INDEX

Name JANJANA NRESH Sub.

Std.: Div. Roll No.

Telephone No. E-mail ID.

Blood Group. Birth Day.

Sr.No.	Title	Page No.	Sign./Remarks
01	LAB - 01 04/10/24	01	
	Send a PDU message and demonstrate ping		
02	LAB - 02 18/10/24		
	- Router IP address configuration		
03	LAB - 03 25/10/24		
	- Default and static routing		
04	LAB - 04 8/11/24		
	- DNS within a LAN		
05	LAB - 05 8/11/24		
	- DHCP		
06	LAB - 06 15/11/24		
	- Error Detection : CRC		
	- Leaky Bucket algorithm		
			26/12
07	LAB - 07 22/11/24		
	- RIP		
	- TTL		
08	LAB - 08 29/11/24		
	OSPF		
09	LAB - 09 20/12/24		
	- ARP		
	- Telnet		
	- WLAN		
	- VLAN		

27/09

LAB - 01

CISCO PACKET TRACER

* **AIM**: Explore Cisco Packet Tracer

① Launched Cisco Packet Tracer Student version.

② Explored Routers, Switches, Hub and End Devices and Connections.

③ Checked the different modes - Realtime or Simulation

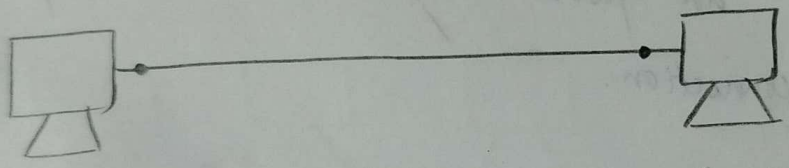
* Create a network

① Select 2 end devices - generic PC in the workspace

② Place a connection between the end devices.

③ Green lines on the link indicate that the connection is working.

④



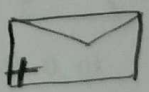
PC-PT
PC1
10.0.0.1

PC-PT
PC2
10.0.0.2

⑤ Click on the PC and switch to Config tab.

⑥ Configure an IP address for both the end devices

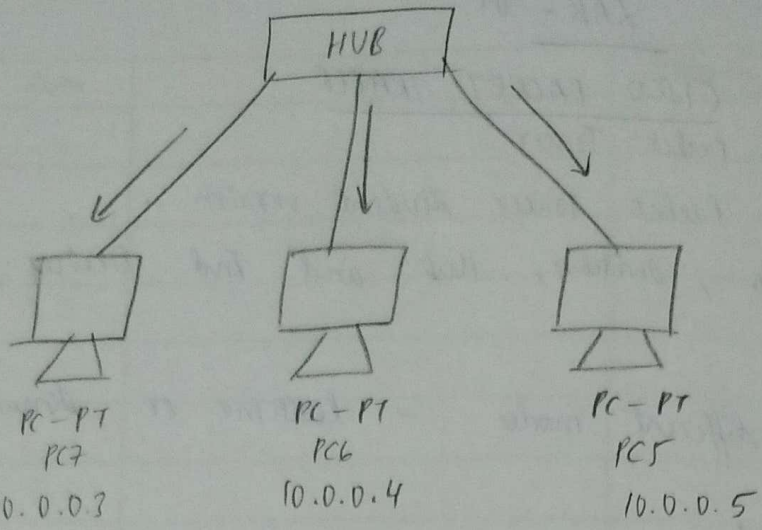
⑦ To sent messages via data packets click on Add Simple PDU (P)



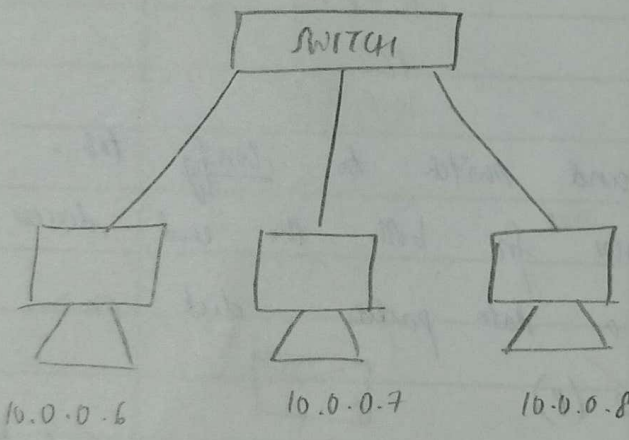
⑧ On the Event List window, check the status and type (ICMP)

⑨ The status should change from In progress to Successful.

OBSERVATION



① Configure each one of the PC's with IP addresses
② Send message from one PC to another via a hub.
③ Make sure green dots are found on the link to for a succesful connection.



④ There is no need to configure any connecting devices - switch or hub.

*

* Real Time and Simulation Mode

- Real time mode is better for network configuration but simulation mode allows you to observe packet in detail.

- Real time mode - Network runs in a mode of real time within the limits of protocol modes used.

- Network responds to your actions immediately as they would on a real device.

- Simulation mode - Direct control over time related to flow of PDU's. You can see the network run step by step.

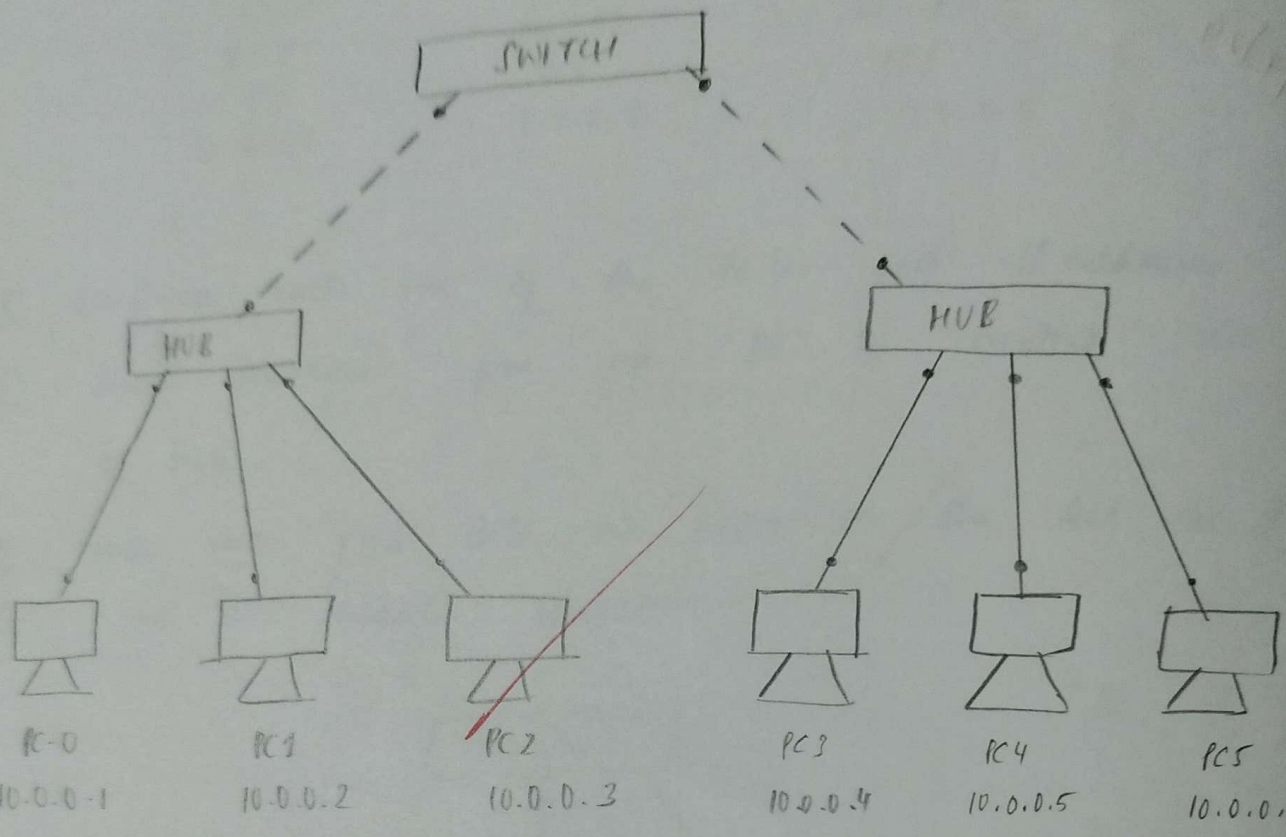
27/7/24

LAB - 01

04/10

EXERCISE - 1

Create a topology involving multiple hubs and a switch connecting them to realize a single PDU.



① Connect multiple end devices - PC to a hub through

② Configure the IP addresses of the end devices - PC's as 10.0.0.1 , 10.0.0.2 , 10.0.0.3

③ Connect 3 PC's to a single hub through a ~~cross~~ copper straight through connection.

④ Similiarly configure the IP address of another set of end devices as 10.0.0.4 , 10.0.0.5 , 10.0.0.6

⑤ Connect this set of PC's to another hub by a copper straight through connection.

⑥ Connect the two hubs to a single switch by copper cross over connection.

⑥ When the connection is available - green dots present on links, send a PDU (data packet) from PC0 to PC 2 - data packets through a hub.

⑦ Similarly, ~~when the conn~~ send another data packet from PC1 to PC4 - through a switch.

OBSERVATIONS

① During the transmission of data packets, last status is 'In progress'. When completed, the last status change to 'Successful'.

② Type in the Event List is ICMP.

③ PDU information can be seen ~~in~~ ~~on~~ by clicking on the message icon which shows layers in OSI model.

TASK - 2

① Click on a PC that has received a data packet, go to Desktop tab and click on command prompt

② Run the command

PC > ping 10.0.0.4 // destination address.

OBSERVATIONS

TTL = Time To Live

① The command prompt say -

PC > ping 10.0.0.3

Reply from 10.0.0.3 : bytes = 32 time = 3ms TTL = 128.

