Program 1

LAB WEEK 1

To demonstrate the transmission of a simple PDU between 2 devices connected using a Hub and a Switch.

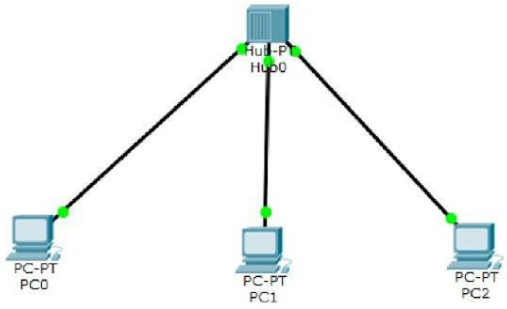


Figure 1: Using Hub

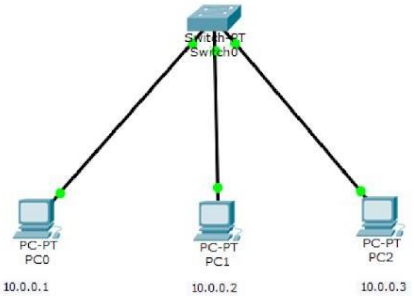
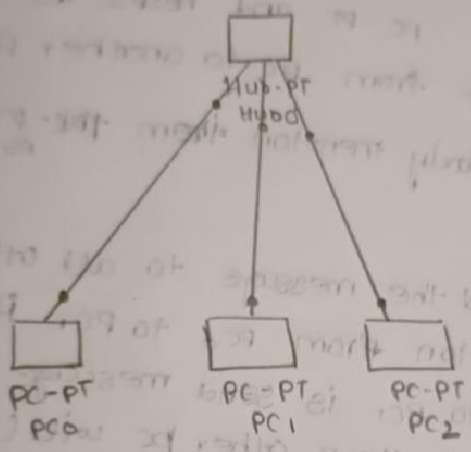


Figure 2: Using Switch

11/10/24 CN LAB

Aim: To demonstrate the transmission of a single pdu between 3 devices connected using a hub and a switch.

Topology: Star Topology



Configuration:

1. First we select the three PC-PT PC0, PC-PT PC1 and PC-PT PC2

2. After we seleded one hub and placed above.

3. And we connected through Automatically choose connection Type from each pc to Hub.

4. From each pc to Hubs.

5. And we select the PC0 and assing the IP adress to pc0 IP Address 10.0.0.3 and click subnet Mask 255.0.0.0 and close the IP address.
and we place the curser on the pc PT can show that pc-PT is connected and

5. And given PDU to the PC-PT $_{PC0}$ and PC-PT $_{PC3}$ and Stimulator mode after connected and click on auto capture/play to observe the transmission of single PDU blw two devices is connected.

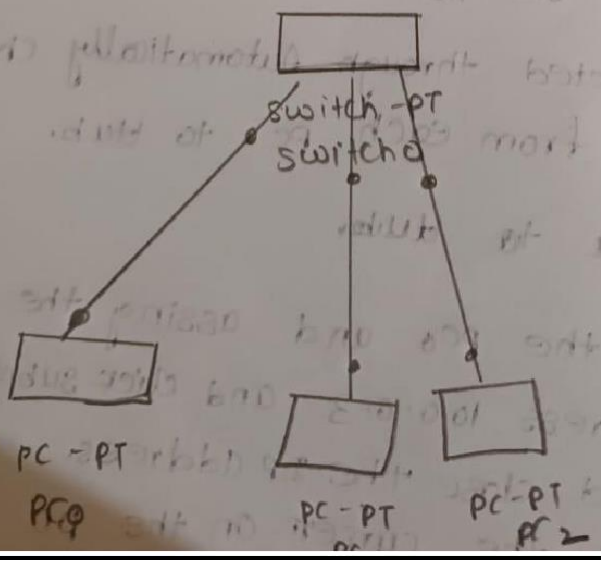
OBservation:

Hub is a unintelligent devices that helps us to connect multiple PC-PT and helps the passes the message from PC to another PC.

And if we practicabaly mention them PC-PT $_{PD}$ and PC-PT $_{P3}$ +

The Hub will send the message to all other PC but we mention from PC_0 to PC_1 it shows that PC_0 to PC_1 is send message tick(V) if will show and then other pc will (X) mark. telling that the message is not sent other PC.

Topology: Star topology:



Observation:

Instead As we see before we used the ~~switch~~ Hub

And now we used switch here the configuration will be same

But here when we used switch it send the message to the PC_0 to PC_2 not to other PC.

8/10/24

Program 2

LAB WEEK 2

i) To demonstrate configuration of IP addresses to the Routers and explore ping command.

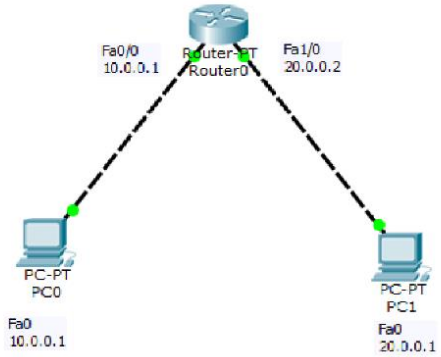


Figure 1: Topology

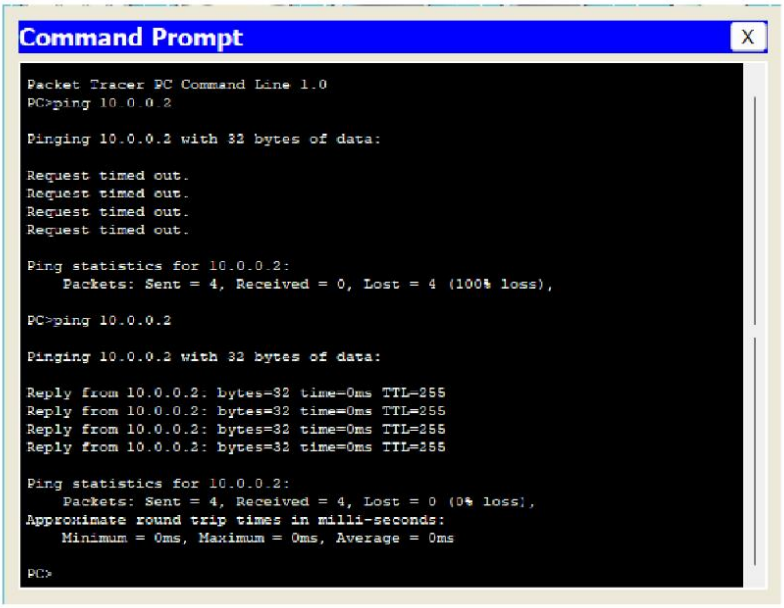


Figure 2: ping command output

LAB -2

8/10/24

Experiment - 2

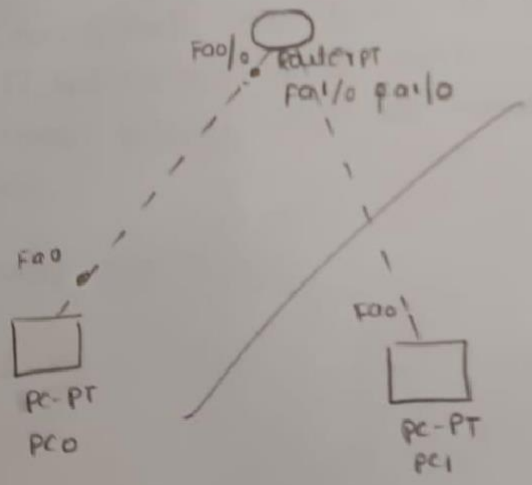
To demonstrate configuration of ip address to the routers and explore the ping command.

Router >erable
Router #
1) Enter
2) Enable
3) config t
4) inter-face -fastetherrd0/0
5) ip address 10.0.02.255.0.0.0
6) no shutdown.

Router >enable
Router # config t
Enter configuration commands, one per line
End with CNTL/z
Router (config)# interface ~~faste~~ fastethernet 0/0
Router (config-if)# ip address 10.0.0.2 255.0.0.0

Configuration:

Topology:



Configuration:

1) Select the Generic PC-PT PC0 and PC1
2) And take one Generic router and the PC-PT PC0 and Router PT through Automatically choose connection Type through connection.
3) Then Give the ipaddress for PC-PT is 10.0.0.1
PC0

routers

line

with ENTIRE

0.2 225.0.0.0

Pco

tion

and give the ip address different for PC PT is PC1
20.0.0.2

4) And Select Router select CLI (command line interface) then First press Enter and write commands.

