$  RIP

$\hookrightarrow$  Network : Add 192.168.3.0  
Add 192.168.4.0  
Add 192.168.7.0

III For Router 2 :

Config  $\rightarrow$  Routing  $\rightarrow$  RIP

$\hookrightarrow$  Network : Add 192.168.5.0  
Add 192.168.6.0  
Add 192.168.7.0



## Virtual LAN:

1. int fa 0/1

switchport access Vlan 10

switchport mode access

int fa 0/2

switchport mode access

switchport access Vlan 20

int fa 0/3

switchport mode access

switchport access Vlan 30

int fa 0/4

switchport mode trunk

show vlan brief

Vlan 10

Vlan 20

Vlan 30

fa0/1

PC0

fa0/2 PC1

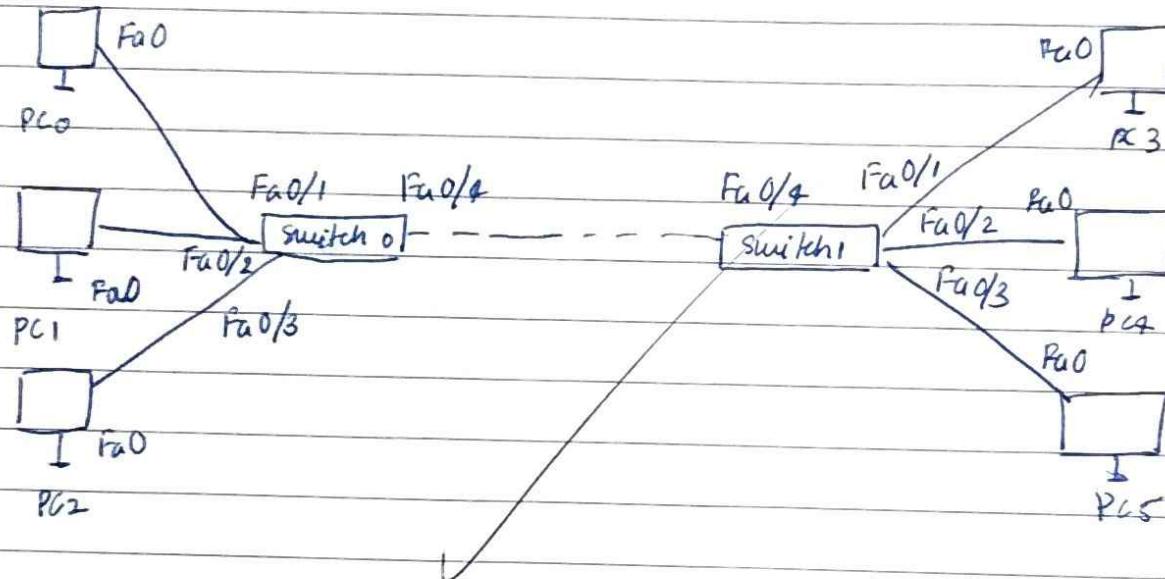
fa0/3 PC2

fa0/1

PC3

fa0/2 PC4

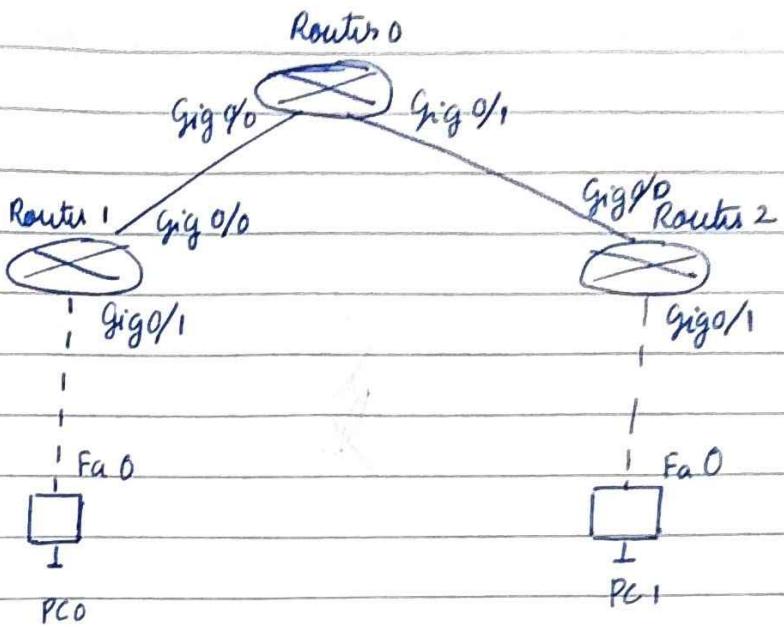
fa0/3 PC5

command line  
enableconfig +  
int fa 0/1

switchport access Vlan 10

[it should bring only system  
on Vlan ]

config OSPF (Open shortest path first) Routing Protocol.



192.168.94.2

10.10.22.2

Default gateway: 192.168.94.1

Default gateway : 10.10.22.1

Router 1

>enable

# config t

# router ospf 1

# network 192.168.55.0 0.0.0.255 area 0

# network 192.168.94.0 0.0.0.255 area 0

# exit

Router 0

>enable

# config t

# router ospf 1

# network 192.168.55.0 0.0.0.255 area 0

# network 172.16.0.0 0.0.255.255 area 0

# exit

Router 2 :

# network 172.16.0.0 0.0.255.255  
area 0

# network 10.0.0.0 0.255.255.255  
area 0

area 0

Checking:

PC0:

↳ command prompt

ping 10.10.22.2

PC1:

↳ command prompt

ping 192.168.44.2

~~xx~~

CRC is most power and easy to implement.

CRC is based on binary division.

Calculate the remainder if remainder is zero in destination, then data is error free or else data has error.