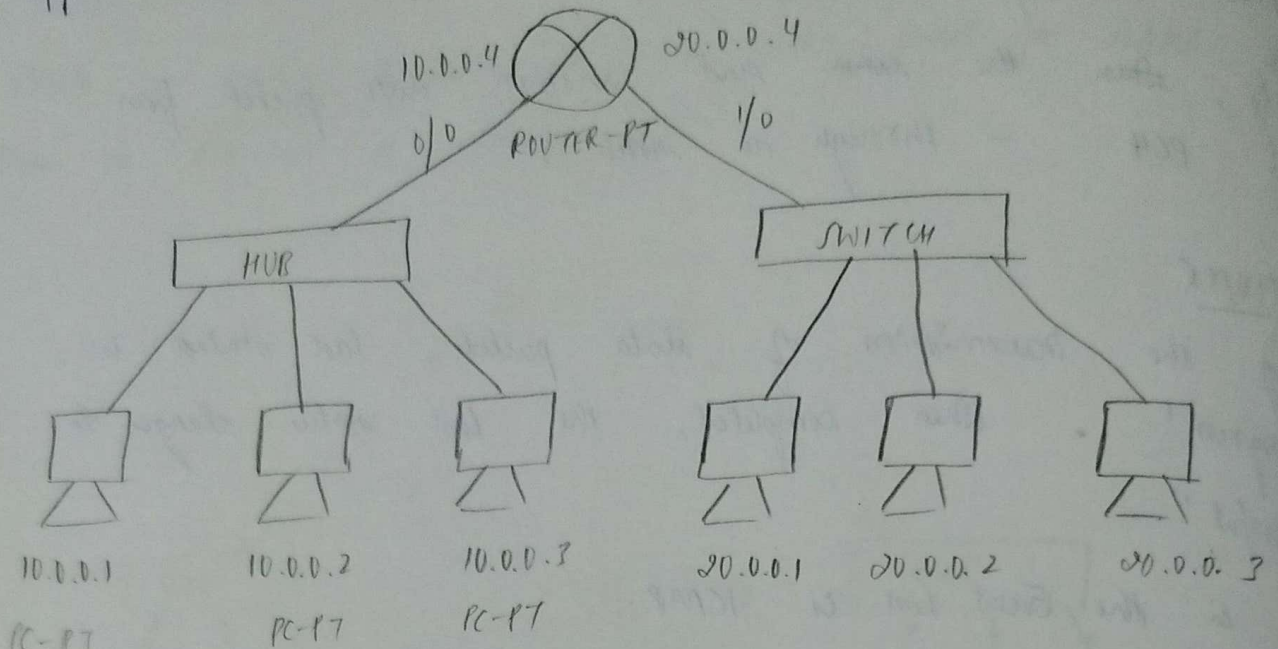
Reply from 10.0.0.3 : bytes = 32 time = 4ms TTL = 128

Reply from 10.0.0.3 : byte = 32 time = 4ms TTL = 128

Ping 4/10 statistics for 10.0.0.3

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

Configure IP address to routers in packet tracer.



① Configure one set of end devices - PC's with IP addresses - 10.0.0.1, 10.0.0.2, 10.0.0.3.

② Let these PC's be connected to a ~~switch~~ hub. by copper ~~cross~~ straight over connection

③ Configure another set of end devices with IP addresses - 20.0.0.1, 20.0.0.2, 20.0.0.3

④ Let these PC's be connected to a single switch by copper cross over connection.

⑤ The hub and switch are connected to a Router - PT using copper straight connections.

⑥ One side of the router (0/0) is configured with IP address 10.0.0.4, while the other side (1/0) is configured with IP address 20.0.0.4

* Router configuration

Open Command Line Interface (CLI) in Router. The next few steps —

- Type no
- Router > enable > config t / configure terminal
- interface fastethernet 0/0
- ip address 10.0.0.4 255.0.0.0
- no shutdown
- exit

* Request timed out – Gateway

- When you try to ping from 10.0.0.1 to 20.0.0.2,

Observation - Response in command prompt is Request timed out

- This is because each PC that is connected to the router, requires to be configured with the gateway of router's address (10.0.0.4/20.0.0.4)
- ~~Once gateway is configur~~

COMMAND PROMPT

PC > ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data.

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.2 :
Packets : Sent = 4, Received = 0, Lost = 4 (100% loss)

- Once gateway has been configured, we can ping from one end device to another end device.

Observation → When we try to ping from 10.0.0.1 to 20.0.0.2, the first response will be Request timed out.

→ This is due to a large network which takes time to recognize end devices.

→ The next response will be a reply from the end device. as after some time all end devices are recognized.

COMMAND PROMPT

PC > ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data :

Request timed out.

Reply from 20.0.0.2 : bytes = 32, time = 0ms TTL = 127

Reply from 20.0.0.2 : bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.2 : bytes = 32 time = 0ms TTL = 127

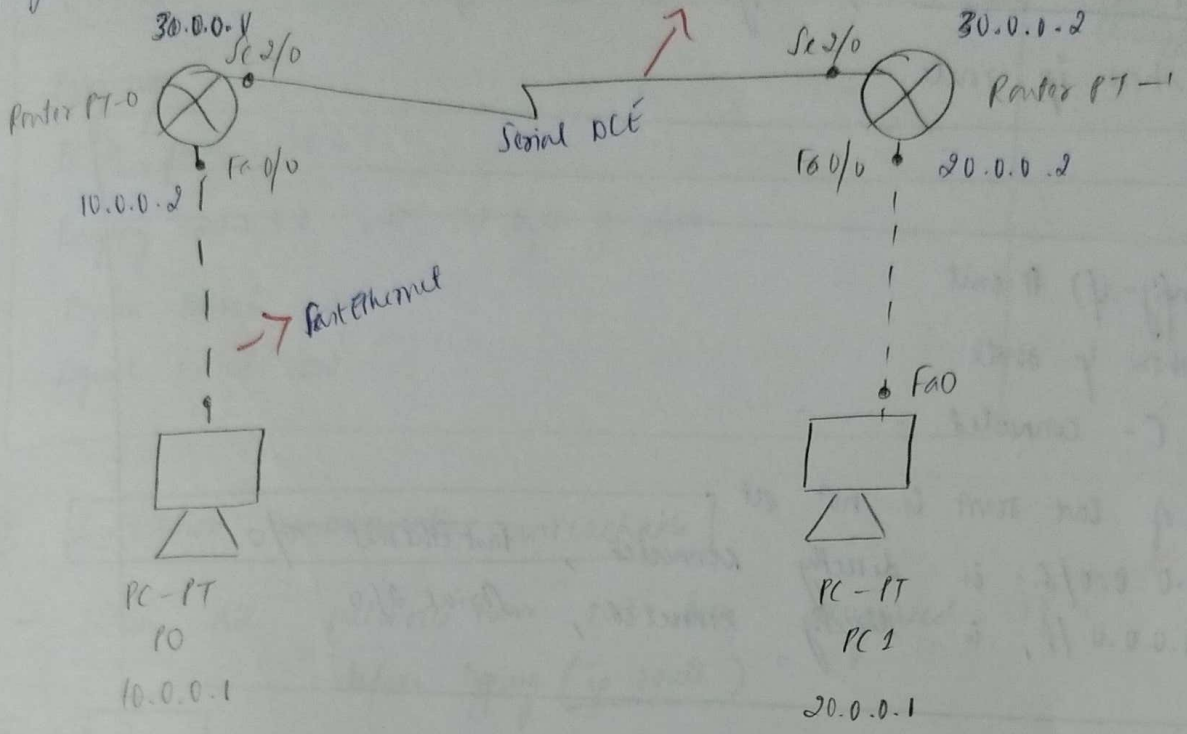
Ping statistics for 20.0.0.2

Packets : Sent = 4, Received = 3, Lost = 1 (25% loss).

LAB - 02

18/10

Q. Configure IP address of routers in packet tracer. Explore the messages - ping responses, destination unreachable, request timed out and reply.



① ~~Con~~ Place 2 end devices - PC's and configure with 2 different type of IP address.

PC0	20.0.0.1
PC1	20.0.0.1

② Place 2 routers. Connect one router (R0) with PC using copper cross over.

③ Configure IP address for one router as 10.0.0.2 in the fastEthernet tab.

④ Configure the gateway for PC0 as 10.0.0.2

⑤ Repeat the steps for Router1 - IP address as 20.0.0.2 in fastEthernet tab and gateway for PC1 as 20.0.0.2

⑥ Use a Serial DCE connection (Se 2/0) between 2 routers and configure IP address under Serial 2/0 tab as 30.0.0.1 and 30.0.0.2 for router1 and router2 respectively.

⑦ Enable port status as ON for all connections.

Observation • When you try to send a packet from PC0 to PC1 status is Failed

To recognize all networks, go to CLI of Router

Router # show ip route

Router 0

CLI

```
Router (config-if) # exit
Router # show ip route
Codes   C - connected . . . .
Gateway of last resort is not set .
C   10.0.0.0/8 is directly connected, FastEthernet 0/0
C   30.0.0.0/8 is directly connected, Serial 2/0
```

• To connect all networks, go to CLI of router0.

Router 0 - CLI

```
Router #
Router # configure terminal
Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.2
```

General syntax

• Router (config) # ip route <unknown network> <subnet mask> <unknown router>

• For Router 1, CLI

ip route 10.0.0.0 255.0.0.0 30.0.0.1

Responses while pinging a message —

① Request time out

- When wrong IP address has been mentioned.
- gateway not configured.

```
Command Prompt

Pinging
PC> ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data :
Request timed out.
Request timed out.
```

② Destination ~~unavailable~~ unreachable

- When all networks have not been recognized. ip route before typing (ip route)

③ Reply

```
PC > ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data :
Reply from 20.0.0.1 : bytes = 32 time = 1ms TTL = 126
Reply from 20.0.0.1 : bytes = 32 time = 7ms TTL = 126
Reply from 20.0.0.1 : bytes = 32 time = 14ms TTL = 126

Ping statistics for 20.0.0.1 :
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)
```