1) Enter
2) Enable
3) config t
4) interface fastethernet 0/0
5) ip address Here the ip address for fasternet 0/0 should the different number at last digit of PC-PT means here 10.0.0.2
6) shut down. then red signal turn to Green
after that give the exit.
start doing 3)command for Fastethernet 1/0.
10.0.0.2. same steps.

5) And after select PC and select config then give the Gateway address as given in the Fa0/0. 10.0.0.2

6) Same for PC-PT PC1 20.0.0.2.

7) Then go to desktop and select command prompt

write ping write destination address of 20.0.0.2

ping statistics for 20.0.0.2:

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

observation:

After typing the ping command in the command prompt

ping 20.0.0.2

pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2 : byte = 32 time = 0ms TTL = 255
Reply from 20.0.0.2 : byte = 32 time = 0ms TTL = 255
Reply from 20.0.0.2 : byte = 32 time = 0ms TTL = 255
Reply from 20.0.0.2 : byte = 32 time = 0ms TTL = 255

Ping statistics for 20.0.0.2:

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

Approximate round trip times in milli-seconds:

Minimum = 0ms, maximum = 0ms, Average = 0ms.

Here the from source to destined it received.

Program 3

LAB WEEK 3

To demonstrate configuration of default and static routes through a connection of routers.

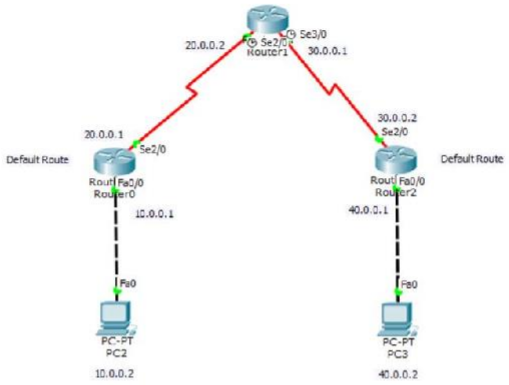


Figure 1: Topology

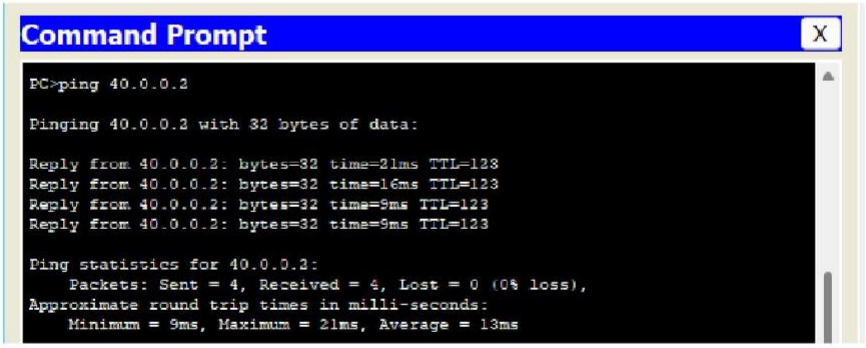
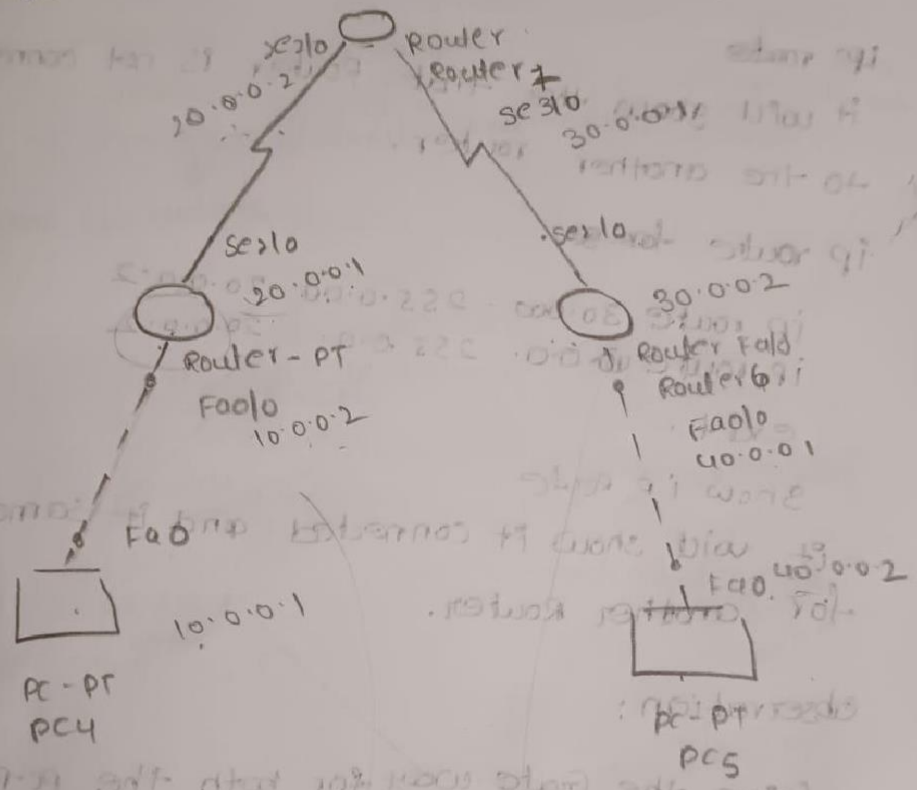


Figure 2: ping command output

Experiment - 3

To demonstrate the configuration of default routes to the router.

Topology:



~~Configuration:~~

~~1) select the Generic PC-PT pc0 and PC1~~

To the Router-PT → Router6 Fa0/0

ip address route
destination network ip
destination subnet mask
Next hub.

} → for reaching the unknown nw

this command is used for.

1) Select PC-PT and Router and connected the PC-PT and Router

1) connect the PC-PT and Router
2) And go to the Router CLI and show ip route

ip route
it will show the that Router is not connected to the another router.

ip route for se

ip route 30.000 . 255.0.00 20.0.0.2
ip route 40.0.0. 255.0.0. 20.0.0.2
exit.
show ip route.
it will show it connected and it same for another Router.

observation:

Give the Gate way for both the PC-PT And command pronpt.

ping 40.0.0.2

pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=9ms TTL=
Reply from 40.0.0.2: bytes=32 time=9ms TTL=
Reply from 40.0.0.2: bytes=32 time=9ms TTL=
Reply from 40.0.0.2: bytes=32 time=5ms TTL=

ping statistics for 40.0.0.2:

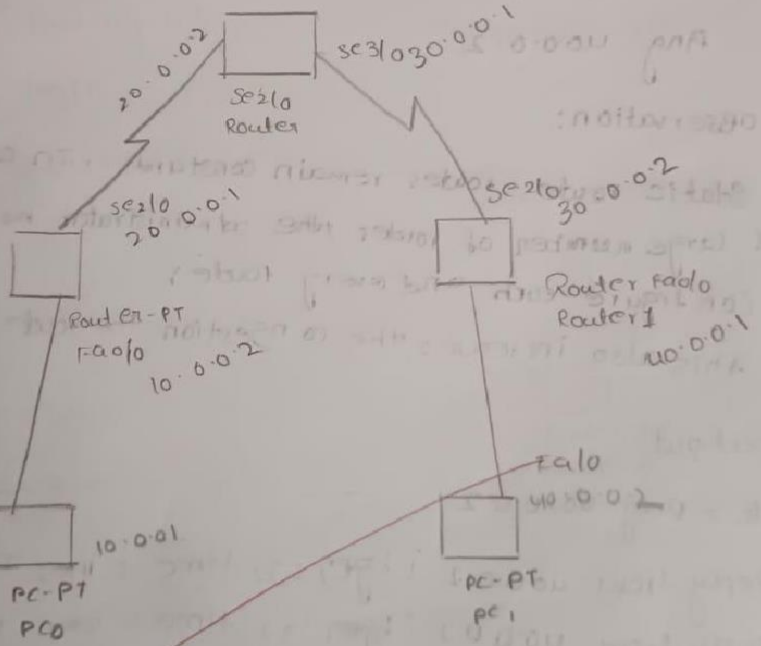
packets: sent = 4, received = 4, lost = 0 (0% loss)

15/1.0/24

22/10/24

Experiment-4:

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



Configuration:

Static → routes remain constant.