a. Frame: 11 001  
data [F]

Polynomial Generator ( $G_1$ ): 101

→ Compute the CRC at the under side:

	111
101	11 0 0 1
	10 1
	<u>0 1 1 0</u>
	1 0 1
	<u>0 1 1 1</u>
	1 0 1
	<u>0 1 0</u>
	⇒ CRC = 10 → 2

~~F + CRC = 1100110 ⇒ must be transmitted across the network~~

→ Compute the CRC at medium side:

	1111
101	11 0 0 1 1 0
	10 1
	<u>0 1 1 0</u>
	1 0 1
	<u>0 1 1 1</u>
	1 0 1
	<u>0 1 0 1</u>
	1 0 1
	<u>0 0 0 0</u>

as remainder is zero, then the data sent is correct.

Q. Frame: 100100

$$\text{Polynomial} : x^2 + x + 1 = 1101$$

Sender side:

	1111 01
1101	100100000
	1101
	01000
	1101
	01010
	1101
	01110
	1101
	00110
	00000
	01100
	1101 → CRC → 1

$$F + CRC = 1001001$$

Receiver side:

	1111
1101	1001001
	1101
	01000
	1101
	01010
	1101
	01111
	1101

0010 → CRC is not zero, thus data is not correct.

## Program:

```

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

int main()
{
    char rem[50], a[50], s[50], c, msg[50], gen[30];
    int i, genlen, t, j, flag = 0, K, n;
    printf("enter the generator \n");
    gets(gen);
    printf("generator polynomial is crs: CCITT : %s \n", gen);
    genlen = strlen(gen);
    K = genlen - 1;
    printf("enter the message \n");
    n = 0;
    while ((c = getchar()) != '\n')
    {
        msg[n] = c;
        n++;
    }
    msg[n] = '\0';
    for (i = 0; i < n; i++)
        a[i] = msg[i];
    for (i = 0; i < K; i++)
        a[n + i] = '0';
    printf("In message polynomial append with zero \n");
    puts(a);
    for (i = 0; i < n; i++)
    {
        if (a[i] == gen[i])
            a[i] = 0;
        else
    }
}

