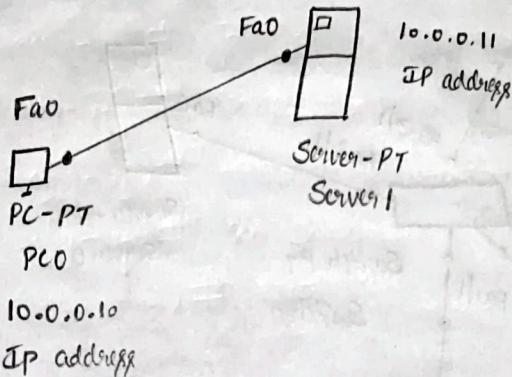


## 6) Configuring HTTP and DNS web server.

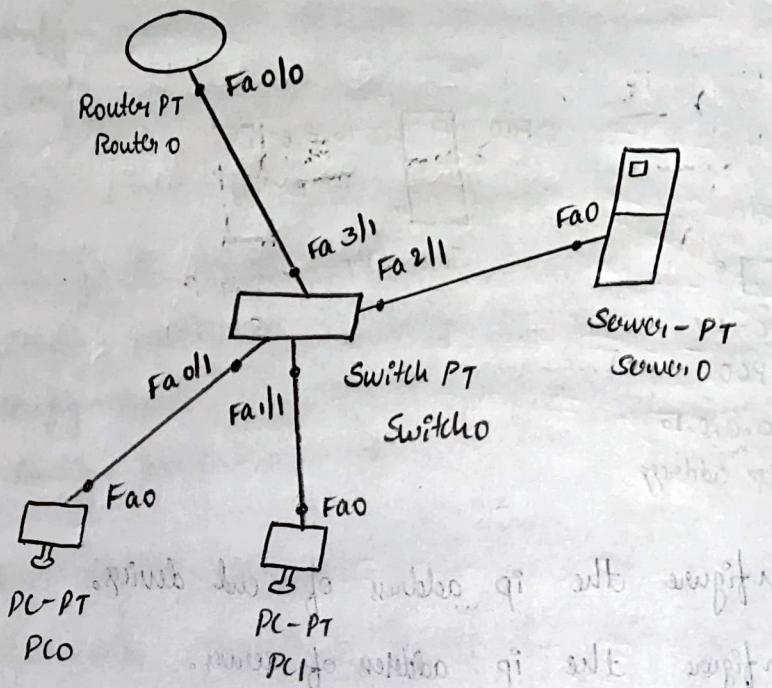
### Topology



- Step 1: Configure the ip address of end devices.
- Step 2: Configure the ip address of server.
- Step 3: Go to Services by clicking on services button, on DNS option, and click on ON.
- Step 4: Add the name (website name) like www.culab.com, address (ip address of server).
- Step 5: Then go to end-devices  
Desktop → Web Browser → enter url there
- Step 6: Go to services click on http we can see the file  
In File Manager  
click edit on index.html and give whatever message you want.
- Step 7: Follow step 5 & enter url we can observe change in main heading name.

Note : nslookup command - is used to know the IP address of any domain names

## 7) Configuring DHCP to Web Server, in same network.



Step 1: Create a topology as shown above

Step 2: Configure all the routers first, without configuring the end devices.

Step 3: Go to Services on Server, choose DHCP & select ON.

Step 4: Give default way on Router's IP address.