default →

for Router 1

Enable

Config t.

ip route 0.000 0.0.0.0 20.0.0.2

and for the Router 2:

Enable
config t
ip route 0.0.0.0 0.0.0.0 30.0.0.1

And for the middle Router (top router)

write the two command 20.0.0.1
ip route 10.0.0.0 255.0.0.0 ~~30.0.0.2~~
ip route 40.0.0.0 255.0.0.0 ~~20.0.0.1~~ 30.0.0.2

open desktop command prompt of pc0 and type

Ang 40.0.0.2

OBservation:

Static route - routes remain constant. In case of a large number of routes the administrator has to configure each and every route; this also increases the congestion the route,

output:

PC > ping 40.0.0.2

Reply from 40.0.0.2 : bytes : 32 time = 11ms, TTL = 123
Reply from 40.0.0.2 : bytes : 32 time = 10ms TTL = 123
Reply from 40.0.0.2 : bytes 32 time = 13ms TTL = 123
Reply from 40.0.0.2 : bytes : 32 time = 16ms TTL = 123

Ping Statistics for 40.0.0.2

Packets: Sent = 4 Received = 4 Lost = 0 (0% loss),
Approximate round trip time in milliseconds:
Minimum = 10ms, Maximum = 18ms, Average = 9ms

Program 4

LAB WEEK 4

i) To understand the operation of TELNET by accessing their router placed in the server room from a PC in IT office.

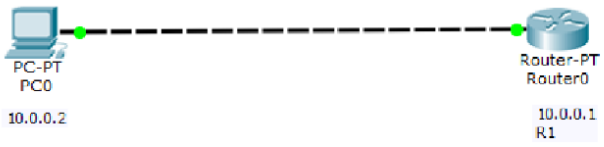


Figure 1: Topology

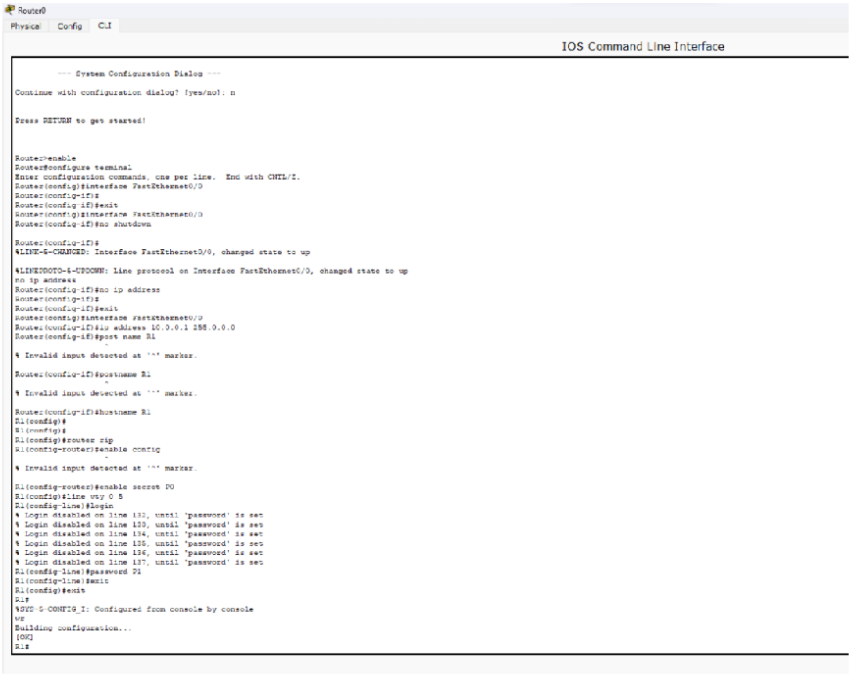


Figure 2: Router CLI

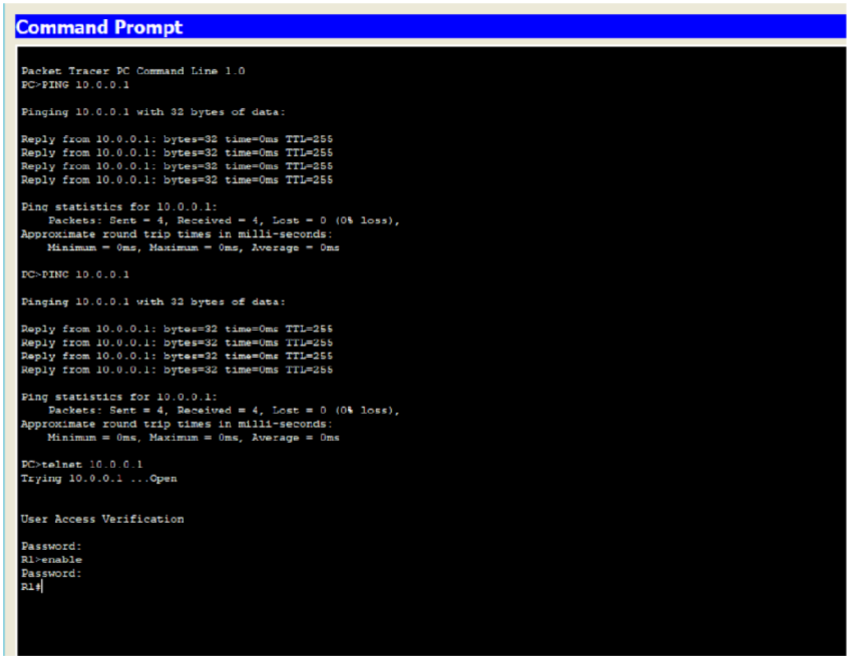


Figure 3: PC Command Prompt

Experiment : 05

AIM: To understand the operation of TELNET by accessing the router placed in the server room from a PC in the office.

After configuration

exit

type this commands.

hostname R1

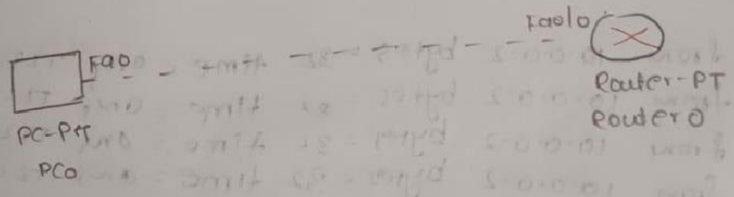
R1(config)# enable secret P0

line vty 0 5

login

password P1

Topology:



Configuration:

select PC and Router And connect these two PC and Router through Automatically Choose connection type

And select PC and set ip address 10.0.0.1

And select Router and give configuration 10.0.0.2

Select PC and give the gate way.

Select Router and type exit

type the commands.

hostname R1

R1(config)# enable secret P0 → assign the password.

line vty 0 5 → this line do here is to enter from user to previlage mode. ↓ mean Router#

login

password P1

then type two times exit ~~exit~~

exit.

R1# wr → write memory is to save change in the memory we have done.

Building configuration

[OK]

R1#

Output:

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 = 0ms, TTL = 255
Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 255

Ping statics for 10.0.0.2:

Packets: sent = 4, Received = 4, lost = 0 (0% loss),

Approximate round trip times in milli seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC > telnet 10.0.0.2 → The PC is for remotely access the telnet

password: P1

R1> enable

password: P0

R1#

12/11/24

Aim: a) To

server pr

b) To con

server

Configura

1) conne

2) to

3) I n

program 5

LAB WEEK 5

a) To configure IP addresses of the host using DHCP server present within the LAN.

