29/10/24

Date __/__/__
Page

LAB - 1

Components:

i) Console cable: 

Connects PC's to routers/switches; requires matching settings for speed, data bits and parity.

ii) Copper-straight-through:

Standard ethernet cable for connecting devices at different OSI layers

iii) Copper-Cross over: 

Ethernet cable for connecting devices at the same OSI layer

iv) Fiber: 

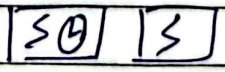
Used for connections between fiber ports (100 mbps or 1000 mbps).

v) Phone: 

Connects devices modem ports, typically for dialling into a network.

vi) Co-axial: 

Connects devices with coaxial ports, like a cable modem to a network.

vii) DCE DTE Serial: 

Serial connections for WAN links; requires clocking on the DCE side.

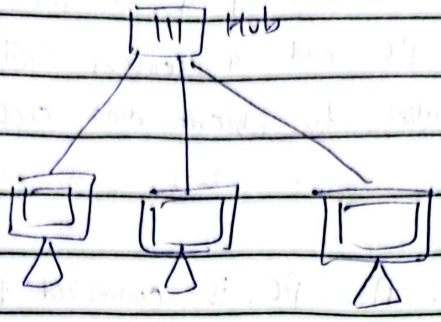
viii) DCE Octal: High density 8-port asynchronous cable.

Date __/__/__
Page

Hub	Switch
Layer 1 (Physical layer)	Layer 2 (Data link layer)
Broadcast to all devices.	Forwards to specific devices.
Less efficient, more collisions	More efficient, less collisions.
Traffic is Unmanaged	Traffic is managed.
Cheaper generally	More expensive
eg: Simple networks	Modern networks.

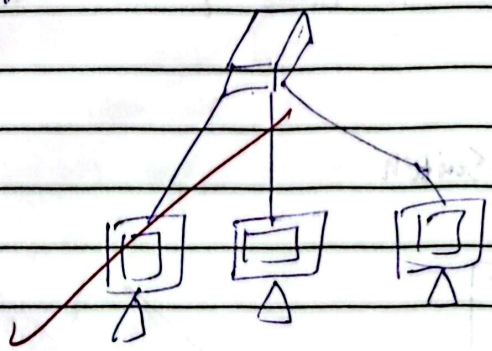
Date __/__/__
Page _____

Hub:



- Layer: operates at layer1 (physical layer) of OSI model.
- It broadcasts data to all connected devices, doesn't filter or manage traffic
- Limited efficiency, more collisions can occur due to simultaneous data transmission

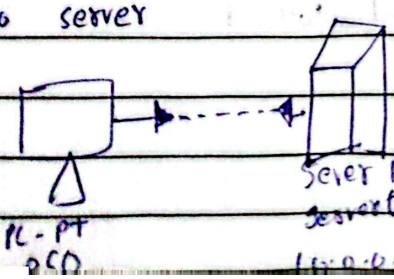
Switch:



- Operates at datalink layer of OSI model.
- It forwards data only to the specific device for which it is intended, based on MAC address.
- More efficient, reduces collision

Experiment 1:

1) PC to server



Date __/__/__
Page _____

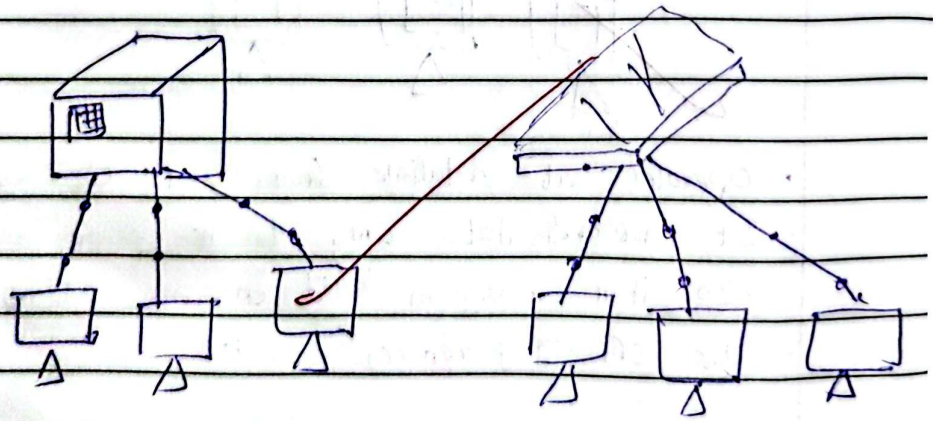
Aim: To setup a point to point network b/w a PC and a server, facilitating direct communication to observe data exchange.