Step 5: Give DNS Server: Server's IP address

Step 6: Give starting IP address (10.0.0.1) - Same as router example IP address

Step 7: Give Subnet mask 255.0.0.0

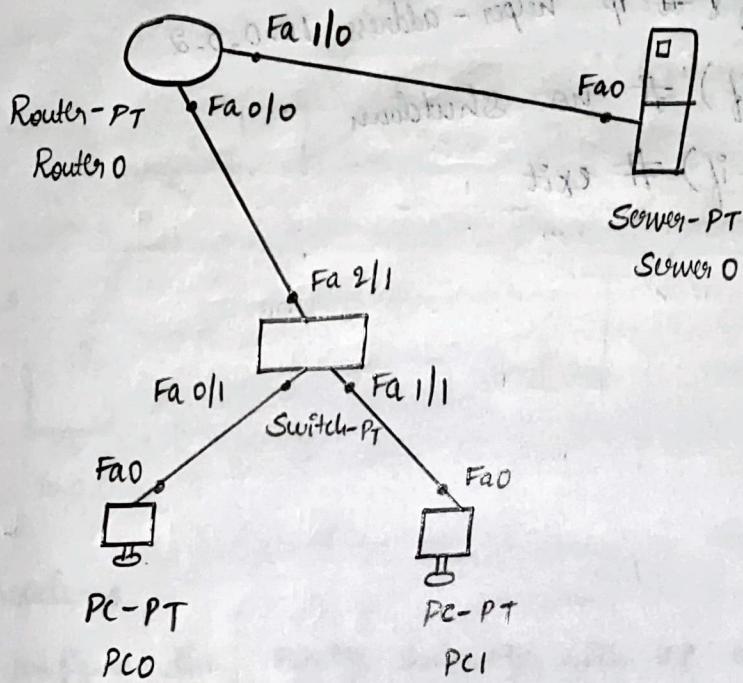
Step 8: Give TFTP Server: IP address of server.

Step 9: Add all details.

Step 10: Go to end device → Desktop → IP configuration → click on DHCP.

## 8) Configuring DRCP to web server, in different network

### Topology



Step 1 : Create a topology as shown above.