```

```
a[t] = '1';
```

```
t++;
```

```
{
```

```
{
```

```
{
```

```
for (i=0; i<k; i++)
```

```
    sum[i] = a[n+i];
```

```
sum[k] = '0';
```

```
printf ("the checksum appended\n");
```

```
puts (sum);
```

```
printf ("\n the message with checksum appended\n");
```

```
for (i=0; i<n; i++)
```

```
    a[i] = msg[i];
```

```
for (i=0; i<k; i++)
```

```
    a[n+i] = sum[i];
```

```
a[n+k] = '0';
```

```
puts (a);
```

```
n=0;
```

```
printf ("enter the received message\n");
```

```
while ((c = getchar ()) != '\n')
```

```
{
```

```
s[n] = c;
```

```
n++;
```

```
{
```

```
s[n] = '0';
```

```
for (i=0; i<n; i++)
```

```
if (s[i] == '1')
```

```
{
```

```
t = i;
```

```
for (j=0; j <= k; j++, t++)
```

```
{
```

```
if (s[t] == gen[j])
```

```
    s[t] = '0';
```

```
else
```

```
    s[t] = '1';
```

```
}
```

```
}
```

```
}
```

```
for (i=0; i < k; i++)
```

```
{
```

```
    if (num[i] == '1')
```

```
        flag = 1;
```

```
}
```

```
if (flag == 0)
```

```
printf ("received polynomial is mworfe in");
```

```
else
```

```
printf ("received polynomial is error in");
```

```
return 0;
```

```
y
```

## PROGRAM : client.c

```
# include <stdio.h>
# include <sys/types.h>
# include <sys/socket.h>
# include <netinet/in.h>
# include <netdb.h>
# include <string.h>

int main (int argc, char* argv[])
{
    int sockfd, newsockfd, portno, len, n;
    char buffer [256], c[20000];
    struct sockaddr_in serv, cli;
    FILE *fd;
    if (argc < 2)
    {
        printf ("Usage: %s portno\n", argv[0]);
        exit(1);
    }
    sockfd = socket (AF_INET, SOCK_STREAM, 0);
    bzero ((char*) & serv, sizeof(serv));
    portno = atoi(argv[1]);
    serv.sin_family = AF_INET;
    serv.sin_port = htons (portno);
    if (connect (sockfd, (struct sockaddr *) & serv, sizeof(serv)) < 0)
    {
        printf ("Server not responding...\n");
        exit(1);
    }
```

```

printf ("Enter the file with complete path \n");
scanf ("%s", &buffer);
if (write (sockfd, buffer, strlen (buffer)) < 0)
printf ("Error writing to socket ..\n");
bzero (c, 2000);
printf ("Receiving ..\n..");
if (read (sockfd, c, 1999) < 0)
printf ("error : read error\n");
printf ("client : display content of %s \n.. \n", buffer);
fflush (c, stdout);
printf ("\n.. \n");
return 0;
}

```

## PROGRAM : Server.c

```

#include <stdio.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <netdb.h>
#include <string.h>
#include <unistd.h>
int main (int argc, char *argv[])
{
    int sockfd, newsockfd, portno, len, n;
    char buffer [256], c [2000], cc [2000];
    struct sockaddr_in serv, cli;
    FILE *fd;
    if (argc < 2)
        printf ("error : no port no in usage : %s /server port no\n");
    exit (1);
}

