VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

MADHURIKA PRIYA(1BM22CS344)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Sep 2024-Jan 2025

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**COMPUTER NETWORKS**" carried out by **MADHURIKA PRIYA(1BM22CS344),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Computer Networks Lab - (23CS5PCCON)** work prescribed for the said degree.

Spoorthi D M
Associate Professor,
Department of CSE,
BMSCE, Bengaluru

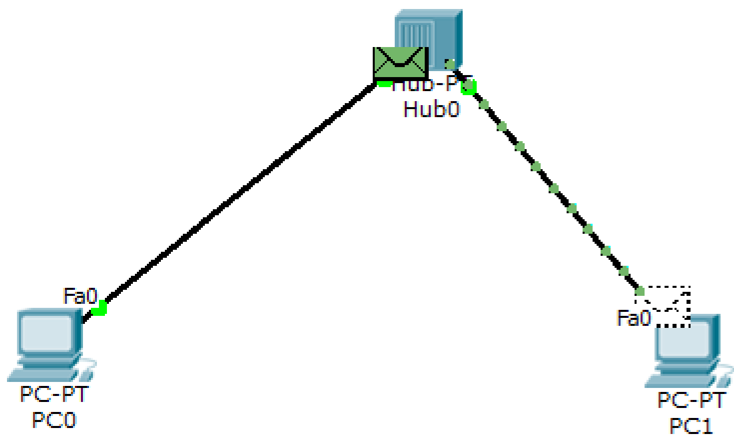
Dr. Kavitha Sooda
Professor and Head,
Department of CSE
BMSCE, Bengaluru

TABLE OF CONTENTS

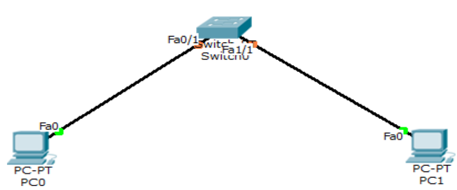
S. No	Title	Page No.
1	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.	1
2	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	8
3	Configure default route, static route to the Router	12
4	Configure DHCP within a LAN and outside LAN.	16
5	Configure RIP routing Protocol in Routers	19
6	Configure OSPF routing protocol	22
7	Demonstrate the TTL/ Life of a Packet	27
8	Configure Web Server, DNS within a LAN.	30
9	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	33
10	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	36
11	To construct a VLAN and make the PC’s communicate among a VLAN	39
12	To construct a WLAN and make the nodes communicate wirelessly	43
13	Write a program for error detecting code using CRC-CCITT (16-bits).	46
14	Write a program for congestion control using Leaky bucket algorithm	49
15	Using TCP/IP sockets	52
16	UDP Sockets	54

Program 1

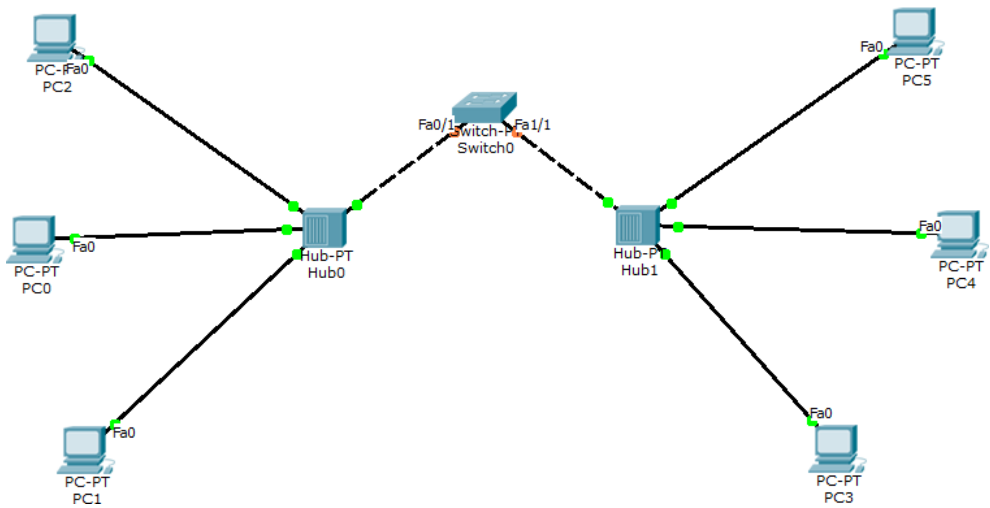
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.



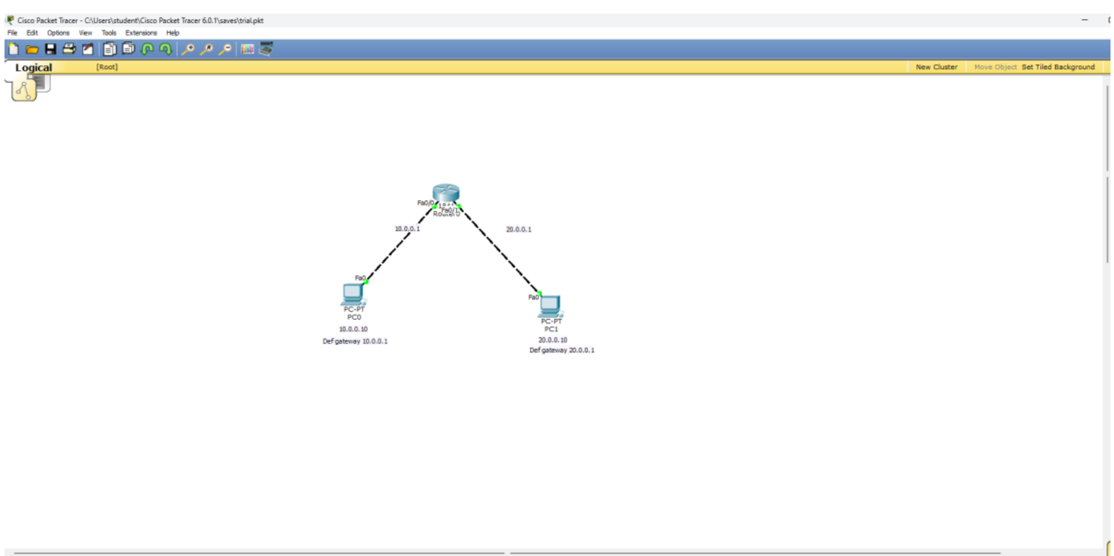
1.1 Hub



1.2 Switch Topology

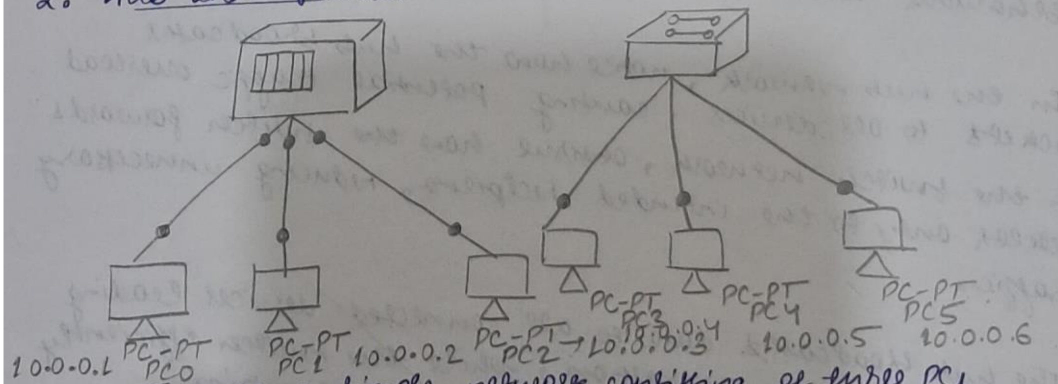


1.3 Switch and Hub Topology



1.4 Router Topology

2. Hub and Switch.



Aim: To create a simple network consisting of three PCs connected to a central hub and another network with three PCs connected to a switch. This configuration will help observe the behaviour of data transmission using hub & switch devices.

Topology:

1. Hub Network: Three PCs (PC0, PC1, PC2) are connected to a hub (Hub 0) using straight-through Ethernet cables

IP addresses: PC0 = 10.0.0.1, PC1 = 10.0.0.2, PC2 = 10.0.0.3

2. Switch Network: Three PCs (PC3, PC4, PC5) are connected to a switch (Switch 0) using straight-through ethernet cables.

IP addresses: PC3 = 10.0.0.4, PC4 = 10.0.0.5, PC5 = 10.0.0.6

Procedure:

1. Add 1 hub, 1 switch and 6 PCs (PC0, PC1, PC2 for the hub; PC3, PC4, PC5 for the switch) to the cisco packet tracer workspace.

2. Use copper straight-through cables to connect PC0, PC1, and PC2 to Hub 0. ||ly connect PC3, PC4 & PC5 to Switch 0 using same type of cables.

3. Assign IP addresses to each PC & obtain subnet mask.

4. Switch to simulation mode to observe data traffic behaviour when packets are sent between the devices.

5. In the hub network, notice how the hub broadcasts packets to all devices, causing potential traffic overload. In the switch network, observe how the switch forwards packets only to the intended recipient, reducing unnecessary traffic.

6. The hub broadcasts data to all connected devices leading to more network congestion, while the switch efficiently sends data only to the correct device, optimizing performance.

Observation:

(5)

1. The hub broadcasts packets to all devices, which may cause unnecessary traffic.

2. The switch forwards packets only to the appropriate device by learning MAC addresses, making it more efficient in reducing traffic.

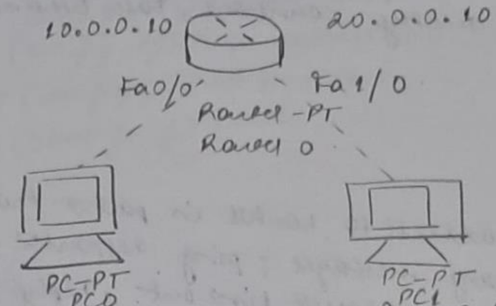
Difference between Hubs and Switches.