Step 2 : Configuring Router

Router # Config terminal

Router (config) # interface fast Ethernet 0/0

Router (config-if) # ip address 10.0.0.10 255.0.0.0

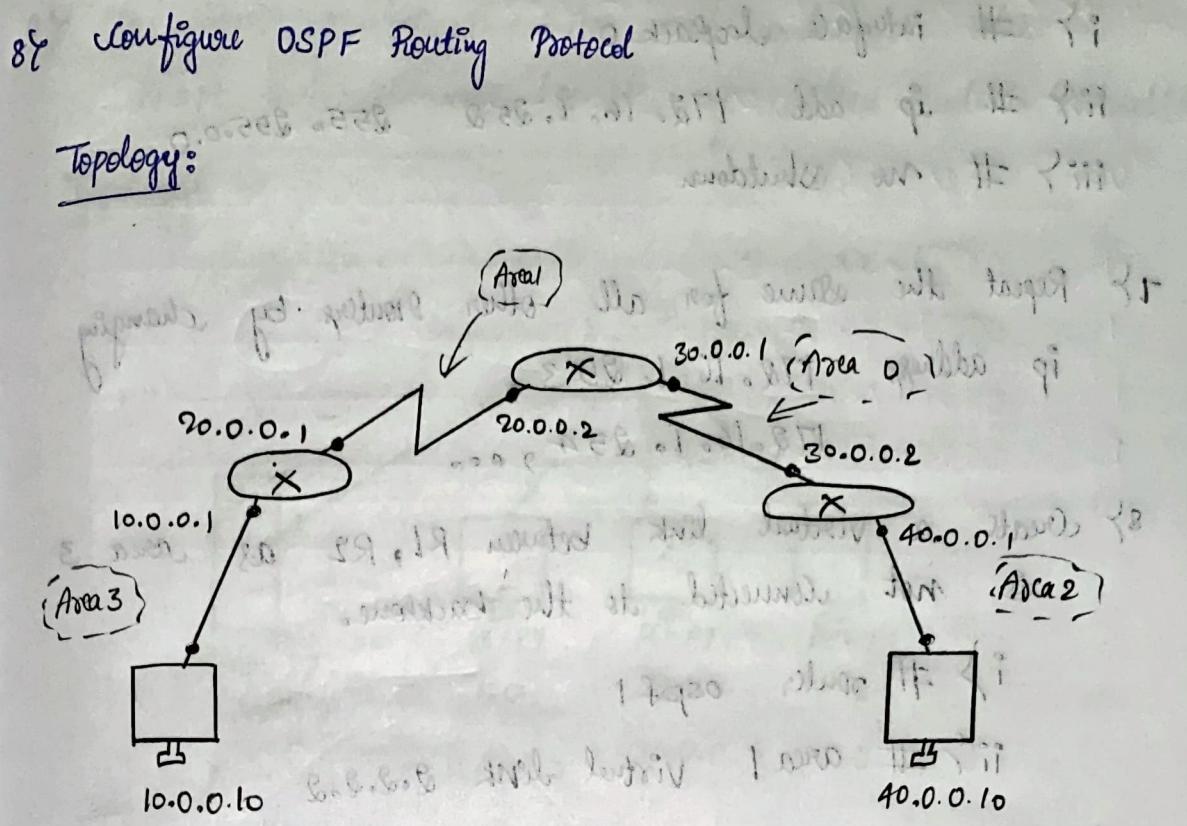
Router (config-if) # no shutdown

Router (config-if) # exit

Now do it for 1/0 fast Ethernet network

follow the given steps.

Router # config terminal  
Router (config) # interface fast Ethernet 1/0  
Router (config-if) # ip address 20.0.0.10 255.0.0.0  
Router (config-if) # ip helper-address 10.0.0.2  
Router (config-if) # no shutdown  
Router (config-if) # exit



### Procedures:

- 1) Configure the routers and PCs with IP addresses and gateway according to topology shown above.
- 2) Configure each router according to topology.
- 3) Encapsulation PPP and clock rate need to be set as done in RIP protocol experiment.
- 4) To set protocol follow these steps:
  - a) # router ospf 1
  - b) # router-id 10.0.0.1
  - c) # network 10.0.0.0 0.255.255.255 area 3
  - d) # network 20.0.0.0 0.255.255.255 area 1
- 5) Repeat the same following steps to each individual router with their respective addresses.
- 6) Now set loopbacks for the routers.  
Follow the given steps.

i) # interface loopback0  
ii) # ip add 172.16.1.252 255.255.0.0  
iii) # no shutdown

7) Repeat the same for all other routers by changing  
ip address 172.16.1.253,  
172.16.1.254, etc.

8) Create a virtual link between R1, R2 as Area 3  
is not connected to the backbone.

i) # router ospf 1

ii) # area 1 virtual link 2.2.2.2

9) Now execute commands in router 2

i) # router ospf 1

ii) # area 1 virtual link 1.1.1.1 weight 100  
router number specified at backbone port type link

Pinging

pc> ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data

Request timed out

Reply from 40.0.0.10 bytes=32 time=4ms TTL=125

Ping statistics for 40.0.0.10:

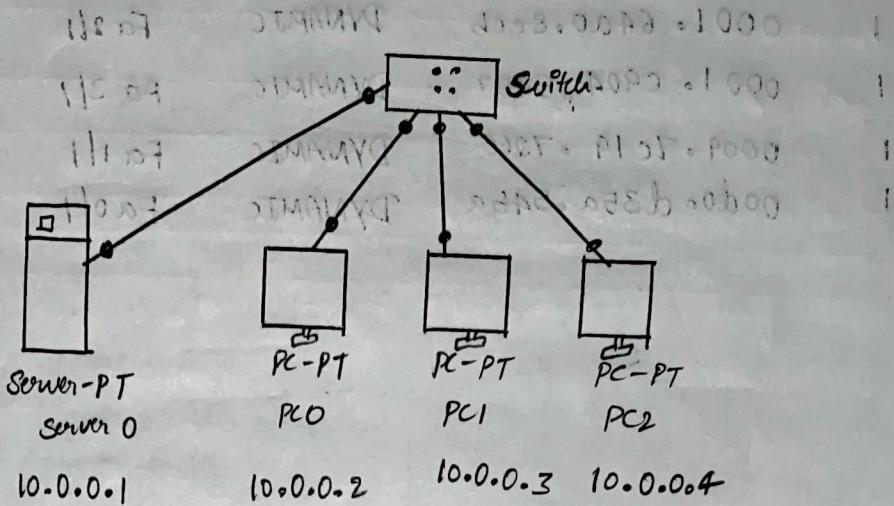
packets sent=4, Received=3, lost=1 (25% loss)