```

sockfd = socket (AF\_INET, SOCK\_STREAM, 0);

portno = atoi (argv[1]);

serv.sin\_family = AF\_INET;

serv.sin\_addr.s\_addr = INADDR\_ANY;

serv.sin\_port = htons (portno);

bind (sockfd, (struct sockaddr \*) & serv, sizeof (serv));

listen (sockfd, 10);

len = sizeof (cli);

printf ("server: In waiting for connection in ");

newsockfd = accept (sockfd, (struct sockaddr \*) & cli, & len);  
bzero (buffer, 255);

n = read (newsockfd, buffer, 255);

printf ("In server received : %s in ", buffer);

if ((fd = fopen (buffer, "r")) != NULL)

{

printf ("server : %s found in opening and reading .. in ", buffer);

printf (" reading .. in .. reading complete ");

fgets (cc, 2000, fd);

while (!feof (fd))

{

fgets (cc, 2000, fd);

strcat (cc, c);

}

n = write (newsockfd, cc, strlen (cc));

if (n < 0)

printf ("server writing to socket ");

printf ("In transfer complete in ");

}

else

{

printf ("server : file not found \n");  
n = write (newsockfd, "file not found", 15);  
if (n < 0)

printf ("error : writing to socket .. \n");  
y

return 0;

3

PROGRAM:

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

```
int min (int n, int y)
```

{

```
    if (n < y)
```

```
        return n;
```

```
    else
```

```
        return y;
```

{

```
int main()
```

{

```
    int drop = 0, mini, nsec, cap, count = 0, i, n[10], process;
```

```
    printf("Enter the bucket size : ");
```

```
    scanf("%d", &cap);
```

```
    printf("Enter the processing rate : ");
```

```
    scanf("%d", &process);
```

```
    printf("Enter the number of seconds you want to simulate : ");
```

```
    scanf("%d", &nsec);
```

```
    for (i=0; i < nsec; i++)
```

{

```
        printf("Enter the size of the packet entering at %d sec : ",
```

```
        i+1);
```

```
        scanf("%d", &nmp[i]);
```

{

```
        printf("In Second %d Packet received | Packet sent | Packet left |
```

```
        Dropped | ");
```

```
        printf("-----\n");
```

```
        for (i=0; i < nsec; i++)
```

{

count += wip[i];

if (count &gt; cap)

{

drop = count - cap;

count = cap;

{

printf ("%d", i+1);

printf ("%d", wip[i]);

mini = min(count, process);

printf ("%d %d", mini);

count = count - mini;

printf ("%d", count);

printf ("%d\n", drop);

drop = 0;

{

for (; count != 0; i++)

{

if (count &gt; cap)

{ drop = count - cap;

count = cap;

{

printf ("%d", i+1);

printf ("%d");

mini = min(count, process);

printf ("%d", mini);

count = count - mini;

printf ("%d", count);

printf ("%d\n", drop);

}

Wireless LAN:

Wireless Devices

↳ WRT 300N

↳ Config

↳ Wireless

↳ SSID : BMSE

Authentication : WPA2 - PSK Phan:bmse123

End Devices

↳ Smart devices

↳ config

↳ Wireless

↳ SSID : BMSE

Authentication : WPA2 - PSK Phan:bmsee123

↳ Generic [ PC-PT ]  
PC0

↳ config

↳ Wireless

↳ SSID: BMSE

Authentication: WPA2 - PSK Phan:bmse0.123

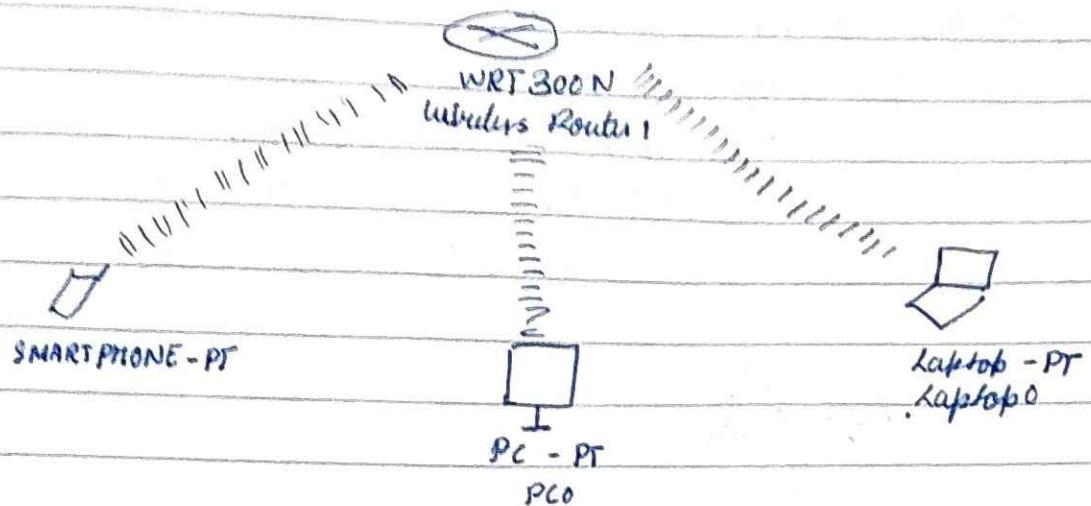
↳ Generic [ Laptop-PT ]  
Laptop0

↳ config

↳ Wireless

↳ SSID : BMSE

Authentication: WPA2 - PSK Phan:bmsee123



### Router (Generic)

↳ Config

↳ Fast Ethernet 0/0

↳ On

IP Address : 192.168.1.1

Subnet mask : 255.255.255.0

### Switch

### Wireless devices

↳ Access Point - PT (generic)

### End Devices

↳ PC - PT  
PC1

↳ Config Desktop

↳ IP Configuration [STATIC]

↳ IP Address : 192.168.1.2

Subnet mask : 255.255.255.0

↳ Laptop - PT  
Laptop1

↳ Desktop

↳ IP Configuration [STATIC]

↳ IP Address : 192.168.1.3 Subnet Mask : 255.255.255.0

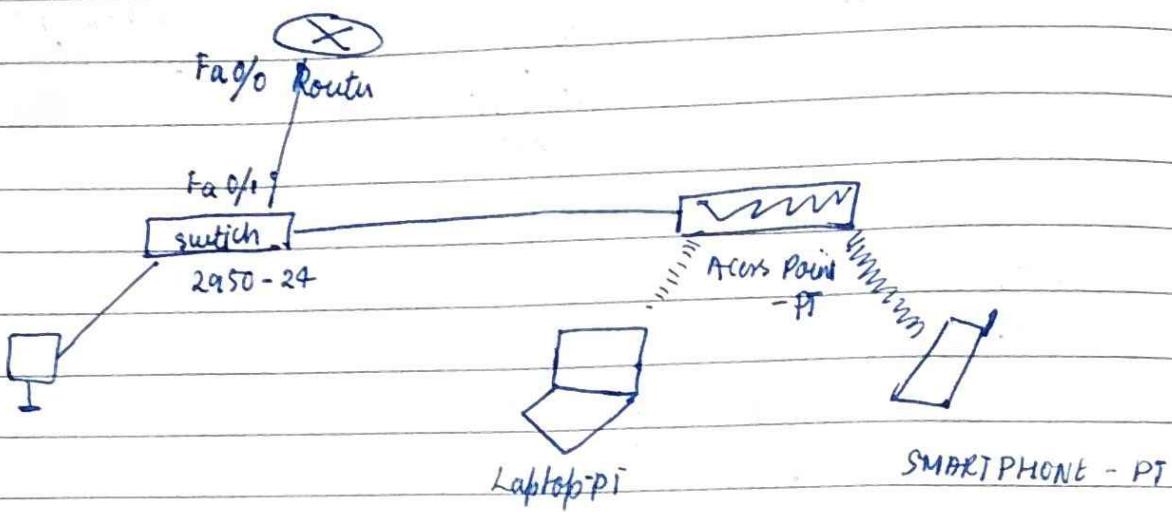
↳ Smart Device

↳ Desktop

↳ IP Configuration [STATIC]