Hubs	Switches
1. Hub broadcast data to all devices	1. Switches send it only to the destination.
2. Hubs create more traffic	2. Switches reduces traffic by directing data.
3. Hubs work at physical layer.	3. Switches operate at the data links layer.
4. Hubs are slower due to shared bandwidth.	4. Switches are faster with dedicated bandwidth.
5. Hubs are cheaper.	5. Switches are more expensive but more efficient.

LAB-1 . (7)

Aim To connect 2 PCs with 2 different routes using Router.

Topology



Default Gateway: 20.0.0.10

Default Gateway: 10.0.0.10

Procedure: (i) Add 2 PCs and 1 generic router

(ii) Connect PC to router using cross-over wire.

(iii) Set up IP addresses, gateway, subnet mask of both the PCs (in fast ethernet & settings (gateways))

(iv) In router set up its IP address (both -based } on 1/0, 0/0)

(v) Open CLI of router and type:

continue with configuration dialog? n .

Router # enable .

Router # config terminal

Router (config) # interface Fast-Ethernet 0/0 .

Router (config-if) # ip address 10.0.0.10 255.0.0.0

Router (config) # no shutdown .

exit

do the same for the ~~other Fast Ethernet 1/0~~ .

(vi) Click on PC $\xrightarrow{}$ PC0 Desktop → Command Prompt

type ping 20.0.0.1 (this sends packet from PC0 to PC1)

Observation:

This sends 32bytes of data

(sent = 4, lost = 1)

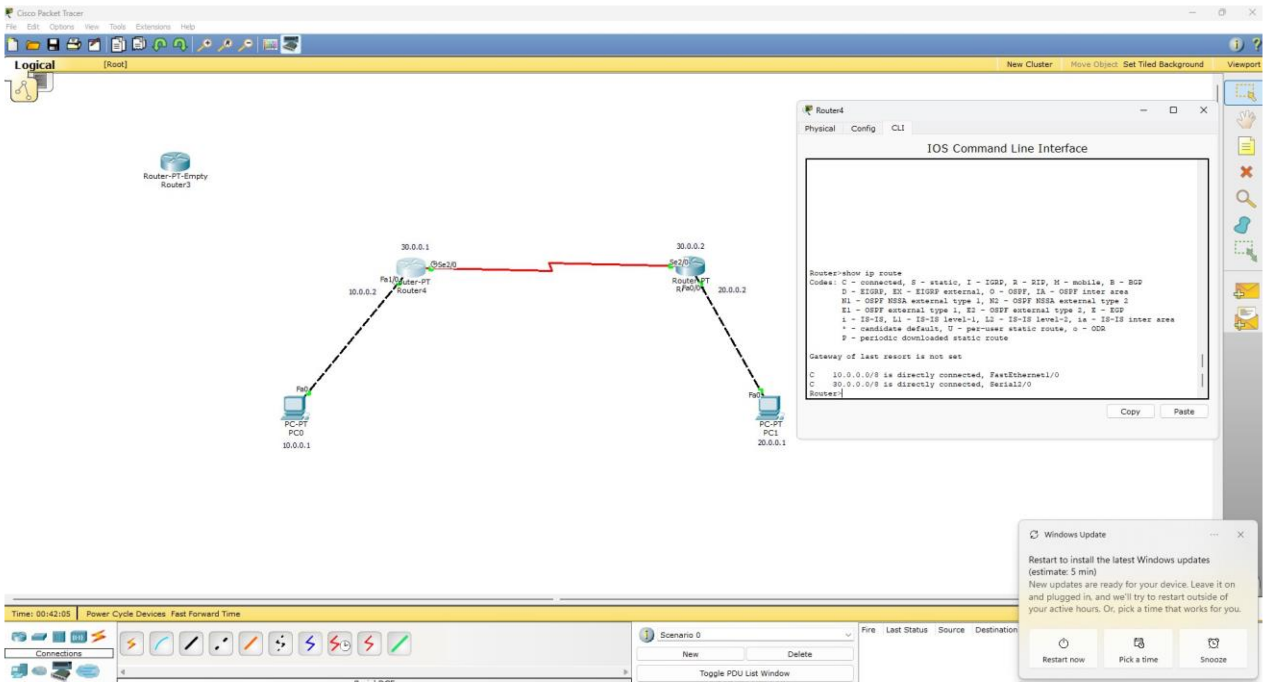
In router CLI

Router (config)#exit.
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, FastEthernet 1/0.

Program 2

Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply



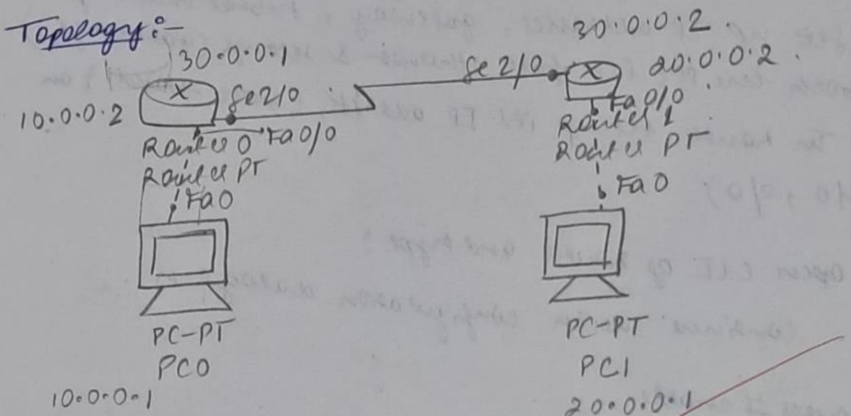
2.1 Router Connection Topology

LAB-2

experiment-2

aim :- configure IP address to routers in packet tracer. Explore the following messages : ping responses, destination unreachable, request time out, reply.

Topology:-



Procedure:-

(1) Create a 2 network of 10.0.0.0 and 20.0.0.0 by connecting a end device to the router.

(2) Set up IP address, gateway, subnet mask of both networks.

(3) In router set up its IP address (fast ethernet).

(4) Set up another network 30.0.0.0 by connecting the 2 routers.

Ion

Commands : for router in CLI.

Router # enable
Router # config terminal
Router (config) # interface fastEthernet 0/0.
Router (config) # no shutdown.
exit.
do the same for router 2.

Output :- Ping 20.0.0.1 .
pinging 20.0.0.1 with 32 bytes of data:
Reply from 10.0.0.3: Destination host unreachable.
Reply from 10.0.0.3: Destination host unreachable.
Reply from " " " " " .
"
Request time out.
Ping statistics for 20.0.0.2 :
Packets: sent = 4, Recieved = 0, Lost = 4 (100% loss),

Ping 20.0.0.2
pinging 20.0.0.1 with 32 bytes of data:
Reply from 10.0.0.3: Destination host unreachable.
" " " " " "
" " " " "
Request time out.
ping statics for 20.0.0.2:
Packets: sent = 4, Recieved = 0, Lost = 4 (100% loss).

Ping. 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data.
Reply from 10.0.0.3: bytes = 32 time = 0 ms TTL = 255
" " " "
" " " "
" " " .
Ping statics for 10.0.0.3.
packets: sent = 4, recieved = 4, lost = 0 (0% loss),
Approximate round trip times in ms:-
Min = 0ms, max = 0ms, Avg = 0ms.

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 = 225 ..
" " " "
" " " "
" " " "

ping statistics for 30.0.0.1:
packets: sent = 4, recieved = 4, lost = 0 (0% loss),
Approximate round trip time in mms:-
Min = 0ms, Max = 0ms, Avg = 0ms .

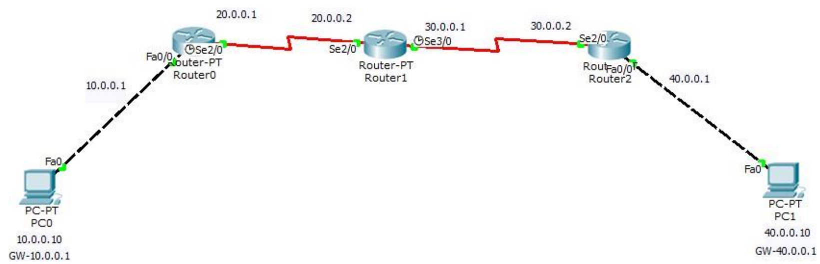
Ping 30.0.0.2 .
Pinging 30.0.0.2 with 32 bytes of data :
Request time out
"
"
"
"
Ping statistics for 30.0.0.2:
packet: sent = 4 , recieved = 0, lost = 4 (100% loss),

Observation:-

* On pinging PC0 to PC1 , it shows that it is unable to ping and shows the devices are not reachable .
shows - "destinations not unreachable"
* On pinging PC0 to Fa0/0 port of router the message is pinged (pinging 10.0.0.2)
* On ping PC0 to Se2/0 router 2, the message is not pinged.

Program 3

Configure default route, static route to the Router



3.1 Static Routing Topology

Experiment-~~2~~3. (11)

10/10.

AIM:- configure default route, static route to router.
configure static connection to route.

TOPOLOGY:- Same as Experiment-2.

PROCEDURE:- Go to CLI of router 0 and in CLI ~~enter~~ enter

Router # enable.
Router # config terminal.
Router (config) # ip route 20.0.0.0 255.0.0.0.
Router (config) # ip route 30.0.0.2 255.0.0.0.

Router (config) # exit.

- Repeat the same for router 0 by changing. 20 to 10 and 30.0.0.2 to 30.0.0.1.