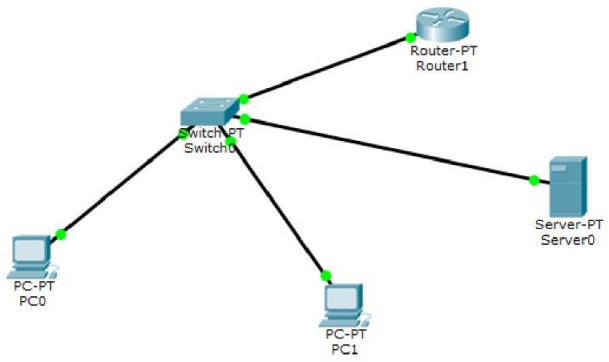


Figure 1: Topology

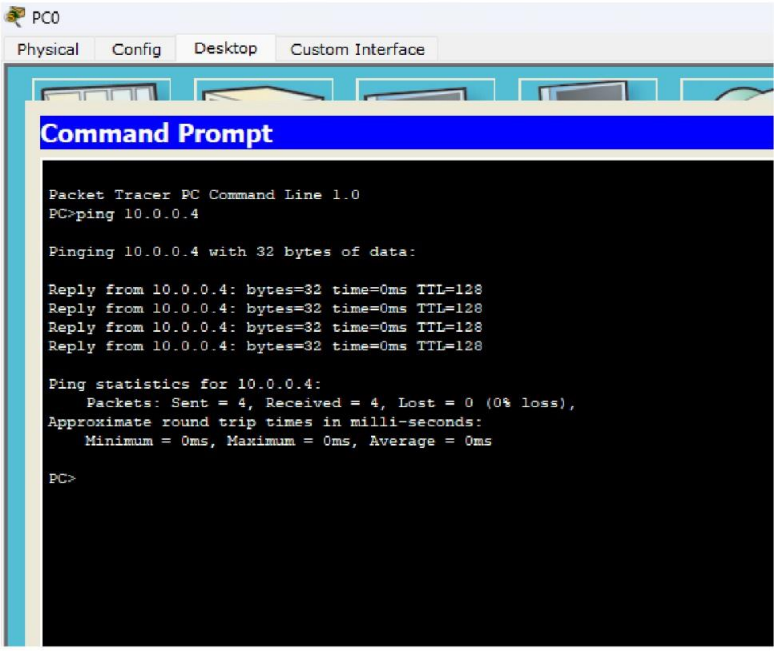
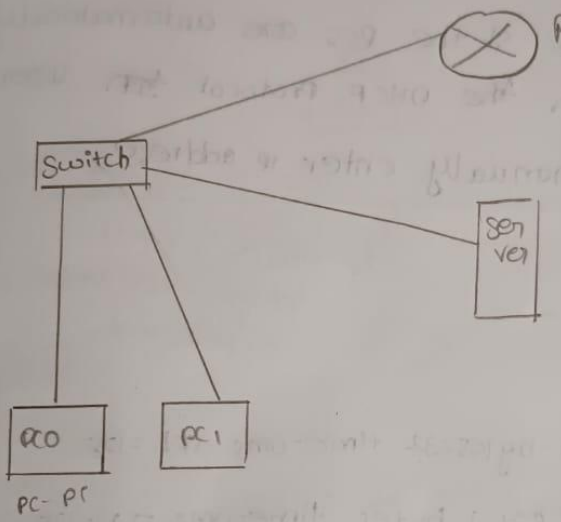


Figure 2: Output

12/11/24 Experiement - 06

Aim: a) To configure IP address of the host using DHCP server present within the network.

b) To configure IP address of the host using DHCP server present in the different network.



Configuration:

1) Connect to element as shown

2) to set router IP address 10.0.0.2

3) In server

ip address = 10.0.0.1

gateway = 10.0.0.2

under servies go to DHCP

pool name: serverpool1

Default gateway = 10.0.0.2

DNS Server = 10.0.0.1

click on save.

4) go to PC0 and PC1 under IP address click on DHCP

PC0 = 10.0.0.3 PC1 = 10.0.0.4.

5) upon pinging PC1 from PC0, there is it 0% loss.

Observation.

The ip address of the PCs are automatically generated via the DHCP protocol user need not manually enter IP address

↓ The main information we got knowledge of this experiment

output:

PC>ping 10.0.0.4

Pinging 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes time=0ms TTL=128
Reply from 10.0.0.4: bytes time=0ms TTL=128
Reply from 10.0.0.4: bytes time=0ms TTL=128
Reply from 10.0.0.4: bytes 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) To configure IP addresses of the host using DHCP server present in different LAN.

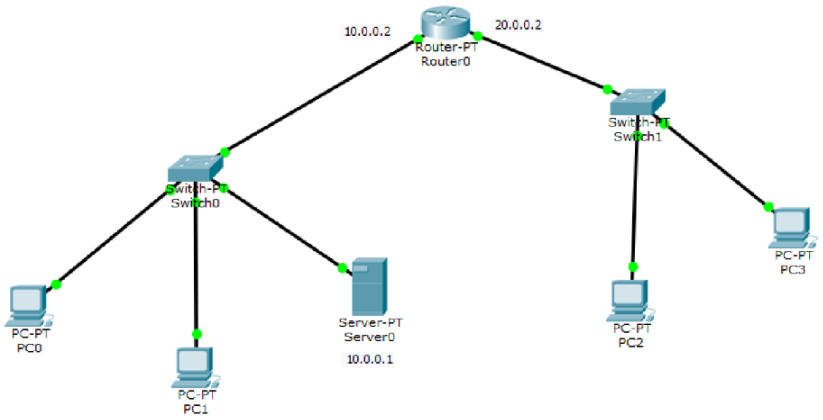
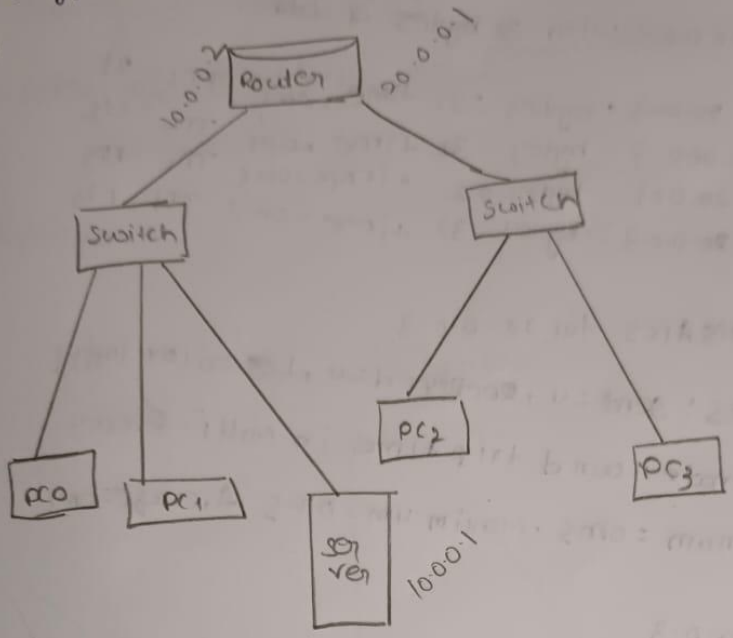


Figure 5: Topology

```
PC>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=3ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 0ms

PC>
```

Figure 6: Output

Topology:



Configuration:

→ Follow the steps of Q4.

→ set router ip address 20.0.0.1 in fastethernet 1/0

→ int fastethernet 1/0.

do ip helper 10.0.0.1

same fastethernet

→ server pool 2

20.0.0.1

10.0.0.1

start ip 1st box = 20

→ generate ip addresses for PC2 and PC3.

Observation.

Successful pinging in 20.0.0. as well as blw 10.0.0. and 20.0.0.

output:

Ping 20.0.0.3

pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.00.3 : bytes = 32 time = 0ms TTL = 128
Reply from 20.0.0.3 : bytes = 32 time = 0ms TTL = 128
Reply from 20.0.03 : bytes = 32 time = 0ms TTL = 128
Reply from 20.0.0.3 : bytes = 32 time = 0ms TTL = 128

ping statistics for 20.0.0.3:

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

Approximate round trip time in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms.

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 = 127
Reply from 10.0.0.3 : bytes = 32 time = 0ms TTL = 127
Reply from 10.0.0.3 : bytes = 32 time = 0ms TTL = 127
Reply from 10.0.0.3 : bytes = 32 time = 2ms TTL = 127

ping statistics for 10.0.0.3:

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

Approximate round trip times in milli-seconds:

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

Program 6

LAB WEEK 6

To configure DNS server to demonstrate mapping of IP addresses and domain names.