Topology: A PC is connected to server using a crossover ethernet cable

IP address of PC - 10.0.0.1, server - 10.0.0.2

Observation:
Direct communication allows PC to communicate with server, which is typical in small networks for such as file sharing, service requests or testing server response to client query.

2. Hub and Switch



PC-PT	PC-PT	PC-PT	PC-PT PT-3	PC-PT	PC-PT
PC0	PC1	EPC2		PC4	PC5
10.0.0.6	10.0.0.7	10.0.0.8	10.0.0.11	10.0.0.12	10.0.0.13

Date __/__/__
Page

Aim:

To create simple network consisting of 3 PCs connected to a central hub and another network with 3 PC's connected to a switch. This connection will help observe the behavior of data transmission using hub & switch device

Topology:

3 PCs are connected to a hub & switch using straight through ethernet cable.

Observation:

Hub broadcasts packets to all devices which may cause unnecessary traffic. Switch forwards packets only appropriate device by learning MAC address, making it more efficient in reducing traffic.

9/10/24

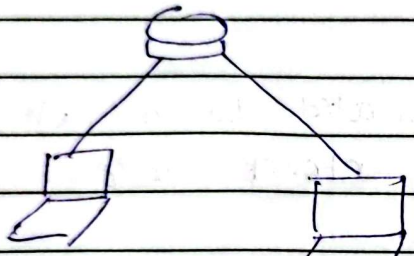
Date / /
Page

LAB - 2

Aim:

To connect 2PC's in two different network using a router.

Topology:



10.0.0.10
Def gateway 10.0.0.1

20.0.0.10
Def gate 20.0.0.1

Procedure:

1) Drag and drop a generic router and 2 PC's

2) Connect the routers and PC's using copper cross over cable.

3) set the gateway 10.0.0.1 & 20.0.0.1 respec.

4) Open CLI in router & enter the following

```
Router > enable
Router # config terminal
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shutdown
```

5) Ping anoth
prompt of
> ping 2

Observation:
Command
with 32

Reply from
"
"
"

Ping stat
Packets :
Approximate
minimu

Date __/__/__
Page

1) Ping another system as interface from the command prompt of PC0 or PC1 using command prompt

> ping 20.0.0.10

Observation:

Command prompt gives output pinging 20.0.0.10 with 32 bytes of data

Reply from 20.0.0.10, bytes = 32 time = 0ms TTL = 127

" "

" "

" "

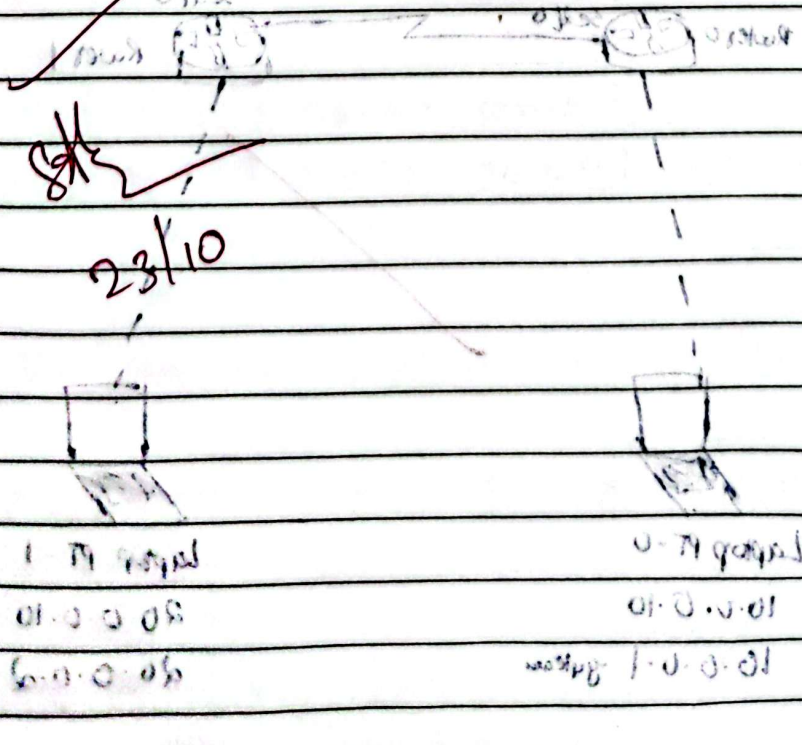
Ping statistics for 20.0.0.10

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

Approximate round trip times in milli-seconds:

minimum = 0ms, maximum 2ms,

Average = 0ms.