↳ IP Address : 192.168.1.4

Subnet Mask : 255.255.255.0



Router

↳ Config

↳ Fast Ethernet 0/0

↳ ON

IP Address : 192.168.1.1

Subnet mask : 255.255.255.0

switch

End Device

↳ PC - PT

PC2

↳ Desktop

↳ IP Configuration [STATIC]

↳ IP Address : 192.168.1.2

Subnet Mask : 255.255.255.0

Default Gateway : 192.168.1.1

↳ PC - PT

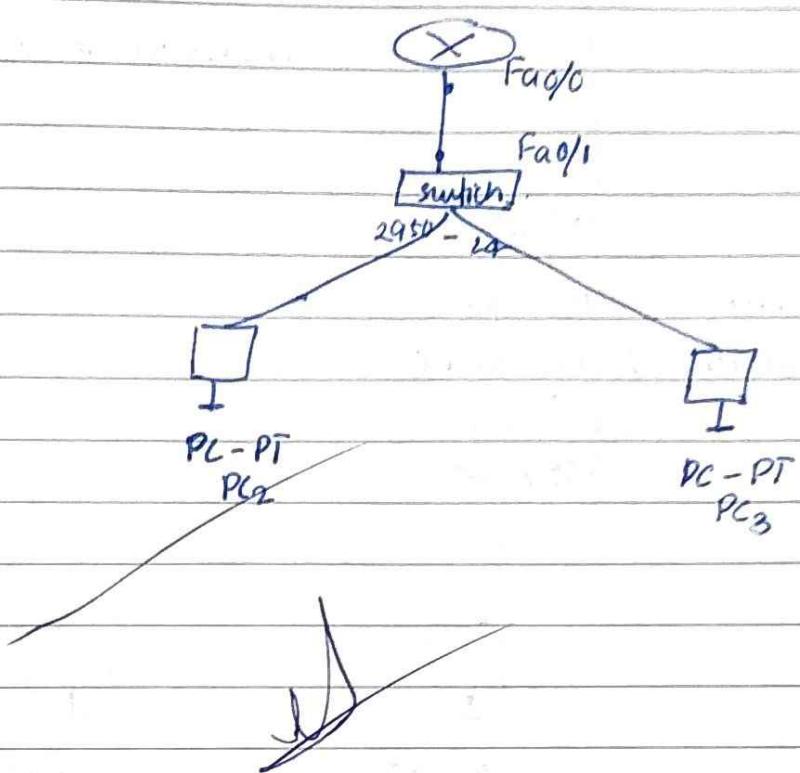
PC<sub>3</sub>

↳ Desktop

↳ IP Configuration [STATIC]

