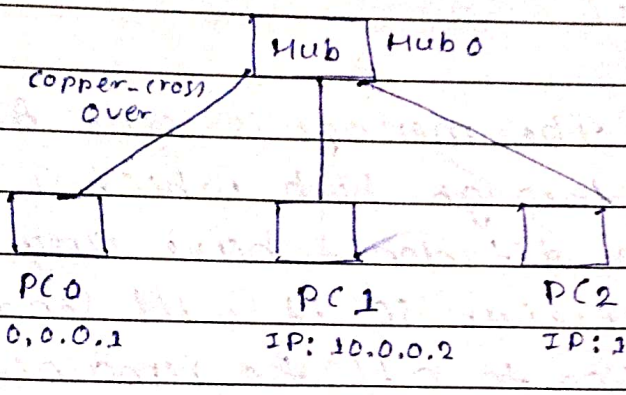
SL NO	Date	Title	Page No	Teacher's Sign / Remarks
1).	1/10/24	The transmission of PDU b/w 2 devices	8	
2).	8/10/24	CLI to Router	10	
	15/10/24	Demonstrate the configuration of Default routes	10	
	22/10/24	configure default & static routes to a connection of routes	10	
	29/10/24	TO understand operation of Telnet.	10	
	5/11/24	a) To configure IP adress of host using DHCP server.	10	
		b) To " " using server in lan.	10	
	19/11/24	configure Rip routing Protocol in server		
	26/11/24	To demonstrate connection b/w two devices using a wireless lan.	10	
	3/12/24	to creat a vlan on top of Physical lan & enable communication b/w physical lan & vlan.	10	
	17/12/24	leaky bucket Algorithm & CRC Algorithm	10	

Aim: To demonstrate the transmission of a simple PDU between 2 devices connected using a hub and a switch

Topology: star topology.



Configuration steps:

1. selecting end devices: select the end devices to be three generic end devices
2. select Hub: select a generic hub
3. Connection type: use connection type copper-~~cross~~ straight through over or automatically choose connection type
4. Set Ip adress: click on end devices to open the configuration window, switch to config tab and select Interface, In that select FastEthernet0, there since we are using IPV4 binding the Ip adress should be of the form 10.0.0.1 with 4 numbers seperated with dots and after that click on subnet mask which will be automatically selected.
5. setting Mode: Make sure when we started experiment we are on real time mode
6. Add simple PDU: select simple PDU and then first click on the sender of message and then the receiver.

7. Simulation mode: Switch to simulation mode to see the simulation for transfer of message from source device to receiver device

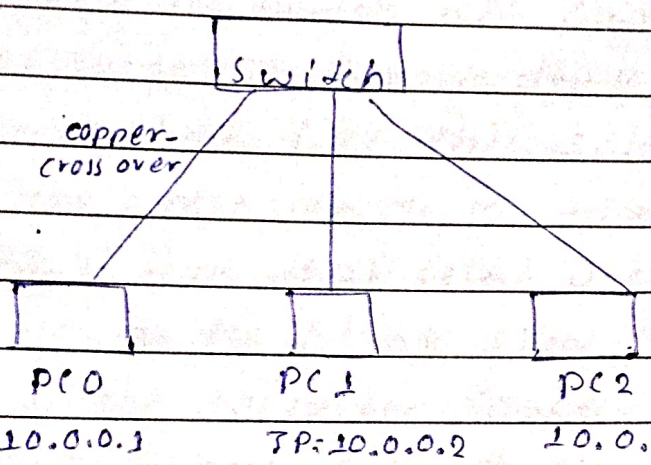
8. Start simulation: In order to start the simulation click on auto capture button on simulation panel

Observation: The message is sent from sender to the receiver through hub which is not a intelligent device, so did don't have any information regarding the receiver, so it will send the received information to all the remaining devices connected to hub

- If the message is reached destination device then it will send success status to hub, which in turn communicate it to remaining devices and the actual sender of message would get success feedback and remaining devices will neglect the signal

Aim: To demonstrate the transmission of a simple PDU between two devices connected using a switch

Topology: star topology



Configuration step:

1. end devices: select 3 generic end devices
2. switch: select a generic switch
3. Connection type: Choose copper-~~over~~ straight through. cross over or automatically choose connection type to establish connection
4. select the end devices which will open configuration tab and in that select config tab and go to interface and select FastEthernet0. set the IP adress in IPV4 gateway. select subnet mask which will be selected automatically.
5. Add simple PDU: select simple PDU and first select the sender and then select the receiver.

Simulation:

- switch to simulation mode which would open simulation panel there start simulation by clicking on auto capture / play button.

Date : / /

Observation: The message is sent from Sender to the receiver through switch, since switch is a intelligent device, it would send the message to only to the destination device which in turn send the success signal back to the sender of message through switch.

Hello 2nd

10/10

Aim: To demonstrate config of Ip adresses to the router and explore ping command

Observations:

* Open cisco packet tracer: Launch the application
* Add devices: From the bottom left corner there is one showing devices select Generic device from that, click on to blank sheet, like this select another one
* Add router: Just above the devices there is one tab called Connections from that select one generic router and place it on the blank sheet
* Connector: Select connecter copper-through straight cable and connect each device with the router
* IP adress: Click on each devices and go to config tab click on the Ip adress for each device (both IP adress should belongs to

PC0 => 10.0.0.1

PC1 => 10.0.0.2

* Config Router:

Click on to the device and go to CLI enter the following commands to config:

1) enable
2) config t
3) interface fastethernet 0/0
4) Ip adress 10.0.0.2 255.0.0.0
5) no shutdown

DO the same for PC1 also.