16/10/24

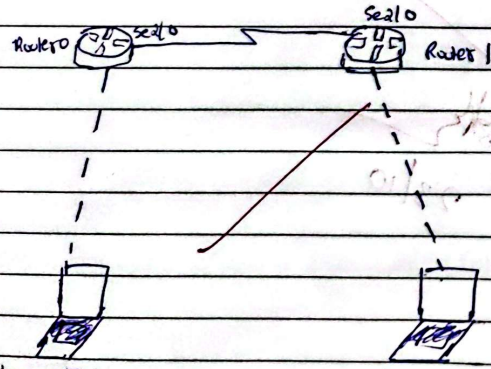
LAB - 3

Aim:

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

b) Configure default route, static route to the router.

Topology:



Procedure:

1) Two routers connect each other with serial DCE wire.

2) Two Laptops connect one to each router using copper cross-over wire.

3) Configure the laptops ip address, gateway etc.

4) Router (1) → CLI
Router > enable
Router # config terminal
Router (config) # interface fastEthernet 0/0.
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shutdown
exit.
→ Do same for router 2. with diff ip and fastEthernet.

5) Open Router again [setting ip of routers]

Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
" " # no shut
" " # exit.
→ Do same for router 2. with diff i/p.

6) Open command Prompt in one of the laptops and enter
> ping 20.0.0.1
ping 30.0.0.1
ping 30.0.0.2.

OBSERVATION: Though the routers are connected physically with wire, the destination is unreachable or there is no connection

b) Procedure:

1) Open routes CLI and enter

Router config)# ip route 10.0.0.0 255.0.0.0 30.0.0.1

Router (config)# exit

Router# show ip route

→ Do same for the seond router.

2) Open command prompt in laptop, enter the ping.

Observation:

Reply from all 3 different pings, showing the connection is valid and working.

23/10/24

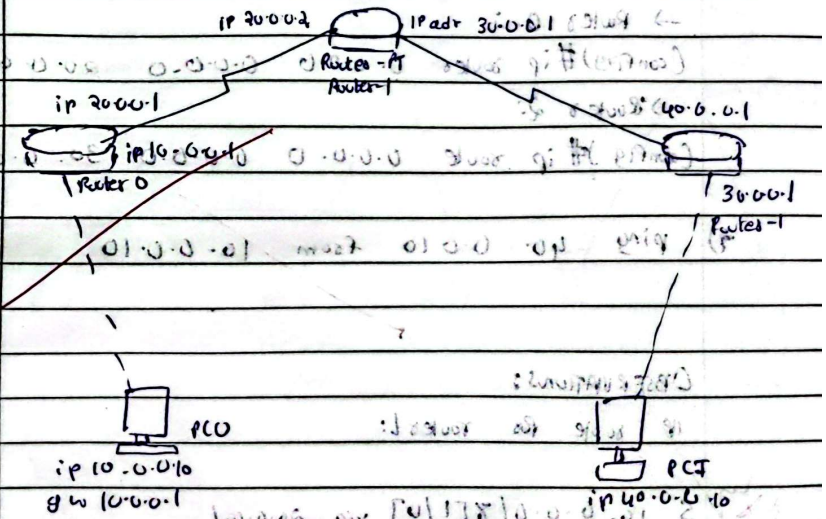
LAB - 04

Static Routing and Default Routing.

Aim:

To configure default route, static route to the routers.

Topology:



Procedure:

1) Place 3 routers, and 2 pcs.

2) Connect pcs to Router 0 and Router 2, copper-cross

5) Static Routing :

to configure ip routes manually rather than dynamic routing.

