DATE :

Interface overview

In this we have a menu bar and in it we have menu bar tool, common tools, Logical / Physical workspace, Realtime / Simulation bar, Device type Selection box, Device Specific Selection box and user created Packet window.

It has 2 workspaces and 2 nodes. In Logical You can build your network and in Simulation you can run controlled networking

You change setting according to your Preferences, you can toggle b/w animation sounds, show link lights etc. In simulation buffer full action, Prompt, clear event list

In admin panal You can disable access to a Particular interface tabs interface looking etc

under Hide panel you can choose to show or hide Phy, config, CLI, Desktop, GUI & HTML. In font You can change the font option. You can set user Profile from the menu bar.

You have multiple algorithm and You can Save the Pakage as PKZ file from the Save as option

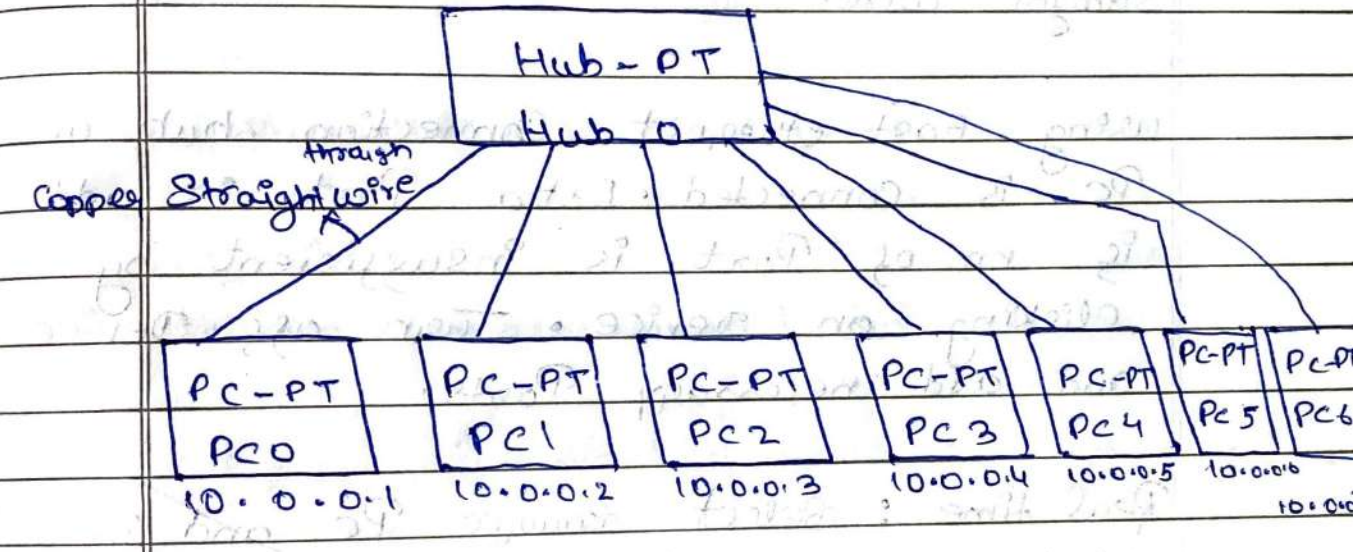
PAGE NO:
DATE:

Lab Work - 1

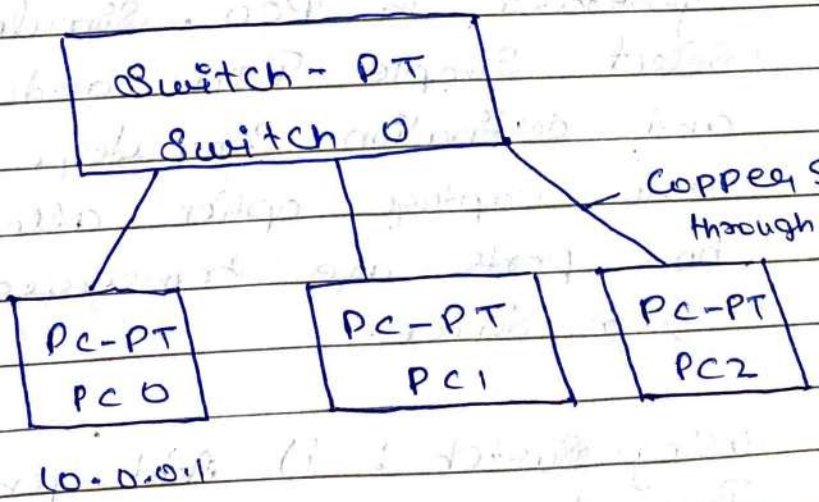
Aim :- Creating a topology and simulate sending simple PDU from source to destination using simple hub and switch as connecting domain.

