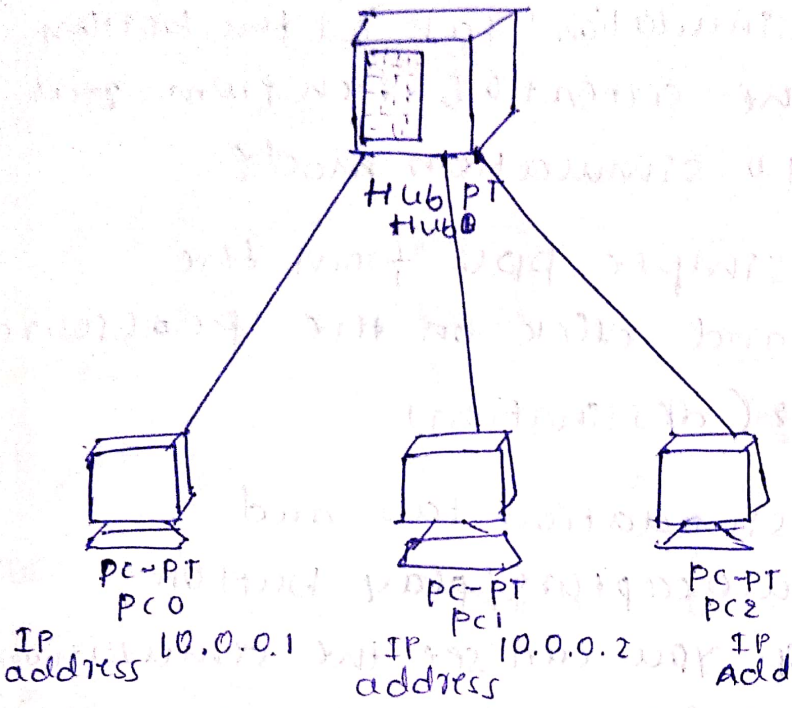
LAB-1 Hub

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

Observation:



port link
port 0 up
port 1 up
port 2 up
port 3 down
port 4 down
port 5 down

i) Opened cisco packet tracer

ii) selected 3 devices from the tab below the window, ~~and~~ selected 3 generic devices

iii) from connecting device, I selected Generic Hub

iv) I ~~choose~~ tap on connection option from the bottom and I selected Automatically choosen connection and connected each device with the Hub

v) Then from the right side I selected simple PDU ~~and~~ from the Icon showed as Add simple PDU and after selecting

E

Assign unique IP Address for each PC for

example

pc0	Address: 10.0.0.1	Subnet mask: 255.0.0.0
pc1	Address: 10.0.0.2	subnet mask: 255.0.0.0
pc2	Address: 10.0.0.3	subnet mask: 255.0.0.0

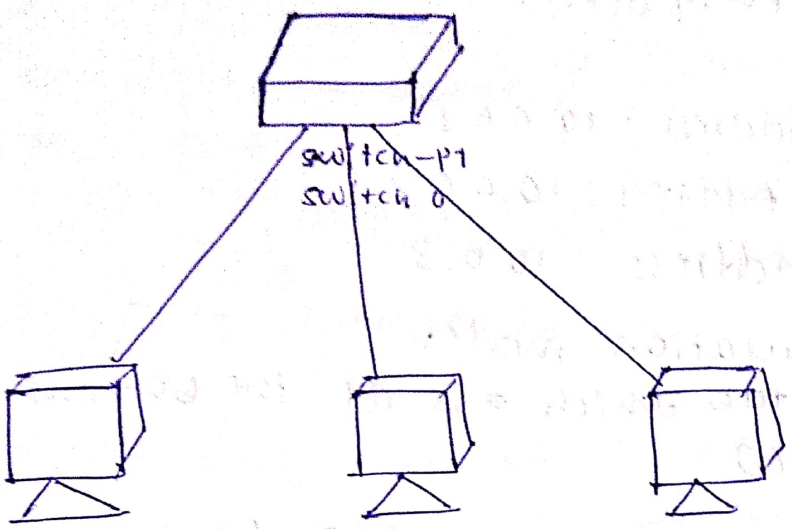
switch to simulation mode:

click on the "simulation" tab at the bottom right of the screen to switch from real-time mode to simulation mode

* after select simple PDU from the right corner and click on the pc0 (source) and after pc2 (destination)
* then go to simulation tab and click on auto capture/play button
* after clicking you can see the simulation of packet transferring from source to destination,
* firstly the packet will goto Hub and then it will go to destination (pc2) you can see (✓) right mark on pc2 and cross (X) mark on pc1

Switch

Observation



Observation:

i) Open cisko packet Tracer: Launch the application

ii) Add devices:

from the bottom left corner select device and click on the blank sheet, and like this select the some more devices (I selected 3 devices)

iii) after that click on the connecting devices and select switch from there (which is available)

Connecting devices:

at the bottom left corner there is one option called connect devices; from there select connecter, choose automatic connecter or you can connect copper through straight" cable

Assign IP addr...
click on each devices to get desktop and select configure tab and set ip address for every device
for example
pc 0 ip Address : 10.0.0.1
pc 1 ip Address : 10.0.0.2
pc 2 ip Address : 10.0.3

* switch to simulation mode:
(click on the tab which is at the bottom right corner)

* after that select simple pdu from the bottom right corner and click the source (pc0) and destination (pc2)

* then go to simulation mode and click on the button called Auto capture, the you can see the simulation

* firstly the packet will go to switch and through switch the packet will go to destination.
you can see this ; there is (✓) mark on the destination pc and (X) mark on the remaining pc

Here you will see how the switch uses its MAC address to intelligently forward the ping (ICMP) packets to the correct destination (pc2), rather than broadcasting to all connected devices like a hub would

Basic commands for IN CLI

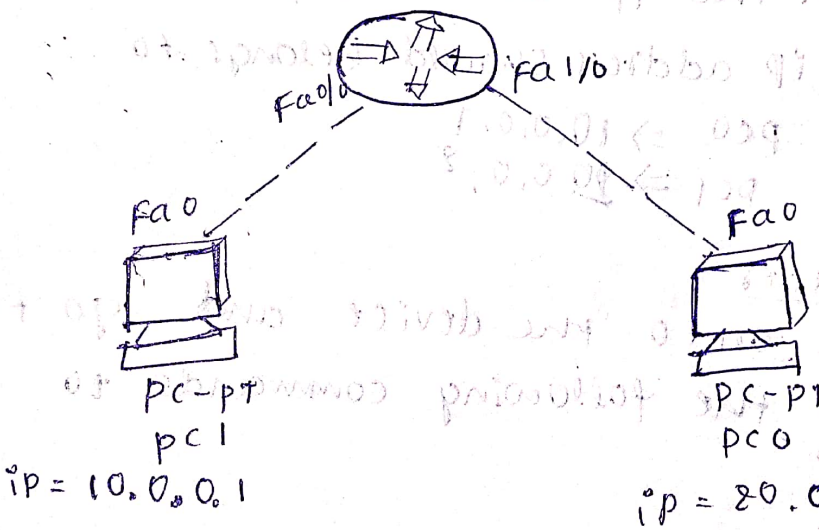
1) Enable
2) #
3) config t
4) interface name (fastethernet 0/0)
5) config ip
6) ip address 10.0.0.1 255.0.0.0
7) no shutdown