- Successful message

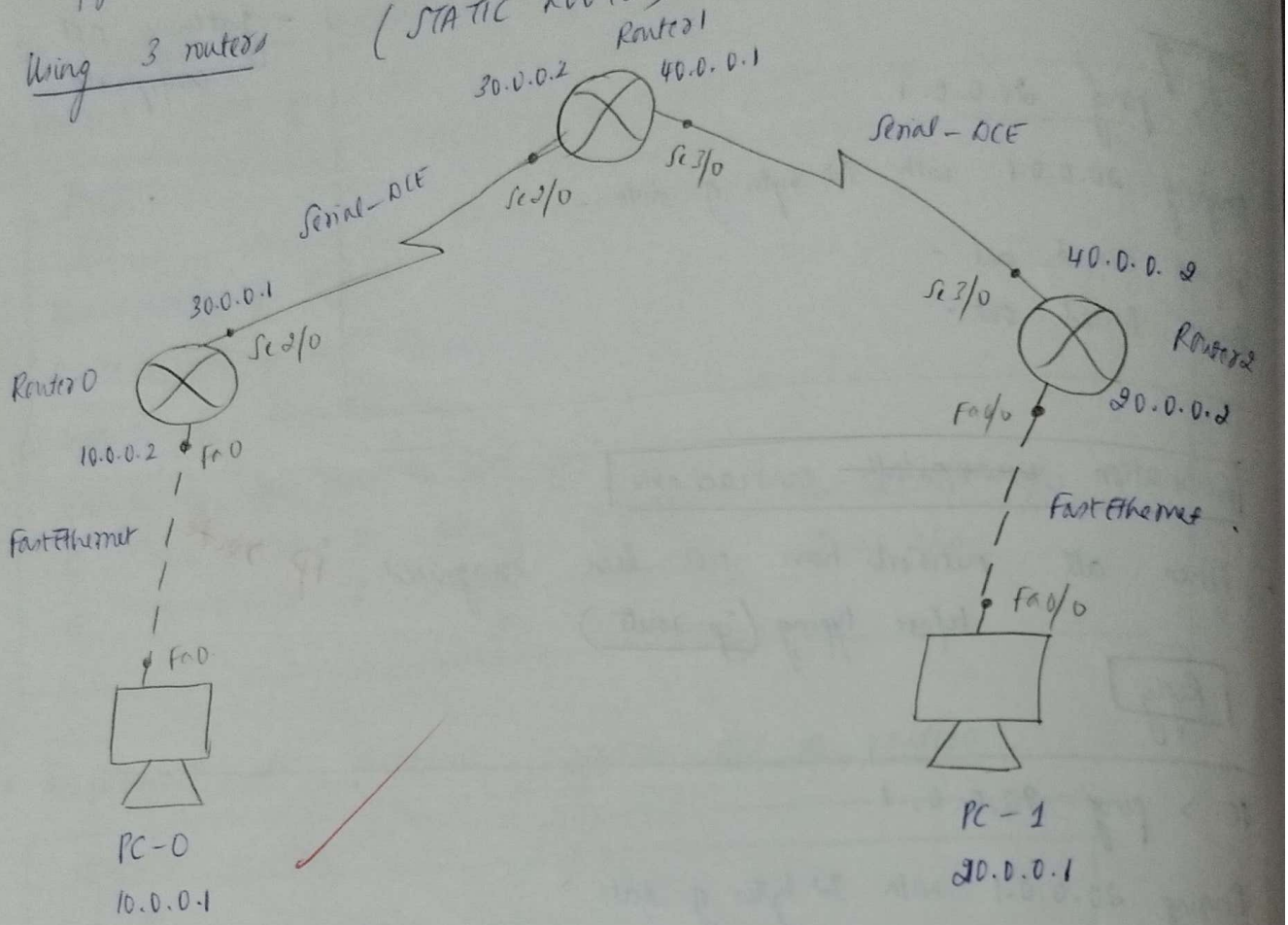
18/10

LAB - 03

25/10/24

Configure default route, static route to router.

Using 3 routers (STATIC ROUTE)



① Place 2 end devices PC-0 and PC-1 and configure them with 2 two different IP address —

PC-0	10.0.0.1
PC-1	20.0.0.1

② For a 3 router configuration, there will be 4 different type of network.

③ Place 3 routers and connect each with Serial - DCE connection and connect Router-0 and Router 2 to end device PC-0 and PC-1 with FastEthernet respectively.

④ Configure Router 0 as 10.0.0.2 with Fa 0/0 port.
30.0.0.1 in Se 2/0 port.

Router 1 with 30.0.0.2 in Se 2/0 port
40.0.0.1 in Se 3/0 port

Router with 40.0.0.2 in Se 2/0 port
20.0.0.2 in Fa 0/0 port.

⑤ Type gateway of PC-0 as 10.0.0.2 and PC-1 as 20.0.0.2

⑥ Enable port status as ON for seperate connections.

OBSERVATION – To recognize all networks for each router —

Router0 > CLI

```
Router #
Router # configure terminal
Router (config) # ip route 40.0.0.0 255.0.0.0 20.0.0.2

Router (config) # ip route 20.0.0.0 255.0.0.0 40.0.0.2
```

Similarly do the same for Router1 and Router2

Router1 > CLI

```
Router # configure terminal
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router (config) # ip route 20.0.0.0 25.0.0.0 40.0.0.2
```

Router 2 > CLI

```
Router # configure terminal
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router (config) # ip route 30.0.0.0 255.0.0.0 40.0.0.1
```

Reponses while pinging a message —

① **Request timeout** — When wrong IP address has been mentioned or when IP address has not been configured properly.

Command prompt

PC > ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data :

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.1 :

Packets : Sent = 4, Received = 0, Lost = 4 (100% loss)

② **Reply** — Successful response

PC > ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data :

Request timed out.

Reply from 20.0.0.1 : bytes = 32 time = 12ms TTL = 125

Reply from 20.0.0.1 : bytes = 32 time = 15 ms TTL = 125

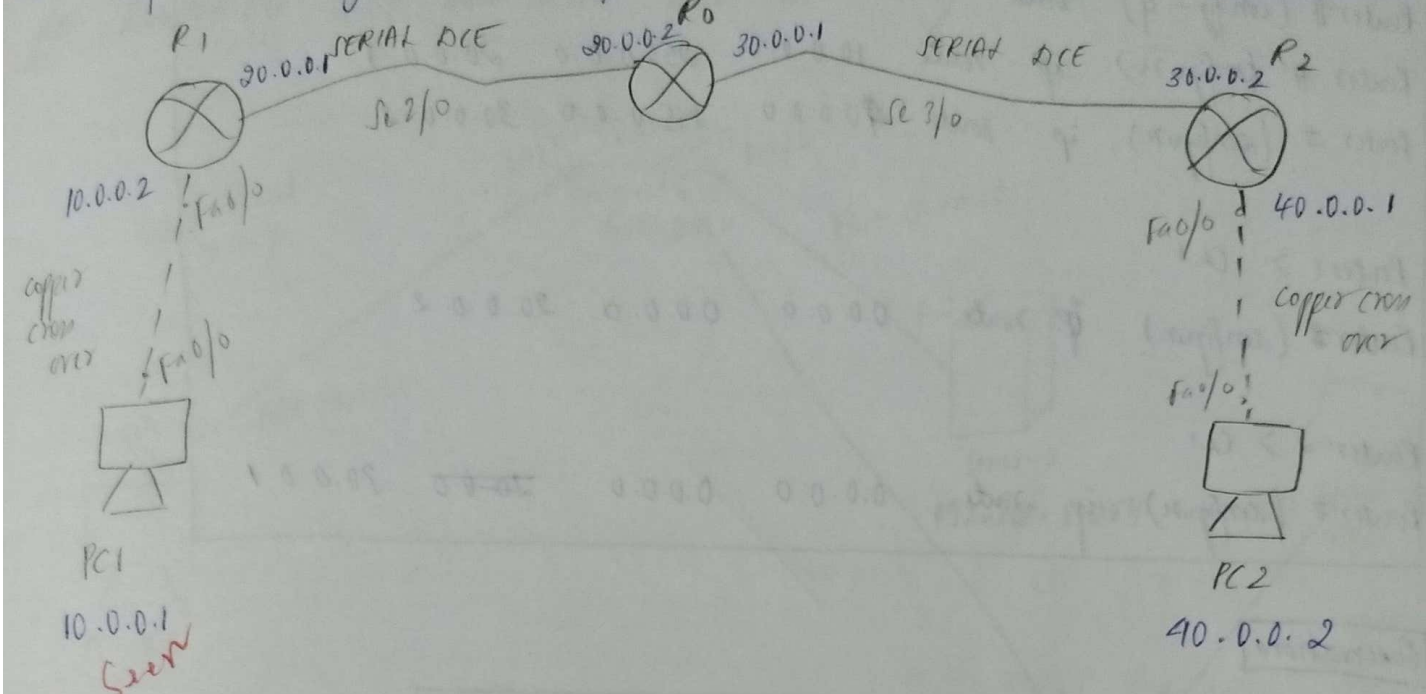
Reply from 20.0.0.1 : bytes = 32 time = 15 ms TTL = 125

Ping statistics for 20.0.0.1 :

Packets : Sent = 4 Received = 3 Lost = 1 (25% loss).

* DEFAULT ROUTE

- Default routing is done for routers connected with end devices



Default route

For R1,

#ip route 0.0.0.0 0.0.0.0 20.0.0.2

For R2,

ip route 0.0.0.0 0.0.0.0 30.0.0.1

Operation -

↳ show ip route (check this)

① Place 2 end devices PC1 and PC2 and configure their IP address -

PC 1 - 10.0.0.1

PC 2 - 40.0.0.2

② For a 3 router configuration, there will be 4 different networks.

③ Configure each of the routers as mentioned above. and specify the gateway for each router.

④ Routers are connected by Serial - DCE Connection and PC-Router is connected by copper cross wire.

To recognize all networks,

Router0 > CLI

Router# (config-if) exit
Router # (configure) ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router # (configure) ip route 40.0.0.0 255.0.0.0 30.0.0.2

Router1 > CLI

Router# (configure) ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router 2 > CLI

Router# (configure) ip route 0.0.0.0 0.0.0.0 ~~20.0.0~~ 30.0.0.1

Observations

Router#1 > CLI

Router# show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 20.0.0.0/8 is directly connected, Serial 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2