Topology :

using hub :



using switch



Procedure

using hub i) First add generic hub and seven Pcs to workspace after that configure the ip address of each Pc in configuration tab and ensure that ip is different for each Pc or (device). In hub all Pc is connected using copper straight wire.

using Fast ethernet Connection hub and Pc is connected. Extra Port is added if no of Port is insufficient by clicking on device. Turn off Device and add necessary Ports.

Real time : Select Source Pc and in desktop tab, Select command Prompt option in Command Prompt type. Ping 10.0.0.3 This Ping Pc_2 and response is generated in Pco. Simulation time Select Simple PDU and Select Source and destination Computer. clicking on auto capture option allow us to see how packets are transferred to and from device.

using Switch : i) Add generic Switch and Pc's to workspace.

ii) Configure IP address of each PCs in the configuration that

PAGE NO:
DATE:

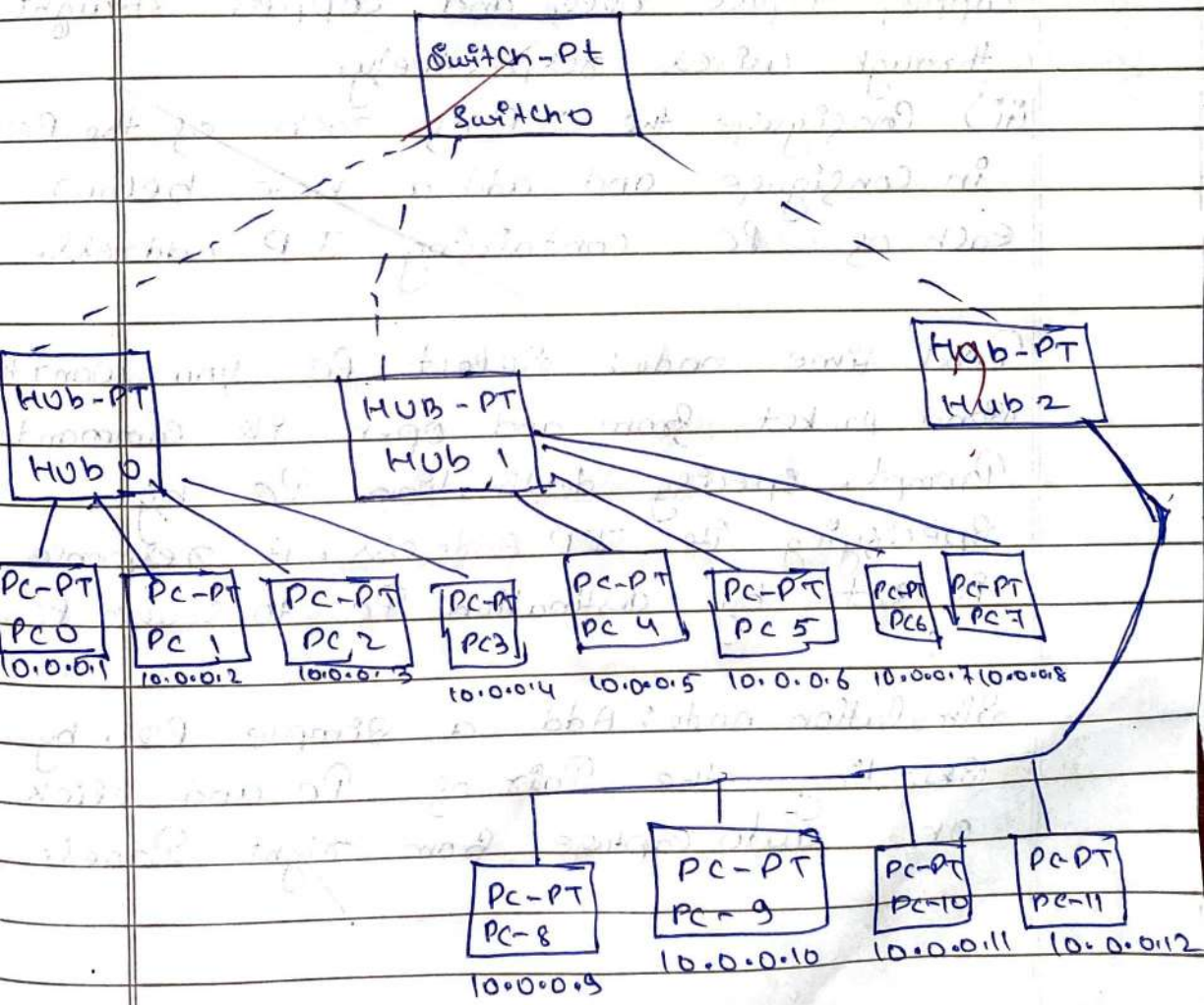
ip is different for each device

iii) Connect all PC's to switch using copper Straight through wire.

iv) If no. of Ports are insufficient then add extra ports by clicking on device. Turn off device and add necessary ports.

v) Write IP's of all device in note below the device.

Hybrid (using hubs and Switches)



DATE :

Real time : Select Source PC and in the desktop tab • Select Command Prompt option In command prompt option ; Ping destination PC by Specifying its Ip.

Simulation time :- Select Simple PDU and Select Source and destination Computer clicking an auto Capture option allows us to See how Packets are transferred

Hybrid mode i) Add a Switch, 3 hub's and 12 PC's to workspace.
ii) Convert three hubs to Switch and 4 PC's to each of the hubs using Copper Cross over and Copper Straight through wires respectively.
iii) Configure the IP of each of the PC's in Configure and add a note below each of PC containing IP address.