TO know ping desktop → command prompt
ping 20.0.0.2

gateway ⇒ Exit Interface

Set gateway config → gateway set 10.0.0.1
20.0.0.2

Topology Router ping ip address of router



Aim: To demonstrate config of ip address to the router and explore ping command.

observation:

* 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 the 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
* and select connecter ("copper-through straight" cable and connect each device with the router

Assign ip Address

click on each device and go to config tab click on the ip addressis for each device (both ip address should belongs to

pc0 => 10.0.0.1
pc1 => 10.0.0.2

config router

click onto the device and go to CLI enter the following commands to config:

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

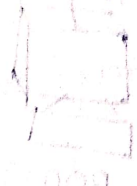
Do the same for pc1 also

2) enable
3) config t
3) interface fastethernet 0/0
4) ip address 20.0.0.1 255.0.0.0
5) 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

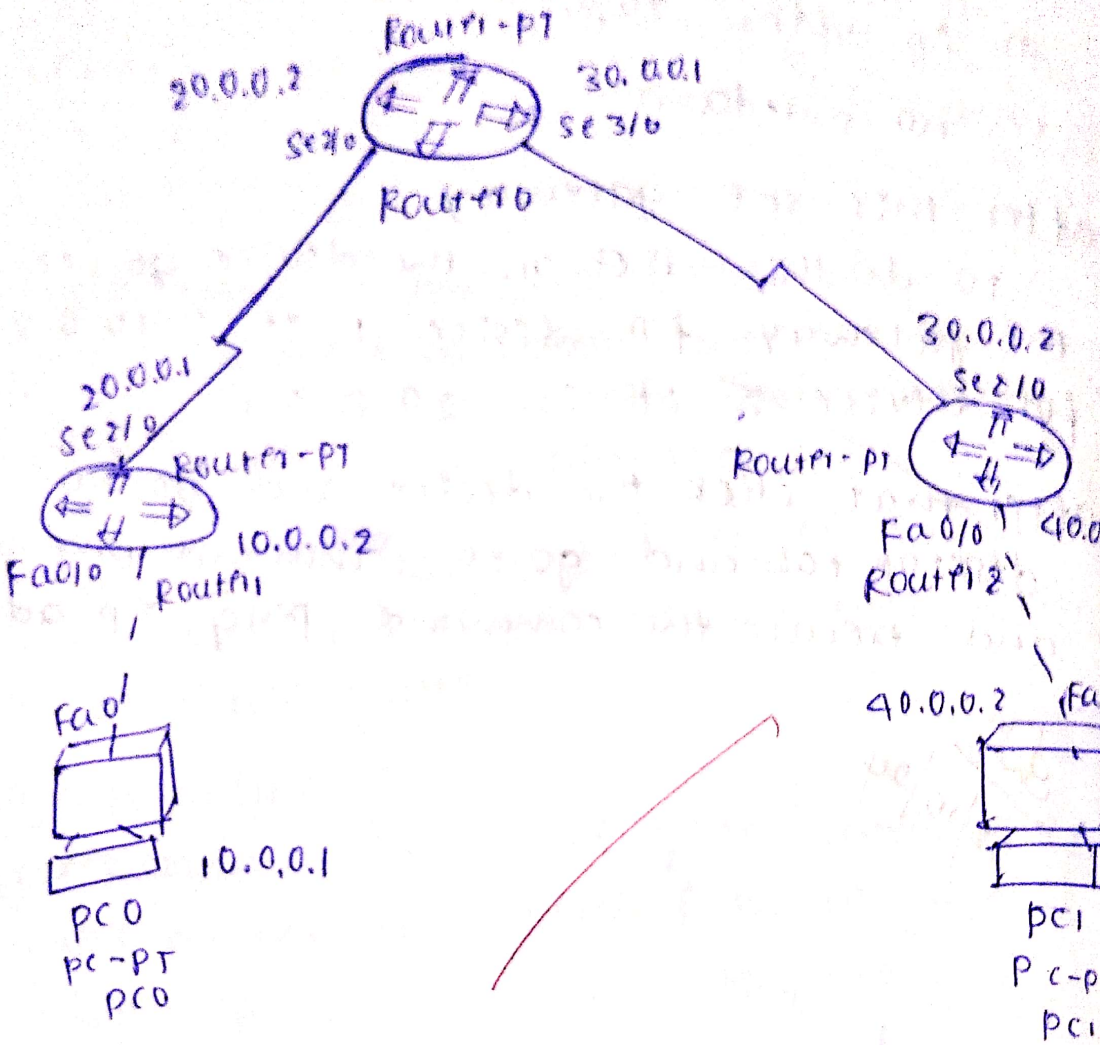
after that click the device and go to desktop tab and go to command prompt and execute the command ping ip address



22/10/24

Experiment 3

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



show ip route ⇒ command to know connections

ip route
destination n/w ip
destination mask
Next stop

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

Aim: to demonstrate the configuration of default route to the routers

procedure:

1. configure the pc0 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 address 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 pc

3) config router0 to connect Router 2 using the following commands

```
Router(config)# interface serial2/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# no shutdown.
```

similarly configure R_2 to connect Router and similarly configure Router 2 to connect router 3 and Router 3 to connect Router 2

4) To check network des router connects to IN CLI type Router> show ip route

5 To add other networks:

```
Router(config)# ip route destination mask
                next hop.
```

* To connect Router

Router(config)# ip route 30.0.0.0 255.0.0.0 20.0.0.2

Router(config)# ip route 40.0.0.0 255.0.0.0 20.0.0.2

Similarly connect other routers with each other.

6) ping a message from PC0 to PC1 through connected prompt.

PC> ping 40.0.0.2

pinging 40.0.0.2 with 32 bytes of data

Reply from 40.0.0.2 bytes 32 time 7ms TTL=125

Ping statistic for 40.0.0.2

Packets: Sent=4 Received=4, Lost=0

Sk
22/10/24

LAB-4 date 22/10/2024

Aim: To configure default and static route to a connection of routers

Router 2;