OBSERVATION :- In CLI
\# show ip route.
(for router 2).
S 10.0.0.0 /8 [1/0] via 30.0.0.1
C 20.0.0.0/8 is directly connected, fast ethernet 0/0
C 30.0.0.0/8 is directly connected, serial 2/0.

OUTPUT :- On pinging end devices, the message is sent, also router all also

Router # enable .
Router # config terminal . (18)
Router (config) # interface serial 2/0 .
Router (config) # ip address 20.0.0.2 255.0.0.0 .
Router (config # no shutdown .
Router (config) # exit .

(7) Now, a serial connection b/w all the routers is established.
(8) Next step is to default route, router 0 and router 2.
In Router 0 CLI enter the below commands:

Router # enable .
Router # config terminal .
Router (config) # ip route 0.0.0.0 0.0.0.0 20.00.2 .
Router (config) # exit .

Repeat the same for the router 2 for ip route 0.0.0.0 0.0.0.0 30.0.0.1 .

OBSERVATION :-

In Router 0 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 .

In Router 2 CLI :-

Router # show ip route

C 30.0.0.0/8 is directly connected, serial 3/0 .
C 40.0.0.0/8 is directly connected, FastEthernet 0/0 .
S* 0.0.0.0/0 [1/0] via 30.0.0.1 .

In Router 1 CLI :-

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 ,

14

→ On pinging from one end device to the other,

ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1 bytes=32 time=2ms TTL=253.

" " " " " " "

" " " " " " "

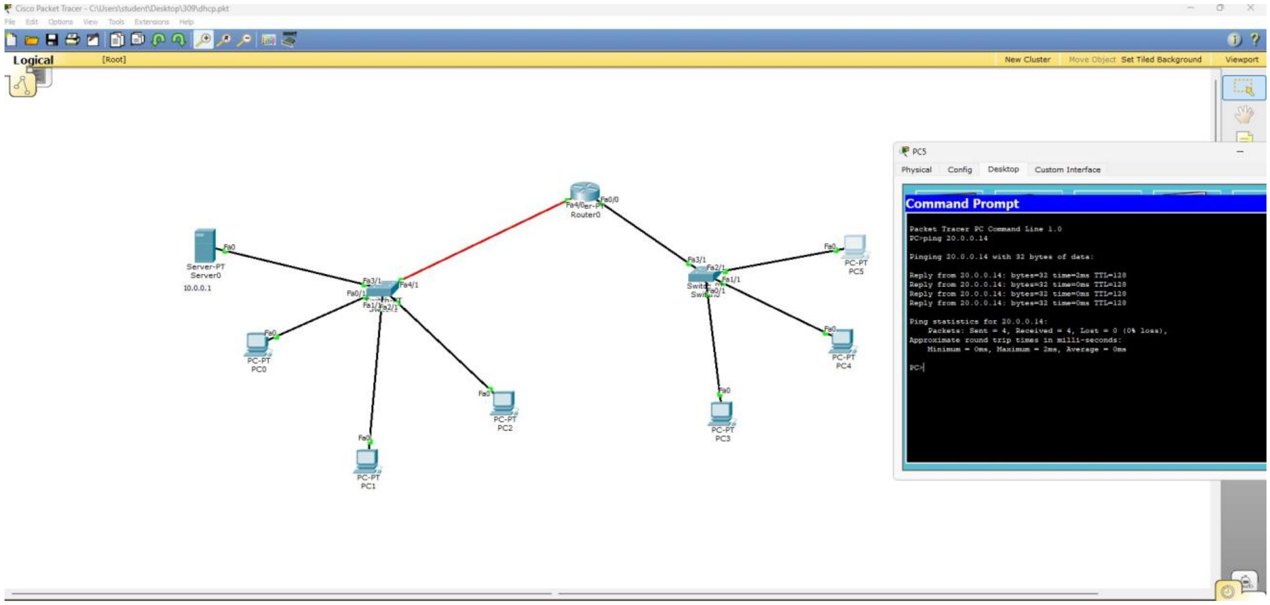
" " " " " " "

Ping statistics for 40.0.0.1:

Packets: sent=4, Received=4 lost=0 (0.1 loss).

Program 4

Configure DHCP within a LAN and outside LAN.



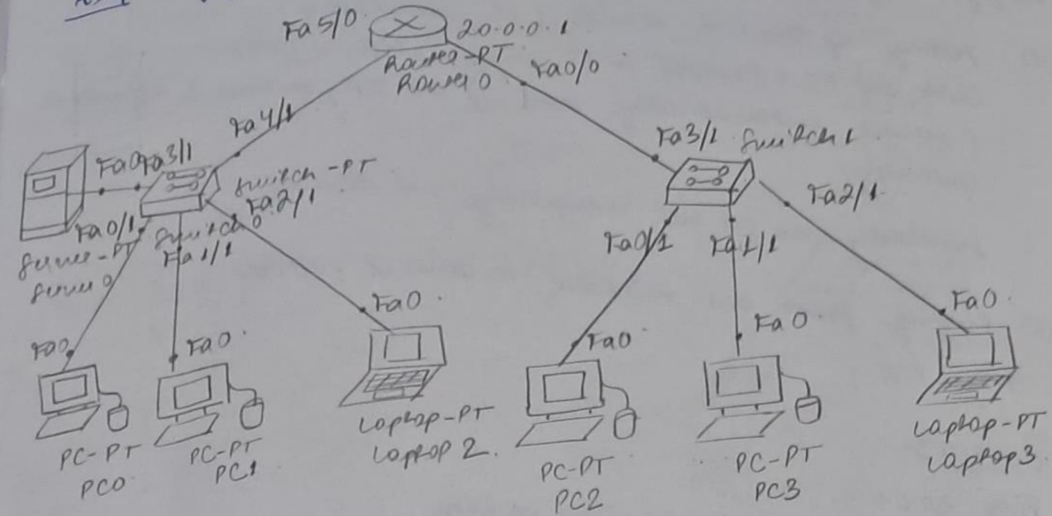
4.1 DHCP Topology

(4) (15) → DHCP Dynamic Host configuration Protocol.

13/11/24.

EXPERIMENT-4

AIM Configure DHCP network in a LAN and outside LAN.



PROCEDURE:-

(1) Connect the devices as shown above.

(2) Click on server.

(3) under desktop → IP configuration → enter IP address (10.0.0.2), subnet mask, Default Gateway (10.0.0.1).

under services → DHCP → Turn on service, change pool name (switch 1), default Gateway (10.0.0.1). Start IP address → 10.0.0.3, max no. of users: 100 and add.

Create another server pool

poolname (switch 2), default Gateway (20.0.0.1)

DNS server 0.0.0.0 start IP 20.0.0.3,

max no. of users : 100 and add.

(3) Click on Router → CLI.

Router > enable.

Router # config terminal

Router (config) # interface fastethernet 4/0

Router (config) # ip address 10.0.0.3 255.0.0.0

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

Router (config-if) # no shutdown .

Configure the same for fast-ethernet 0/0 router .

(4) Setting up the end devices (PCs and laptop) .
Click on PC → Desktop → IP configuration → DHCP .
(servers automatically sets up the IP, subnet & default gateway)

Similarly do for all remaining end devices .

(5) Pinging from one network to other is possible .

From PC0 .

OUTPUT:-

Ping 20.0.0.3.

Pinging 20.0.0.3 with 32 bytes of data .

Request timed out .

Reply from 20.0.0.3 : Bytes : 32 time : 1ms TTL=128 .

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

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

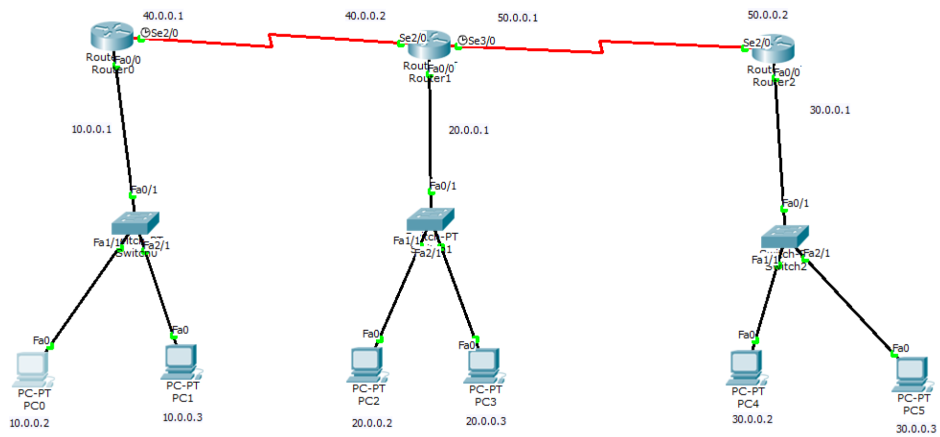
Ping statistics

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

27/11/24

Program 5

Configure RIP routing Protocol in Routers

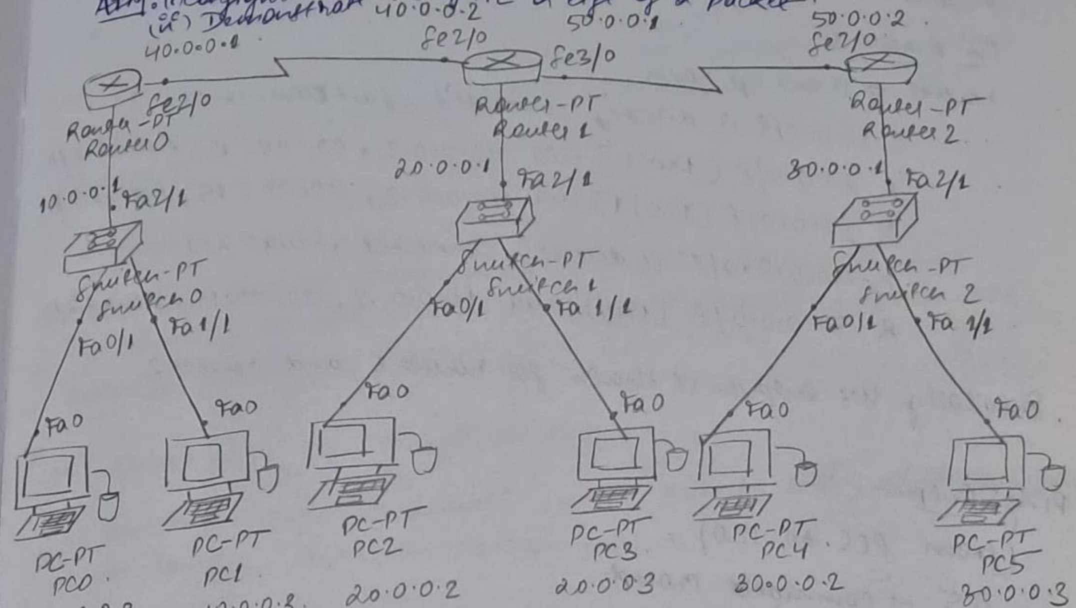


5.1 RIP Topology

(16) 20/11/24. EXPERIMENT-5 (17)

LAB-6.