↳ IP Address : 192.168.1.3

Default Gateway : 192.168.1.1

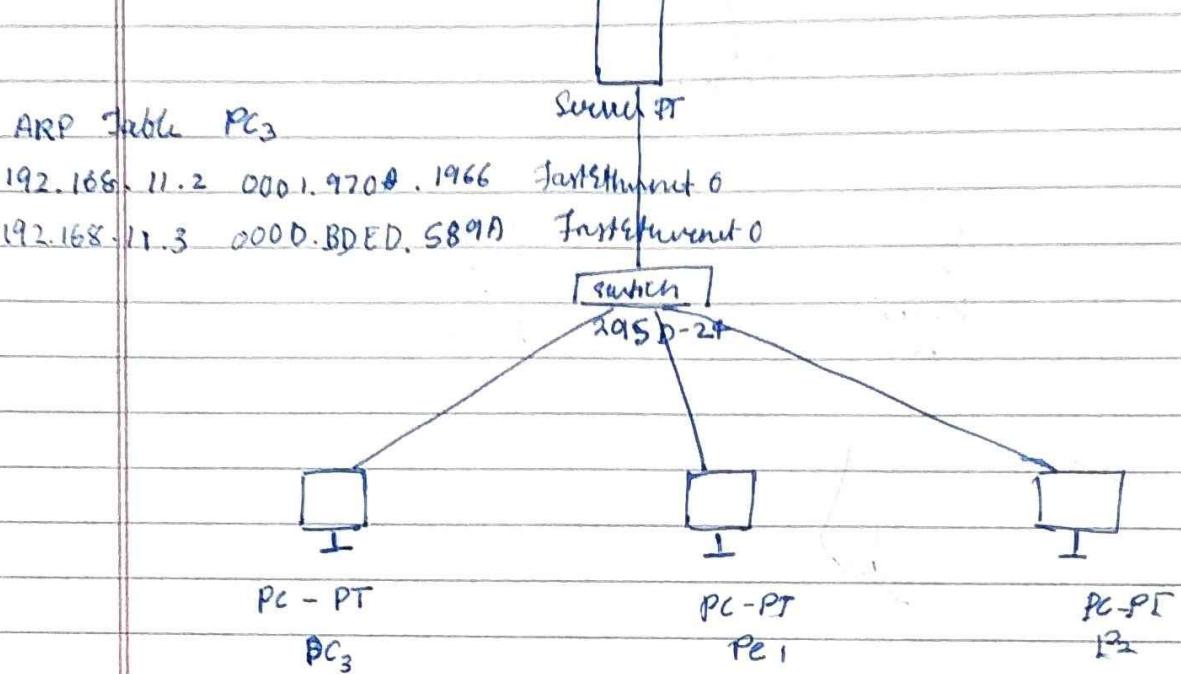


To construct simple LAN and understand the concept and also operation of Address Resolution Protocol (ARP)

- ARP is used to map an IP address to a MAC address.
- ARP is used to get data link layer address, MAC address with help of IP address.

ARP Table - Scenario

192.168.11.1 0090.0B015C90 FastEthernet0



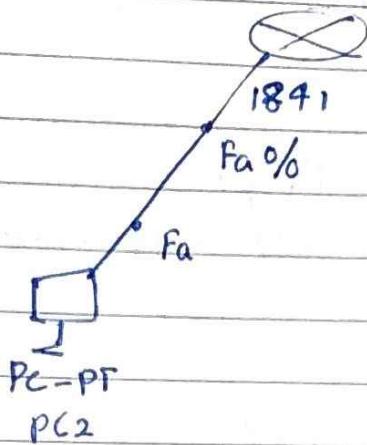
go to magnifying (a)

↳ select PC<sub>3</sub>, PC<sub>1</sub>, PC<sub>2</sub>, switch  
↳ ARP Table

At first open and ping of PC<sub>3</sub>  
↳ arp -a (No ARP Entries Found)

Then ping / send msg from PC<sub>3</sub> to other PC or server.

- Construct a Topology to demonstrate concept of TELNET
- It is used to access remote servers.
  - It is simple command line tool that runs on computer and allow you to send command remotely to a server.
  - TELNET is also used to manage other devices like routers, switches, to check if ports are open or close on a server.



Router

↳ CL

↳ no

enable

# configt

# enable secret rp

# int Fa0/0

# ip add 192.168.1.1 255.255.255.0

# no shutdown

# linc vty 0 5

# login

# password tp

# exit

# int

wr

# show ip interface brief

Pg  
6

Command prompt

↳ Telnet 192.168.1.1

R>telnet 192.168.1.1

password : tp

R1 >en

password : tp

R1# show ip interface brief

R1# config t

R1(config)# int fa 0/1

R1(config-if)# ip add 192.168.1.2 255.255.255.0

Route

↳ show ip interface brief

FastEthernet 0/0	192.168.1.1	YES	manual	up
FastEthernet 0/1	192.168.1.2	YES	manual	administratively
Vlan1	unassigned	NO	unset	