// default routing

Router0 > CLI

Router# show ip route

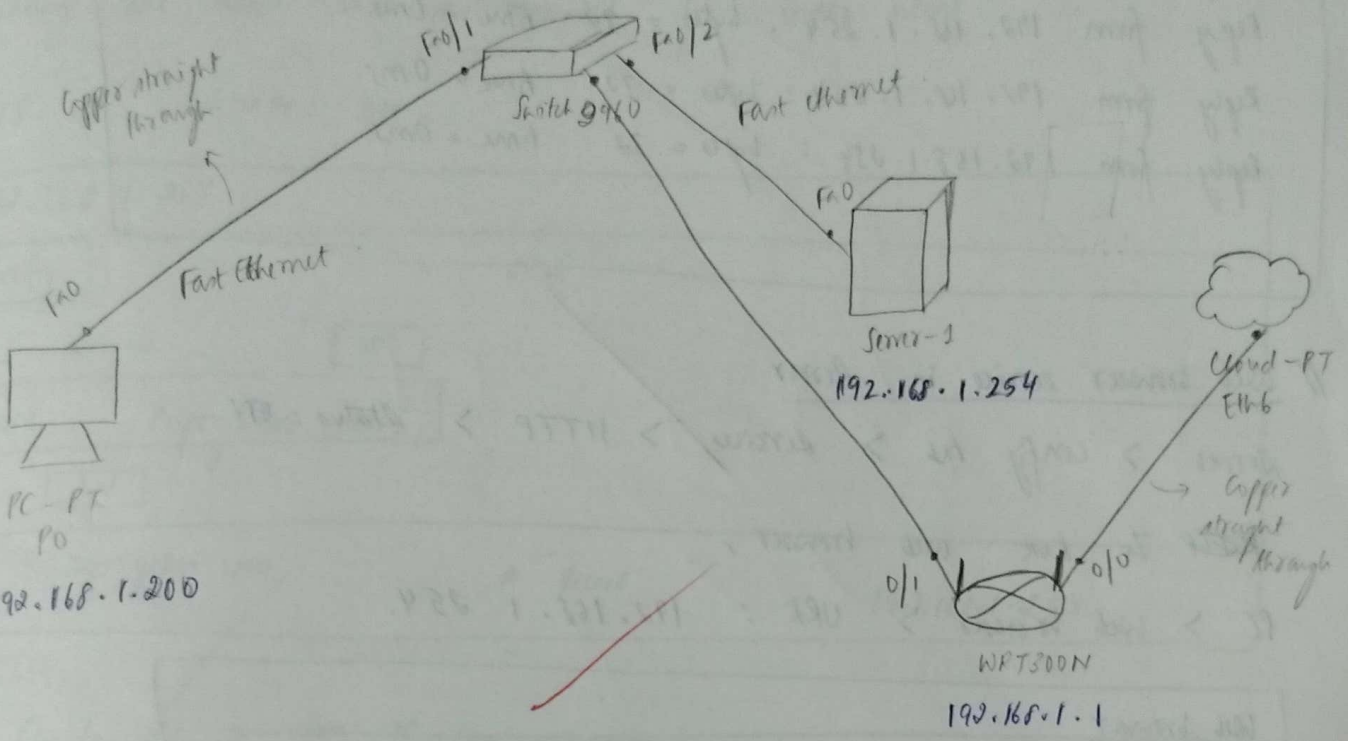
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial 2/0
C 30.0.0.0/8 is directly connected, Serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

8/11

05/11/24

LAB - 04

~~For~~ Demonstrate WEB ~~DNS~~ server and DNS using Packet Tracer —



① Place a generic PC, switch 2960, generic server in the logical workspace.

② Configure the end devices a.

PC 0	192.168.1.200
Server 1	192.168.1.254

③ Connect the PC to switch using straight through cable, switch to server also through straight through.

④ Ensure Port status is ON for all connections.

⑤

PC0 > Config.

IP address	192.168.1.100
Subnet mask	255.255.255.0

- Ping from PC to server (to ensure successful connection)

```
PC > ping 192.168.1.254.

Reply from 192.168.1.254 : bytes = 32  time = 0ms  TTL = 128
Reply from 192.168.1.254 : bytes = 32  time = 0ms
Reply from 192.168.1.254 : bytes = 32  time = 0ms
Reply from 192.168.1.254 : bytes = 32  time = 0ms
```

// Web browser service in Server

Server > config tab > Service > HTTP > Status : ON

To test web browser,

PC > Web browser > URL : 192.168.1.254.

Web browser

URL : http://192.168.1.254.

Cisco Packet Tracer

Welcome to Cisco Packet Tracer!
Quick Links:

② Connections including web wireless router and internet

Wireless router : 192.168.1.1

In Router > Configure.

Change IP addon from 192.168.0.1 ⟶ 192.168.1.1.

To check for successful connection, ping from PC to router.

PC > ping 192.168.1.1.1

- Configure the default gateway of PC as 192.168.1.1

// To view a different website called superyahoo.com

⇒ Setup a DNS server

Server > Service > ~~DNS~~ HTTP.

Change the html code as required for index.html.

DNS > Service: ON.

192.168.1.254

Config

DNS

DNS Service ON

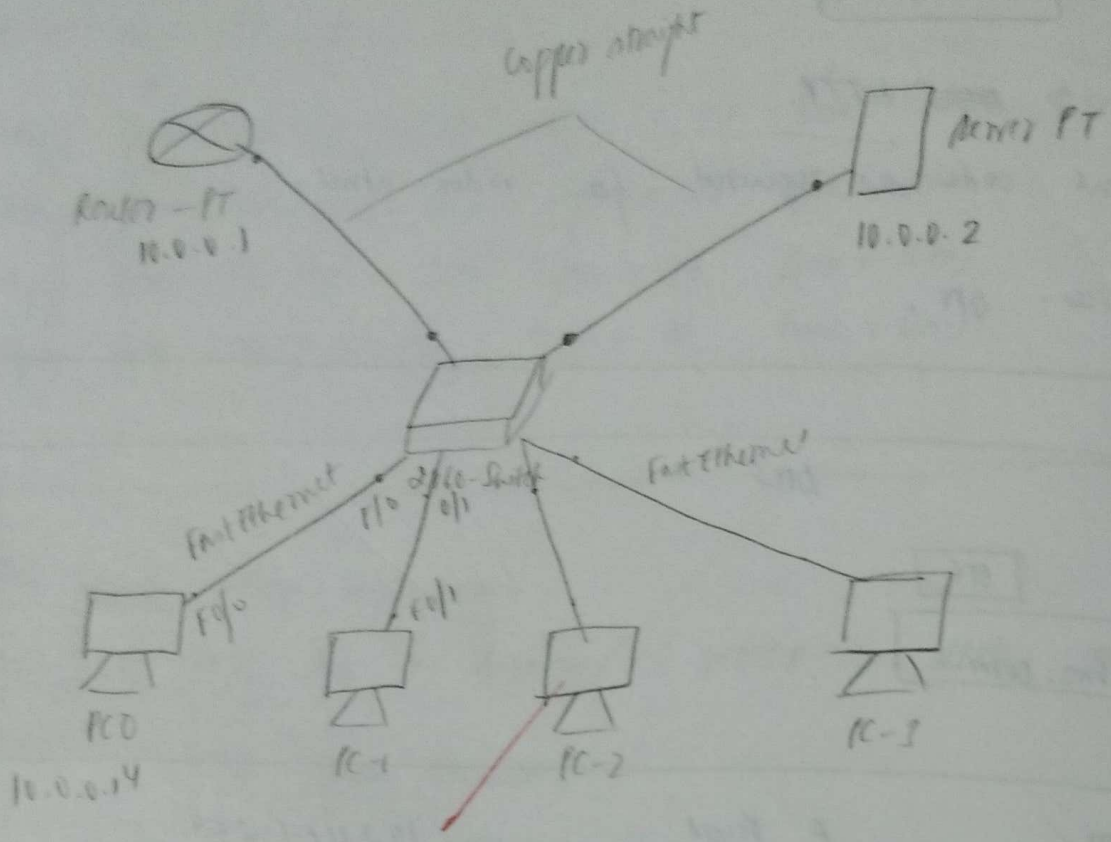
Name : Superyahoo.com

Add

1.	Superyahoo.com	A Record	192.168.1.254

• Go to PC > ~~IP~~ IP Configuration.

To configure DHCP within a LAN in packet tracer



- Create a LAN like the above topology with router, server, PC and switch.
- Configure router interface with IP address 10.0.0.1
  subnet mask 255.0.0.0
- In Server > Config, assign gateway as 10.0.0.1
  IP address 10.0.0.2
  subnet mask 255.0.0.0
- Click on DHCP, gateway - 10.0.0.1
  DNS - 10.0.0.2.
  Start IP address 10.0.0.10.
  Maximum number of users 500.
  TFTP server IP address 10.0.0.2

Server0 > Config

Service – DHCP

Service ⊙ On ○ Off

Pool Name – serverPool

Default Gateway – 10.0.0.1

DNS Server – 10.0.0.2

Start IP address – 10.0.0.16

Subnet mask – 255.0.0.0

Maximum number of users – 100

TFTP server – 10.0.0.2

Add Save

- Click on any PC > Desktop > IP Configuration > DHCP.
- Wait for some time till request is successful.

PC-0 > Desktop.

IP Configuration

⊙ DHCP DHCP request successful.

○ Static

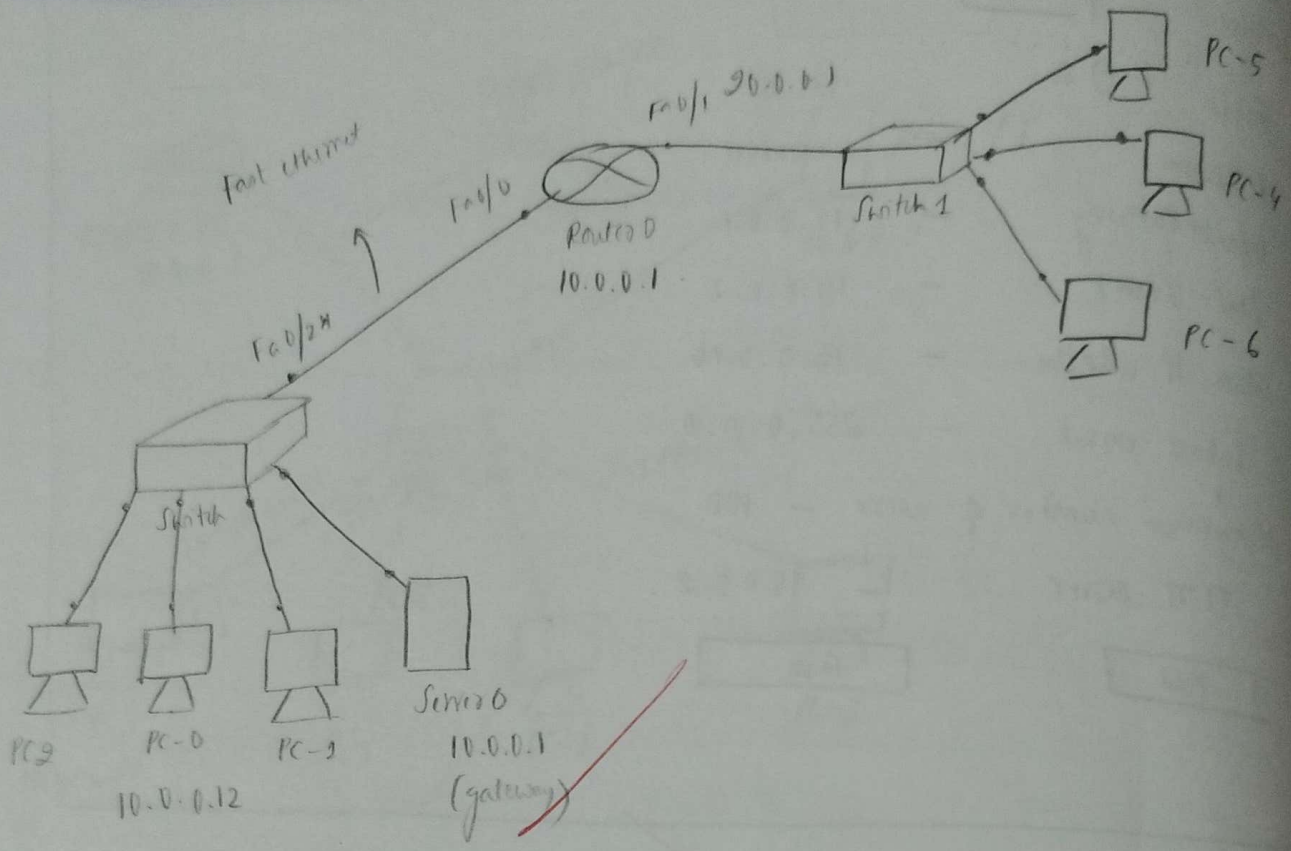
IP address 10.0.0.14

Subnet mask 255.0.0.0

Default gateway 10.0.0.1

DNS Server 10.0.0.2

To configure DHCP outside a LAN. using IP Hyper Address.