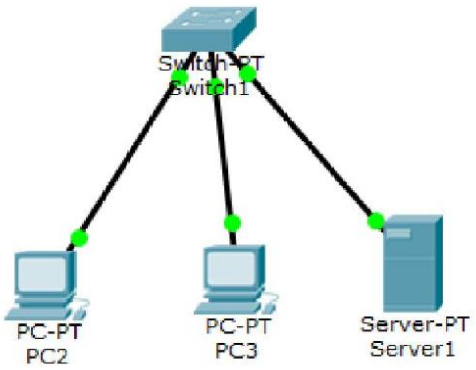


Figure 1: Topology

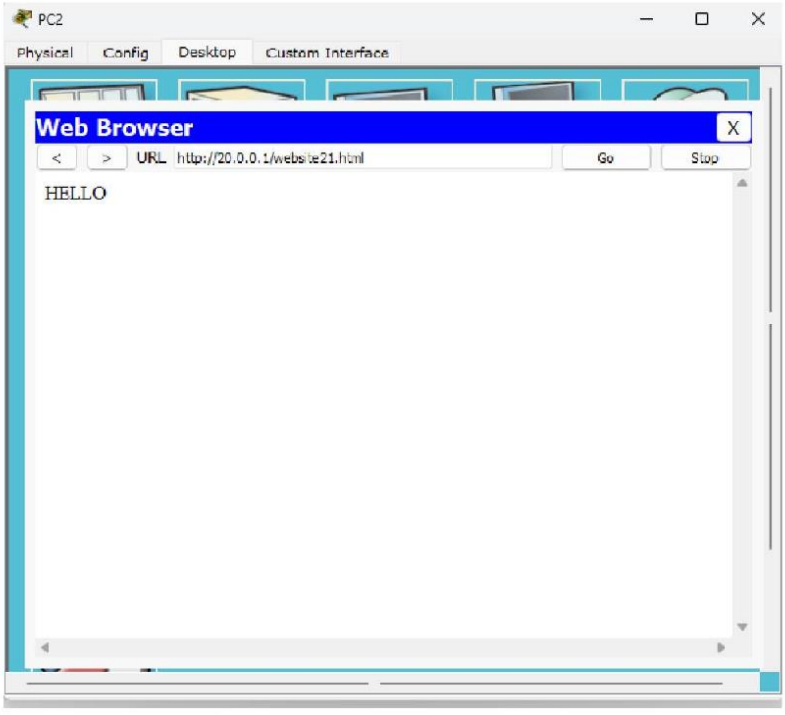
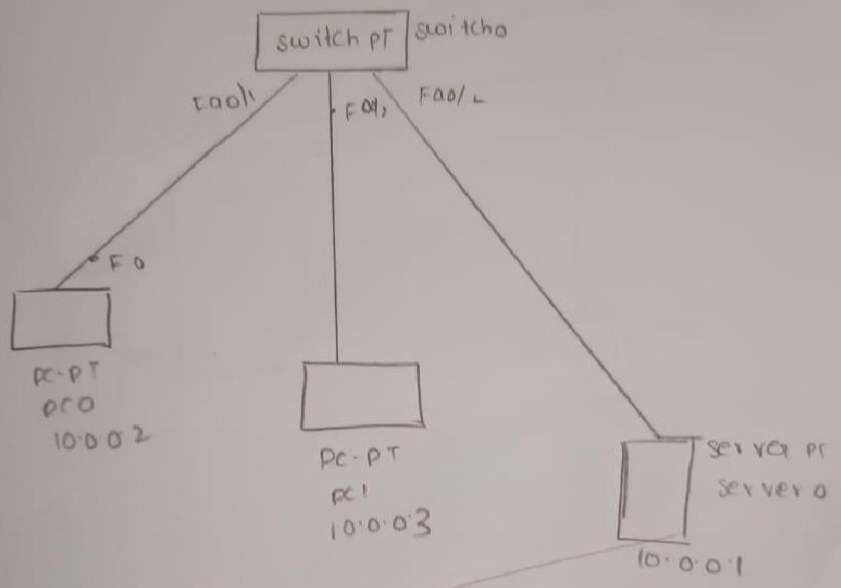


Figure 2:Output from PC2

To configure DNS server to demonstrate the Mapping of IP address and domain names

Topology:



configuration:

i) configure the IP address for the server as 10.0.0.1 under services configure DNS

First turn on the DNS servies and give a name web1 and address as 10.0.0.1

ii) Go to HTTP and edit the index.html if needed.

iii) configure the end devices by giving them IP address as 10.0.0.2 10.0.0.3

iv) under PC0 open the web browser and

search for web 1.

OBServation:

On searching the name on the webbrowser of the PCO we will be able to see the webpage.

Program 7

LAB WEEK 7

To configure RIP routing protocol in Routers.

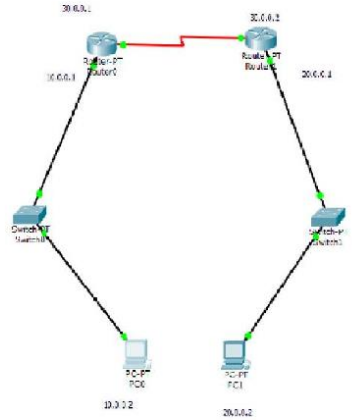


Figure 1: Topology

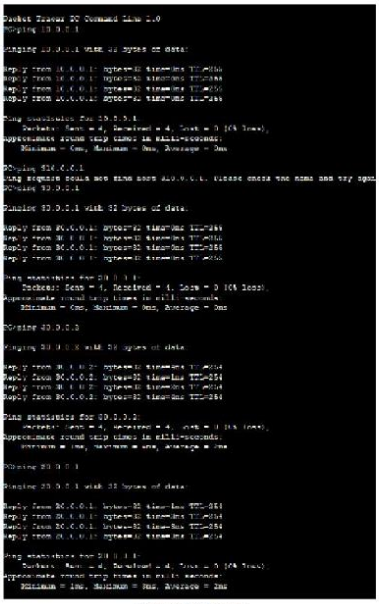
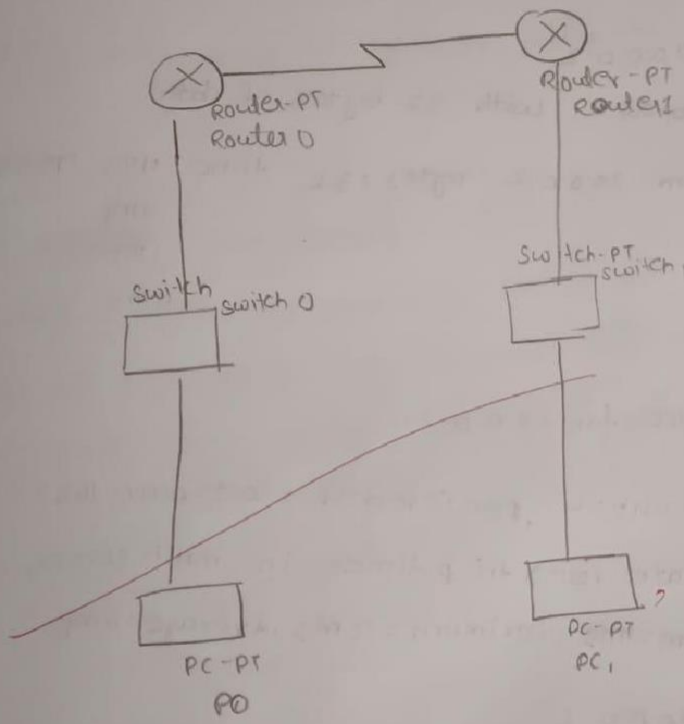


Figure 2: Output

19/11/24 Experiment - 8

Aim: Configuring RIP (routing information protocol)

Topology:



Configuration:

1. select 2 pcs 2 switches & 2 Routers & connect all the end devices

2. Set Ip address & Gateway for Pc & Router

Ip : Pc0 - 10.0.0.2 Pc-1 - 20.0.0.2 Router2 - 30.0.0.1

Router 1 - 30.0.0.2

Gateway pc0 - 10.0.0.1 pc-1-20.00.1

3. select Router 0, select RIP & add networks as 10.0.0.0 & 30.0.0.0, follow same for Router 1

4) Go to settings; In NVRAM save the changes

5) ping from pc0 to pc1

observation:

pc> ping 20.0.0.2

pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2 : bytes = 32 time = 4ms TTL = 12
= 18ms
1ms
4ms

ping statistic for 20.0.0.2:

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

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 18ms, Average = 6ms.

pc> ping 30.0.0.2

pinging 30.0.0.2 with 32 bytes of data

Reply from 30.0.0.2 : bytes = 32 time = 1ms TTL = 254
5ms
4ms
3ms

ping statistics
packets: sent
Approximate
Minimum =
pc> ping 3
pinging 3
Reply from
Repo
ping statistic
Packes:
Approximat
Minimum =

ping statistics for 30.0.0.2:

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

Approximate round trip times in milliseconds:

Minimum = 1ms, Maximum = 5ms, Average = 3ms.