AIM: (i) Configure RIP routing protocol in Routers
(ii) Demonstrate the TTL a life of a packet.



(TOPOLOGY)

PROCEDURE:-

1. Connect the devices as shown above such that all the links turn green.
   For end devices → FastEthernet () → set IP, subnet → settings → enter gateway.

   For routers → In CLI → write the commands to setup all the IP addresses of the router

2. For routing
   For each router go to CLI and enter
   ex: Router 1.
   router > enable.
   router # config terminal.
   router (config) # router rip.
   router (config - router) # network 40.0.0.0.
   router (config - router) # network 50.0.0.0.
   router (config - router) # network 20.0.0.0.

Similarly for
Router 0 → connect to network 10.0.0.0 and 40.0.0.0,
Router 2 → connect to network 50.0.0.0 and 30.0.0.0.

3. Once this setup is complete, we can the message from one device to any other end device.

Output :-

In Router 0

router # show ip route.

C 10.0.0/8 is directly connected, fastEthernet 0/0,
R 20.0.0/8 [120/1] via 40.0.0.2, 00:00:15, serial 2/1
R 30.0.0/8 [120/1] via 40.0.0.2, 00:00:15, serial 2/0
C 40.0.0.0/8 is directly connected, serial 2/0.
R 50.0.0.0/8 [120/1] via 10.0.0.2, 00:00:15, serial 2/0

Similarly the output is shown for router 1 and router 2.

Ping Output

(from PC5 to PC0).

PC5 → command prompt.

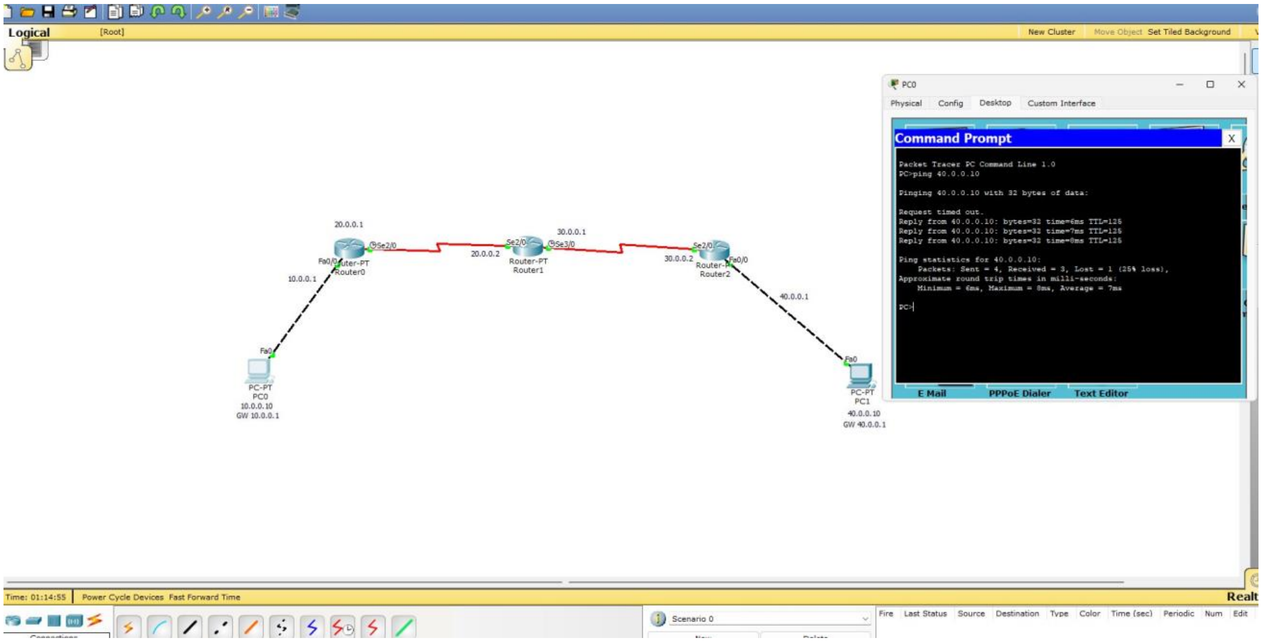
PC> Ping 10.0.0.2.

pinging 10.0.0.2 with 32 bytes of data.
Reply from 10.0.0.2 : bytes = 32 time 2ms TTL=125
Reply from 10.0.0.2 : bytes = 32 time 2ms TTL=125
Reply from 10.0.0.2 : bytes = 32 time 2ms TTL=125
Reply from 10.0.0.2 : bytes = 32 time 2ms TTL=125

Ping statistics 10.0.0.2
Packets sent = 4, Recieved = 4, lost = 0 (0% Loss).

Program 6

Configure OSPF routing protocol

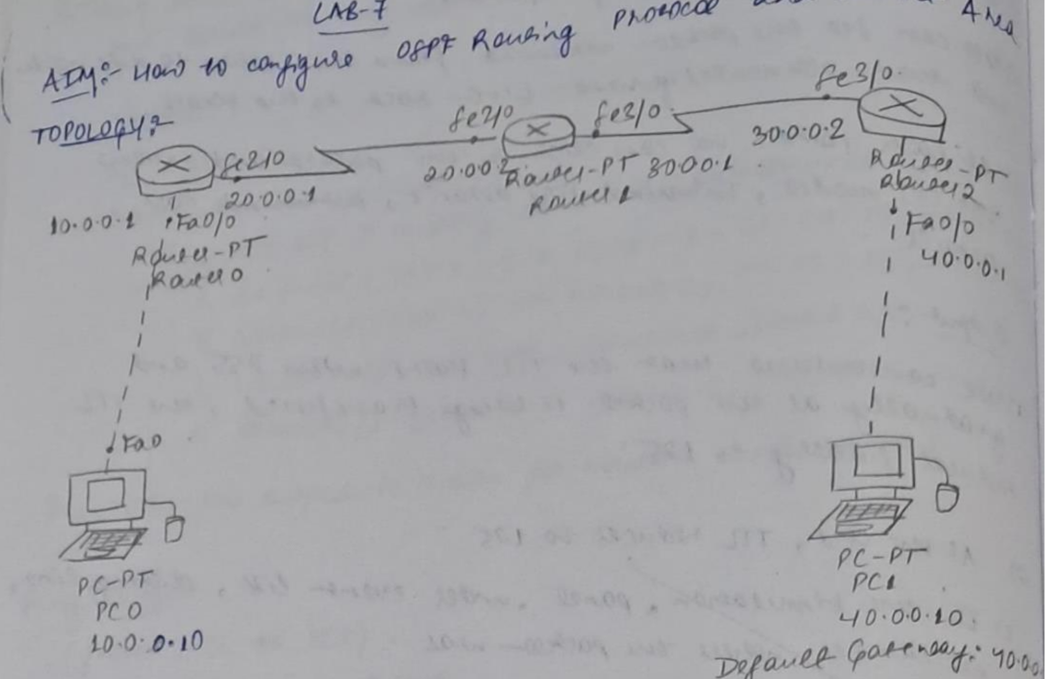


6.1 OSPF routing protocol

EXPERIMENT - 6
LAB-7

AIM:- How to configure OSPF Routing protocol and Connect Area

TOPOLOGY:-



PROCEDURE:-

(1) Create the above given topology such that the links turn green.

For end devices → FastEthernet ( ) → set IP, subnet → settings → enter gateway.

For routers → In CLI → Write the commands to set up all the IP addresses of the router.

Set default gateways.

(2) In Router 0,

Router(config) # interface fastethernet 0/0.
Router (config-if) # ip address 10.0.0.1 255.0.0.0.
Router (config- if) # no shutdown.
Router (config- if) # exit.

R0 (config) # interface serial 2/0.
R0 (config) # ip address 20.0.0.1 255.0.0.0.
R0 (config-if) # encapsulation PPP.
R0 (config-if) # clock rate 64000
R0 (config-if) # no shutdown.
R0 (config-if) # exit.

(28)

Similarly we set up the IP's of R1 and R2 while the setup of (2) fastEthernet remains same, the setting up of serial connections has 2 extra lines (encapsulation PPP, clock rate 64000).

clock rate 64000 must only be written if the serially connected port show a (c) symbol.

Here, we write clock rate command for R0, serial 2/0, R1 serial 3/0.

After this step all the connections must have turned green.

3. To enable IP routing by configuration of OSPF routing protocol in all routers.

Router R0 → CLI.

R0(config)# router OSPF 1.

R0(config-router)# router-id 1.1.1.1.

R0(config-router) # network 10.0.0.0 0.255.255.255 area 3.