① Create a topology like the one above

② Configure router interface fastethernet 0/0 with IP address 10.0.0.1
fast ethernet 0/1 with 20.0.0.1
Subnet mask - 255.0.0.0

③ For server, gateway ip addrss is 10.0.0.1
Fastethernet ip addrss 10.0.0.2
subnet mask 255.0.0.0

④ Then for PC0 in LAN with server, click on
PC > Desktop > IP configuration > DHCP.
IP address is given from DHCP server for this PC

PC 0

Desktop > IP Configuration

IP Configuration

⊙ DHCP DHCP request successful.

○ Static

IP address	10.0.0.12
Subnet mask	255.0.0.0
Default Gateway	10.0.0.1
DNS Server	10.0.0.2

⑤ To get IP address for PC without server –

Server > Config > DHCP

Default gateway :	20.0.0.1
DNS Server	10.0.0.2
Start IP address	20.0.0.10
Subnet mask	255.0.0.0
TFTP	10.0.0.2

Add > Save.

⑥ In router, global configuration mode.

Router (config) # interface fastethernet 0/1

Router (config-if) # ip-helper address 10.0.0.2

Router (config-if) # exit.

15/11

15/11/24

LAB - 05

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

```
#include <iostream>
#include <string>
using namespace std;

int xor(char *ip, char *op, char *poly, int mode)
{
    strcpy(op, ip);
    if (mode)
    {
        for (int i = 0; i < strlen(poly) - 1; i++)
        {
            strcat(op, "0");
        }
    }
    int msg_len = strlen(op);
    int poly_len = strlen(poly);
    for (int i=0; i<= msg_len - poly_len; i++)
    {
        if (op[i] == '1')
        {
            for (int j = 0; j< poly_len; j++)
            {
                op[i+j] = (op[i+j] == poly[j])? '0' : '1';
            }
        }
    }
    for (int i = msg_len - poly_len + 1; i< msg_len; i++)
    {
        if (op[i] == '1')
        {
            return 0;
        }
    }
    return 1;
}
```

```
int main()
{
  char ip[100], op[200], recv[200];
  char poly[] = "10001000000100001";

  cout << "Enter the input message in binary : ";
  cin.getline(ip, 100);
  crc(ip, op, poly, 1);
  cout << "The transmitted message is : " << ip;
  for(int i = strlen(ip); i < strlen(op); i++)
  {
      cout << op[i];
  }
  cout << endl;

  cout << "Enter the received message in binary : ";
  cin.getline(recv, 200);

  if(crc(recv, op, poly, 0))
  {
     cout << "No error in data " << endl;
  }
  else
  {
     cout << "Error in data transmission has occurred" << endl;
  }
  return 0;
}
```

OUTPUT

* **Without error**

Enter the input message in binary : 10101100111111000111011010

The transmitted message is : 10101100111111000111011010000000

Enter the received message in binary : 10101100111111000111011010

No error in data.

* **With error**

Enter the input message in binary : 10101100111111000111011010

The transmitted message is : 10101100111111000111011010000000

Enter the received message in binary : 10101100111111000111011011

Error in data transmission has occurred.

Implement 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 ;

  cout << " Enter bucket size " ;
  cin >> bucket_size ;

  cout << " Enter input packet size " ;
  cin >> input_pkt_size ;

  cout << " Enter output packet size " ;
  cin >> output_packet_size ;

  cout << " Enter number of queries " ;
  cin >> no_of_queries ;

  storage = 0 ;

  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
    {
      cout << " Packet loss = " << input_pkt_size << endl ;
    }

    cout << " Buffer size = " << storage << " Out of bucket size = "
    << bucket_size << endl ;