config t

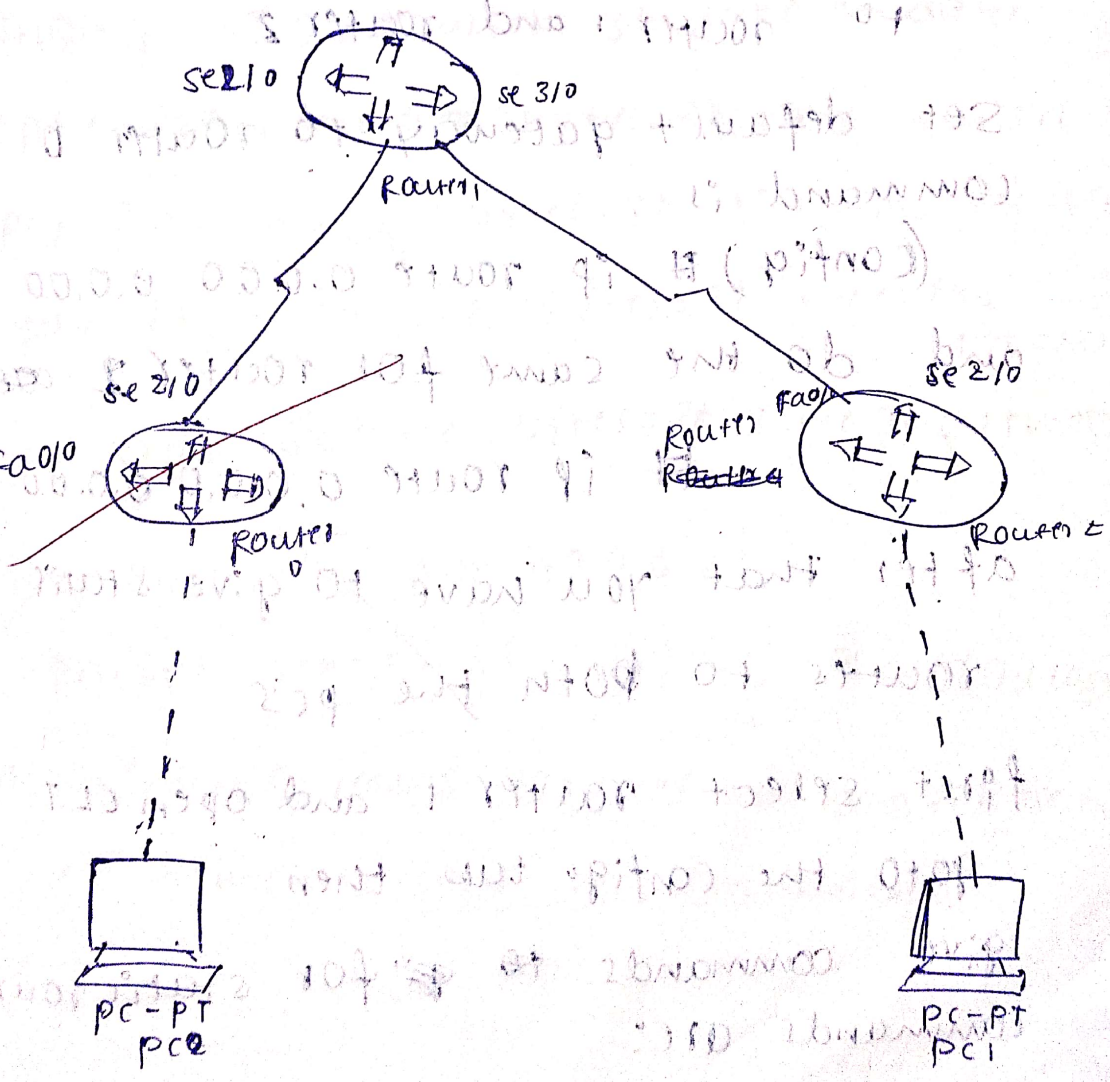
(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.1

R1: ip route 0.0.0.0 0.0.0.0 30.0.0.1

destination

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



procedure

* configure pc0 and pc1 by giving ip address as 10.0.0.1 and 40.0.0.2 respectively
* also give ip address to pc0
* connect router 1 to pc0 by giving ip address of router1 as a gateway of pc0 and do the same for pc1
* for router 0: ip address is 10.0.0.2 and serial 2/0 as 20.0.0.1

for router 1; ip addrs is 40.0.0.1
serial 2/0 as 30.0.0.2

3) after that give ip configurations to router 1 and router 2

set default gateway to router 0
command is

(config) # ip route 0.0.0.0 0.0.0.0 20.0

and do the same for router 2

ip route 0.0.0.0 0.0.0.0 30.0

after that you have to give static routes to both the pc's

first select router 1 and open CLI
go to the config tab then
give commands for static routes
commands are:

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

after that

In CLI give command # show ip route

you can see all the routes of the network

after at last we need to check the connections of destination routers

to do this ~~execute~~ go to the command prompt

give command # Ping destination ip address

here # ping 40.0.0.2

output

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=14ms TTL=253

Reply from 40.0.0.2: bytes=32 time=7ms TTL=253

Reply from 40.0.0.2: bytes=32 time=2ms TTL=253

Reply from 40.0.0.2: bytes=32 time 7ms TTL=253

Ping statistics for 40.0.0.2:

Packet: sent=4, Recieved=4, Lost=0(0% loss)

Approximate round trip times in milli-seconds

minimum= 2ms, maximum=14ms,

Average=7ms

To know IP route

show ip route

S 10.0.0.0/8 [1/0] via 2.0.0.1

C 20.0.0.0/8 is directly connected, Fast

C 30.0.0.0/8 is directly connected, Serial

S 90.0.0.0/8 [1/0] via 30.0.0.2

S* 0.0.0.0/0 [1/0] via 2.0.0.1
[1/0] via 30.0.0.2

SK
23/10/24

LAB-5

Aim: To understand the operation of TELNET by understanding the router placed in the server room from a PC in IT office

commands

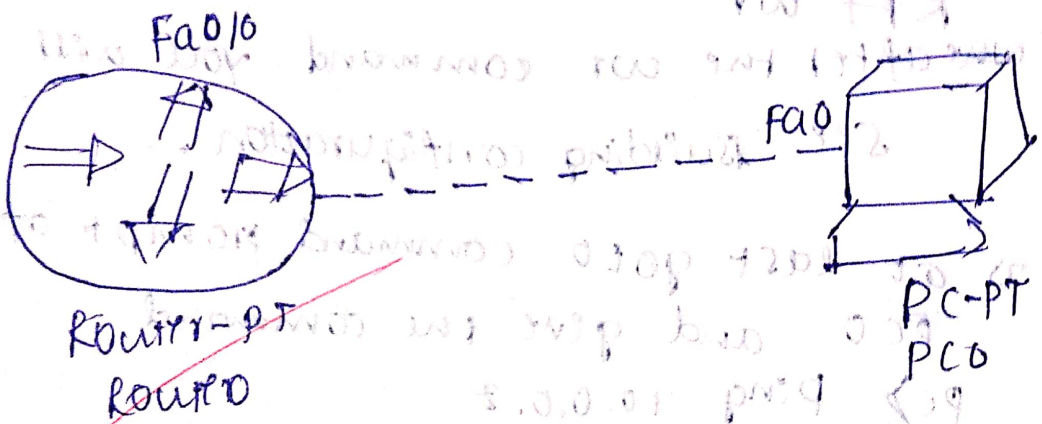
Router(config) # hostname R1

R1(config) # enable secret P0

R1(config-line) # login

R1(config-line) # password P1

R1(config-line) # exit



procedure:

1) select one PC from the devices
2) select one router from the connecting devices
3) config PC with ip address
PC0 10.0.0.1 255.0.0.0
4) config router with ip addrs 10.0.0.2 255.0.0.0
5) enable the connection and give the gateway to router and make sure the connection is successful.