Real time mode : Select PC you want to Send packet from and open its Command Prompt. Specify destination PC by Specifying its IP Address. A response is set by destination PC to source PC.

Simulation mode : Add a simple PDU by Selecting the Pair of Pc and click on auto Capture from right Panel:

DATE :

Observation :

→ Hub

Learning outcomes → i) when Send a Packet in network the hub source the Packet and ends broad cast over the network i.e it Sends data to all the end devices in network and node where it matches with the specified address accepts the Packet and acknowledge it. Remaining nodes ignore the message.

ii) Common between hub and end devices is established through copper straight through wire as they belong to different layer.

iii) No of Ports can be added is needed by clicking on the device and adding the necessary Ports.

Result : PC > Ping 10.0.0.3

Pinging 10.0.0.3 with 32 byte of data

Reply from 10.0.0.3 byte = 32 time = 0ms

Reply from 10.0.0.3 byte = 32 time = 0ms

Reply from 10.0.0.3 byte = 32 time = 0ms

Reply from 10.0.0.3 byte = 32 time = 0ms

Ping Statistics for 10.0.0.3

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

→ Hybrid mode :

PAGE NO :
DATE :

Learning outcomes i) Switch and hub are connected through copper cross over as they belongs to the same network layer but PC and hub are connected through copper straight through as they belong to different network layers.

24/11/22

Exp 2:-

Aim:- Configure IP address to routers in Packet tracer, explore Ping, responses destination unreachable, reply, request, timed out.

Procedure:- End devices are connected to routers then Ip address is configure to end devices. Config Ip address and Subnet mask using command enable, Config terminal interface Fa 0/0, ip address 10.0.0.2 255.0.0.0 no shut down. Gateway is config for end device. End devices and interface are Pinged to check connection.

Topology:- Star topology

Result:- Successfully pinged end devices.

PC$ Ping 10.0.0.1

Pinging 10.0.0.1 with 52 bytes of data.