PC> ping 30.0.0.1

pinging 30.0.0.1 with 32 bytes of data:

Reply from 30.0.0.1: bytes=32, time=0ms TTL = 255

2ms

0ms

~~Repl~~

ping statistic for 30.0.0.1:

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

Approximate round trip time in milli-seconds:

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

&

19/11/24

Program 8

LAB WEEK 8

a) To demonstrate communication between two devices using a wireless LAN

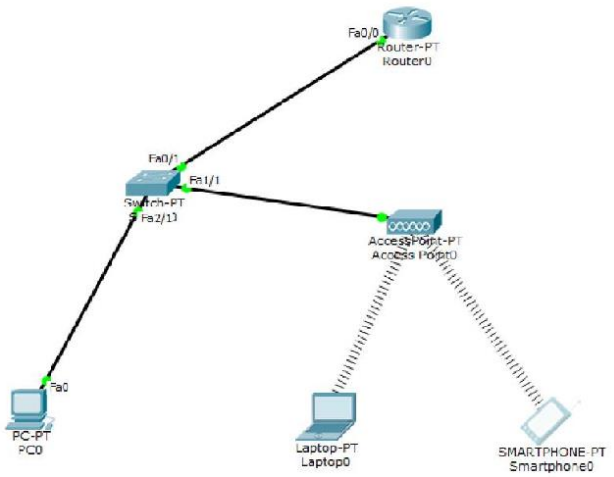


Figure 1: Topology

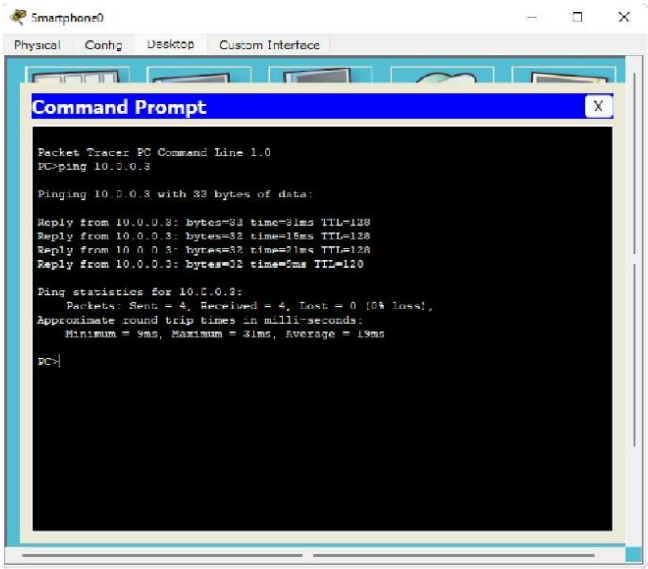
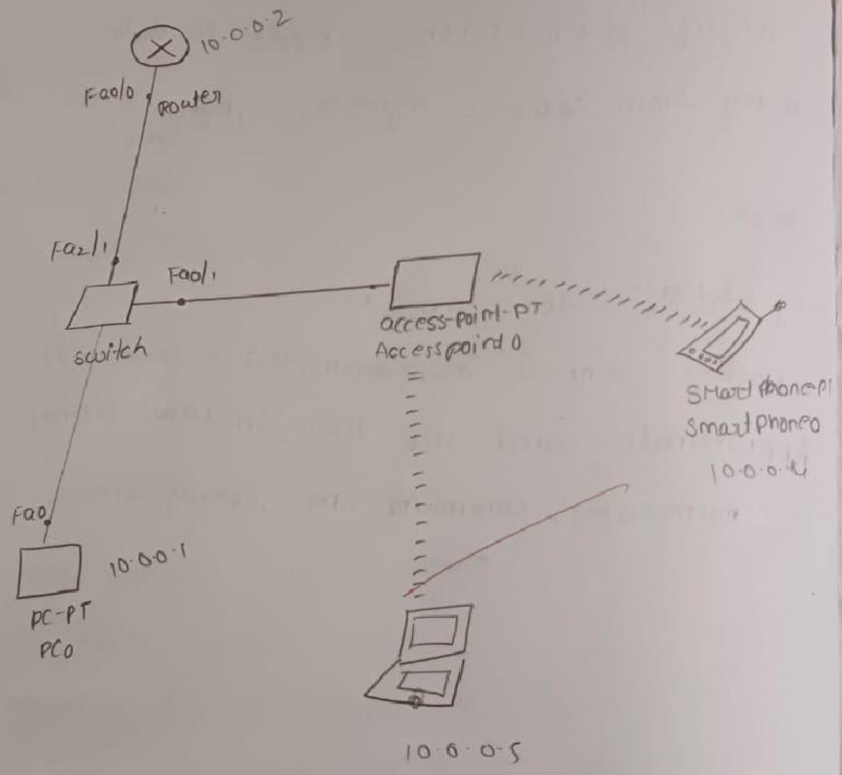


Figure 2: Output

26/11/24 Experiment-9

a) Aim: To demonstrate communication b/w two devices using a wireless LAN.

Topology:



Observation:

1) Select the PC, Router, switch and Access-point smartphone and laptop.

2) connect through Automatically choose connetion Type.

3) Give the configuration for PC and router

PC -) 10.0.0.1 Router 10.0.0.2

4) Go to the
Port 0 mak
Go to port 1
Authenticati
WPA 2 -
password

5) Go to smart
Give the
Authentica
Give the

6) Go to to
Give the
perform

But in th
there
Go to
Go go to
Switch
and add
then

Observation
With the
will be c
Connected

4) Go to the accesspoint go to config

port 0 make it Auto on.

Go to port 1 Give the SSID wlan 1

Authentication

WPA 2 - PSK

password is 345678.

5) Go to smartphone

Give the SSID wlan 1

Authentication WPA2 - PSK password 12345678.

Give the IP address 10.0.0.4

6) Go to laptop

~~Give the~~ perform same Give the IP address 10.0.0.5.

But in the ~~pysical~~ config wireless 0 will not be there to add that.

Go to pysical.

Gogo to physical device view.

Switch off button drag the wired connection and add wireless and on the switch.

then Access point and laptop will be connected.

Observation:

With the help of wireless connection we will be able to ping from pc no different wired Connected devices like the laptop and smartphone.