1) enable
2) config t
3) interface ethernet

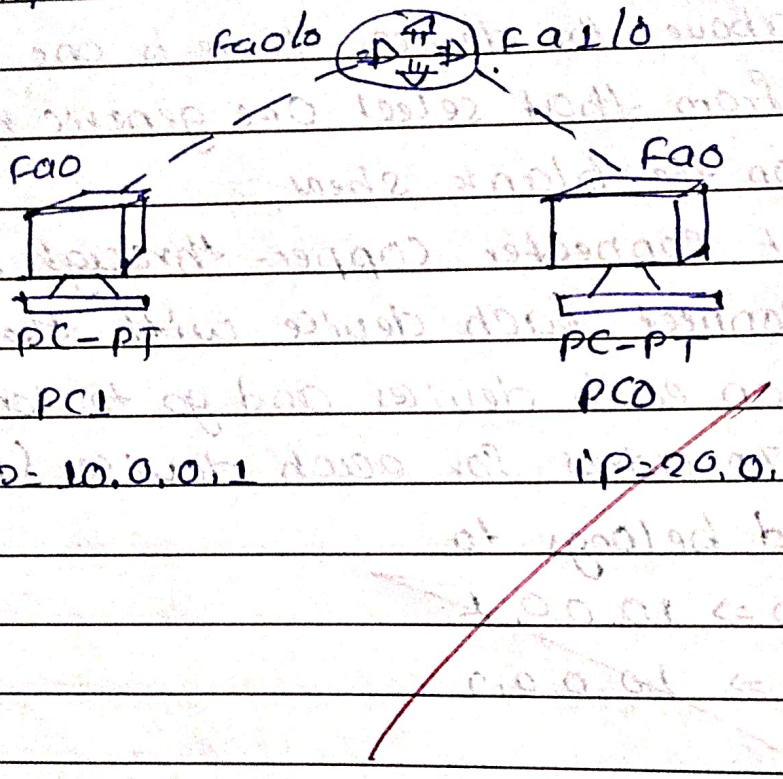
IP adress 10.0.0.1 255.0.0.0

no shutdown

after this set gateway

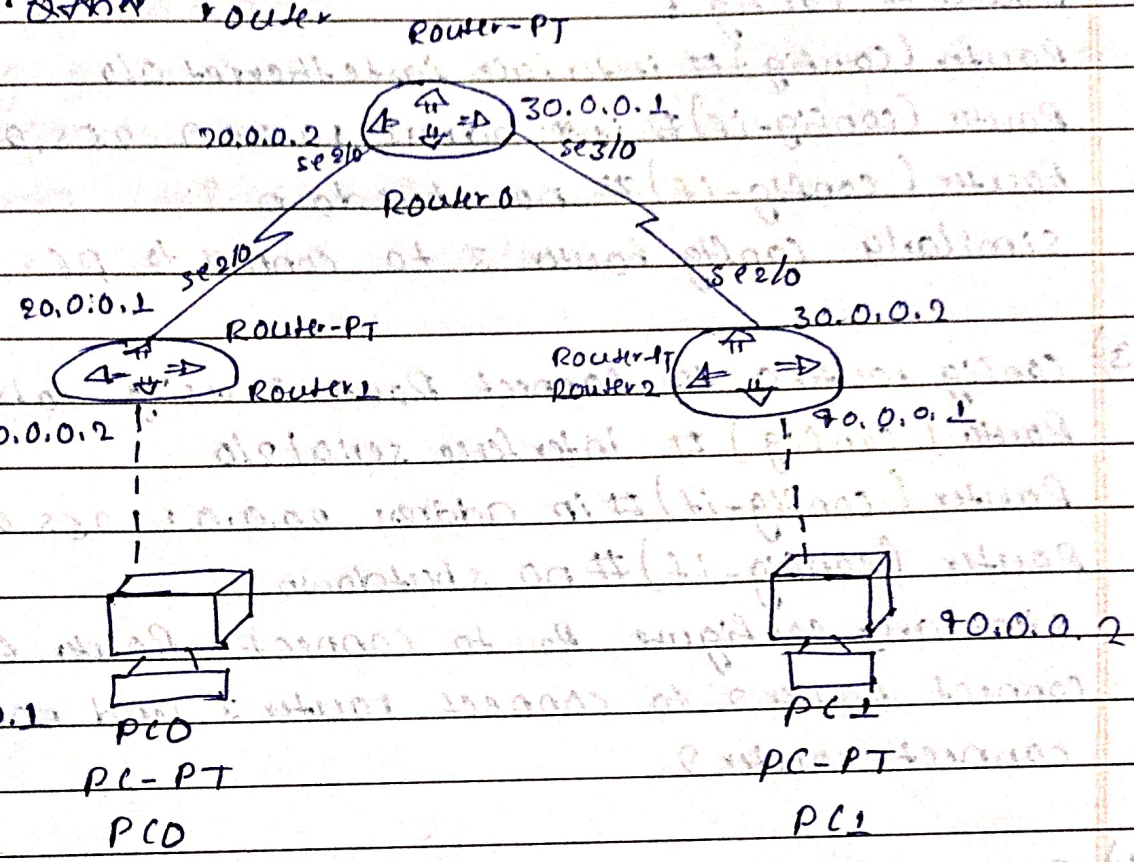
to do this click on the device set the gateway for device 1 it is 10.0.0.2 for device 2 it is 20.0.0.1

after that click the desktop and go to command prompt and execute the command ping IP adress



SWE
22/10/24

Aim: To demonstrate the config of default routes to the router



=> show ip route => command to know connections

ip route
destination nslw ip
destination mask
next stop

eg ip route 30.0.0.0 255.0.0.0 20.0.0.2
ip route 40.0.0.0 255.0.0.0 20.0.0.2

Procedure:

1. Configure the PC0 network with IP address set as 10.0.0.1, set gateway address as 10.0.0.2 similarly configure PC1 with ip as 40.0.0.2 set gateway as 40.0.0.1

2) Config router0 to connect PC0 using CLI as follows

Router enable
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
similarly config router 3 to connect to PC

3) Config router 0 to connect Router 2 using following com
Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
Router (config-if) # no shutdown
similarly configure R2 to connect Router & similarly connect Router 2 to connect router 3 and router 3 to connect router 2

4) To check network des router connects to in CLI type the router > show ip route

5) To add other networks:
Router (config) # ip route destination
destination mask
next hop

6) To connect Router to network 30 & 40
Router (config) # ip route 30.0.0.0 255.0.0.0
20.0.0.2
Router (config) # ip route 40.0.0.2 255.0.0.0
20.0.0.2
similarly connect other routers with each other

6) ping a message from pc0 to pc1 through command prompt

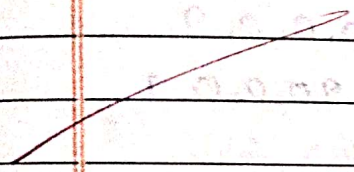
PC> ping 10.0.0.2

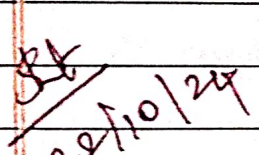
pinging 10.0.0.2 with 32 bytes of data from 10.0.0.2

bytes 32 time

ping statistic for 10.0.0.2

packets: sent = 4 received = 4





22/10/24

Aim: To configure default and static routes of routers.