Reply from 10.0.0.1 byte = 32 time = 6ms TTL = 128

Reply from 10.0.0.1 byte = 32 time = 6ms TTL = 128

Reply from 10.0.0.1 byte = 32 time = 5ms TTL = 128

Reply from 10.0.0.1 byte = 32 time = 3ms TTL = 128

Ping Statistics for 10.0.0.1
Packet : Sent = 4, Received = 4
lost = 0 (0% Loss)
Approximate round trip times in milli Secs
minimum = 2ms maximum = 6ms Avg = 4ms

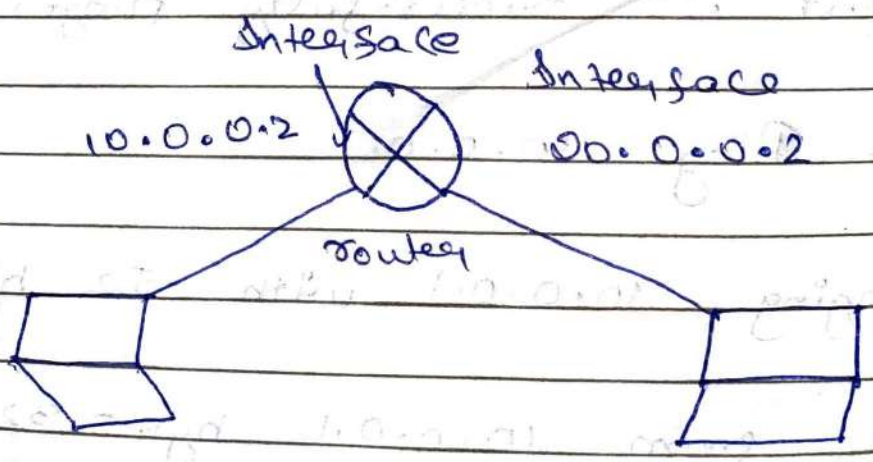
Pc> Ping 20.0.0.1

Pinging 20.0.0.1 with 32 byte of data:

Request time out :
Request time out :
Request timed out :
Request timed out :

Ping Statistics for 20.0.0.1

Packet Sent = 4, Received = 0, lost = 4
(100% Loss)



IP: 10.0.0.1
GateWay: 10.0.0.2

IP 2: 20.0.0.1
GateWay 20.0.0.2

PAGE NO :
DATE :

Observation

When we config both end devices and routers with appropriate ip address and by configuring subnet mask of interface of router as 255.0.0.0 and gateway of PC0 set as 10.0.0.2 which is of ~~PC0~~ Fa0/0 interface followed by same for PC1.
Then we could successfully Ping.