R0(config-router) #network 20.0.0.0 0.255.255.255 area 1.

R0(config-router)#exit.

Router R1 → CLI

R1(config) #router OSPF 1.

R1(config-router)# router-id 2.2.2.2.

R1(config-router) # network 20.0.0.0 0.255.255.255 area 1.

R1(config-router) # network 30.0.0.0 0.255.255.255 area 0.

R1(config-router) # exit.

Router R2 → CLI

R2(config) # router OSPF 1.

R2(config-router)# router-id 3.3.3.3.

R2(config-router) # network 30.0.0.0 0.255.255.255 area 0

R2(config-router) # network 40.0.0.0 0.255.255.255 area 2.

R2(config-router) #exit.

4. Once the setting up of networking area is done, we configure loopback address to router

R0(config-if) # interface loopback 0.

R0(config-if) # ip add 172.16.1.252 255.255.~~255~~ 0.0

R0(config-if) # no shutdown.

R2(config-if)#interface loopback 0.
R2(config-if)# ip add 172.16.1.254 255.255.0.0.
R2(config-if)# no shutdown.
R1(config-if)# interface loopback 0.
R1(config-if)# ip add 172.16.1.253 255.255.0.0.
R1(config-if)# no shutdown.

5. On checking routing table of R3 using 'show ip route' we can see that R3 doesn't know about area 3.
Gateway of last resort is not set.

OIA 20.0.0.0/8 [110/128] via 30.0.0.1 serial 1/0.
C 40.0.0.0/8 is directly connected, fastEthernet 0/0
C 30.0.0.0/8 is directly connected, serial 2/0.

Since R3 doesn't know about area 3 we have to create a virtual link between R0 & R1.

6. Creating virtual link between R1, R0.
In Router R0.
R0(config)# router ospf 1.
R0(config-router) # area 1 virtual link 2.2.2.2.
R0(config-router) #exit

In Router R1.
R1(config)# router ospf 1.
R1(config-router) # area 1 virtual-link 1.1.1.1.
R1(config-router) #exit.

7. Now, check routing table of R3. Once all these steps are completed, the message can now be pinged from 1 end-device to other.

Observation

In R2
Router # show ip route.
OIA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:57:25, serial 2/0.
C 40.0.0.0/8 is directly connected, fastEthernet 0/0.
OIA 10.0.0.0/8 [110/129] via 30.0.0.1 00:57:25, serial 2/0.
C 30.0.0.0/8 is directly connected, serial 2/0
C 172.16.0.0/16 is directly connected, Loopback

(22) Similarly the output is shown for Router0 and 1. (23)

Ping output.
(From PC0 to PC1).
PC0 → Command prompt
C:\> ping 40.0.0.10.
Pinging 40.0.0.10 with 32 bytes of data.

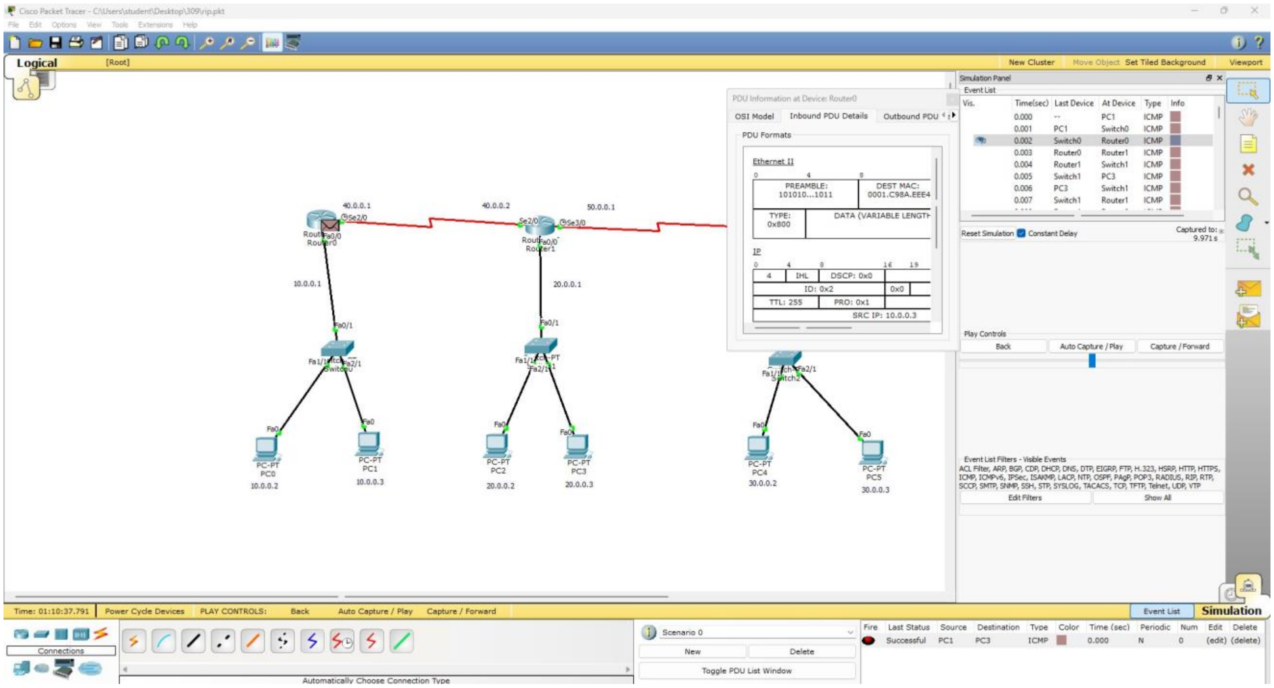
Request time out
Reply from 40.0.0.10 : bytes = 32 time = 21 ms TTL = 125
Reply from 40.0.0.10 : bytes = 32 time = 2ms TTL = 125
Reply from " " " " = 28ms TTL = 125.
" " "

Pinging statistics for 40.0.0.10.
Packets: 7 sent = 4, Recieved = 3, lost = 1 (25% loss).

18/12/24

Program 7

Demonstrate the TTL/ Life of a Packet



7.1 TTL Demonstration

(ii) Topology.

Same as previous -

Procedure:-

(1) Click on Simulation mode

(2) Click on Packet option on the source to destination end device.

(3) Enter Auto Capture/Play.

(Time to leave)

(19)

Demonstrate a TTL as life of a packet

(4) We can see the packet movement from the source to destination and then acknowledgement sent back to the source.

(5) At each point, we can click on the packet and view the OSI model, Inbound PDU details, outbound PDU details.

Output :-

(1) We can observe that the TTL starts with 255 and gradually as the packet is being transferred, the TTL reduces finally to 125.

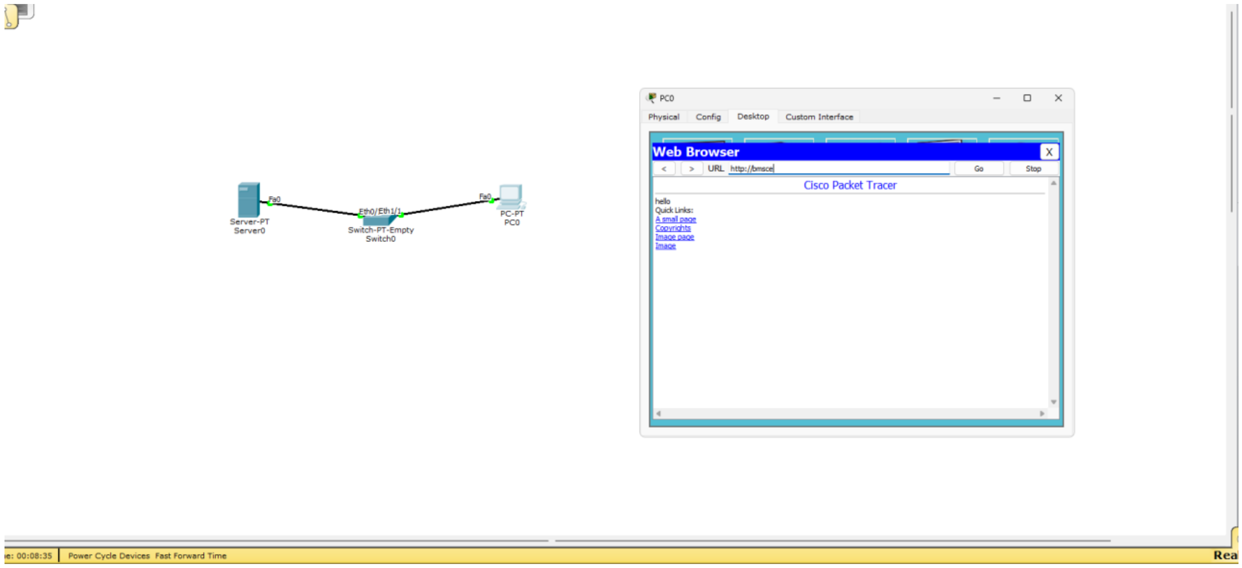
(2) At the end, TTL reduces to 125.

(3) In the simulation panel, under event list, clicking line, we can see where the packet was.

27/11/24

Program 8

Configure Web Server, DNS within a LAN.



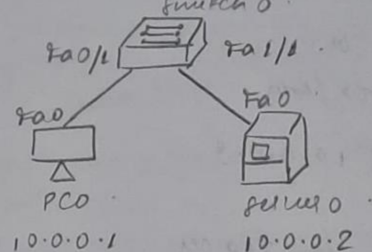
8.1 DNS Configuration

EXPERIMENT 12.

configure web server, DNS within a LAN.

AIM :- To configure DNS server to demonstrate the mapping of IP addresses & domain names.

TOPOLOGY :-



Connect a PC & a server to a switch, assign ip address as 10.0.0.1 & 10.0.0.2 respectively.