~~router 0/2 ip~~

Config t

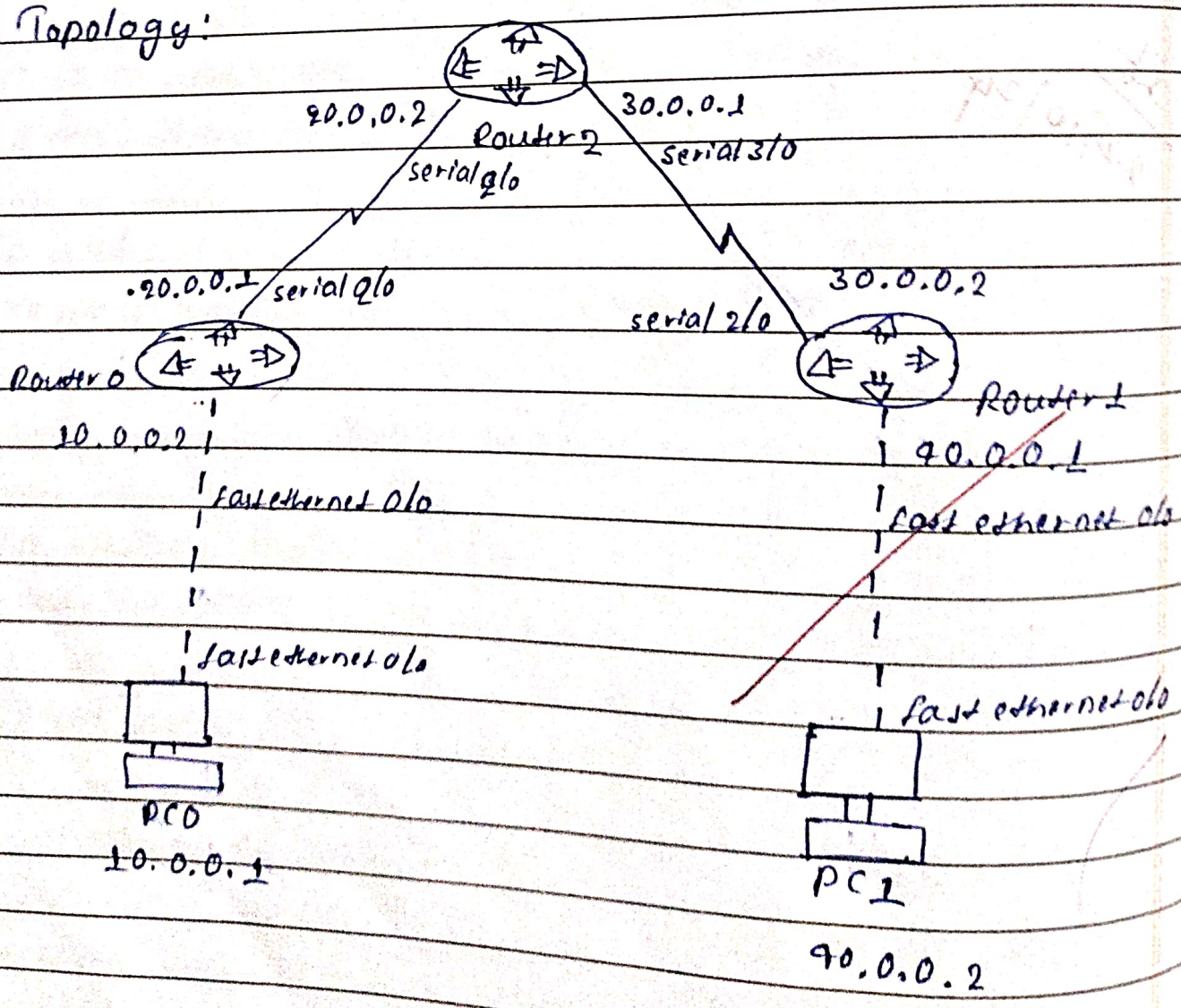
R2: (config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

R1: ip route 0.0.0.0 0.0.0.0 30.0.0.1

~~ip route 10.0.0.0 255.0.0.0 20.0.0.2~~

R2: ip route 40.0.0.0 255.0.0.0 30.0.0.2
ip route 10.0.0.0 255.0.0.0 20.0.0.1

Topology:



EXCELLENT
Page No.
Date : / /

Procedure:

1) configure PC0 & PC1 by giving ip address as 10.0.0.1 and 40.0.0.2 respectively and also give default gateway as 10.0.0.2 and 40.0.0.1 respectively

2) Configuring routers

Router 0: give ip for fastethernet interface as 10.0.0.2 and serial 2/0 as 20.0.0.1

Router 2: serial 2/0 ip 20.0.0.2
serial 3/0 ip 30.0.0.1

Router 1: serial 2/0 ip 30.0.0.2
fastethernet 0/0 ip 40.0.0.1

3) Set default gateways from Router 0
(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2
it means that any messages from router 0 will be forwarded to default gateway 20.0.0.2

4) Set default gateway from Router 1
(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1
it means that any message from router 1 will be forwarded to default gateway 30.0.0.1

5) setting gateways at Router 2
(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2
(config)# ip route 10.0.0.1 255.0.0.0 20.0.0.1
it means that any packets addressed to 40's network will be routed to gateway 30.0.0.2 in first command, & the similar to second command.

6) ping from PC0 to PC1
>ping 40.0.0.2

Pinging 40.0.0.2 with 32 byte of data.
Reply from 40.0.0.2; bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=10ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125

packets sent: sent=4, Received=4, lost=0

To know the connections established from any given router, write the command in CLI (from Router2)
~~show~~ Router> show ip route
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, serial2/0
C 30.0.0.0/8 is directly connected, serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
S* 0.0.0.0/0 [1/0] via 20.0.0.1
[1/0] via 30.0.0.2

8 bit
22/20/24

TTL → time to live.

Aim: To understand the operation of telnet, by accessing the ~~router, server~~ router placed in the server room, from a PC in IT office.

Router (config)# hostname R1

R1 (config)# enable secreat P0
↳ password.

R1 (config)# line vty 0 5

R1 (config-line)# login

password P1

wr → to save configuration.

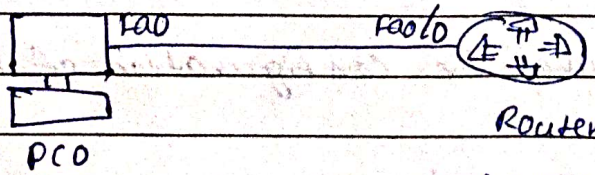
access router from PC

Command prompt

telnet 10.0.0.2
↳ Router IP.

password: P1
↳ it will not be visible, we just have to enter correct password & press enter.

Topology:



Observation:

Set the ipadress of PC and Router PC, make sure that both have same network that is 10's network or 20's

Set router configuration: in CLI enter the commands.

Router# config t

Router(config)# host name R1

↳ it is alias name used for given router.

R1(config)# enable secret P0

here p0 is password, we can give any password we want.

R1(config)# line vty 0 5

R1(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

% login disabled on line 137, until 'password' is set

R1(config-line)# password p1

here p1 is any password we can choose.

R1(config-line)# exit

R1(config)# exit

wr

↳ it would save the configuration of router

Open command prompt

PC> ping 10.0.0.2

Reply from 10.0.0.2. bytes=32 time=1 ms TTL=255
Reply from 10.0.0.2 bytes=32 time=0ms TTL=255
Reply from 10.0.0.2 bytes=32 time=0ms TTL=255
Reply from 10.0.0.2 bytes=32 time=0ms TTL=255

Packets: sent=4, Received=4, lost=0.

PC> telnet 10.0.0.2
Trying to open 10.0.0.2.