Program 9

LAB WEEK 9

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

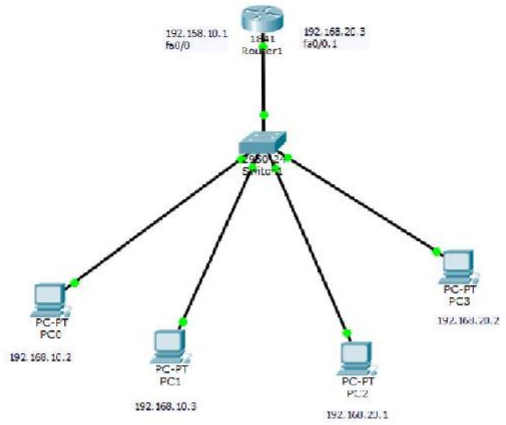


Figure 1: Topology

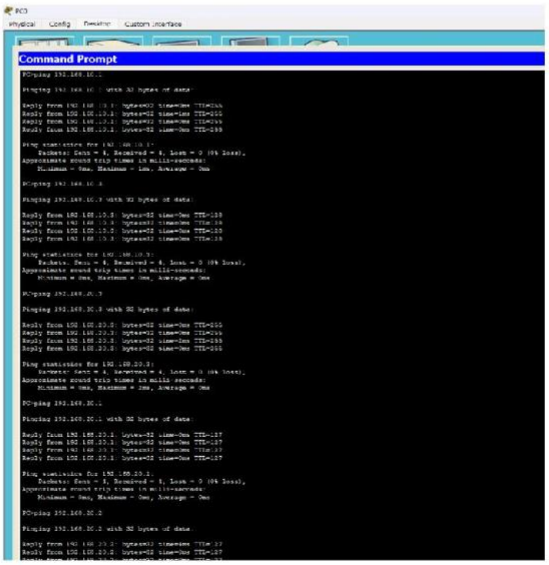


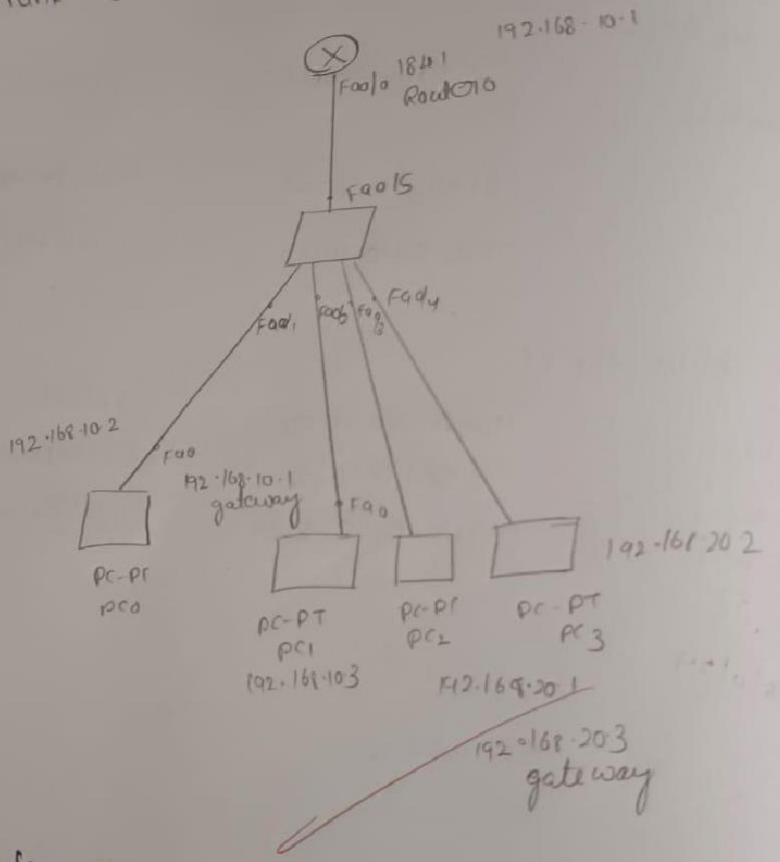
Figure 2: PC Command Prompt

8/12/24

Experiment- 11

Aim: To create a virtual LAN on top of the physical LAN and enable communication b/w physical LAN and virtual LAN

Trunk is a interface that passes messages



Configuration:

1) select the 4 pc and one switch 2950T and router 1841

2) config router and ipaddress 192.168.10.1 and the select pc give the ip address 192.168.10.2 and 192.168 and give the gateway.

3) go to switch -> vlan database -> vlan 1 -> add

4) go to Fa0/5

Change access
select vlan
go to Fa0/3
only sele

5) set ip addr

6) go to
enter

7) set ip ad
then exit

8) type:
vlan d
->v la

9) enter
type

AB
3/12/24

11) again

12) type
ip add
no shut
exit

13) ping
Observatio
Success
physical

Change access to trunk
Select vlan 20
go to Fa0/3 & Fa0/4
Only select vlan 20.

5) set ip address and gateway for Pc2 and Pc3

6) go to router
enter fastethernet 0/0.1 interface.

7) set ip address 192.168.20.3 255.255.255.0

then exit config mode ie config-subif. and config.

8) type:
vlan database
->vlan 30 vname vlan1

9) enter config-subif mode.

type:
encapsulation dot1q 20
no shutdown
exit.

10) again enter interface to fastethernet 0/0.1

12) type encapsulation dot1q 20
ip address 192.168.20.3 255.255.255.0
no shutdown.
exit

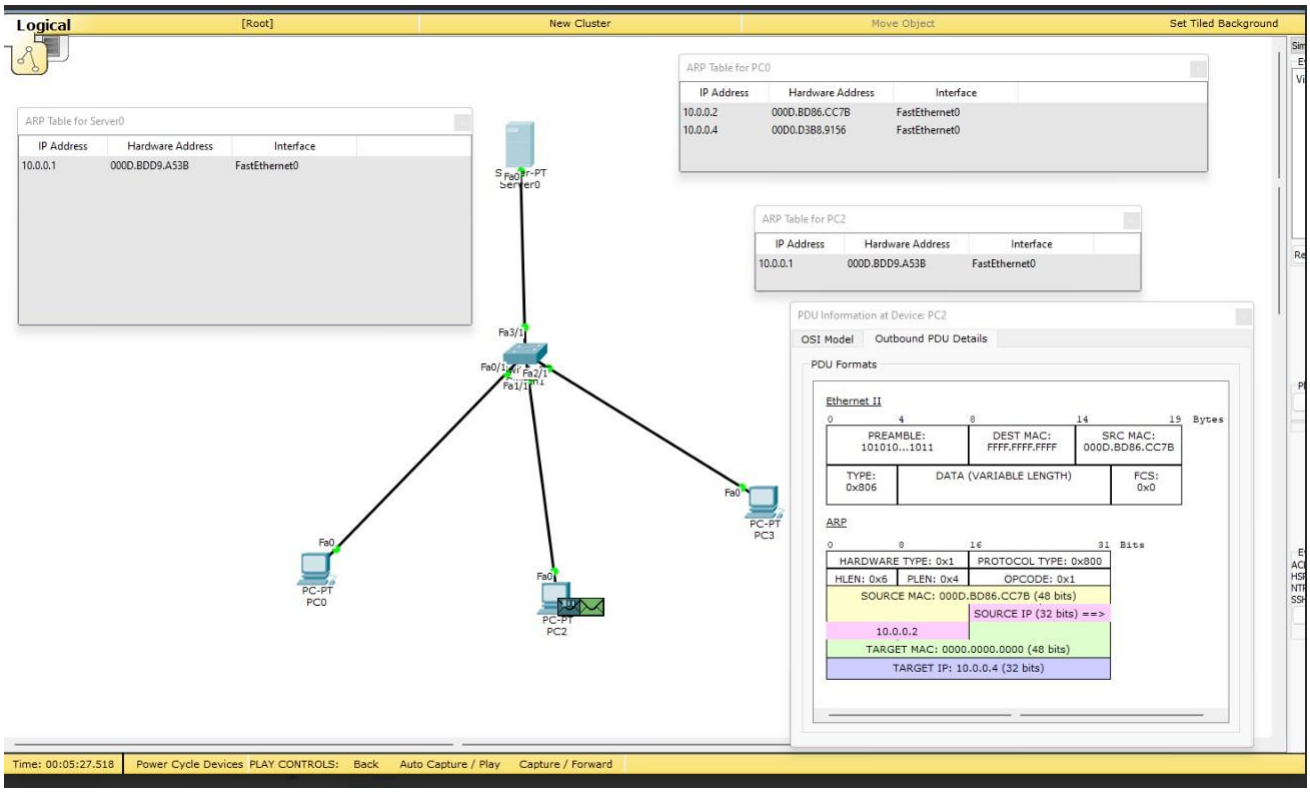
13) ping Pc3

Observation:
Successful pinging within the network and the physical network and virtual network Pc

Program 10

LAB WEEK 10

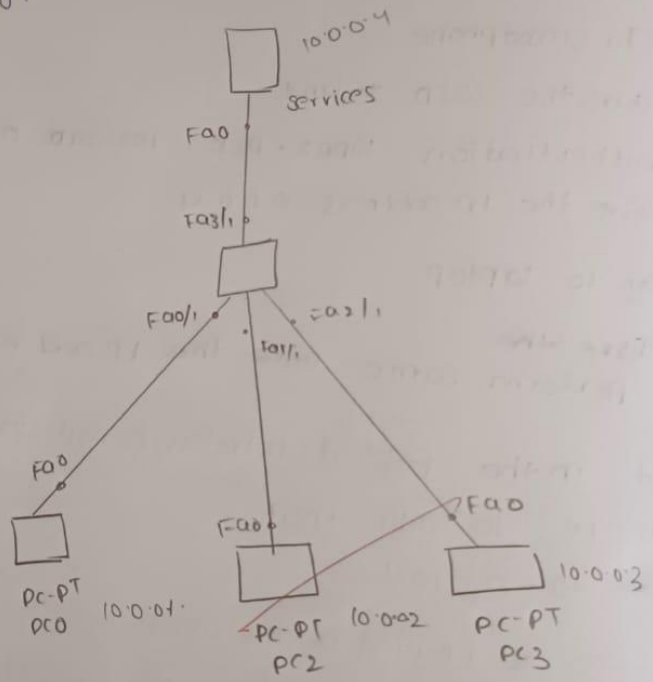
To Demonstrate the working of address resolution protocol(ARP) for communication within a LAN



26/11/24. Experiment - 10

Aim: To demstrate the working of address resolution protocol (ARP) for communication within a LAN.

Topology:



~~Observe~~ configuration:

1) select the 3 pc and assign the IP address
2) one switch and services.
3) Give the simple PDU from PC0 to server and PC2 to Services then check the ARP table.