Configuration :-

open cisco packet tracer and arrange as given in topology and configure the devices as given below :-

PC0 :

IP address : 10.0.0.2.

server 0 :

IP address : 10.0.0.3.

Connect PC0 & server 0 via a switch PT

PC0 connects to switch on interface Fa0 & switch on Fa0/1.

server connects to switch on interface Fa0 & switch on Fa1/1.

server 0 :

Go to server → services → DNS

Enable on

In the text fields add :

name : abc

address : 10.0.0.3.

click add

go to HTTP
click edit for index.HTML [change if needed] (34)
click save.

PROCEDURE:-

(1) Go to PC0 → Desktop → Webbrowser
(2) Search 'abc' in the bar (or)
(3) Search 10.0.0.2 in the box/bar.

Output: for both 'abc' & 10.0.0.3

Cisco packet Trace.

Welcome to cisco packet trace. Opening doors to new opportunities. Mind wide open.

Quick Links:

A small page

copyrights

Image page

Image

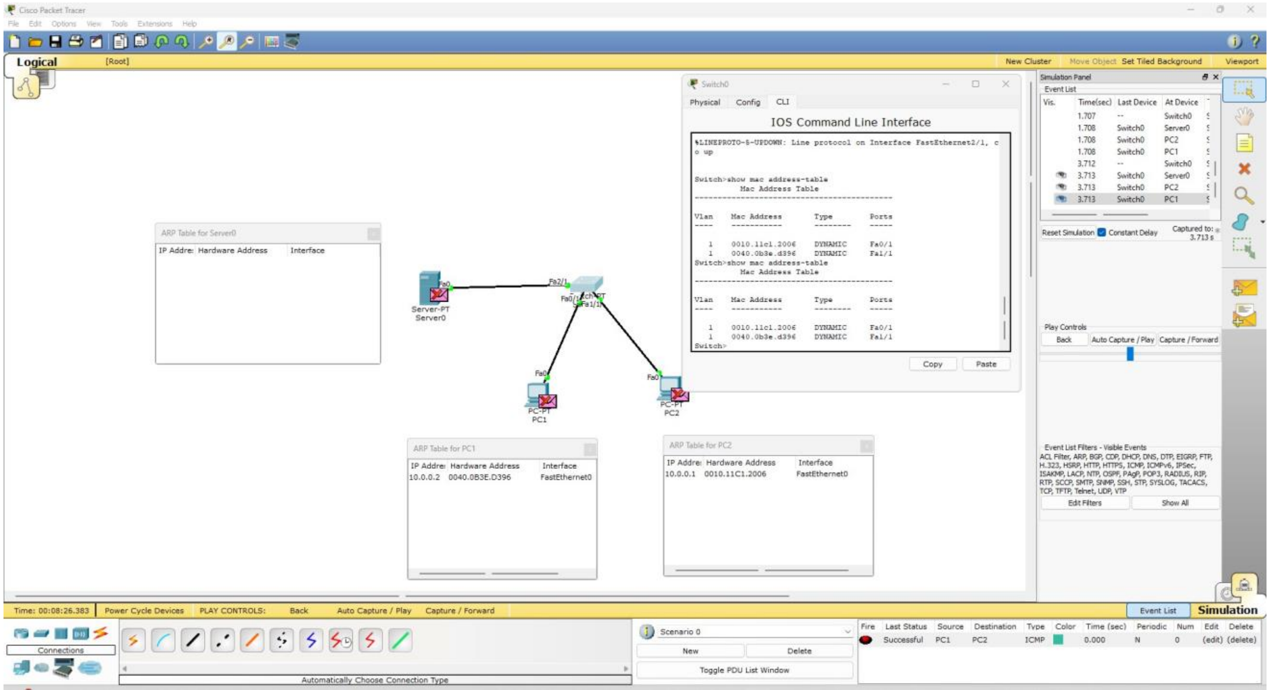
Observation:-

DNS translates domain names to ip addresses. It simplifies accessing websites by using human-readable names.

In this experiment, a web server & DNS were configured within a LAN to map domain names to ip addresses. The PC0 successfully accessed the server0 by both its ip address and the configured domain name 'abc'. The configurations was successful allowing the webpage to be accessed via both methods.

Program 9

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



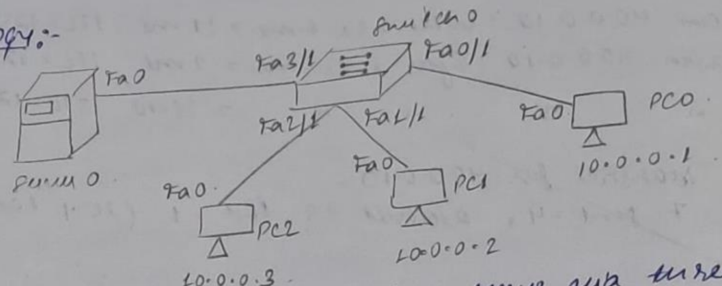
9.1 ARP Topology

EXPERIMENT - 8.

To construct a simple LAN & understand concept and operation of ARP.

AIM:- Construct a simple LAN simulate operation of Address Resolution Protocol.

TOPOLOGY:-



1. Switch connected to 3 PCs and a server via three fast ethernet interface and one ethernet interface respectively.

2. All connections made via copper straight through cable.

PROCEDURE:-

(1) Open cisco packet tracer & drag the following switch, PC: place 3 PC's, each connected to switch 0 and server: place 1 server and connect it to switch 0.

(2) Assign an IP address and subnet mask to all the devices then connect them via a switch,

(3) Use the inspect tool ('Q'), click on a PC to view ARP table.
Display the ARP table of all the devices.

(4) Initially ARP is empty for all.

(5) Also in CLI of switch, the command = show mac address-table can be given on every transaction to see how the switch learns from transactions and build the address table.

(6) Use the capture button in the simulation panel to go step by step so that changes in ARP can be clearly noted.

7. Observe the switch as well as nodes updated the ARP table as and when communication starts

Observation:

- As the message travels from one source host to its destination host the ARP table of all devices get updated.

ARP maps an IP address to a MAC address.
It ensures communication within a local network.

ARP table for PC0 (source):-

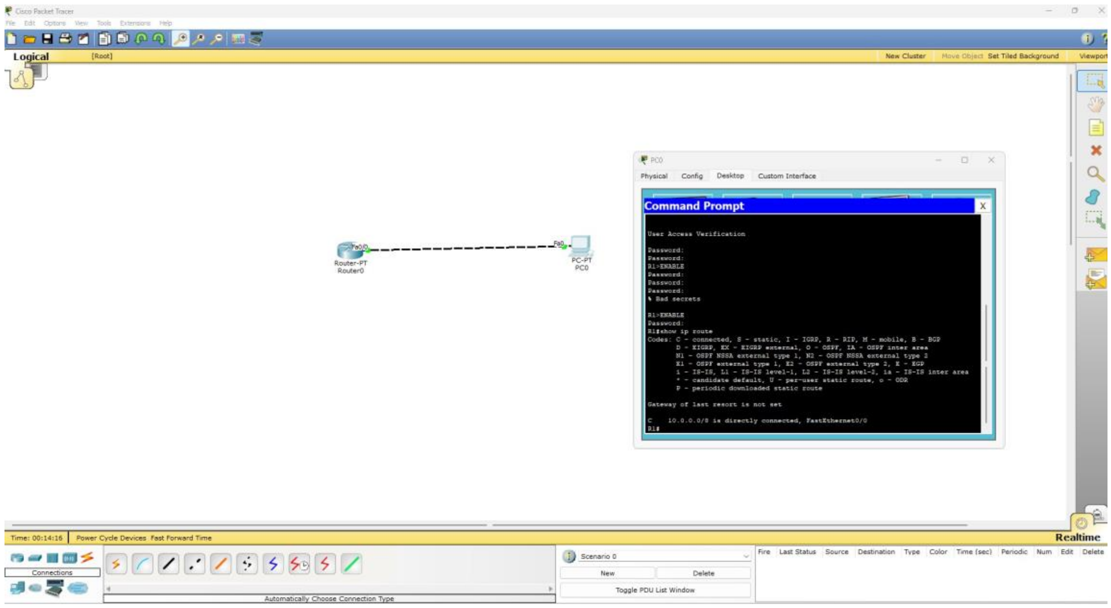
IP address	Hardware Address	Interface
10.0.0.3	0060.2F29.2CB8	FastEthernet0

ARP table for PC2 (destination):-

IP address	Hardware Address	Interface
10.0.0.1	00D0.D302.96DB	FastEthernet0

Program 10

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



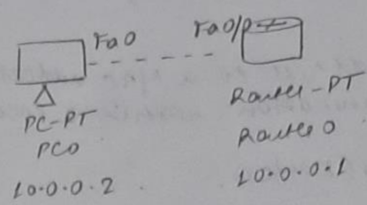
10.1 TELNET Configuration

EXPERIMENT-9

To understand the operation of TELNET.

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

TOPOLOGY:-



A router connected to a single PC via a fastethernet interface with copper cross-over cable.

PROCEDURE:-

(1) Open cisco packet tracer and drag a PC & a router.

(2) Connect the PC to the router via fastethernet interface with a copper cross-over cable.

(3) Assign the IP address to the PC - 10.0.0.2 with gateway as 10.0.0.1.

Configure the router:

Router > enable

Router # config.

Router (config) # hostname r1.

r1 (config) # enable secret p1.

r1 (config) # ~~enable~~ interface fastethernet 0/0.

r1 (config-if) # ip address 10.0.0.1 255.0.0.0.

r1 (config-if) # no shutdown.

r1 (config-if) # line vty 0 5 -- to allow virtual term

r1 (config-if) # login

r1 (config-line) # password p0.

r1 (config-line) # exit.

r1 (config) # exit

r1 # wr - to save changes in router.