User access verification.
password:
here enter the password set for the router.
R1>

18/11

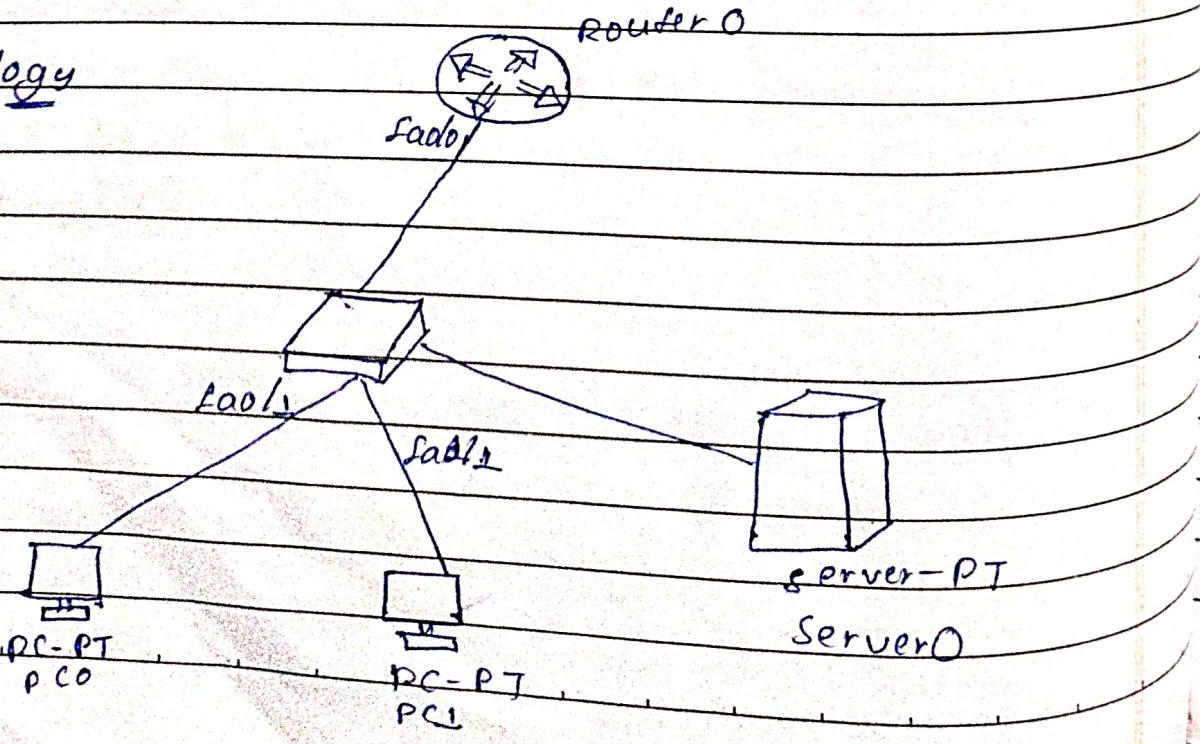
AIM: (a) to configure IP addresses of Host using DHCP server present within the lan
(b) to configure IP adress of Host using DHCP server present in the different LAN

DHCP → Dynamic Host Configuration protocol

→ assign IP adress to server Manually : 10.0.0.1
→ configure router, assign IP adress to router : 10.0.0.2
→ set gateway for server, (routers interface adress)
→ configure DHCP protocol
server → service → DHCP → turn on service
pool name : serverpool1
Default : 10.0.0.2
DNS : 10.0.0.1

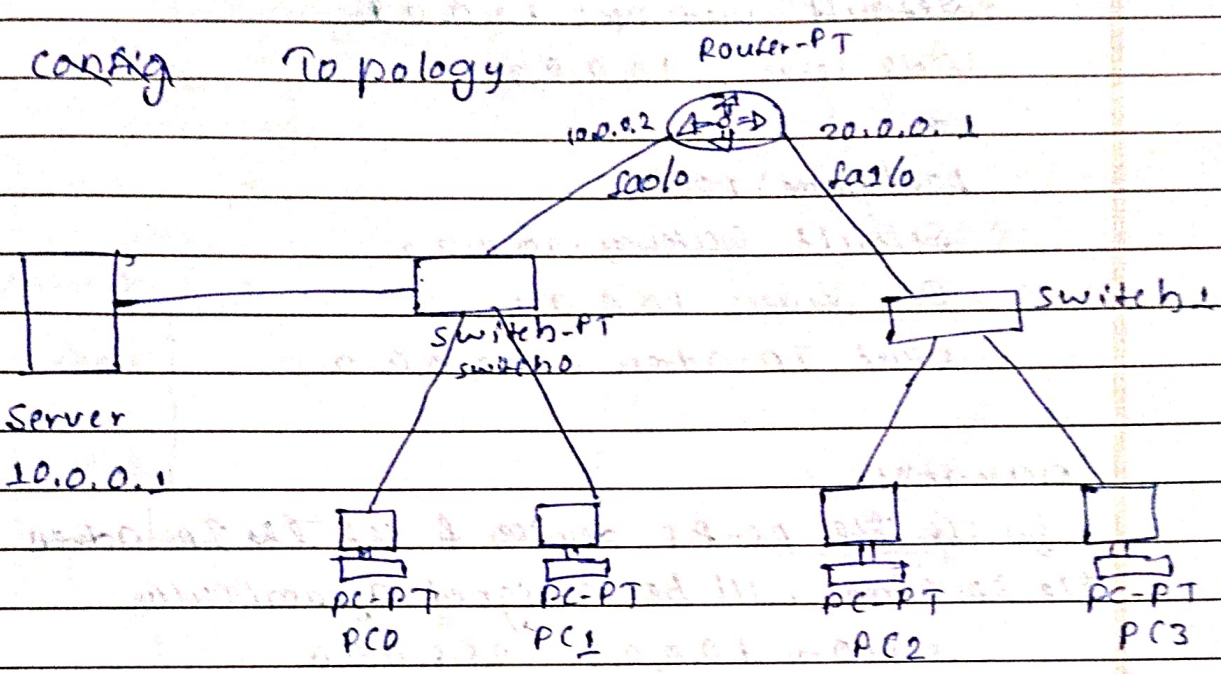
→ Dynamically configure IP adress for end device
select end device > Interface > FastEthernet0
IP configuration
↓ select DHCP
it will automatically set IP adress to end device

Topology



Observation:
When we try to configure IP address for end devices, select DHCP server will automatically select IP adress for PCs.

(b) Config Topology



Configure Server.
IP adres: 10.0.0.1
poolName: serverpool
Default: 10.0.0.2
DNS: 0.0.0.0

With CLI
enable
Router#
Router# config t
Router(config-t)# interface fastEthernet fa0/0
Router(fa0/0)- 10.0.0.2 255.0.0.0
Router(fa0/0)- 20.0.0.2 255.0.0.0

Set gateway in server PT. as 10.0.0.2

7) Now, go to the server in server-PT device & go to the DHCP & turn on the service & save the following details.