6) after that go to CLI of router and give the below commands in config tab

R1(config-if)# hostname R1

R1 (config)# enable secret P0

R1 (config-line) vty 0 5

R1 (config-line) #login

R1 (config-line) # password P1

R1 (config-line) #exit

R1 (config) # exit

R1 #

R1# wr

after the wr command you will see Building configuration...

7) at last goto command prompt of PC0 and give the command

PC> ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

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=4ms TTL=255

Reply from 10.0.0.2: bytes=32 time=4ms TTL=255

Ping statistics for 10.0.0.2

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

Approximate round trip times in milli-seconds

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

Trying 10.0.0.2 ... Open

User Access Verification

Password: pi

R1>

[Connection to 10.0.0.2 closed by foreign host]

A) AIM: To configure ip addresses of the host using DHCP server present within the line

B) AIM: To configure ip addresses of the host using DHCP server present in the different Line

DHCP => Dynamic host configuration protocol

⇒ assign IP address to server manually 10.0.0.1

-> configure router assign IP address to router 10.0.0.2

→ set gateway for server (routers interface address)

→ configure DHCP protocol

server → service → DHCP → turn on service

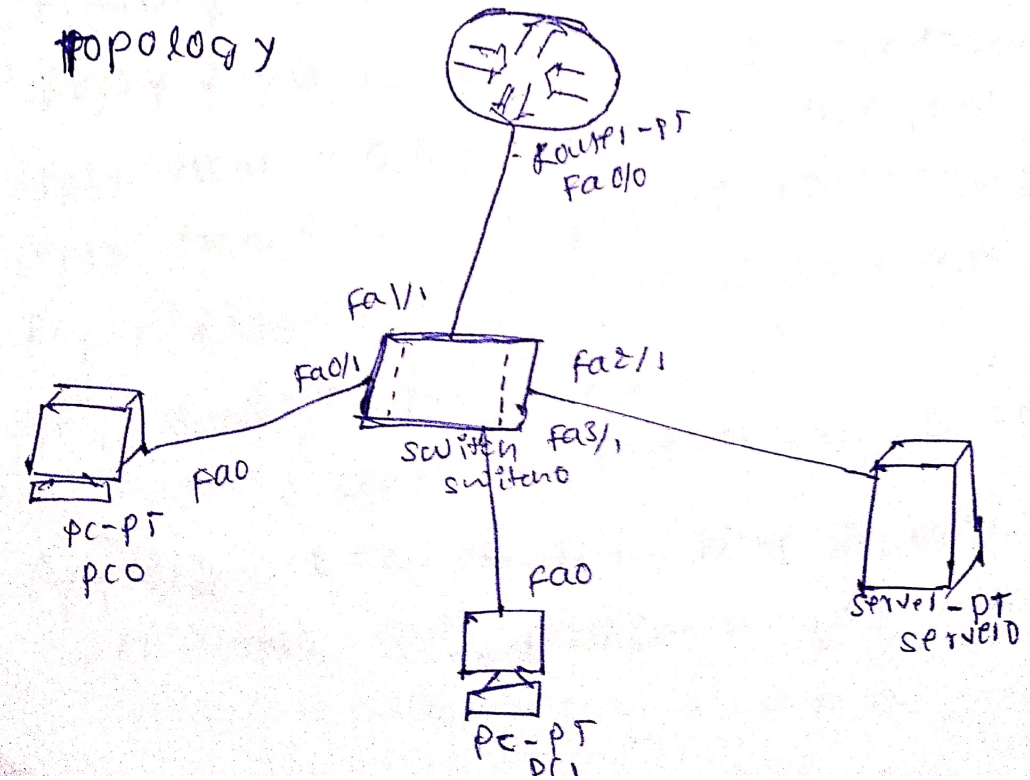
poolname: serverpool

Default: 10.0.0.2

DNS: 10.0.0.1

→ Dynamically configure the end devices by selecting DHCP in configuration tab of devices

Topology



check ping ⇒ command prompt

Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Replay from 10.0.0.4 : bytes = 32 time=0ms TTL=128

Replay from 10.0.0.4 : bytes = 32 time= 0ms TTL=128

Reply from 10.0.0.4 : bytes = 32 time=0ms TTL=128

Reply from 10.0.0.4 : bytes = 32 time = 2ms TTL=128

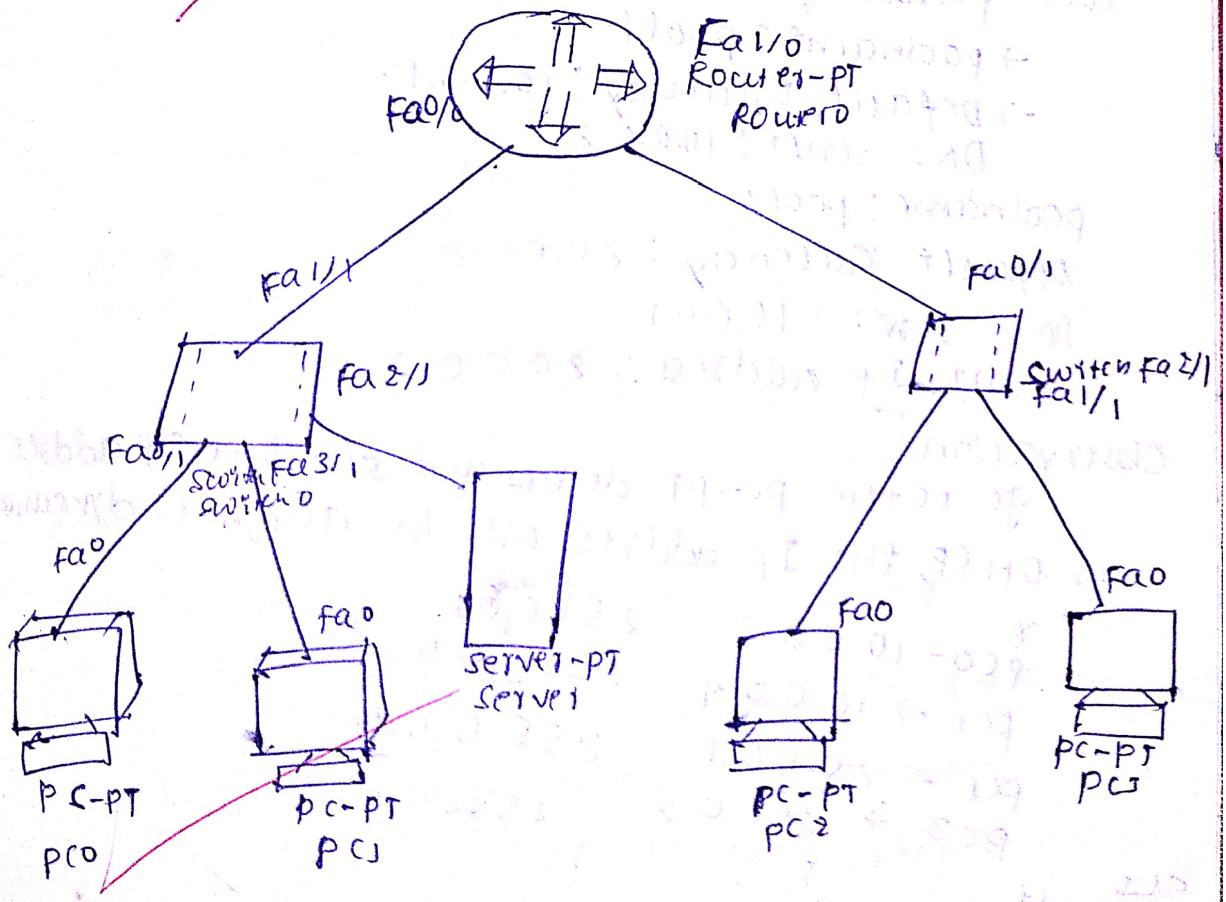
ping statistics for 10.0.0.4

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

Approximate round trip times in milli-seconds:

minimum = 0ms, maximum = 2ms, Average = 0ms

B)



configure server
IP address : 10.0.0.1
poolname : server pool
default : 10.0.0.2
DNS : 0.0.0.0

write 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

4) Now, go the server in server-PT device and go to the DHCP and turn on the service & give the following details

→ poolname : pool1
→ Default Gateway : 10.0.0.1
DNS server : 10.0.0.2

poolname : pool2
Default Gateway : 20.0.0.2
DNS server : 10.0.0.1
Start IP address : 20.0.0.2

Observation:

go to the PC-PT devices and set the IP addr as DHCP, the Ip address will be assigned dynam

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
enable
Router #
Router # config t
Router(config-t) # interface FastEthernet0/0
Router(cont-if) # ip helper 10.0.0.1
Router(config-if)# no shutdown

WT
18/11

AIM: To configure DNS server, to demonstrate the mapping of ip addresses and domain names

Router (config) # exit

Router (config-t) # interface fast Ethernet 1/0

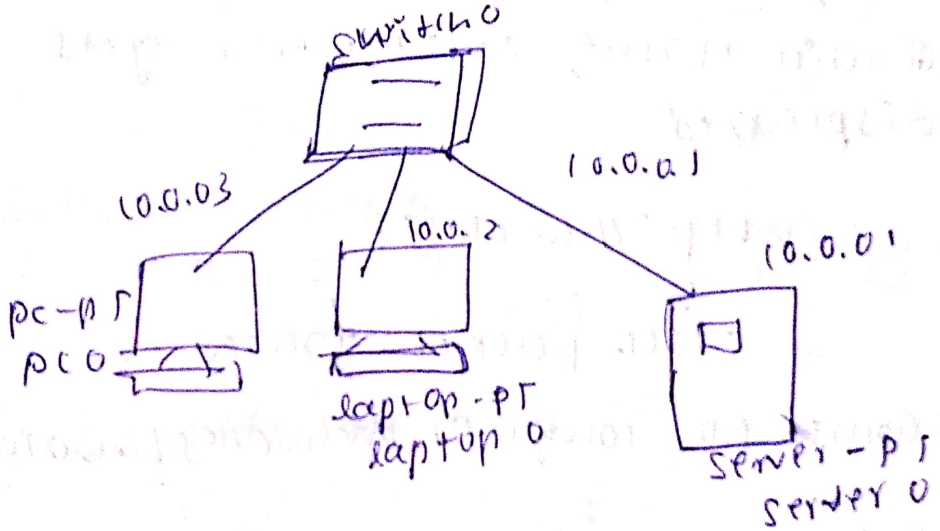
Router (config-t) ip helper 10.0.0.1

no shut down

exit

LAB program - 8

AIM: TO configure DNS server, to demonstrate the mapping of ip address and domain names



configuration

connection: copper straight through

At server 0:

In fastEthernet-0: IP address 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

Address: 10.0.0.1

In HTTP:

- ment.

In PC0: click on DHCP, the IP address get configured automatically

- In laptop: on clicking DHCP, the IP address get configured automatically

Thus all the configuration are done

Destination:

- using any end device, click on "web browser" in desktop menu
- There, one 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

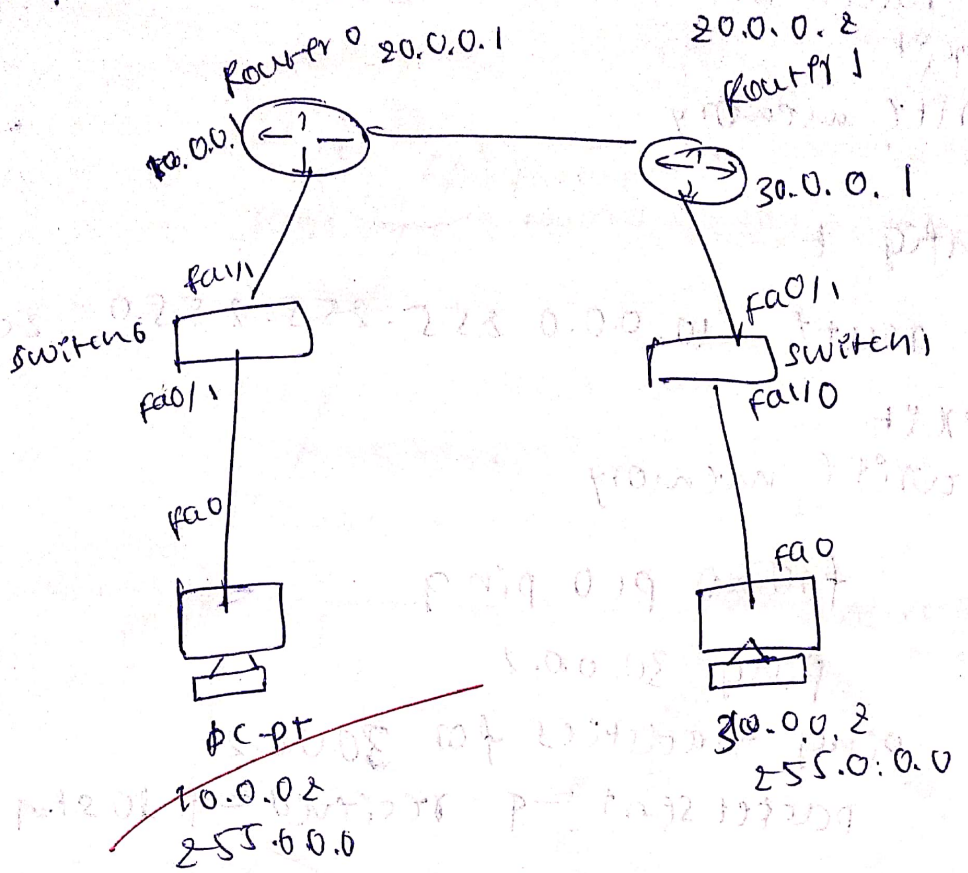
copy rights

Image page

Image

Aim : configure RIP routing protocol

Topology



configure router 0

fa 0/0 : 10.0.0.1 255.255.255.0

S@ 1/0 : 20.0.0.1 255.255.255.