→ Route-1 must know about 10.0.0.0 network and also the 40.0.0.0 network

→ Open CLI in router 1:

Router (config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
(hop address)

Router (config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

6) Default Routing :

default route when nothing else is available:

→ Router 0 :

(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

→ Router 2:

(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

7) ping 40.0.0.10 from 10.0.0.10

OBSERVATIONS:

ip route for router 1:

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 " " , serial 3/0

S 40.0.0.0/8 [1/0] via 30.0.0.2

C → connected S → static

Router 0: C 10.0.0.0/8 is directly connected, FastEthernet 0/0

S 30.0.0.0/8

20/11/24

LAB-6

Date
Page

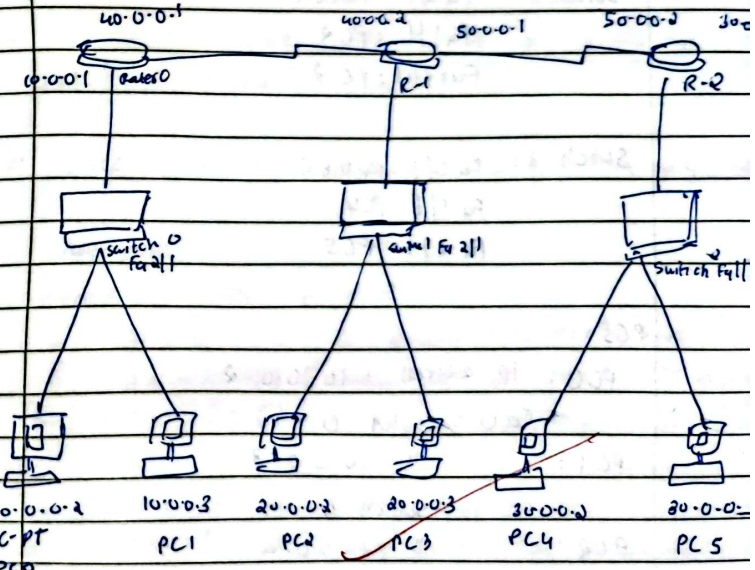
Routing Information Protocol

(RIP routing protocol in Routers)

Q. Configure RIP routing protocol in Routers.

Aim: To obtain connection in the network of 3 routers 3 switches and 6 PC using RIP.

Topology :



Routers :

Router 0: IP address : 10.0.0.1

Fa0/0 : with switch (10.0.0.1)

se 2/0: with router1 (40.0.0.1)

Router 1 : IP address : 20.0.0.1

Fa0/0 : with switch 1 (20.0.0.1)

se 2/0 : with router 0 (40.0.0.2)

Router 2: IP address : 30.0.0.1
Fa 0/0 with switch 2 (30.0.0.1)
se 2/0 with router 1 (50.0.0.2)

Switches:
Switch 0: Fa 0/1 Router 0
Fa 1/1 : PC 0
Fa 2/1 : PC 1

Switch 1: Fa 0/1 : Router 1
Fa 1/1 : PC 2
Fa 2/1 : PC 3

Switch 2: Fa 0/1 : Router 2
Fa 1/1 : PC 4
Fa 2/1 : PC 5

PCs:
PC 0: ip address : 10.0.0.2
Fa 0 : switch 0
PC 1: " " 10.0.0.3
: Switch 0
PC 2: " " 20.0.0.2
Switch 1
PC 3: 20.0.0.3
: Switch 1
PC 4: 30.0.0.2
: Switch 2
PC 5: 30.0.0.3
: Switch 2

Procedure:

1. Connect 3 routers 3 switches and 6 PCs using appropriate connections.
2. configure end-devices: 6 PCs ip address and gateway
3. Configure the interface of routers with all connections turn green.
4. For each routers configure the rip routes and add all connected networks.
5. Check with show ip route command and ping comm

Observation:
Router ## show ip route
C 10.0.0.0/8 is directly connected FastEthernet 0/0.
R 20.0.0.0/8 [120/1] via 40.0.0.2 00:00:26, ser 2/0
R 30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:26, Serial 2/0

C 40.0.0.0/8 is directly connected 2/0
R 50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:26 Serial 2/0

PC 0:
ping 10.0.0.3
pining with 32 bytes of data.
Reply from 10.0.0.3 : bytes 32 ttl 7.7L:/ar
:
:
ping 20.0.0.2

Date / / Page

EXP - 6

Aim: Demonstrate TTL

TTL means time to for a packet it tells that for how many time units the packet will be here in the network.

Topology:

Same as the prev exp.

Procedure:

→ send a simpy PDU from PC1 to PC4

→ Auto capture the screen list then observed in TTL of each router in PDU Information.