-> poolname: pool1
-Default Gateway: 10.0.0.2
DNS server: 10.0.0.2

poolname: pool2
Default Gateway: 20.0.0.2
DNS server: 10.0.0.1
start IP adress: 20.0.0.2

Observation:
go to the PC-PT devices & set the IP adress as DH
the IP adress will be assigned dynamically.

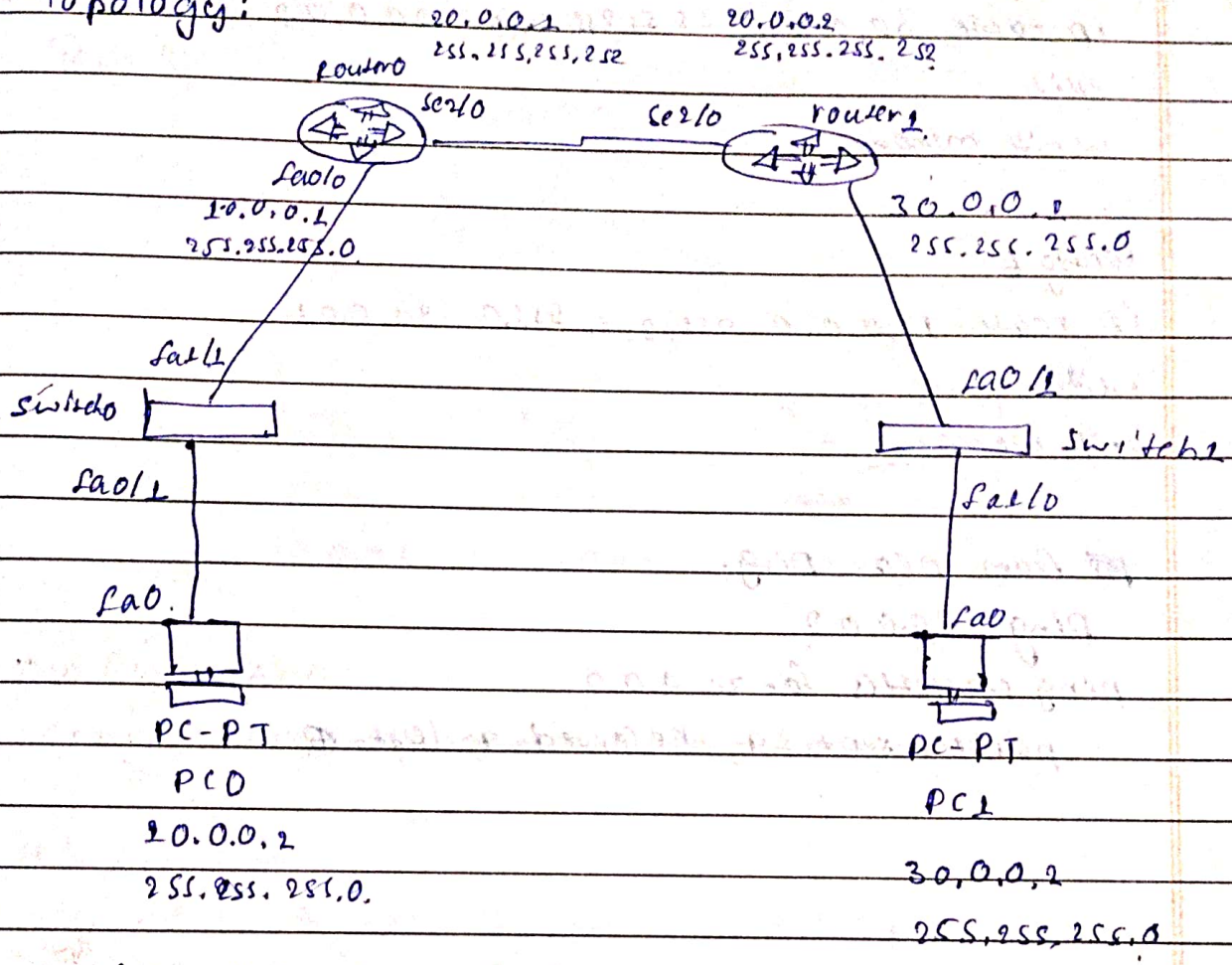
PC0 → 10.0.0.3 255.0.0.0
PC1 → 10.0.0.4 255.0.0.0
PC2 → 20.0.0.1 255.0.0.0
PC3 → 20.0.0.2 255.0.0.0

CLI
Renable.
Router#
Router# config t
Router(config-t)# interface fastEthernet 0/0
Router(con-IF)# ip helper 10.0.0.1
Router(con-IF)# no shutdown
Router(con-IF)# exit
Router(config-t)# interface fastEthernet 1/0
Router(config-IF) ip helper 10.0.0.1
no shut down
exit

EXCELLENT
Page No.
Date : / /

Aim: Configure RIP routing protocol in Routers.

Topology:



Configure router 0.

- ip addr fa0/0 : 10.0.0.1 255.255.255.0
  se2/0 : 20.0.0.1 255.255.255.252

CLI: interface serial 2/0
ip addr 20.0.0.1 255.255.252
clock rate 64000
no shutdown
exit
write memory

Configure router 1:

fa0/0 30.0.0.1 255.255.255.252
serial 20.0.0.2 255.255.255.252

enable routing:

Config t

ip route 30.0.0.0 255.255.255.0 20.0.0.2

exit

write memory

Config t

IP route 10.0.0.0 255.255.255.0 20.0.0.1

exit

write memory

from pc0 ping:

ping 30.0.0.2

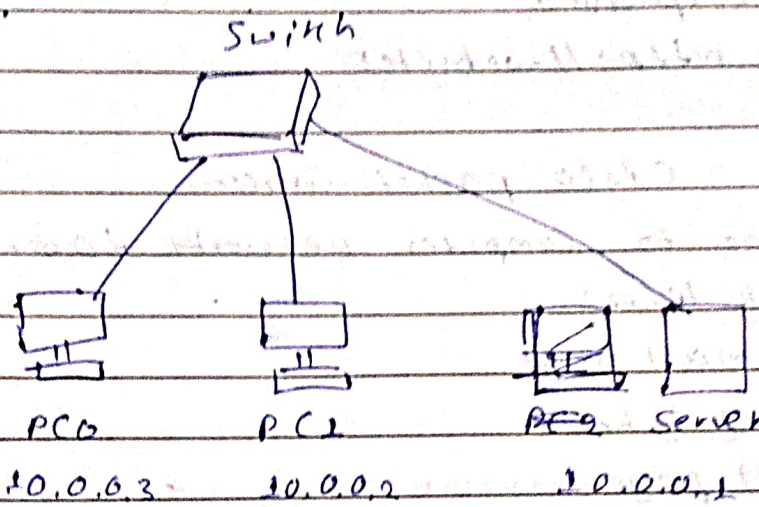
ping statistics for 30.0.0.2

packets: sent = 4, received = 4, lost = 0

Sat
18/11

Aim: Configure DNS server to demonstrate the mapping of IP adresses & domain names.