    storage -= output_pkt_size ;
  }
  return 0 ;
}
```

OUTPUT

Enter bucket size : 10

Enter input packet size : 4

Enter output packet size : 1

Enter number of queries : 5

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

Packet loss = 4

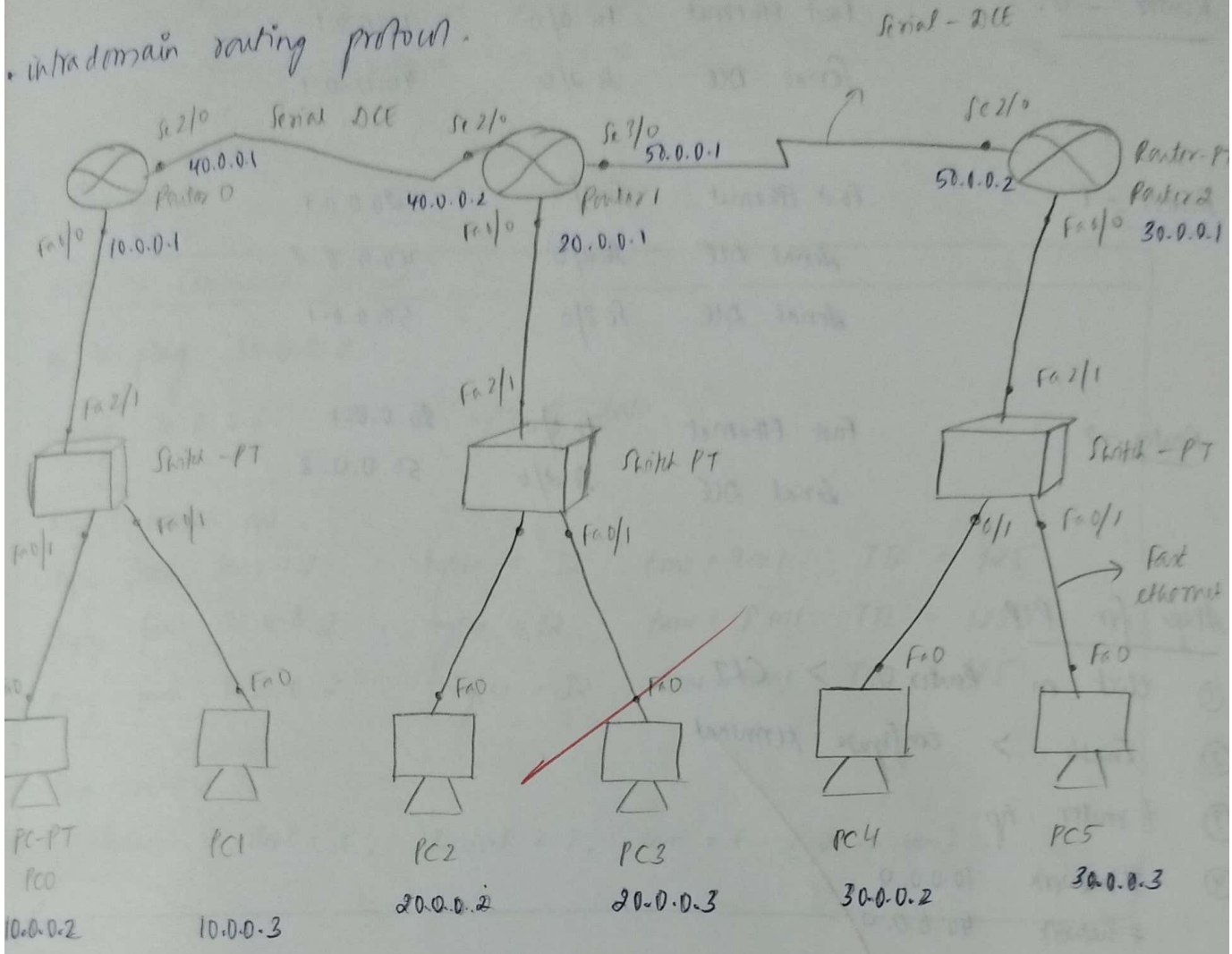
Buffer size = 8 out of bucket size = 10

20/11/24

LAB - 06

ROUTING INFORMATION PROTOCOL (RIP)

- intradomain routing protocol.



① Create a topology like the one above. Connect two end devices to a switch which is further connected to a router.

② The same setup is followed three times. The end devices are connected to the switch with copper straight through cable. The routers are connected with Serial - DCE connection

③ The end devices are configured with IP address, and gateway.

PC-0	10.0.0.2	]	Gateway 10.0.0.1
PC-1	10.0.0.3		
PC-2	20.0.0.2	]	Gateway 20.0.0.1
PC-3	20.0.0.3		
PC-4	30.0.0.2	]	Gateway 30.0.0.1
PC-5	30.0.0.3		

④ Each router is configured with ip address for Fa0/0 and Serial DCE.

Router - 0	Fast Ethernet	Fa 0/0	10.0.0.1
	Serial DCE	Se 2/0	40.0.0.1
Router - 1	Fast Ethernet	Fa 0/0	20.0.0.1
	Serial DCE	Se 2/0	40.0.0.2
	Serial DCE	Se 3/0	50.0.0.1
Router - 2	Fast Ethernet	Fa 0/0	30.0.0.1
	Serial DCE	Se 2/0	50.0.0.2

Steps for RIP

① Click on Router 0 > CLI

② Enable > configure terminal

③ #router rip

④ #network 10.0.0.0
#network 40.0.0.0

⑤ exit

Router0 > CLI

Router > enable
Router # configure terminal
Router (config) # router rip
Router (config-router) # network 10.0.0.0
Router (config-router) # network 40.0.0.0
Router (config) # exit
Router # show ip route

C 10.0.0.0/8 is directly connected, Fast Ethernet 0/0
R 20.0.0.0/8 via 40.0.0.2, Serial 2/0
R 30.0.0.0/8 via 40.0.0.2, Serial 2/0
C 40.0.0.0/8 is directly connected, Serial 2/0
R 50.0.0.0/8 via 40.0.0.2, Serial 2/0

similarly for Router1

Router (config-router) # network 20.0.0.0
network 40.0.0.0
network 50.0.0.0

② Pinging

PC0 > Command prompt

PC > ping 30.0.0.2

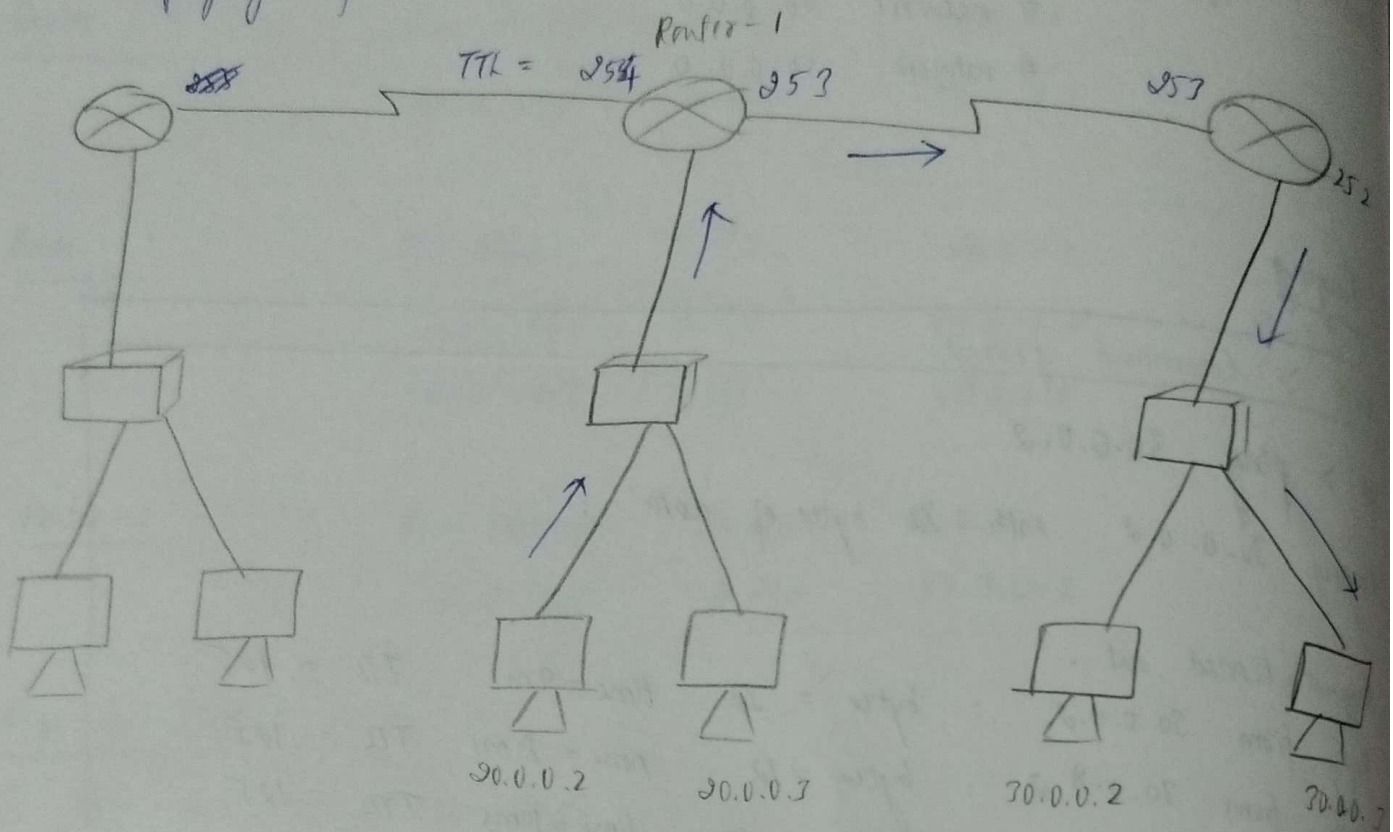
Pinging 30.0.0.2 with 32 bytes of data :

Request timed out.
Reply from 30.0.0.2 : bytes = 32 time = 9ms TTL = 125
Reply from 30.0.0.2 : bytes = 32 time = 8 ms TTL = 125
Reply from 30.0.0.2 : bytes = 32 time = 10ms TTL = 125

Ping statistics :
Packets : Sent = 4, Received = 3, Lost = 1 (25% loss)

* TTL

When pinging from 20.0.0.2 to 30.0.0.3



Value of TTL decrease by 1 in outbound communication.

At inbound, the value of TTL remains same whereas once the packet cross the ~~outb~~ router, value of TTL is decremented by 1.

22/11

29/11/24

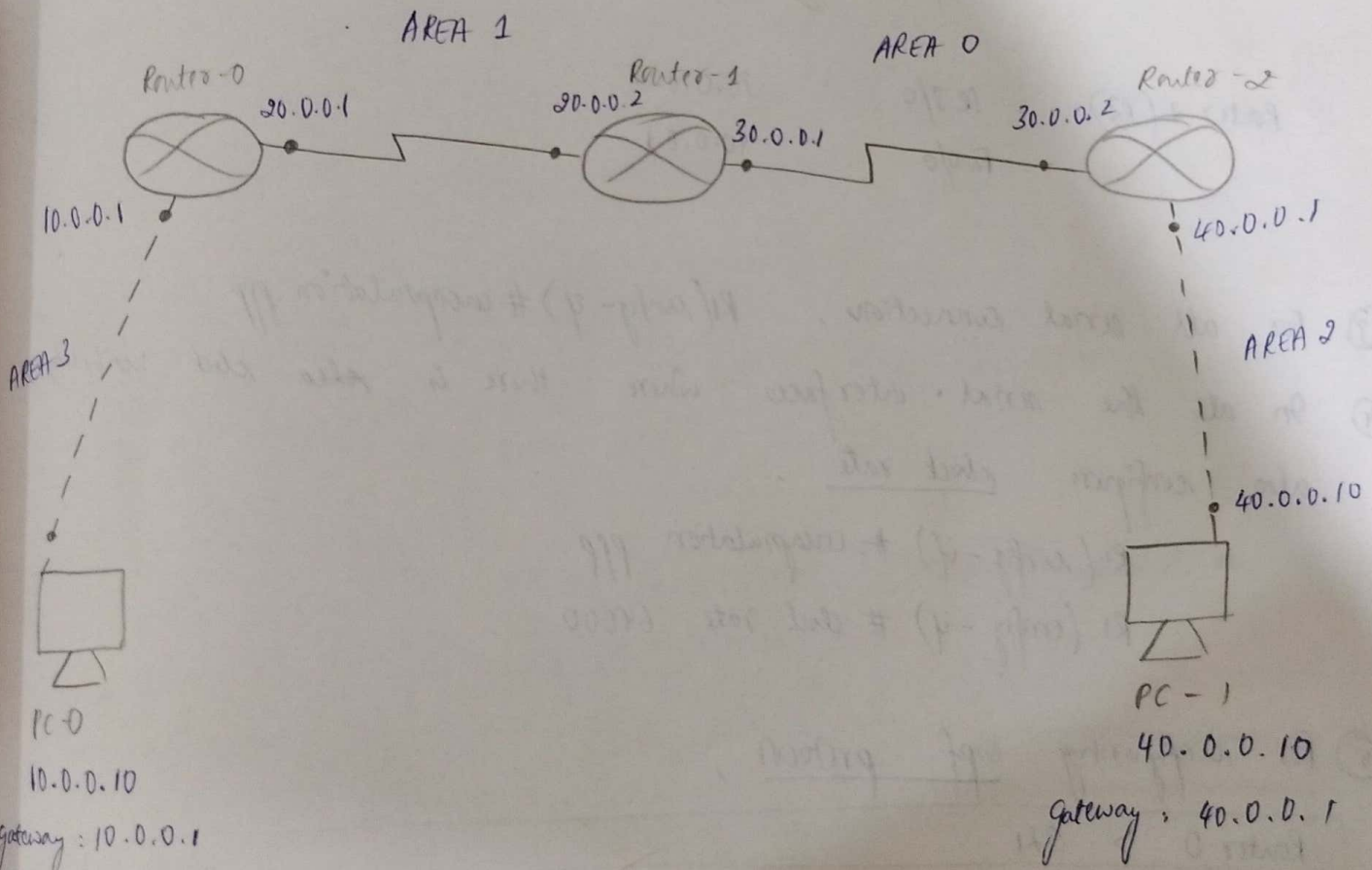
LAB-07

Open Shortest Path First (OSPF)

- dynamic routing protocol.
- intradomain routing
- to establish autonomous system
- Areas - instead of one controlling agent for all networks, divide into areas

Area 0 is called backbone network.

TOPOLOGY



0.255.255.255 - focus is on network (instead of default address for host)

Clock rate only in those interfaces which has clock visibility.

0 - OSPF

① Create a topology with two end devices and three gateways.

② Configure all IP address and gateway.

PC 0	Fa 0/0	10.0.0.10
	gateway	10.0.0.1
PC-1	Fa 0/0	40.0.0.10
	default gateway	40.0.0.1
Router 0 (R1)	Fa 0/0	10.0.0.1
	⑤ Se 2/0	20.0.0.1
Router 1 (R2)	Se 2/0	20.0.0.2
	Se 3/0	30.0.0.1
Router 2 (R3)	Se 3/0	30.0.0.2
	Fa 0/0	40.0.0.1

③ For all serial connection, R1(config-if)# encapsulation ppp

④ On all the serial interfaces where there is clock with also configure clock rate.

R1(config-if)# encapsulation ppp
R1(config-if)# clock rate 64000

⑤ For configuring ospf protocol,

```
Router 0 > CLI
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.0.0.0 0.255.255.255 area 3
R1(config-router)# network 20.0.0.0 0.255.255.255 area 1
R1(config-router)# exit
```

Router2 > CLI

```
R2(config-if) # router ospf 1
R2(config-router) # network 20. router-id 2.2.2.2
R2(config-router) # network 20.0.0.0 0.255.255.255 area 1
R2(config-router) # network 70.0.0.0 0.255.255.255 area 0
R2(config-router) # exit.
```

⑥ To configure loopback address to routers —

```
R1(config-if)# interface loopback 0
R1(config-if) # ip add 172.16.1.252 255.255.0.0
R1(config-if) # no shutdown