Approximate round trip in milli-seconds:

minimum=4ms, maximum=10ms, average=7ms

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

### Topology:



### Procedure:

- 1) Create a topology of 3 PCs and Server
- 2) Assign IP addresses to all devices.
- 3) Connect them through a switch.
- 4) Using Inspect tool to click on PC & Server do see ARP table
- 5) In command prompt, of PC type arp -a
- 6) In CLI of switch type show mac address-table can be given on transition, we can see a address-table

Pinging output

pc> arp -a

Internet Address	Physical Address	Type
10.0.0.1	00:0C:90:40:54:97	dynamic

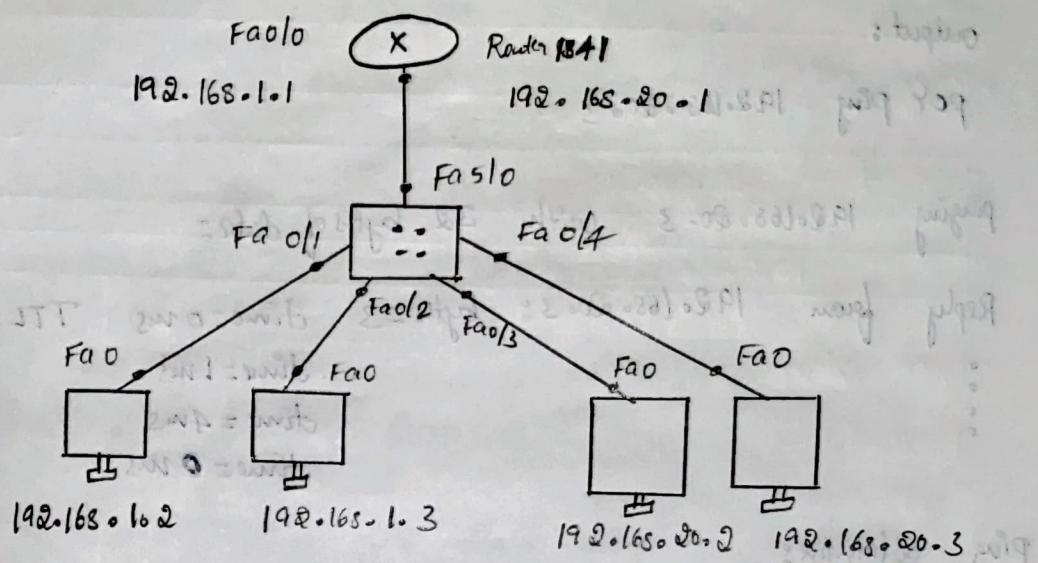
Switch 7 show mac address-table

(un)known student with MAC Address-table two type

Vlan	Mac Address	Type	Ports
1	0001.6400.8ecb	DYNAMIC	Fa 2/1
1	0001.c904.5457	DYNAMIC	Fa 3/1
1	0009.7c19.7260	DYNAMIC	Fa 1/1
1	00d0.d35a.546a	DYNAMIC	Fa 0/1

108 Construct a VLAN & make PCs communicate among VLAN

Topology:



Procedure:

- 1 Create a topology as shown above.
- 2 Connect all 4 PCs to Switch & now connect this router to switch.
- 3 Now create a VLAN database in switch with a VLAN number and VLAN names.
- 4 Now go to routers CLI mode.
- 5 Select interface fast Ethernet 0/5 which is towards or between switch and Router, and make VLAN trunk.
- 6 Go to the PCs which are set of VLAN & set ethernet to new VLAN.
- 7 Go to CLI of router, change it to config mode  
# interface 0/0.1  
# encapsulation dot1q 2  
# ip address 192.168.20.1 255.255.255.0  
# no shutdown

84 This will create a sub interface 0/0/1 setting  
192.168.20.1 as its gateway.

a) Now ping a pc from one network to another.