Topology:



Configuration:

Connection: Copper Straight through.

At server:

In fast Ethernet ip adress 10.0.0.1

In services > use DHCP protocol:

Default gateway 0.0.0.0

DNS server 10.0.0.1

> In DNS server.

Name: Website

Adress: 10.0.0.1

> In HTTP:

Edit index.html, as per your requirement

IP PC0: Click on DHCP, the IP adress get configured automatically.

In Laptop0: on Clicking DHCP. IP adress get configured automatically.

Observation:

Using any end device, click on web browser in Desktop, Then, on typing website, the domain name, index.html gets displayed.

URL: http://website/

Cisco packet Tracer

Welcome to Computer networks Laboratory

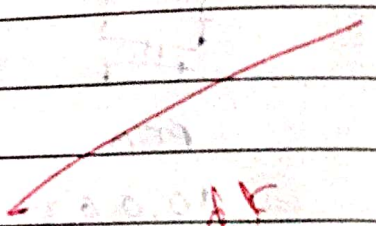
Quick links:

a small page

Copyrights

Small page

image

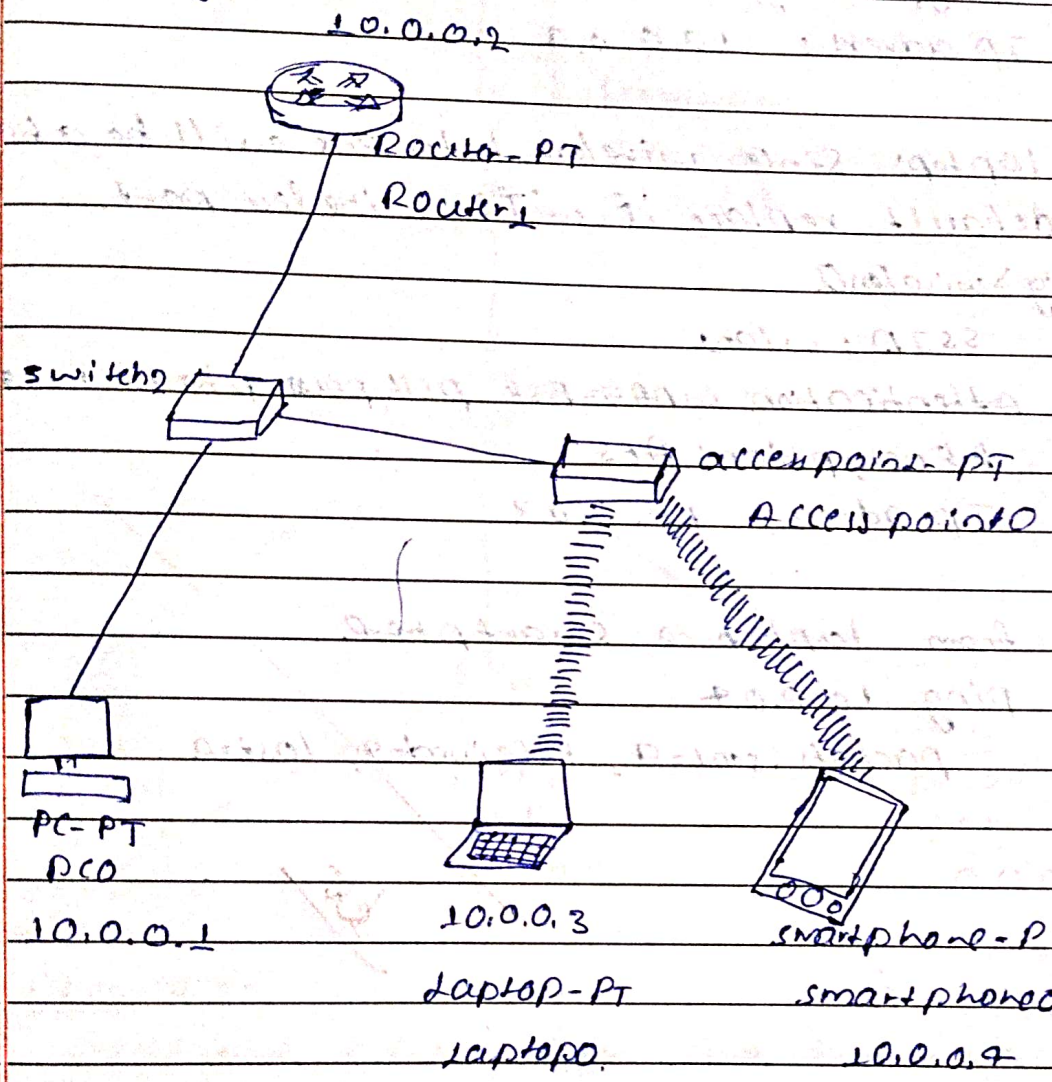


18/11

EXCELLENT
Page No.
Date: / /

Aim: to demonstrate communication between two devices using a wireless lan

Topology



Observation

configure the devices as shown in figure and topology and also give corresponding Ip address. for PC0, give gateway as 10.0.0.2

for accesspoint:

port0: Select Band width as auto and turn on.

port1: SSID: wlan1

Channel: 6

Authentication: WPA2-PSK PSK Passphrase 12345678

Encryption type: AES

for smartphone

wireless0: SSID: wlan1

Authentication: WPA2-PSK PSK pass phrase 123456

Encryption type: AES

IP adress: 10.0.0.4

for laptop: Since wireless interface will be absent by default, replace it with wireless port.

config > wireless0

SSID: wlan1

Authentication: WPA2-PSK PSK pass phrase 1234567

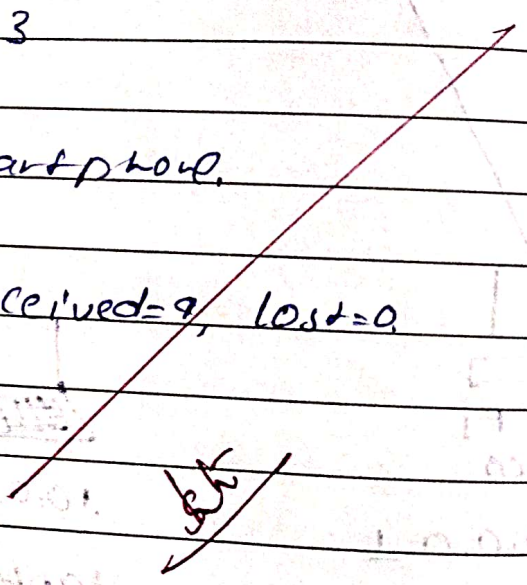
Encryption: AES

IP adress: 10.0.0.3

ping from laptop to smartphone.