CLI : Interface serial 2/0

ip address 20.0.0.1 255.255.25

clock rate 64000

no shutdown

exit

write memory

config router router 1:

fa0/0 30.0.0.1 255.255.255.0

serial 20.0.0.2 255.255.

enable routing
config t
ip route 30.0.0.0 255.255.255.0 20.00.2
exit
write memory

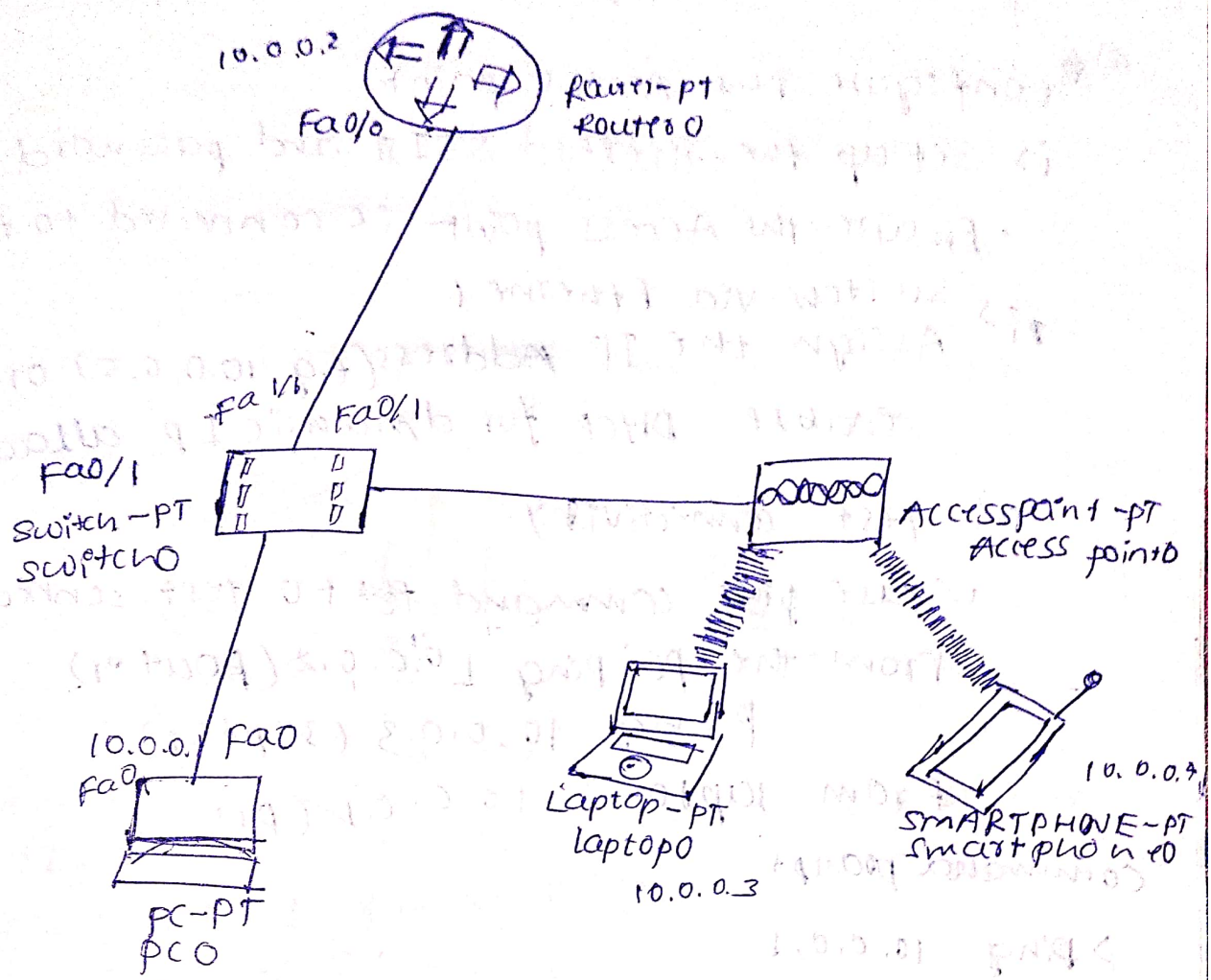
config t
ip route 10.0.0.0 255.255.255.0 200.
exit
write memory

from pc0 ping
ping 30.0.0.2
ping statistics for 30.0.0.2
packet sent = 4 received = 4 lost = 0

18/11

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

TOPOLOGY



Observation

Select router, switch, wirelessdevice, laptop and pc and smartphone ~~for configure~~

1) configure the Router: Ip address: 10.0.0.2

2) configure PC0 ~~router~~ Ip address: 10.0.0.1

Default Gateway: 10.0.0.2

3. configure the laptop and smartphone

i) Ensure they are connected to the Access point wirelessly

ii) Assign IP Addresses manually or set them to receive ~~add~~ addresses via DHCP if configured.

Example for laptop:

Ip Address: 10.0.0.3

Subnet mask: 255.255.255.0

Default gateway: 10.0.2

4) Configure the Access point

i) set up the wireless SSID and password

- Ensure the Access point is connected to the switch via Ethernet

ii) Assign the IP Address (e.g 10.0.0.5) or enable DHCP for dynamic IP allocation

5) Test connectivity

i) use ping command to test connectivity

From the pc: ping 10.0.0.2 (Router)

ping 10.0.0.3 (laptop)

From laptop: 10.0.0.1 (pc)

command prompt

> ping 10.0.0.1

pinging 10.0.0.1 with 32 bytes of data:

Replying from 10.0.0.1: bytes = 32 time=2ms TTL

Replying from 10.0.0.1: bytes = 32 time = 8ms TTL

Replying from 10.0.0.1: bytes = 32 time = 13ms TTL

Reply from 10.0.0.1: bytes = 32 time, TTL=128

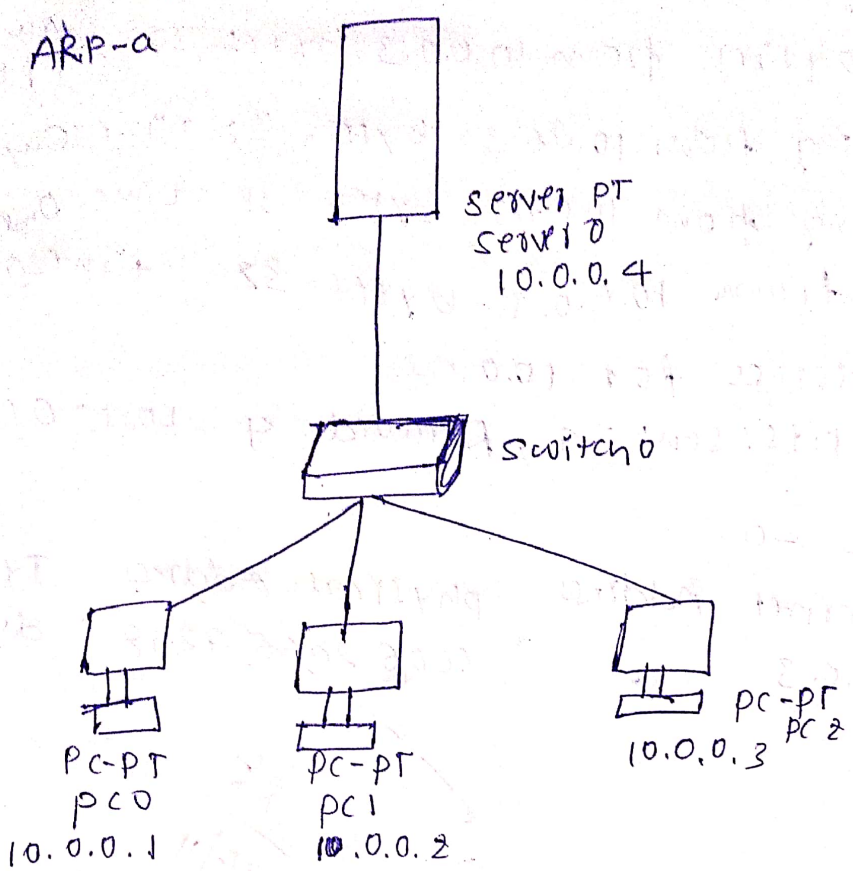
ping statistics for 10.0.0.1:

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

SK 26/11

Aim: To demonstrate the working of Address resolution protocol for communication within a LAN

ARP-a



Observation:

configure PC's address are shown in figure

configure ip for each pc

PC 0 => 10.0.0.1

PC 1 => 10.0.0.2

PC 2 => 10.0.0.3 server 0 => 10.0.0.4

To check ARP table execute command arp -a in command prompt after pinging.

you can see the arp table.

and also configure server with ip address 10.0.0.4

* select simple PDU and choose source and destination, you can also capture to see the results.

then click on the ☑ mark on the source devices then goto inbound device and you can see ARP table

Commands in cmd

ping 10.0.0.3

Pinging with 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

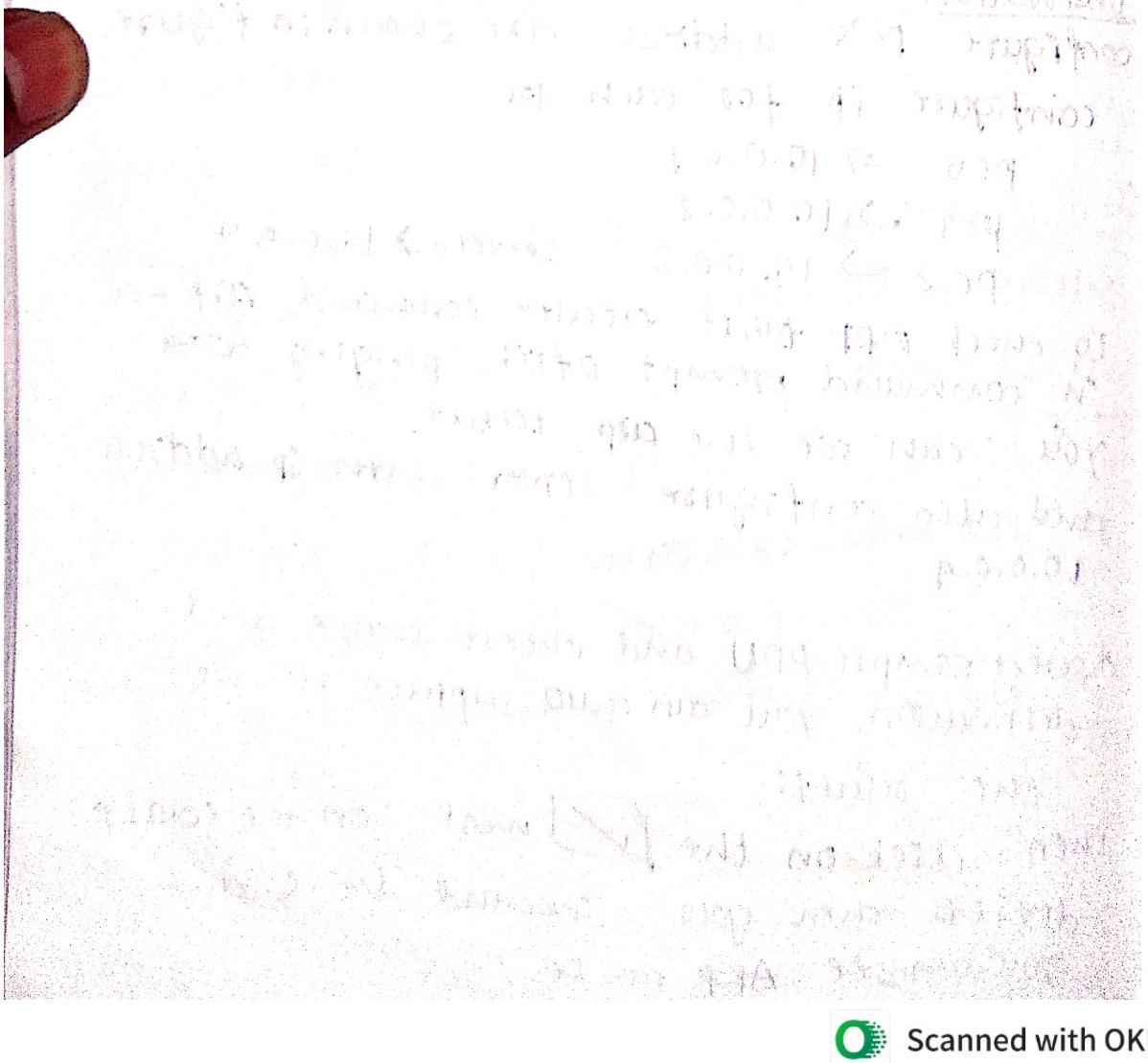
Reply from 10.0.0.3 : bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3