R2(config-if) # interface loopback 0
R2(config-if) # ip add 172.16.1.253 255.255.0.0
R2(config-if) # no shutdown

R3(config-if) # interface loopback 0
R3(config-if) # ip add 172.16.1.254 255.255.0.0.
R3(config-if) # no shutdown
```

⑦ ~~Routing to~~ Create a virtual link to connect area 2 to area 0. between R1, R2.

In router R1 > CLI

```
R1(config) # router ospf 1
R1(config-router) # area1 virtual-link 2.2.2.2
R1(config-router)#

* 1:03:52 : %. OSPF-5-ADJCHG : Process 1, Nbr 2.2.2.2 on
  OSPF_VL0 from LOADING to FULL Loading Done.
```

On router R2 > CLI

R2 (config - router)#

R2 (config - router)# area 1 virtual-link 1.1.1.1

R2 (config - router)# exit.

01:01:50 : %-OSPF-5-ADJCHG : Process 1, Nbr 1.1.1.1 on OSPF

④ ~~R3 #~~ show ip route of router R3

R3 > CLI

R3 # show ip route.

O IA 30.0.0.0/8 [110/128] via 30.0.0.1 00:01:50, Serial3/0

C 40.0.0.0/8 is directly connected, FastEthernet0/0

O IA 10.0.0.0/8 [110/129] via 30.0.0.1 00:01:50 Serial3/0

C 30.0.0.0/8 is directly connected, serial 3/0.

C 172.16.0.0/16 is directly connected, Loopback 0

③ Check connectivity

PC0 > Command prompt

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 = 8ms TTL = 125

Reply from 40.0.0.10 : bytes = 32 time = 7ms TTL = 125

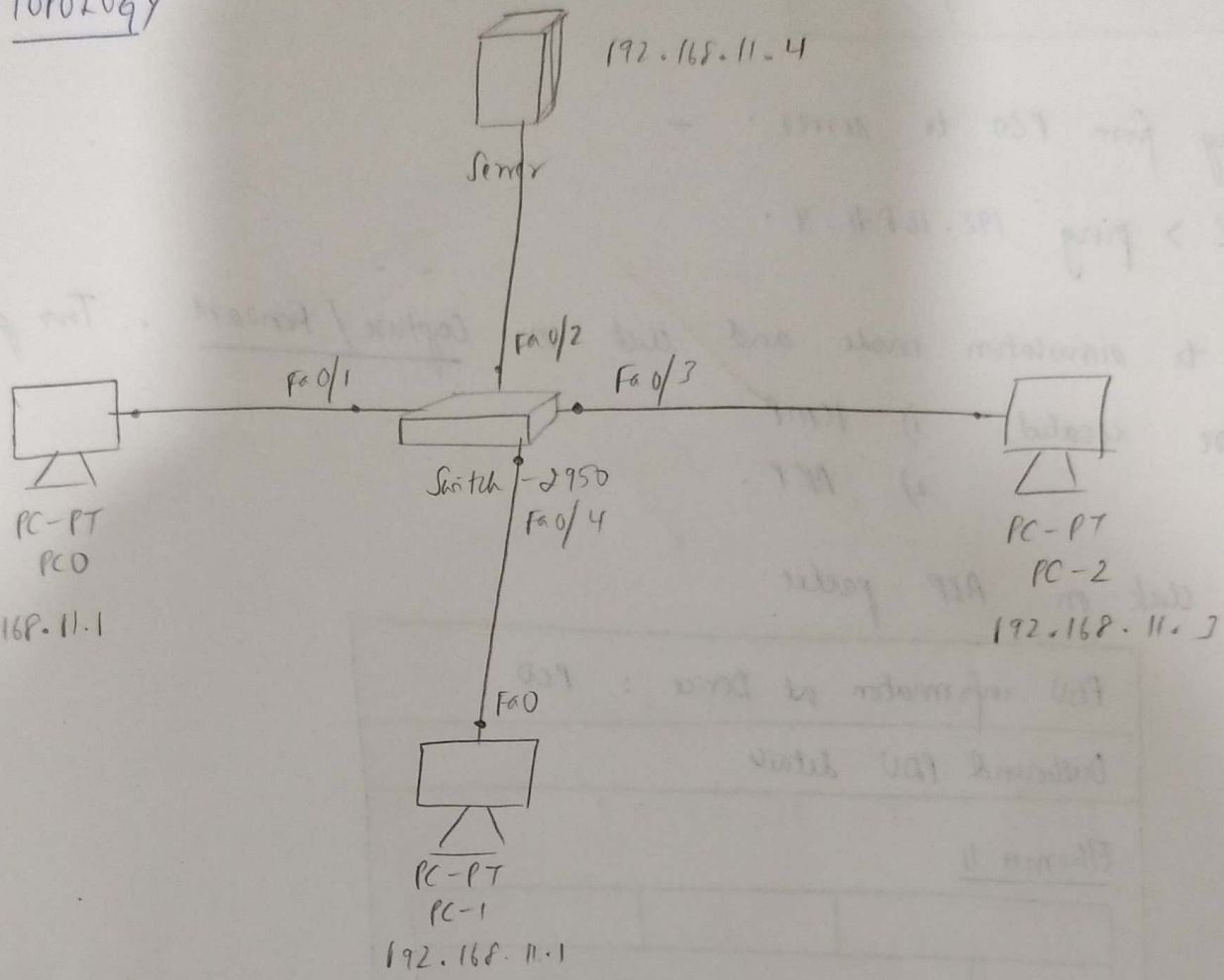
Reply from 40.0.0.10 : bytes = 32 time = 2ms TTL = 125

20/12

LAB - 08

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

* TOPOLOGY



① Create a topology similiar to the above diagram with 3 end devices and a server. Connect all of this via Switch - 2950

PC0 — 192.168.11.1
PC1 — 192.168.11.2
PC2 — 192.168.11.3
Server — 192.168.11.4

② Configure IP addresses to all devices.

③ Go to the simulation on the right hand side of the window and click on Inspect and right click on PC0 > ARP Table.

④ Observe that there are no entries in the ARP table of the PC or server

⑤ To the Alternate way to check for ARP entries,

PC0 > Command prompt
PC > arp -a No ARP entries found

⑥ Ping from PC0 to server :-

PC > ping 192.168.11.4 .

⑦ Go to simulation mode and click on Capture / Forward . Two are created 1) ICMP
2) ARP .

⑧ To click on ARP packet .

PDU information at Device : PC0
Outbound PDU details
Ethernet II
ARP
OPCODE : 0x1
192.168.11.1 SOURCE IP
TARGET IP : 192.168.11.4

⑨ Keep clicking on Capture / Forward to see all simulations and track ICMP packet movements.

Observation

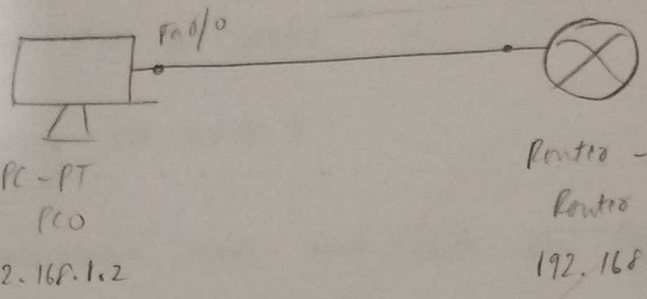
ARP Table for Server0	
IP address	Hardware address
192.168.11.1	0DE0.F9AC.14[illegible]

ARP table for PC0	
IP address	Hardware address
192.168.11.4	0090.2[illegible].DA[illegible]

To understand the guration of TELNET by accessing the routers server room from a PC in IT office —

TELNET Protocol

* TOPOLOGY



① Create a topology as above diagram with a PC and generic router

② Configure IP addresses and gateway.

PC-0 IP: 192.168.1.2

Gateway : 192.168.1.1

Router IP : 192.168.1.1

③ Configure the router in CLI

Router0 > CLI

```
Router > en
Router # config t.
Router (config-t) # hostname R1
R1 (config) # enable secret cp
R1 (config) # ip addres 192.168.1.1 255.255.255.0
R1 (config-if) # no shut
R1 (config-if) # line vty 0 5
R1 (config-line) # login.
% Login disabled on line 192, until 'password' is set
    :
R1 (config-line) # password cp
R1 (config-line) # exit
```

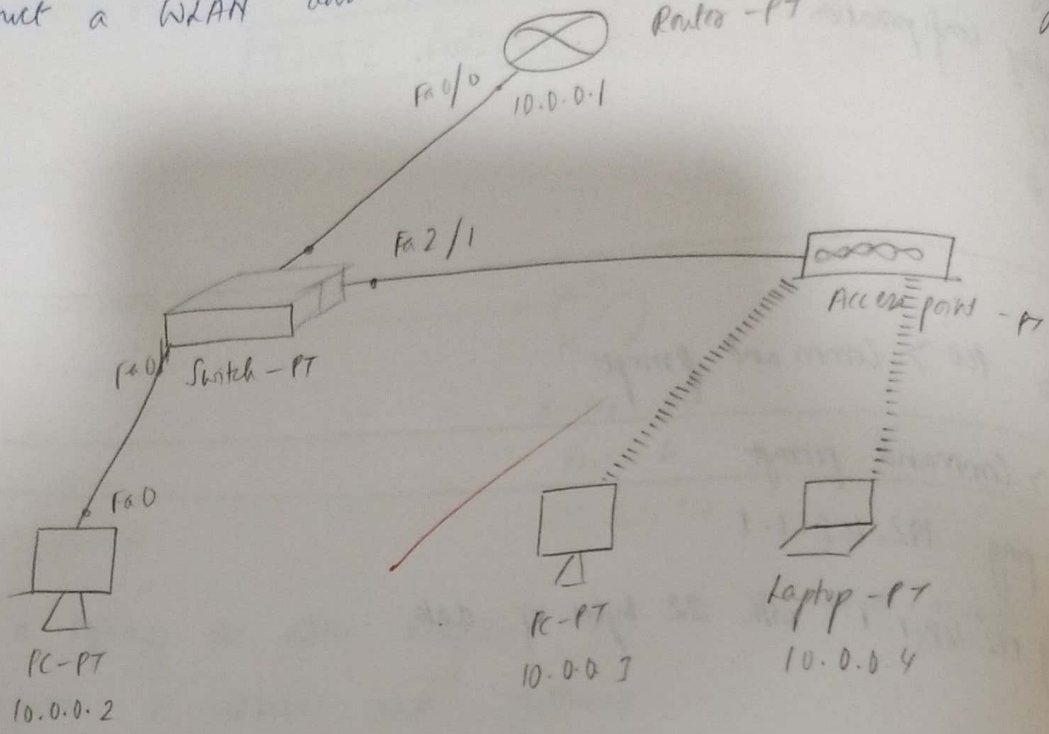
```
R1(config)# exit
R1# wr
Building configuration....
[OK]
R1#
```

④ Go to PC0 > Command prompt