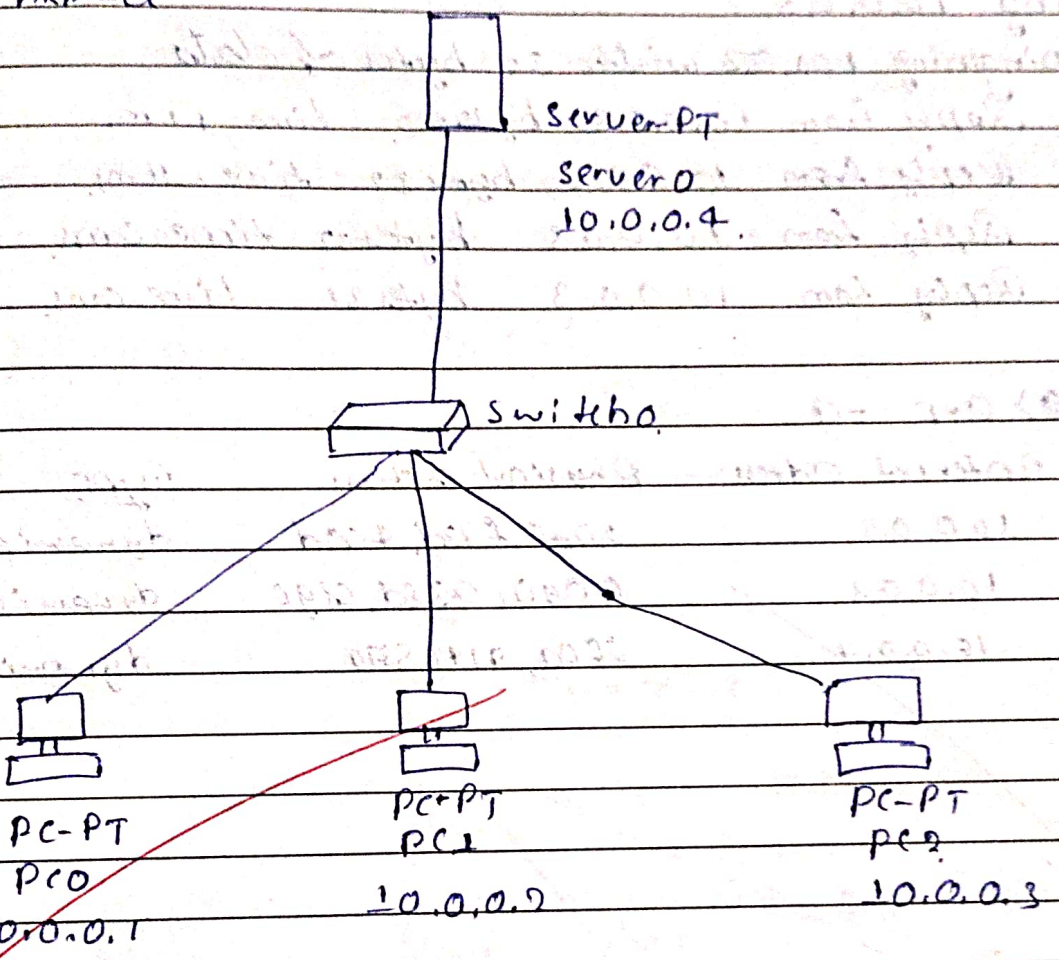
ping 10.0.0.4

packets: sent=4, received=4, lost=0



Aim: Demonstrate the working of adress resolution protocol for communication within a lan.

ARP -a



Observation:

Configure PC's adress as shown in figure and also assign IP adress to server.

Open ARP table of PC0 it will have no content as we haven't sent any PDU

Open command prompt

PC>arp -a

No ARP Entries found

Open simulation mode and start auto capture. When PDU reaches server ARP table is set for PC and server, it will assign mac address as PDU moves.

again take another PDU from PC0 to PC1. It will assign MAC address in ARP Table.
open Command Prompt of PC0
ping 10.0.0.3.
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=11ms
Reply from 10.0.0.3, bytes 32 time=11ms
Reply from 10.0.0.3, bytes 32 time=0ms
Reply from 10.0.0.3 bytes 32 time=0ms

PC0> arp -a

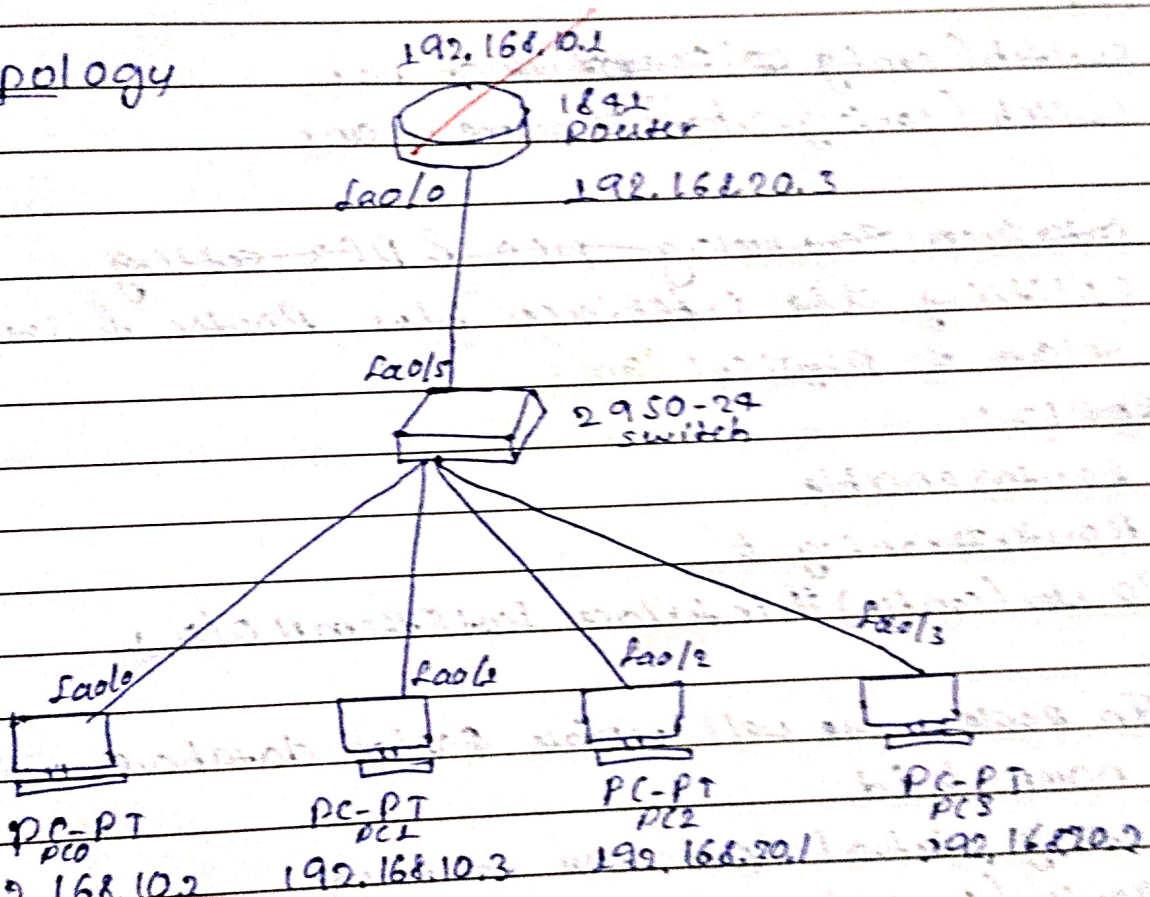
Internet adress	Physical address	Type
10.0.0.2	0000.f73d.4c0d	dynamic
10.0.0.3	0000.a38d.c8gc	dynamic
10.0.0.4	000a.9147.5890	dynamic

Aim: To Create a virtual lan on top of the physical lan and enable communication b/w physical lan and virtual lan.