Observations:

When the packet passes Router 0:

inbound TTL = 255ms

outbound TTL = 254ms

When the packet pass from router 0 to Router 1

inbound TTL = 254ms

outbound TTL = 253ms

When the packet passes:

inbound TTL = 253ms

inbound TTL = 252ms

Hence we can conclude here will be decrement in TTL for 1ms then it passes across router.

Re

Date / / Page

27/11/24

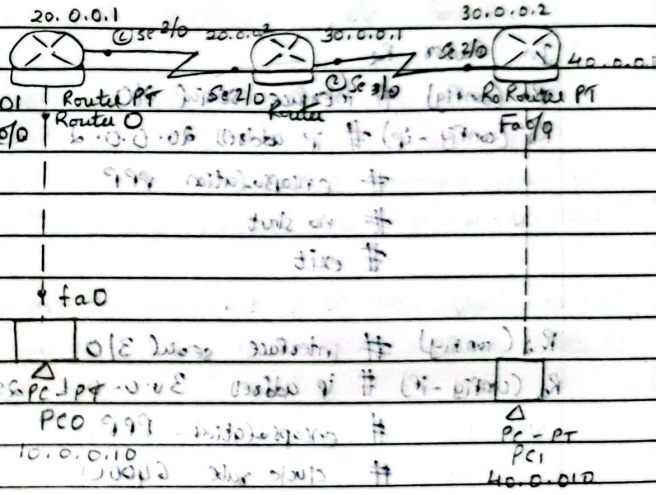
LAB - 7

Q. Demonstrate OSPF routing protocol using different routes and networks

Aim:

Demonstrate OSPF routing protocol using different routes and networks.

Topology



Procedure

→ Create the above topology

· Configure ip address to all interfaces.

For all connections to end devices just enter the respective ip address and sub networks.

For one end of serial interface enter commands

interface serial 2/0

ip address 20.0.0.1 255.0.0.0

no shut
exit

and for other end of router

R1 (config)# interface serial 2/0
R1 (config-if)# ip address 20.0.0.1 255.0.0.0
R1 (config-if)# encapsulation ppp
clock rate 64000
no shut
exit.

In Router R2

R2 (config)# interface serial 2/0
R2 (config-if)# ip address 20.0.0.2 255.0.0.0
encapsulation ppp
no shut
exit

R2 (config)# interface serial 3/0
R2 (config-if)# ip address 30.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shut
exit

In router R3,

R3 (config)# ip address 30.0.0.2
R3 (config-if)# encapsulation ppp
no shutdown
exit,

R3 (config)# interface fastethernet 0/0
R3 (config-if)# ip address 40.0.0.1 255.0.0.0
R3 (config-if)# no shut
exit

Step 4 → Now check routing table of R1 router # show ip route.

Here R1 known three network
20.0.0.0 connected to R1 from R1

Gateway of last resort is not set
C 10.0.0.0/8 is directly connected 0/0
C 20.0.0.0/8 is directly connected serial 0/0
O IA 40.0.0.0/8 via 20.0.0.2 00:04:23

There must be interface up of keep OSPF process up, so is better to configure loopback address to routers.

R1
R1 (config-if)# interface loopback 0
ip address 112.16.1.253 255.0.0.0
no shut

R2
R2 (config-if)# interface loopback 0
ip address 112.16.253 255.0.0.0
no shut

R3 (config-ip) # interface loopback 0
R3 (config-ip) # ip address 153.14.1.254 255.255.0.0
no shut

Step 5: Now check routing of R2.

R3 # show ip route

Step 6: Create a virtual link b/w R1, R3 by this we create a virtual link to connect area 3 to area 0.

R1
router OSPF 1
area 1 virtual link 2.2.2.2
exit

R2 area 1 virtual-link 1.1.1.1
exit

Step 7 → R2 and R3 get updates about all
Now check routing table of 3
R3# show ip route

output → o/p ping from 10.0.0.10
ping 10.0.10
ping 10.0.0.10 with 32 byte of data.
Reply from 10.0.0.10 byte = 32 time = 9ms TTL=12
" "
" "
" "

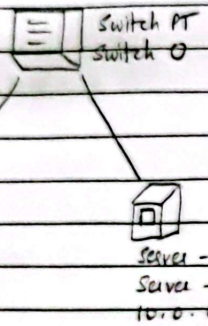
ping statistic for 10.0.0.10
packets = sent - 4 recieved = 4 lost = 0 [0% loss],
Approx round trip times in milli-seconds
minimum = 7ms, maximum = 9ms, Average = 8ms.