In command prompt:

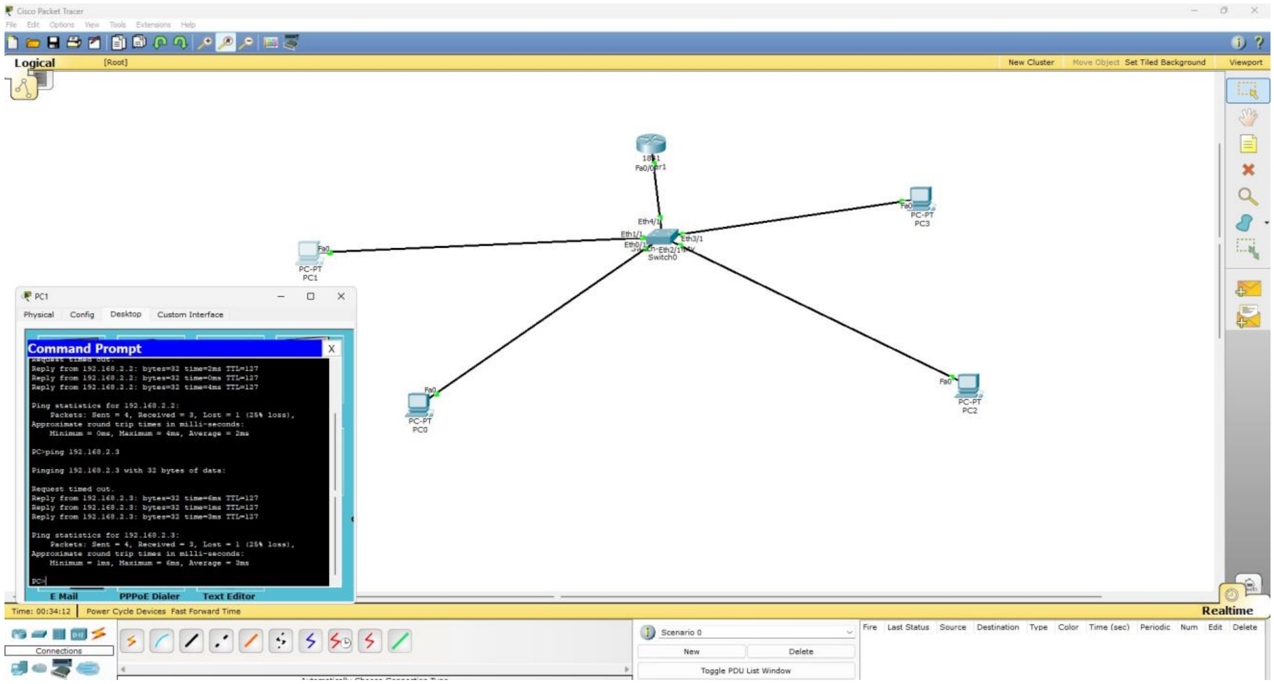
ping 10.0.0.1.
Password for user authentication is p0.
Password for enable is p1.

Observations:
Telnet is a protocol for remote access to server.
It allows command-line communication over a network.
The PC is able to send the data to the router and
indicates that the gateway is available and connected.

access for 6. users

Program 11

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



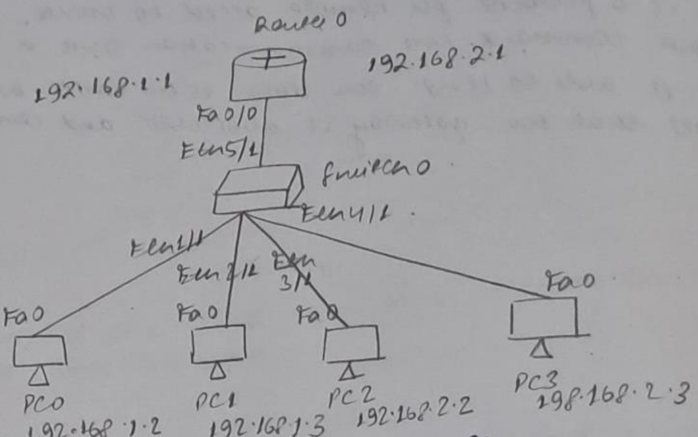
11.1 Virtual LAN Topology

EXPERIMENT-10

AIM:-

To construct a VLAN and make the PC's communicate among a VLAN.

TOPOLOGY:-



PROCEDURE:-

(1) Place a 1841 Router, a switch and 4PCs.

(2) connect the four PCs to the switch via fastethernet.

(3) Since only 4 fastethernet ports are available in the switch, we have to add an ethernet port.

(4) Switch off the power button of switch.
Add the ethernet port to the switch.
Switch on the power button.
Connect the router to the switch via Ethernet 6/1.

(5) In the switch, go to config tab and
- select VLAN Database
- Give VLAN number say 2.
- Give VLAN name say (cse cse)
- Add it to the Database.

(6) Select the switch:
- Go to config
- Go to Ethernet 6/1, i.e., connected to router
- Make it the trunk.

(7) Configure the PCs as shown in the topology

(8) Select switch.
- Go to config
- Go to FastEthernet 2/1.
- Set VLAN number as 2, i.e., 'CSE CSE'
- Similarly set VLAN 2 for fastEthernet 3/1 interfaces.

(9) Config the Router:

Router (config) # interface fastEthernet 0/0.

Router (config-if) # ip address 192.168.1.1 255.255.255.0

Router (config-if) # no shut

Router (config-if) # exit

Now, to configure the router's VLAN interfaces

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

Router (config-subif) # exit.

(10) Ping devices within same VLAN and to devices of different VLAN.

OBSERVATIONS:-

(1) When devices are pinged within same VLAN:
- Pinging 192.168.1.3 from 192.168.1.2
- the data packet doesn't go to the router.
- the switch forwards the packet without the need of the router.

(2) When a device pings a device of another VLAN
- Pinging 192.168.2.3 from 192.168.1.2.
- the data packet's journey is as follows:-

192.168.1.2 → switch → Router

192.168.2.3 ← switch ←

(3) VLANs divide a single switch into multiple logical switches

- Devices in one VLAN cannot directly communicate with devices in another VLAN without a router

(4) Traffic Isolation:-

- Each VLAN maintains its own broadcast domain
- Broadcasts sent by devices in on VLAN do not reach devices in another VLAN.

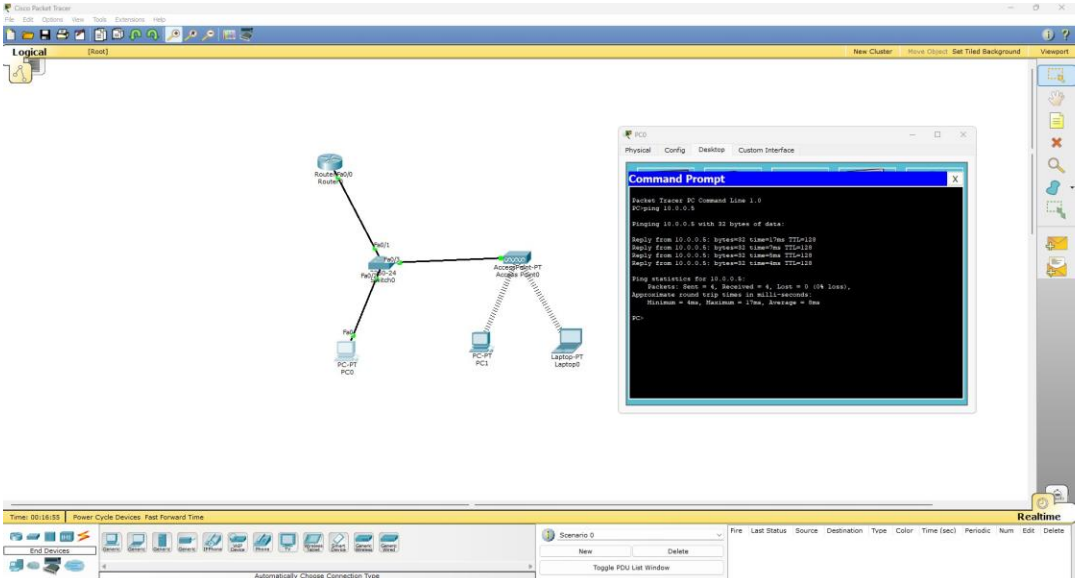
(5) VLAN trunking allows switches to forward frames from different VLANs over a single link called trunk.

- This is done by adding an additional header information called tag to the ethernet frame.

- VLAN tagging.

Program 12

To construct a WLAN and make the nodes communicate wirelessly



12.1 WLAN Topology

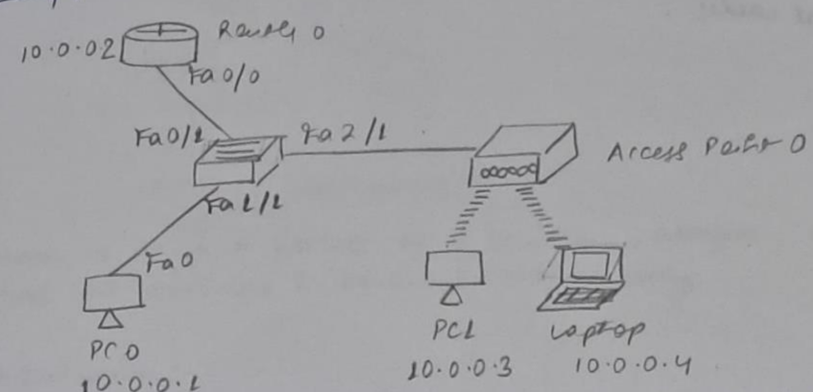
EXPERIMENT - 10

~~To construct WLAN and the PCs communicate along a VLAN~~

To construct a WLAN and make the nodes communicate wirelessly.