```
PC0 > Command prompt

PC > ping 192.168.1.1

Pinging 192.168.1.1 with 32 byte of data :

:

PC > telnet 192.168.1.1

Trying 192.168.1.1 ... Open

User Access Verification

Password :
Password : tp
R1 > en
Password : tp
Password : tp
R1 #
```

WLAN

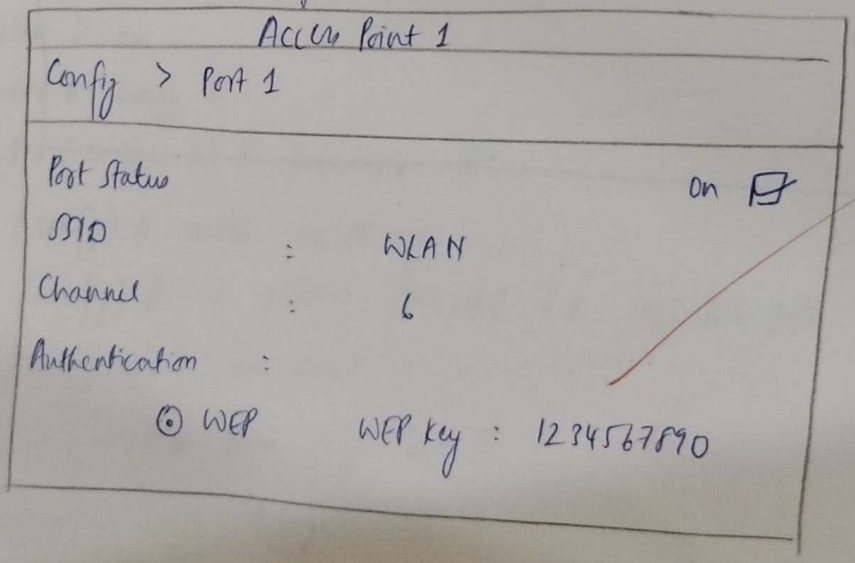
To construct a WLAN and make the nodes communicate wirelessly



① Construct the above topology and configure IP addresses.

PC 0	-	10.0.0.2
PC 1	-	10.0.0.3
Laptop	-	10.0.0.4
Router	-	10.0.0.1

② Click on Access point 1 -> Port > SSID Name



③ Configure PC2 and laptop with wireless standards.

④ In the Physical tab, switch off the device. Drag the existing PT-HOST-NM-1AM to the left hand panel. Drag WMP300N wireless interface to the empty port. Switch on device

③ In the Config tab, a new wireless interface is added.

④ Configure SSID, WEP, WEP key, gateway for both PC and Laptop

PC0 > Config > Wireless0

Wireless 0

Port Status ☑ On

SSID WLAN

Authentication

◉ WEP WEP key 1234567890

IP Configuration

○ DHCP

◉ Static

IP address 10.0.0.3

Subnet mask 255.0.0.0

⑤ Ping from all devices to check connection.

PC0 > Command Prompt

PC > ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data :

Reply from 10.0.0.3 : bytes = 32 time = 20ms TTL=128

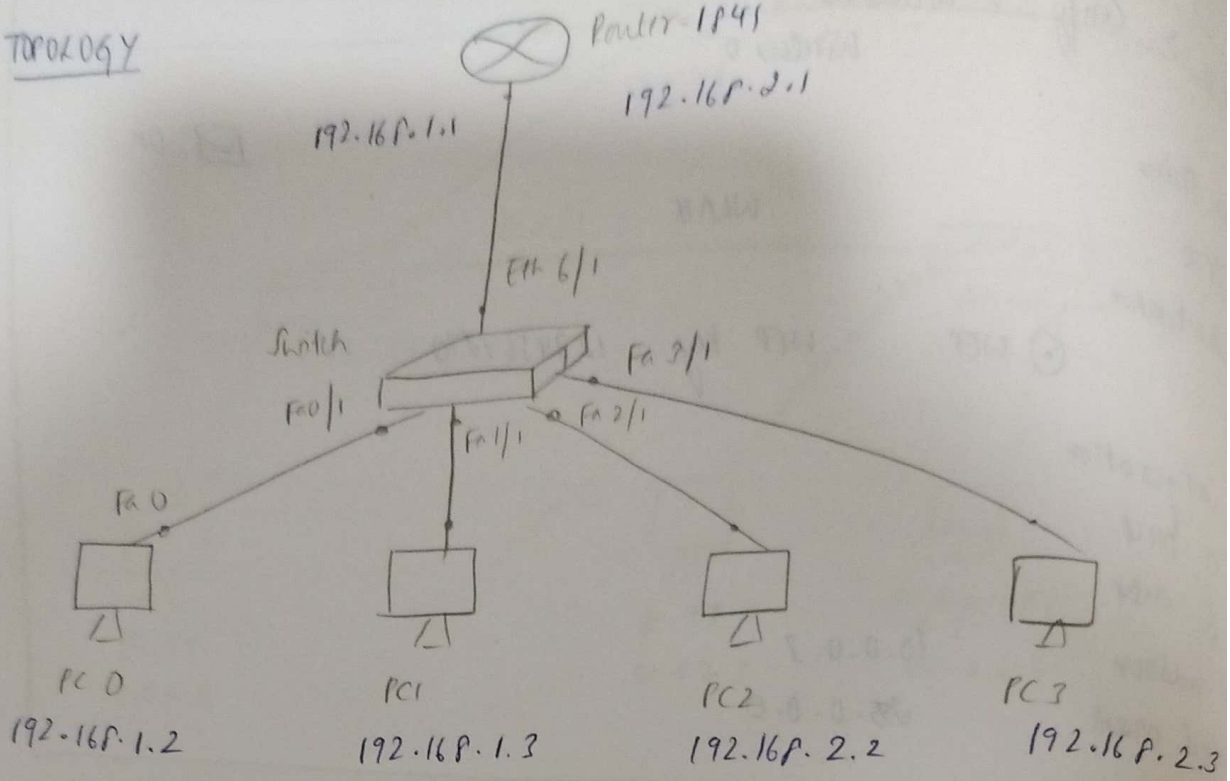
⋮

PC > ping 10.0.0.4

VLAN

To construct a VLAN, and make PC's communicate among a VLAN

* TOPOLOGY



① Construct the above topology with a generic switch, 4 end devices and a 1841 router. Add an extra port in switch to connect to router (Eth 6/1)

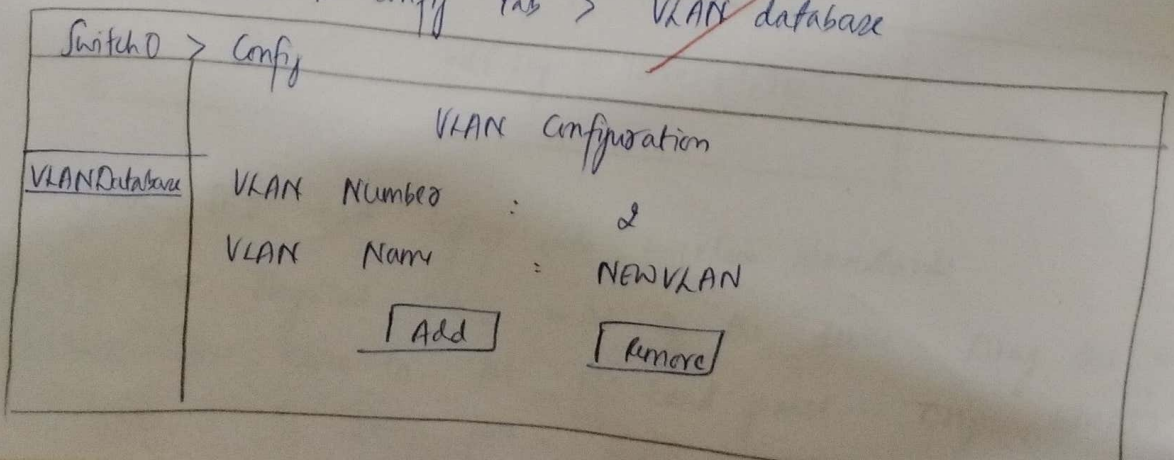
② Configure IP addresses for all devices.

PC 0 — 192.168.1.2]
PC 1 — 192.168.1.3] Gateway : 192.168.1.1

PC 2 — 192.168.2.2]
PC 3 — 192.168.2.3] Gateway : 192.168.2.1

Router-1841 — 192.168.1.1

③ On the switch > Config tab > VLAN database



④ Select the interface – ethernet 6/1 and make it trunk and check all in VLAN field.

⑤ Now, for fastethernet 2/1 and fastethernet 3/1.
select Access and VLAN : 2

⑥ Go to Router > Config tab > VLAN Database.
Enter the name and number of VLAN created.

Router > CLI

```
Router (vlan) # exit
APPLY completed.
Exiting...
Router # config t
Router (config) # interface fastEthernet 0/0.1
Router (config-subif) # encapsulation dot1q 2
Router (config-subif) # ip address 192.168.2.1 255.255.255.0
Router (config-subif) # no shut
Router (config-subif) # exit
Router (config)) # exit
```

⑦ Ping a message from one device to another.

CYCLE - 2

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 —

Server TCP.py

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

ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket (AF_INET, SOCK_STREAM)
clientSocket.connect ((serverName, serverPort))
sentence = input ("\n Enter file name: ")
```

```
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print('\n From Server : \n')
print(fileContents)
clientSocket.close()
```

OUTPUT —

```
The server is ready to receive.        // run ServerTCP.py

Enter file name : ServerTCP.py         // run ClientTCP.py

from server :

from socket import *
serverName =
.
.
.
connectionSocket.close()

Sent contents of ServerTCP.py          // in ServerTCP.py
```

Using UDP sockets, write a client - server program to make client send the file name and the server to send back the contents of the requested file if present.

~~CLIENT~~

clientUDP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input('\n Enter file name : ')
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print('\n Reply from server')
print(filecontents.decode('utf-8'))
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = ~~'127.0.0.1'~~ 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print('\n The server is ready to receive')
while 1 :
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode('utf-8')
    file = open(sentence, "r")
    con = file.read(2048)
    serverSocket.sendto(bytes(con, 'utf-8'), clientAddress)
    print('\n Sent contents of ', end = ' ')
    print(sentence)
```

26/8