LAB - 8

a) Configure with server, DNS with a LAN.

AIM: configure DNS server

Topology:



Procedure:

→ select the PC, switch and a server connect it using cable.

→ Assign IP address to PC and server.

→ In server go to services DNS turn it on, write the name and address and press add.

→ In services select HTTP choose or edit the file.

→ Then in PC got Desktop in web browser. write the domain name to get result.

OBSERVATION:

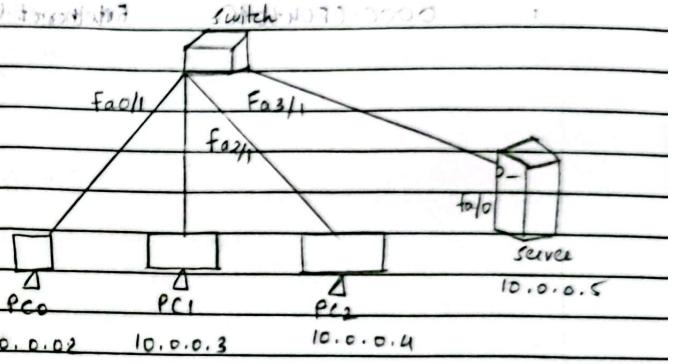
* The domain name system marks each IP address with domain name.

+ When entered the domain name the contacts of the specific IP address comes from server.

ART

10. To construct simple LANs and understand the concept and operation of Address Resolution Protocol (ARP).

AIM: To understand concept of ARP



Procedure:

→ select 3 PCs, one server and a switch.

→ Assing IP address to every element (PC & server).

→ Connect them to switch.

→ Link a inspect tool and then click on PCs to see the inspect ARP table.

→ Command line in CLI in arp-a.

→ Initially ARP table is empty.

→ In the CLI of switch write the command: show mac address table.

→ select a simple PDU and click on source and destination PC to see and dred on capture the observe the transfer of the PDU.

Observation:

Observed that switch as weak as nodes update the ARP table as and when how communication starts.

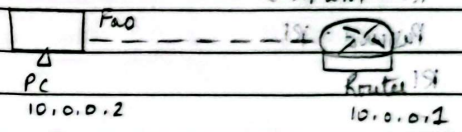
VLAN	MAC address	Port
1	0001.6420.C014	FastEthernet 1/1
1	0021.9938.C630	FastEthernet 2/1
1	0002.4A1639.3F7	FastEthernet 3/1
1	000C.CF04.C03C	FastEthernet 0/1

TELNET

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

Aim:
To understand and design TELNET to connect PC to router

Topology:



Procedure:

* Connect PC to the routers as shown in topology.
* Assign IP address to the PC and routers.
* Configure Routers:

Router CLI:

```
# interface fast ethernet 0/0
# ip address 10.0.0.1 255.0.0.0
# no shut
```

Then commands in Routers.

```
# enable
# config t
# host name R1
# enable server R1
# interface fast ethernet 0/0
# ip address 10.0.0.1 255.0.0.0
# no shut
    line vty 0 3
```

Observation:

→ It is observed that through telnet the hostname and password is given to access CLI modes in any other device.

→ In PC type pins ftd to know its connected R1 telnet 10.0.0.1 command used access to R1 host in PC Cputo

User Access verification

Password: PO

R1 > enable

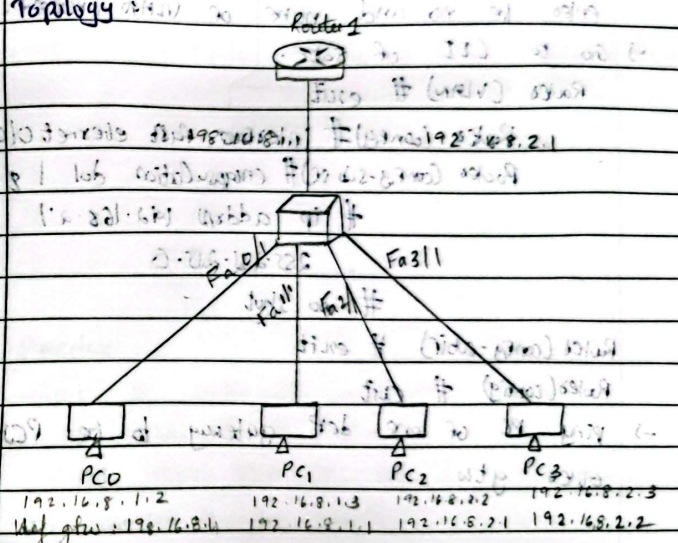
Password: R1

R1

Q. Creating a VLAN and make PCs communicate using the VLAN

Aim: To configure VLAN and make PCs communicate each other

Topology



Procedure:

→ select 4 nos and connect to switch connect 4 PCs to switch

→ Assign IP address and gtw to the PCs as shown in the topology.