Output:

pc> ping 192.168.20.3

Pinging 192.168.20.3 with 32 bytes of data:

Reply from 192.168.20.3: bytes=32 time=0ms TTL=128

:



time=1ms

time=4ms

time=0ms

Ping statistics:

packets: Sent=4, Received=4, lost=0 (0% loss)

Appx round trip in milliseconds

Minimum=0ms Maximum=4ms, Average=2ms

MAX 0 ms. MIN 0 ms. via 0 ms. well  
MIN 0 ms. via 0 ms. well

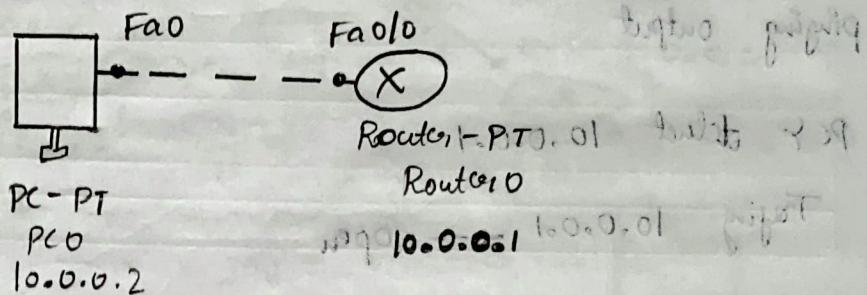
MAX 0 ms. via 0 ms. well

MAX 0 ms. via 0 ms. well

MAX 0 ms. via 0 ms. well

1) Understanding operations of TELNET by accessing the router.

Topology:



Procedure :

- 1) Create a topology as shown above
- 2) Set gateway of PC to 10.0.0.1 and IP address of PC to 10.0.0.2
- 3) Go to CLI of Router & set IP address of interface Fa 0/0 to 10.0.0.1
- 4) After setting up topology go to router & enable config mode.
- 5) Type hostname r1, then type enable secret p1 here p1 is password of your choice.
- 6) Now type IP address 10.0.0.1 and execute # no shutdown
- 7) In interface mode itself execute line vty 05 command which creates 6 times & now execute login & in next command, password P0 is password for login of router.

8) Go to cmd of PC & execute telnet 10.0.0.1  
which will get administrator access of router.  
Enter password & access router from a connected  
end device.

Pinging output

PC & telnet 10.0.0.1

Traffic 10.0.0.1 --- open

User Access Verification

Password:

enable

Password:

8) # show ip route at R1 for protocol 62

C 10.0.0.0/8 is directly connected, fast Ethernet 0/0  
via the RT 62 & R1 for E1 at IP 10.0.0.1

10.0.0.1 is 0.0.0.0 via interface

over RT of R1 for protocol 62  
via the RT 62 & R1 for E1 at IP 10.0.0.1

19 shows address 0.0.0.0 via interface 62  
via the RT 62 & R1 for E1 at IP 10.0.0.1