Observation:

ARP table for server 0.

IP address	Hardware address	Interface
10.0.0.1	0001.C9AB.3390	FastEthernet1
10.0.0.2	0010.11A0.EA77	FastEthernet1

ARP for PC0

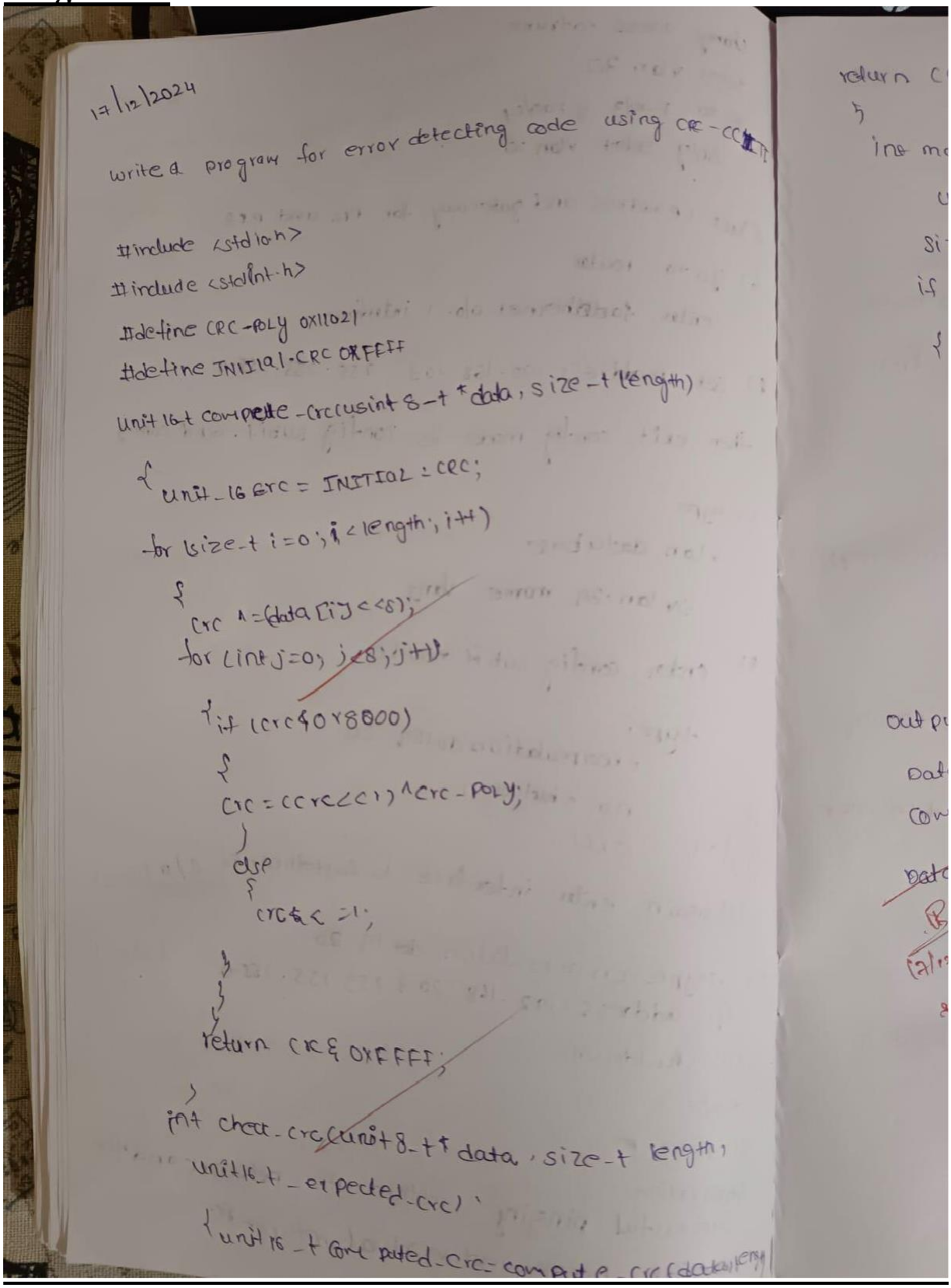
IP address	Hardware address	Interface
10.0.0.2	0010.11A0.EA77	FastEthernet1
10.0.0.4	0000.0390.44C7	FastEthernet1

ARP table for PC.

Ipaddress	Hardware address	Interface
10.0.0.1	0001.C9AB.3390	FastEthernet1
10.0.0.4	0000.0390.0UC7	FastEthernet1

Leela.B
26/4/24

Program 11



17/12/2024

write a program for error detecting code using CRC-CCITT

```
#include <stdio.h>
#include <stdint.h>
#define CRC_POLY 0x1021
#define INITIAL_CRC 0xFFFF
uint16_t compute_crc(uint8_t *data, size_t length)
{
    uint16_t crc = INITIAL_CRC;
    for (size_t i=0; i<length; i++)
    {
        crc ^= (data[i]<<8);
        for (int j=0; j<8; j++)
        {
            if (crc&0x8000)
            {
                crc = (crc<<1)^crc_POLY;
            }
            else
            {
                crc<<=1;
            }
        }
    }
    return crc & 0xFFFF;
}
int check_crc(uint8_t *data, size_t length,
uint16_t expected_crc)
{
    uint16_t computed_crc = compute_crc(data,length);
```

```
return (computed_crc == expected_crc);
}
int main()
{
    uint8_t data[] = "Hello, world";
    size_t received_length = sizeof(received_data);
    if (check_crc(received_data, received_length, crc))
    {
        printf("Data received correctly with no errors!\n");
    }
    else
    {
        printf("Error detected in received data\n");
    }
    return 0;
}
```

Output:

Data: Hello, world!

Computed CRC-CCITT: 0x6F0A.

Data received correctly with no errors.

17/12/24

Leaky bucket Algorithm.

```
#include <stdio.h>

int main()
{
int incoming, outgoing, buck_size, n, store=0;
printf("Enter bucket size, outgoing rate and no of
inputs:");
scanf("%d %d %d", &buck_size, &outgoing, &n);
while(n!=0){
printf("Enter the incoming packet size:");
scanf("%d", &incoming);
printf("Incoming packet size %d\n", incoming);
if(incoming<=(buck_size-store)){
store+=incoming;
printf("Dropped %d no of packets\n", incoming-
(buck_size-store));
printf("Bucket buffer size %d out of %d\n", stor,
buck_size);
store=buck_size;
}
store=store-outgoing;
printf("After outgoing %d bytes left out of %d in
buffer\n", store, buck_size);
n--;
}
```

output:

Enter bucket size, outgoing rate and no of inputs: 5 3 2.

Enter the incoming packet size: 3

Incoming packet size 3

Bucket buffer size 3 out of 5

After outgoing 1 bytes left out of 5 in buffer

Enter the incoming packet size: 2

Incoming packet size 2

Bucket buffer size 3 out of 5

After outgoing 1 byte left out of 5 in buffer.

18
17 12 12 14

Sorry

program 12

In [1]:
```
from socket import *
from threading import Thread

def start_server():
    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 True:
        connectionSocket, addr = serverSocket.accept()
        print(f"Connection from {addr}")
        sentence = connectionSocket.recv(1024).decode()
        try:
            with open(sentence, "r") as file:
                l = file.read(1024)
                connectionSocket.send(l.encode())
        except FileNotFoundError:
            connectionSocket.send("File not found.".encode())
        connectionSocket.close()

# Run the server in a separate thread
server_thread = Thread(target=start_server, daemon=True)
server_thread.start()
```

```
The server is ready to receive
```

In [2]:

```python
from socket import *

serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))

# Prompt for the filename
sentence = input("Enter the file name: ")
clientSocket.send(sentence.encode())

# Receive file content or error message
filecontents = clientSocket.recv(1024).decode()
print('From Server:', filecontents)

clientSocket.close()
```

```
Connection from ('127.0.0.1', 63844)
Enter the file name: filename.txt
From Server: Hello
```

22/12/24.

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.

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)
connectSocket.send(l.encode())
file.close()
connectionSocket.
```

Output:

The ser

Connection

Enter th

From

ogram to
er to send
ent.

cceot()
decode()

output:

The server is ready to receive.

connection from ('127.0.0.1', 63844)
Enter the file name.txt.
From server: Hello.

24/12/24

2. 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.

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.recvfrom(2048)
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, 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()
```

Output:

The server
connection
Enter the
from S