→ configure Router for 2 PCs for: def gtw:

192.168.

enable

config t

interface fastethernet 0/0

ip address 192.168.1.1 255.255.255.0

no shut

exit.

→ Go to switch and in to config select database
→ set the VLAN number to 2
→ set VLAN name bmsit and click ADD.
→ Go to fastethernet 4/1 and set access as trunk.
→ Go to fastethernet 0/1 and set access as trunk.
→ Go to config tab of router select VLAN Database enter the no and name of VLAN created.
→ Go to CLI of router.

Router (VLAN) # exit

Router (config)# interface fast ethernet 0/0.1

Router (config-subif)# encapsulation dot1q 2

ip address 192.168.2.1

255.255.255.0

no shut

Router (config-subif) # exit

Router(config) # exit

→ ping PC of one dep gateway to the PC of other gtw.

OBSERVATION:

→ VLAN truncating allows switch to forward per dif VLANs over single link enabled through.

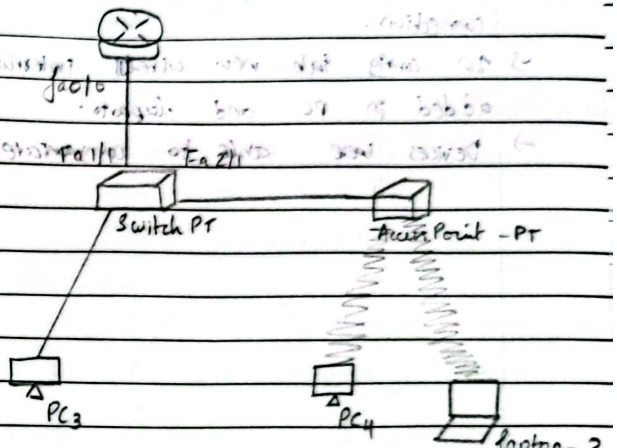
→ This is done by adding index info called tag to ethernet frame. This process is called tagging.

→ The switch understands VLAN.

→ The router understands VLAN.

Q. To construct a VLAN and make nodes communicate wirelessly.

Topology : Router



Procedure :

→ select the router and connect it to switch
→ connect pc to the switch.
→ select Access Point-PT from the wireless devices and connect it to switch.
→ configure Access point c-post-L-SSID Name as any name.
→ select WEP and give any 10 digit hexakey.
→ Configure PC1 and laptop with wireless standard.
→ Go to PC1 switch off the device Now existing PT-Host-Nm. 1AN to the component listed in the CHS.
→ Debug WPC300N wireless interface to empty post.
→ switch on the device.
→ Repeat for the same PCs laptop also.
→ Ping frame every wireless device to every other device and observe the router.

Date / /
Page

OBSERVATION:

→ Wireless LAN use WPA Protocol.
→ It requires SSID and key to be present.
→ It uses Access point to establish wireless connection.
→ In config tab new wireless interface have been added in PC and laptop.
→ Devices were able to communicate successfully.

13/11/24

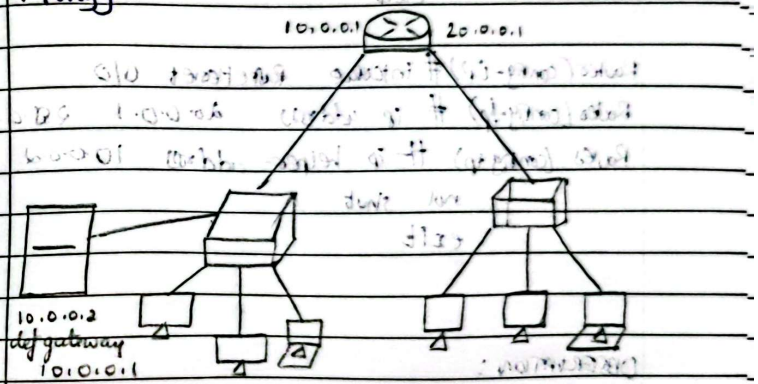
Date / /
Page

LAB - 5

Q. Configure DHCP with a LAN and outside LAN

Aim: To configure DHCP within a LAN and outside LAN

Topology:



Procedure:

1. Select a switch from device type selection.
2. Add 3 end devices on workbench and one servers.
3. Connect all end to switch through copper straight through cable.
4. Repeat same topology to set other network with diff servers.
5. Place on routers on workbench and connect 2 switches to it.