-> logical network build on top of physical network

ip adress of fourth interface
ip adress 192.168.10.1 255.255.255.0

Topology



-> we can split the above lan into physical lan & virtual lan. left 2 pc's are part of physical lan and right 2 pc's are part of virtual lan for pc0 and pc1 follow the regular procedure

pc0: 192.168.10.2
pc1: 192.168.10.3
Router: 192.168.10.1

→ to make the switch to be part of virtual lan, another vlan to database.

switch > config > vlan database
vlan number: 101
vlan name: valan1

→ Choose the interface blw Router and switch Fa0/1 switch to Trunk.

CLI:
switch(config-~~vlan~~) # ~~vlan~~ valan 101
switch (config-vlan) # name vlan1

~~Interfaces connecting PC2 & PC3 setting~~
Splitting the interface blw Router & switch for vlan & physical lan

CLI:
Router > enable
Router # config t
Router (config) # interface fastEthernet 0/0.1

In Router we will update & vlan database with same name & id.
setting virtual ip address to router
Router (config) # interface fastEthernet 0/0.1
Router (config-subif) # encapsulation dot1q 101
Router (config-subif) # ip adres 192.168.20.3 255.255.255.0

PC2: IP 192.168.20.1
Subnet: 255.255.255.0
Default gateway: 192.168.20.3

PC3: IP address: 192.168.20.2
Subnet: 255.255.255.0
Default gateway: 192.168.20.3

Command Prompt

PC2

PC> Ping 192.168.20.2
Packets Sent=4, Received=4, Lost=0.

PC> ping 192.168.20.3
Packets Sent=4 Received=4, Lost=0

>Ping 192.168.20.2
Pinging 192.168.20.2 with 32 bytes of data.

Rep

Observation: transmission of messages b/w physical Lan and vlan with respect to simple pdu for successful.

3/12/24

Leaky Bucket algorithm.

```
#include<bits/stdc++.h>
using namespace std.

int main(){
    int no_of_queries, storage, output_pkt_size;
    int input_pkt_size, bucket_size, size_left;
    storage=0;

    no_of_queries = 9;
    bucket_size = 10;
    input_pkt_size=4;
    output_pkt_size=1;
    for(int i=0; i<no_of_queries; i++){
        size_left= bucket_size - storage;
        if(input_pkt_size <= size_left){
            storage+=input_pkt_size;
        }
        else{
            printf("packet loss= %d\n", input_pkt_size
        }
        printf("Buffer size=%d out of bucket size=%d
                storage, bucket_size);
        storage-= output_pkt_size;
    }
    return 0;
```

output

Buffer size =4 out of bucket size= 10

Buffer size=7 out of bucket size=10

Buffer size=10 out of bucket size=10

Packet loss=4

Buffer size=9 out of bucket 10.

EXCELLENT
Page No.
Date : / /

CRC algorithm

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

char exor(char a, char b){
  if(a==b)
    return '0';
  else
    return '1';
}
void crc(char data[], char keys[]){
    int datalen= strlen(data);
    int keylen= strlen(key);
    for(int i=0; i<keylen-1; i++)
      data[datalen+i]= '0';
      data[datalen+keylen-1]= '\0';
      int codelen= datalen+keylen-1;

      char temp[20], rem[20];
      for(int i=0; i<keylen; i++)
        rem[i]= data[i];

      for(int j=keylen; j<=codelen; j++){
        for(int i=0; i<keylen; i++)
          temp[i]= rem[i];
        if rem[0]=='0'){
          for(int i=0; i<keylen-1; i++){
            rem[i]= temp[i+1];
          }
```

```
        else {
            for(int i=0; i<keylen-1; i++)
                rem[i]= exor (temp[i+1], key[i+1]);
        }
        if(j!= codelen)
            rem[keylen-1]= data[j];
        else
            rem[keylen-1]= '\0';
    }
    for(int i=0; i<keylen-1; i++)
        data[datalen-i]=rem[i];
    data[codelen]= '\0';
    cout<<" CRC="<<rem<<"\n Dataword="<<data;
}

int main() {
    char key[20], data[20];
    cout<< "Enter data:";
    cin>>data;
    cout<< "Enter key:";
    cin>> key;
    crc(data, key);
    return 0;
}
```

Output:

Enter the data: 10110

Enter the key: 1101

CRC= 110

Dataword= 10011010

Procedure:

① CRC used a n-bit generator polynomial which works as divisor.

Generator = 10101 then n=5

② append n-1 no. of zeros to dataword.

dataword = 110010101

Appended dataword = 110010101 + 0000

= 1100101010000

divide the appended data by generator using binary division

```
        111110111 → quotient
10101 ) 1100101010000
        10101
        011000
         10101
         011011
          10101
          011100
           10101
           010011
            10101
            0011000
              10101
               11010
               10101
                11110
                10101
                 1011
```

24/12/24

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

client.py

```
from socket import *
serverName = "127.0.0.1"
serverport = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverport))
sentence = input("Enter filename")
clientSocket.send(sentence.encode())
fileContents = contentSocket.recv(1024).decode()
print('From Server:', fileContents)
ClientSocket.close()
```

Server.py

```
from Socket import *
serverName= "127.0.0.1"
serverport= 12000
serverSocket= socket(AF_INET, SOCK_STREAM)
serverSocket.bind((ServerName, serverPort))
serverSocket.listen(1)
print("The server is ready to receive")
while (1):
    connectionSocket, addr= serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file= open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
file.close()
connectionSocket.close()
```

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.

ClientUDP.py

```
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
ClientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter the file name")
clientSocket.sendto(bytes(sentence, "utf-8"), (servername,
                                                serverPort))
fileContents, serverAdress = clientSocket.recvfrom(2098)
print('From server:', fileContents)
clientSocket.close()
```

ServerUDP.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, ClientAdress = serverSocket.recvfrom(2098)
    file = open(sentence, "r")
    l = file.read(2098)
    serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
    print("Sent back to Client", l)
file.close()
```