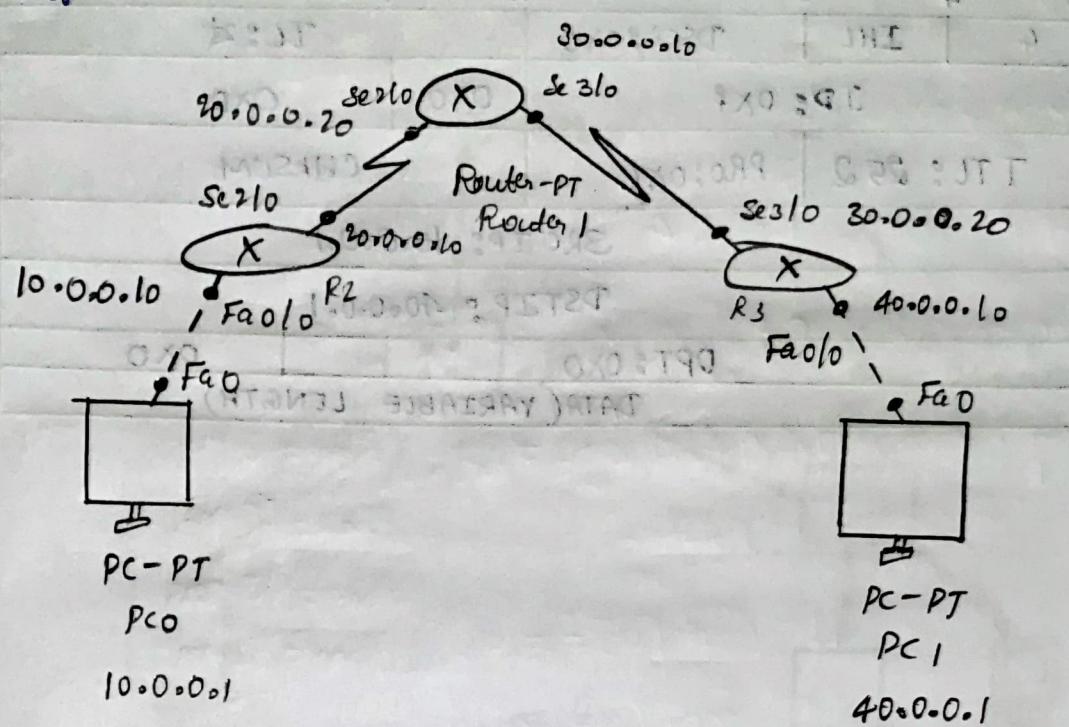
has 10.0.0.1 address RT 62 & R1 for E1 at IP 10.0.0.1

and similarly on the R1

so ping and done

12) Demonstrate the TTL / life of a packet.

Topology:



Procedure:

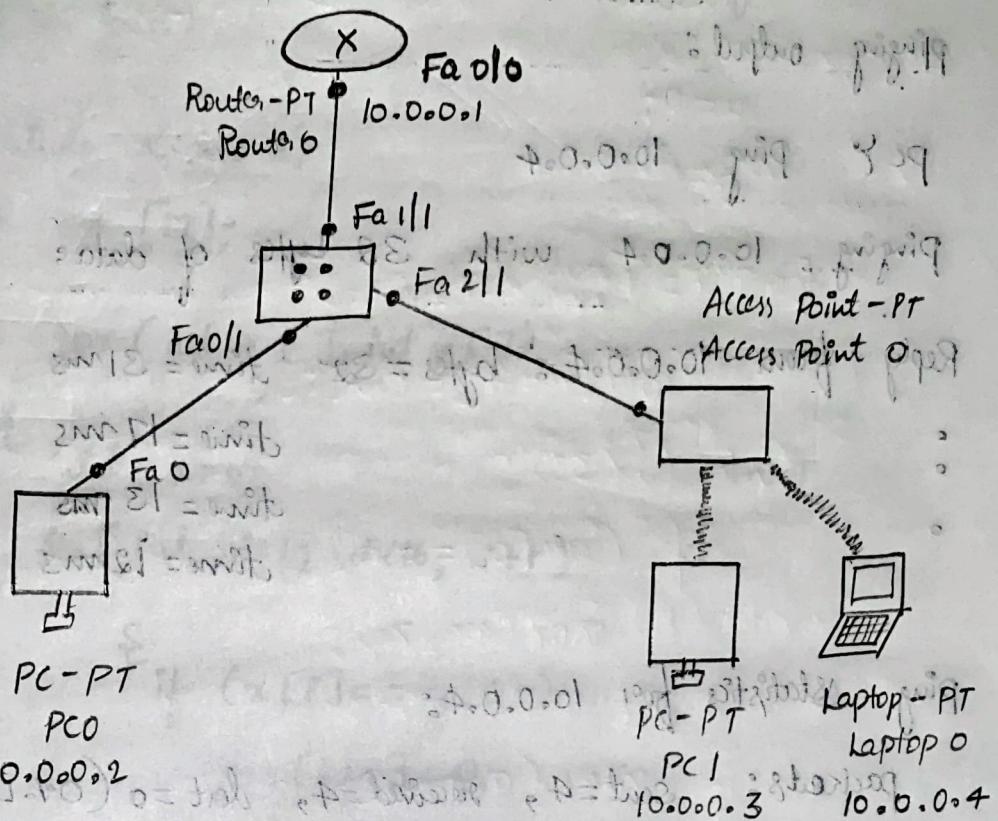
- 1) Create topology as shown above.
- 2) Go to simulation mode, Select Simple PDU & send message from PC0 to PC1
- 3) Click on capture button each time & check packet header each times
- 4) Each time when capture is clicked we can observe the simulation of packet from 1 device to other.
- 5) click on packet each time & observe the header file for TTL.