AIM:- To construct WLAN and make nodes communicate wirelessly

TOPOLOGY:-



Connect a router and access point to a switch through fastethernet interface. Connect a PC and set its ip address. Take a PC and a laptop & set their ip addresses.

PROCEDURE:-

(1) Drag a switch and connect it to a PC, router & an access point.

(2) Place a PC and laptop without any wired connection.

(3) Configure PC0 with ip address 10.0.0.1 and router 0.

(4) Configure Access Point :- Port-1 → SSID → Name → Enter any name → select WEP & give any 10 digit hex key - 1234567890

(5) Configure PC4 & laptop with wireless standards.

(6) Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag WMP300N wireless interface to the empty port. Switch on the device.

(7) In the config tab, a new wireless interface would have been added. Now, configure SSID, WEP, WEP key, IP address and gateway to the device.

(8) Ping from every device to every other device and see the results

Observations:

WLAN enables wireless n/w comm. It uses radio waves for connectivity. WLAN connects devices wirelessly within a local area. It eliminates the need of physical cables.

Program 13

Part B

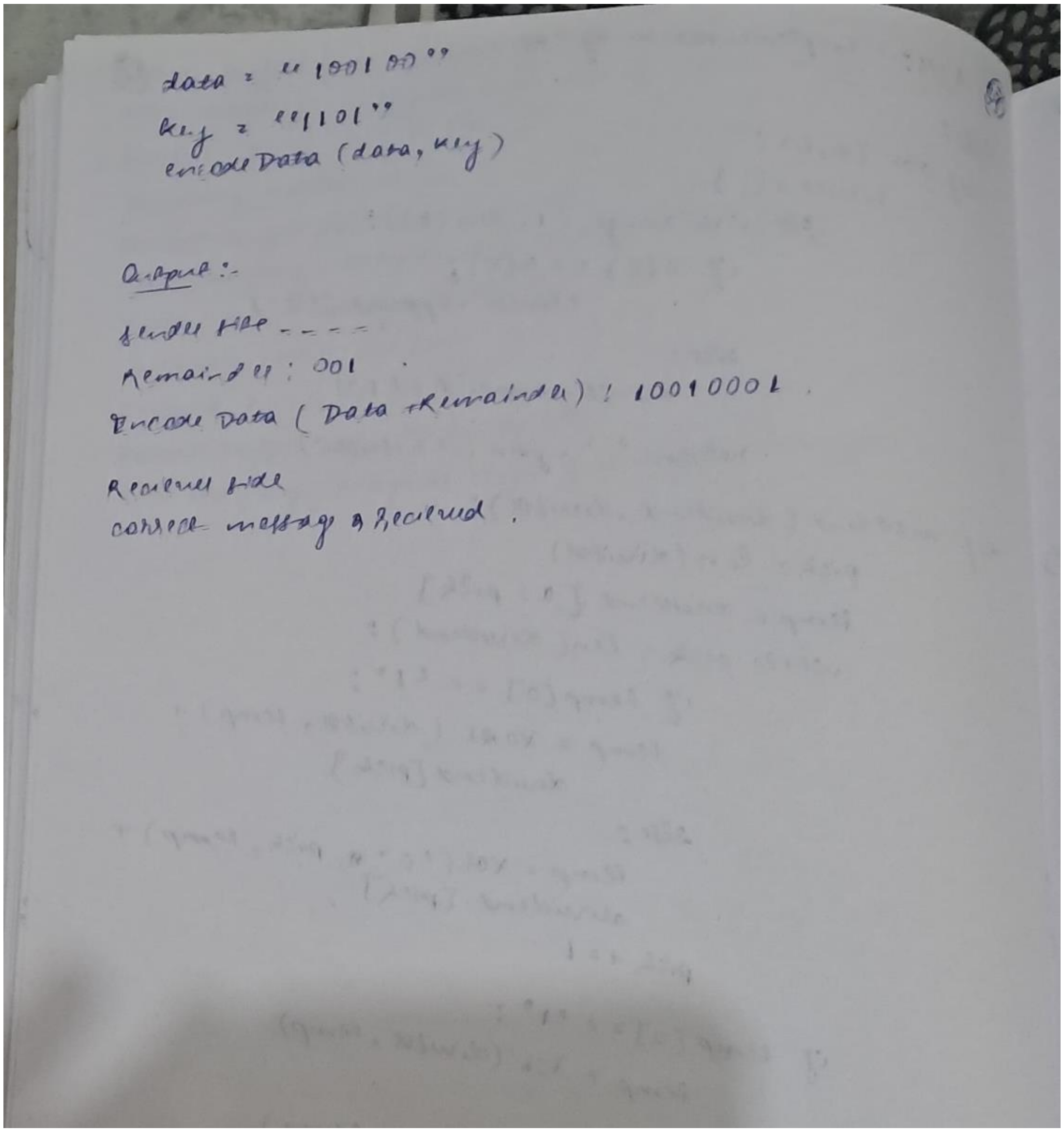
Write a program for error detecting code using CRC-CCITT (16-bits).

(36)

2. AIM:- Implementation of CRC. (37)

Code:-

```
def xor (a,b):
    result=[ ].
        for i in range (1, len(b)):
            if a[i] == b[i]:
                result.append ('0')

            else:
                result.append ('1')

        return ''.join ('result')

def mod2div (dividend, divisor):
    pick= len (divisor)
    temp= dividend [0 : pick]
    while pick < len( dividend ):
        if temp[0] == '1':
            temp = xor (divisor, temp) +
                dividend [pick]
        else:
            temp = xor ('0' * pick, temp) +
            dividend [pick]
        pick += 1

    if temp [0] == '1':
        temp = xor (divisor, temp)

    else:
        temp = xor ('0'* pick, temp)

    checkword = temp
    return checkword

def encodeData (data, key):

    l-key = len(key)
    append_data = data + '0' * (l-key-1)
    remainder = mod2div (append_data, key);
    codeword = data + remainder
    print ("Remainder", remainder)
    print ("Encode Data  (Data + Remainder) : "
                        , codeword)
```


data = "100100"
key = "1101"
encodeData (data, key)

Output :-
Sender side ----
Remainder : 001
Encode Data (Data +Remainder) : 100100001
Receiver side
correct message is received.

OUTPUT:

```
Enter the input message in binary: 1101
The transmitted message is: 1101110100011010110l
Enter the received message in binary: 1101
No error in data
```

Program 14

Write a program for congestion control using Leaky bucket algorithm

(34)

Cycle 2

(35)

AIM:- Implementation of Leaky Bucket Algorithm :

CODE:-

```
#include <stdio.h> .
int main(){
  int incoming, outgoing, bucket_size, n, store=0;
  printf("Enter bucket size, outgoing node and no. of
         input");
  scanf("%d %d %d", &bucket_size, &outgoing, &n);
  while(n!=0){
  printf("Enter the incoming packet size: ");
  scanf("%d", &incoming);
  printf("Incoming packet size %d\n", incoming);
  if(incoming <= (bucket_size - store)){
        store += incoming;
        printf("Bucket buffer size %d out of %d\n",
         store, bucket_size);
   } else {
      printf("Dropped %d no. of packets \n",
      incoming -(bucket_size - store));
      printf("Bucket buffer size %d out of %d\n",
       store, bucket_size);
      store = bucket_size;
   }
  store = store - outgoing;
  printf("After outgoing %d bytes left out of %d in
       buffer \n", store, bucket_size);
      n--;
  }
}
```

Output :-

Enter bucketsize, outgoing rate & no. of inputs : 100 20 3.

Enter the incoming packet size : 30.

Incoming packet size 30.

Bucket buffer size 30 out of 100.
After outgoing 10 bytes left out of 100 in buffer.
Enter incoming packet size: 50.
Incoming packet size 50.
Bucket buffer size 60 out of 100.
After outgoing 40 bytes left out of 100 in buffer.
Enter the incoming packet size : 80.
Incoming packet size 80.
Dropped 20 no. of packets
Bucket buffer size 40 out of 100.
After outgoing 80 bytes left of out of 100 in buffer.

P.T.O

```
Generated packet sizes:
packet[0]: 50 bytes
packet[1]: 30 bytes
packet[2]: 20 bytes
packet[3]: 10 bytes
packet[4]: 50 bytes
packet[5]: 60 bytes
packet[6]: 50 bytes
packet[7]: 30 bytes
packet[8]: 40 bytes
packet[9]: 50 bytes

Enter the Output rate: 100
Enter the Bucket Size: 50

Incoming Packet size: 50 bytes
Bytes remaining to Transmit: 50
Time left for transmission: 10 units
```

2.1 Leaky Bucket Output

```
Packet of size 50 bytes Transmitted
----Bytes Remaining to Transmit: 0

Incoming Packet size: 30 bytes
Bytes remaining to Transmit: 30
Time left for transmission: 20 units

Packet of size 30 bytes Transmitted
----Bytes Remaining to Transmit: 0

Time left for transmission: 0 units
No packets to transmit!!

Incoming Packet size: 20 bytes
Bytes remaining to Transmit: 20
Time left for transmission: 40 units

Packet of size 20 bytes Transmitted
----Bytes Remaining to Transmit: 0
```

2.2 Leaky Becket Output

Program 15

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.

(38) 3. AIM: Implementation of TCP/IP. (39)

code:-

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()
```

Output:

sender side ----

sender is ready to receive

client side -----

Enter file Name: hello.txt

from server: Hello world.

Program 16

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.

4. Aim:- Implement UDP.

Code:

client UDP.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)
    server.socket . sendto ( bytes (l, "utf-8"), clientAddress)
    print ( "sent back to client", l)
file.close ( ) .
```

Output:

server side . . . . .

The server is ready to receive .

sent back to client : hello world .

Client side. . . . .

Enter file name : hello.txt .

from server : hellow world .