* In servers:

set IP address 10.0.0.1
Routers set IP address 10.0.0.2 (for 2 switches set)
Default gateway : 10.0.0.1

Router configuration:

Router > enable
Router # config terminal
Router (config-ip) # interface fastethernet 4/0.
Router (config-ip) # ip address 10.0.0.1 255.0.0.0
Router (config-ip) # ip helper-address 10.0.0.2
no shut
exit.

Router (config-ip) # interface fastethernet 0/0
Router (config-ip) # ip address 20.0.0.1 255.0.0.0
Router (config-ip) # ip helper-address 10.0.0.2
no shut
exit

OBSERVATION:

ping 10.0.0.4
packets: sent=4, Recieved=4, lost=0

1/1/25

CYCLE 2

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

```
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 '1'.join(result)

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

        pick += 1
    if tmp[0] == '1':
        tmp = xor(divisor, tmp)
    else:
        tmp = xor('0'*pick, tmp)
    checkword = tmp
    return checkword
def encode(data, key):
    l_key = len(key)
    appended_data = data + '0'*(l_key-1)
    remainder = mod2div(appended_data, key)
```

codeword = data + remainder

printf("Remainder : ", remainder)

printf("Encoded Data(Data + remainder) : ", codeword)

return codeword.

```
def decode_data(encoded_data, key)
    remainder = mod2div(encoded_data, key)
    print("Remainder after decoding :", remainder)
    if '1' not in remainder:
        print("No error detected in recieved data")
    else:
        print("Error detected in recieved data").
```

```
data = "10010001001001100"
key = "1101"
encoded_data = encode(data, key)
decoded_data = decode_data(encoded_data, key)
```

O/P :

Remainder = 111

encoded_data (data + remainder)

= 10010001001001111

Remainder after decoding = 000

No error detected in recieved data.

1/12/25

14. Write a program for congestion control using leaky bucket algo.

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int no_of_queries = 4, storage = 0, output_pkt_size = 1;
    int input_pkt_size = 4, bucket_size = 10, size_left;

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

O/P :

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

1/11/25

Date __/__/__
Page

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

Client (TCP):

```
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")
print(fileContents)
clientSocket.close()
```

Server (TCP):

```
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 recieve")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('Sent contents of' + sentence)
    file.close()
    connectionSocket.close()
```

Date __/__/__
Page

UDP:

Server side:
- The server is ready to recieve
- sent contents of server TCP.py
- the server is ready to recieve

Client side:
- Enter file name: server TCP.py
- Reply from server.

16 Using UDP sockets, write a client-server program to make client sending the filename & the server to send back the contents of the requested if present

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 ("reply from server")
print (fileContent.decode ("utf-8"))
clientSocket.close()
```

Server (udp.py):

```
from socket import *
serverName = "127.0.0.0"
serverPort = 12000
serverSocket.bind (("127.0.0.1", serverPort))
print ("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 ("sent contents of", end = ' ')
    print (sentence)
    file.close()
```

UP:

The server is ready to receive

Sent contents of server UDP.py } Server side
The server is ready to receive.

Enter file name: server UDP.py } client side
Reply from service:

1/11/25

Date __/__/__
Page ____

17. Tool Exploration - Wireshark

It is a powerful and widely used network protocol analyzer. It allows you to capture & inspect data packets travelling over a network in real-time, making it a crucial tool for studying computer networks.

Key features:

1. Packet Capture: Captures live network traffic from various interfaces.
2. Protocol Analysis: Supports 100's of protocols.
3. Filtering: Offers powerful filters to isolate specific packets or traffic types.
4. Visualization: Displays packet details with hierarchical layers.

Use case of wireshark

1. Network trouble shooting
2. Security Analysis
3. Protocol Study

Common:

* http: show only HTTP traffic
* tcp.port = 80: show traffic on TCP port 80
* ip addr = 192.168.1.1 : show packets to or from a specific address.
* udp: show only UDP traffic.