## Output

0	4	8	16	19	32 64
4	IHL	DSCP: 0x0	TTL: 28		
	ID: 0xf		0x0	0x0	
TTL: 252	PRO: 0xc			CHKSUM	
		SRC IP: 10.0.0.1			
		DST IP: 40.0.0.1			
	DPT: 0x0		0x0		
	DATA (VARIABLE LENGTH)				

13) construct a WLAN and make nodes communicate wirelessly.

Topology:



Procedure:

- 1) Create the above topology
- 2) Now set the gateway & ip address for all the PCs
- 3) Connect router, access point & PC directly to a switch.
- 4) Before setting IP address of PC1 & Laptop 0 switch them off and change the LAN port with wireless ~~modem~~.
- 5) Configure Port 1 of Access point, provide SSID Name, channel size & password.
- 6) Now go to PC & Laptops wireless interface in config tab & connect it to created WLAN.

7) Now topology is ready. Go to command prompt of directly connected PC & ping wireless one  
pinging output:

PC > Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=31ms TTL=128  
.  
.  
.  
time=17ms  
time=13ms  
time=12ms

Ping statistics for 10.0.0.4:

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

Avg round trip in milli-seconds:

minimum=12ms, maximum=31ms, Average=18ms

Details: Originated = ping to wireless 7E failed along with the wireless card & trigger reason customer found.  
After 200 ms, it will go to idle/7E failed along with the wireless card & trigger reason customer found.  
Details: Originated = ping to wireless 7E failed along with the wireless card & trigger reason customer found.  
After 200 ms, it will go to idle/7E failed along with the wireless card & trigger reason customer found.

cycle-2

Q Write a program for error detection using CRC - CCITT  
(16-bits)

```
#include <stdio.h>
int arr[17];
void xor(int x[], int y[])
{
    int k=0;
    for (int i=0; i<16; i++)
    {
        if (x[i]==y[i])
            arr[k++]=0;
        else
            arr[k++]=1;
    }
}

void main()
{
    int dd[17], div[33], ze[17], i, k;
    printf("Enter dataword\n");
    for (i=0; i<17; i++)
        scanf("%d", &div[i]);
    for (i=0; i<33; i++)
        div[i]=0;
    for (i=0; i<17; i++)
        ze[i]=0;
}
```

printf ("Enter dividend");

for ( $i=0; i<17; i++$ )

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

$i=0;$

$k=0;$

for ( $i=1; i<17; i++$ )

$arr[k+i] = arr[i] \text{ mod } 10;$

while ( $i<33$ )

if ( $arr[0] == 0$ )

xor (arr, zero);

else

xor (arr, dd);

$arr[16] = arr[i+16];$

$\}$

$k=0;$

for ( $i=17; i<33; i++$ )

$div[i] = arr[k+i];$

printf ("Codeword %d",

for ( $i=0; i<33; i++$ )

printf ("%d", div[i]),

for ( $i=0; i<17; i++$ )

$arr[i] = 0;$

printf ("At receiver, end transmission"),

$k=0;$

$i=0; arr[i] = 0;$

for ( $i=1$ ;  $i < 17$ ;  $i++$ )

arr [ $k++$ ] = div [ $i$ ];

while ( $i < 33$ )

if (arr [ $0$ ] == 0)

xor (arr, ze);

else

xor (arr, dd);

arr [ $16$ ] = div [ $i++$ ];