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

PC> arp -a

Internet Address	physical Address	Type
10.0.0.3	0006.2a66.9298	dynamic



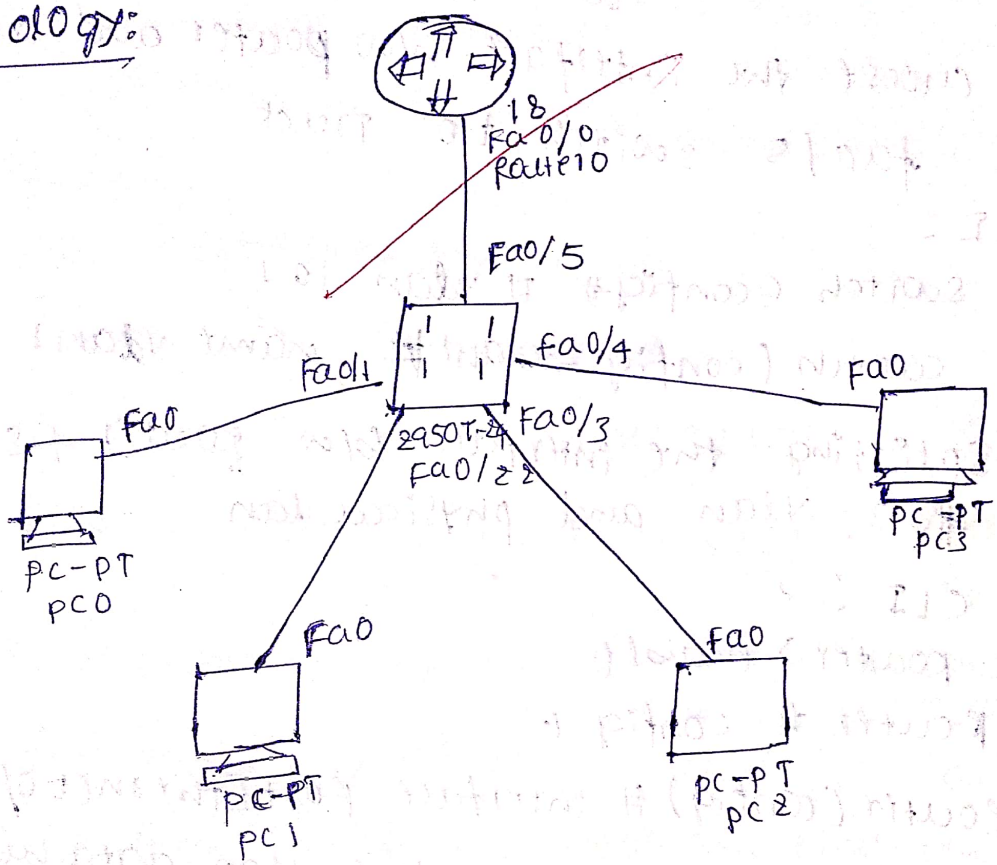
Virtual LAN

Aim: To create a virtual LAN on top of the physical lan and enable communication between physical LAN and virtual LAN

⇒ ~~pkt~~ logical network built on top of physical layer

encapsulation dot1q 101

Topology:



1) First do the topology as shown above
2) we can devide this into 2 layers, 1 is physical layer and another one is virtual layer
3) First configure PC0 with ip 192.168.10.1 and PC1 with ip 192.168.10.2
4) ~~Give interface of to router~~ config of router with ip 192.168.10.3

the pc0 pc1 together forms the physical layer and give the interface to each pc that is fastEthernet0/0 192.168.10.3

6) To make the switch to be part of virtual lan add another vlan to database

switch > config > valan database
valan number : 101
valan name : vlan1

choose the interface b/w router and switch
fa0/5 switch to track

CLI :

switch (config-t # vlan 101
switch (config-vlan) # name vlan1

Splitting the interface b/w Router & switch to vlan and physical lan

CLI :

Router > enable
Router # config t

Router (config) # interface fastEthernet0/0.1

In Router as well update vlan data base with same name & id

setting virtual ip address to router

Router (config) # interface fastEthernet0/0.1
Router (config-subif) # encapsulation dot1q 101
Router (config-subif) # ip address 192.168.20.3 255.255.255.0

pc2 ip 192.168.10.1
subnet : 255.255.255.0
default gateway : 192.168.20.3

pc 3: 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

Observation

command prompt >

pc> ping 192.168.20.3

Pinging 192.168.20.3 with 32 bytes of data

Repeying from 192.168.20.3: bytes=32 time=0ms TTL=255

Reply from 192.168.20.3 bytes=32 time=0ms TTL=255

Reply from 192.168.20.3: bytes=32 time=0ms TTL=255

Reply from 192.168.20.3: bytes=32 time=0ms TTL=255

Observation: transmission of messages between physical lan and vlan with respect to simple pdu for successful

3/12/24

CRC

```
#include<iostream>
#include<vector>
#include<string>
using namespace std;
uint16_t calculateCRC(const vector<uint8_t>& data,
        uint16_t polynomial = 0x1021,
        uint16_t initialCRC = 0xFFFF) {
    uint16_t crc = initialCRC;
    for (uint8_t byte:data){
        crc ^= (byte << 8);
        for(int i=0; i<8; i++){
            if (crc & 0x8000){
                crc = ((crc<<1)^polynomial);
            else{
                crc<<=1;
            }
            crc &= 0xFFFF;
        }
    }
    return crc;
}
vector<uint8_t> sender(const string& inputdata){
    vector<uint8_t> data(inputdata.begin(),
                         inputdata.end());
    uint16_t checksum = calculateCRC(data);
    cout<<"sender_data"<<inputdata<<"checksum
    <<hex<<uppercase<<checksum<<endl;
    data.push_back(checksum>>8);
    data.push_back(checksum & 0xFF);
    return data;
}
```

```
bool receiver(const vector<uint8_t>& receivedData) {
    if (receivedData.size() < 2) {
        cout << "Receiver-Data is too short to
                contain a checksum" << endl;
        return false;
    }
    vector<uint8_t> data(receivedData.begin(), receivedData.
                end() - 2);
    uint16_t receivedChecksum = (receivedData[receivedData.
                size() - 2] << 8) |
                receivedData[receivedData.size() - 1];
    uint16_t calculatedChecksum = calculateChecksum(data);
    return receivedChecksum == calculatedChecksum;
}
int main() {
    string inputData = "1234567";
    vector<uint8_t> transmittedData = sender(inputData);
    if (receiver(transmittedData)) {
        cout << "Receiver-Data is valid (No errors)" << endl;
    } else {
        cout << "Receiver-Data is invalid (Errors detected)"
                << endl;
    }
    return 0;
}
```

Output

Sender-Data: 1234567, Checksum: 29B1

Receiver - Received checksum: 29B1, Calculated checksum: 29B1

Receiver-Data is valid (No errors detected)

17/12/24

LEAky BUCKET

```
#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 = 4;
    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\n",
        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 size = 10
```

Observation for CRC

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

Generator = 10101 then n=5

2 Append n-1 number of zeros to the data word

data word = 11001010

Appended data word = 11001010 + 0000 =

110010100000

3) Divide the appended data word by the generator by using binary division

we get remainder 1011

4) The remainder is n-1 bit CRC code

n-1 bit CRC code = 1011

5) Replace n-1 zeros in data word

final data word = 11001010 + 1011

= 110010101011

shift

17/2/24

1) 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.

SOLUTION: 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 file name")
clientSocket.send (sentence.encode())
fileContents = clientsocket.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()
```

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

SOLUTION:

clientUDP.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.recv
    from(2048)
print("From server:", fileContents)
clientSocket.close()
```

serverUDP.Py

```
from socket import *
serverPort = 12000
serverSocket.bind(("127.0.0.1", serverPort))
print("the server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom
                                (2048)
    file = open(sentence, "r")
    l = file.read(2048)
    serverSocket.sendto(bytes(l, "utf-8"),
                        clientAddress)
    print("sent back to client", l)
    file.close()
```