$k = 0$ ;

for ( $i=17$ ;  $i < 33$ ;  $i++$ )

div [ $i$ ] = arr [ $k++$ ];

printf ("Code word: %s", arr);

for ( $i=0$ ;  $i < 33$ ;  $i++$ )

printf ("%d", div [ $i$ ] );

### Output:

Enter dataword

1 0 1 1 0 0 1 1 1 0 0 1 0 1 1 1

Enter dividend

1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1

Code word: 101100111100101110000000000011011

At receiver end

Code word: 101100111100101110000000000000000000000

Q) Write a program for Congestion Control using leaky bucket algorithm.

```
#include <stdio.h>
Void main () {
    int buckets, outlets, k=1, num, remaining;
    printf ("Enter Bucket size & Outstream size \n", num, outlets);
    scanf ("%d %d", &buckets, &outlets);
    remaining = buckets;
    while (k) {
        num = rand() % 1000;
        if (num < remaining)
            remaining = remaining - num;
        printf ("Packet of %d bytes accepted \n", num);
        else
            printf ("Packet of %d bytes is discarded \n", num);
        if ((buckets - remaining) < outlets)
            remaining += outlets;
        else
            remaining = buckets;
        printf ("Remaining bytes: %d \n", remaining);
        printf ("If you want to Stop input, press 0, otherwise press 1 \n");
        Scanf ("%d", &k);
    }
}
```

while (remaining > buckets)

if (buckets - remaining > outlets)

remaining + = outlets;

else

remaining = buckets;

point f ("Remaining bytes: %d\n", remaining);

Enter Bucket size & OutStream size : = 2000, 100

Packet of 41 bytes accepted

Remaining bytes: 2000

If you want to stop input, press 0, otherwise press 1

Packet of 467 bytes accepted

Packet of 334 bytes accepted

Remaining bytes: 1399

Remaining bytes: 1499

Remaining bytes: 1599

Remaining bytes: 1699

Remaining bytes: 1799

Remaining bytes: 1899

Remaining bytes: 1999

Remaining bytes: 2000

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

client TCP.py

from socket import \*

Server Name = '127.0.0.1'

Server Port = 12000

Client socket = socket (AF\_INET, 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 ('From Server: ', file contents)

print (file contents)

Client socket. close ()

Client side program

Python

Server TCP.py

from socket import \*

Server Name = "127.0.0.1"

Server Port = 12000

Server Socket = socket (AF\_INET, SOCK\_STREAM)

Server Socket.bind ((Server Name, Server Port))

Server Socket.listen(1)

while 1:

(Mark point (see The server is ready to receive))

Connection Socket, addr = Server Socket.accept()

Sentence = Connection Socket.recv(1024).decode()

file = open (Sentence, "r")

(File open, read mode)

L = file.read(1024)

Connection Socket.send(L.encode())

Point ("In Sent contents of" + Sentence)

(File close)

Connection Socket.close()

Server side Program

Python

47 Using UDP Sockets, write a client - server program to make client sending the file name & server to send back the contents of requested file if present.

Client UDP.py

from socket import \*

Server Name = "127.0.0.1"

Server Port = 12000

Client Socket = socket (AF\_INET, SOCK\_DGRAM)

Sentence = input ("Enter file name: ")

clientSocket.sendto (bytes (Sentence, "utf-8"),  
(Server Name, Server port))

file (contents), Server Address = Client Socket.recvfrom (2048)

print ("Reply from Server")

print (file.decode ("utf-8"))

Client Socket.close()

Client Socket.close()

Server UDP.py

from socket import \*

Server Port = 12000

Server Socket = socket (AF\_INET, SOCK\_DGRAM)

Server Socket.bind ((127.0.0.1, Server Port))

print ("The Server is ready to receive")

Server Socket.listen(1)

while (1):

Sentence, Client Address = Server Socket.recvfrom(2048)

Sentence = Sentence.decode("utf-8")

file = open (Sentence, "r")

Con = file.read(2048)

Server Socket.sendto (bytes (Con, "utf-8"), Client Address)

print ("Sent Con of", end = "\n")

print (Sentence)

file.close()