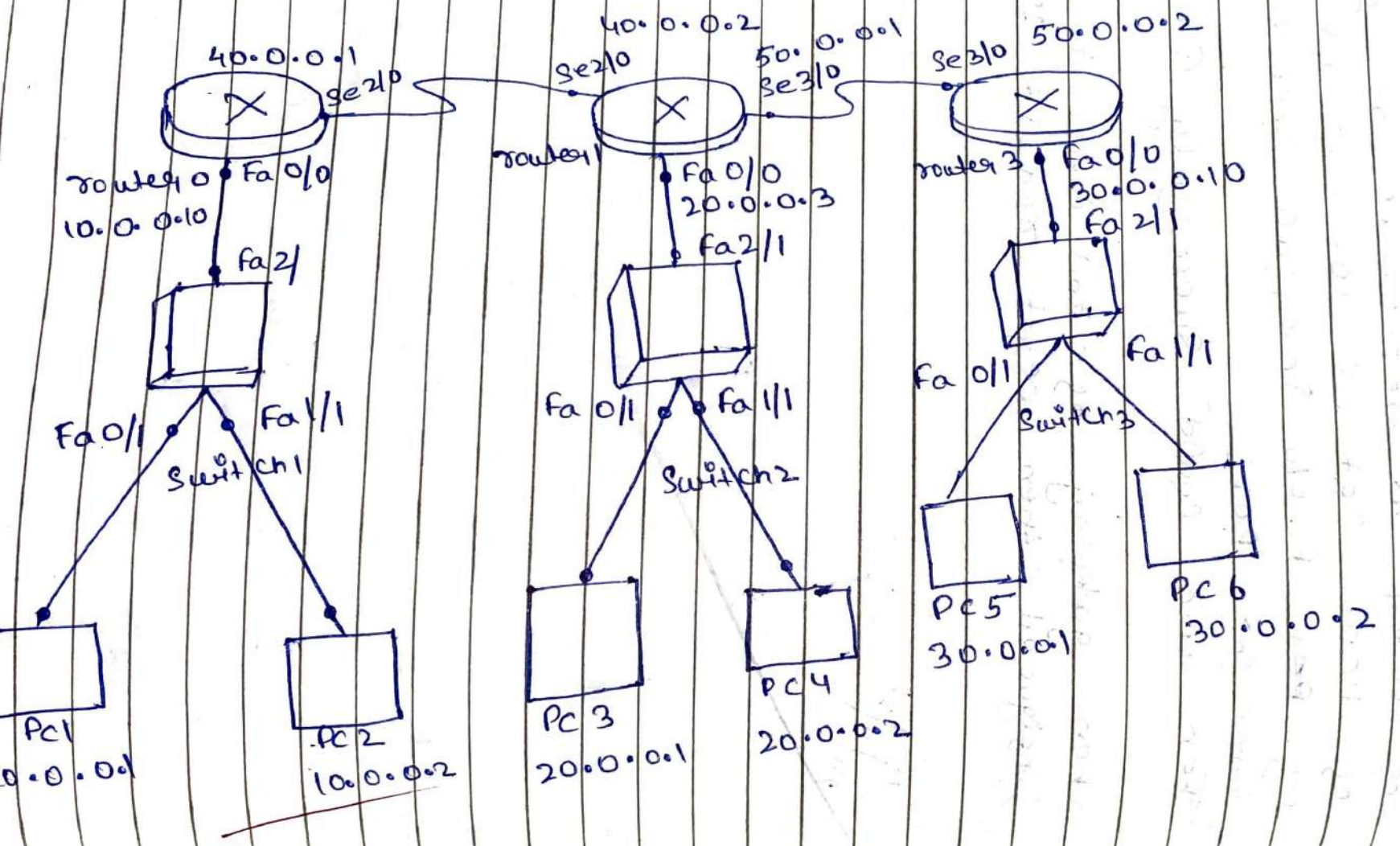
when gateway of end device is not configured then we get request time out.

24/11/22

Lab Work - 3

Aim :- Configure default route to router

Topology :-



PAGE NO :
DATE :

Procedure

→ 6, genric Pc & 3 switches, 3 routers are placed with respective notes with Ip address and Ports of routers. routers are connected through Serial DCE and coppers Straight wire is used for routers connections.

→ Set Ip address gateway, Subnet mask for each Pc by clicking on it repeat it for all 6 Pc's.

→ Configure routers using CLI IP address and Subnet is interfaced for both fastethernet 0/0 as 10.0.0.10 and 225.0.0.0 and Serial 2/0 as is default routers for router1 and this is done by Ip route 0.0.0.0 0.0.0.0 4.0.0.0.2.

→ Configure router 2 by setting 3-interfaces fastethernet 0/0 as 20.0.0.3 and 225.0.0.0 and Serial 2/0 as 40.0.0.2 & 225.0.0.0 and Serial 3/0 as 50.0.0.1 and 255.0.0.0.

router 0 does not have any default routers and static routing is done for the network 10 and 40 by the following Command

Ip route 10.0.0.0 255.0.0.0 40.0.0.1
Ip route 30.0.0.0 255.0.0.0 50.0.0.2

→ Configure for router 3 with ip address and subnet mask as Fastethernet 0/0 with 30.0.0.10 and 255.0.0.0 and Serial 2/0 with 50.0.0.2 & 255.0.0.0. The default router for router 3, router 2 and this set by the command.

Ip route 0.0.0.0 0.0.0.0 50.0.0.

observation

* One router cannot have 2 default router.

* The default router for first router is the middle router because any Packets which have to be delivered will go to middle router.

* Similarly for 3rd router default router is middle router

* No default routes for middle router because if one of the router is made default then there is a chance that the Packets which are to be sent to the Switch are sent to the router.

PAGE NO :
DATE :

Result

Ping 00.0.0.1

Pinging 00.0.0.1 with 32 bytes of data.

request timed out
reply from 20.0.0.1 byte = 32, time = 1ms
reply from 00.0.0.1, bytes = 32, times = 2ms
reply from 20.0.0.1 bytes = 32, times = 6ms

Ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data request timed out.

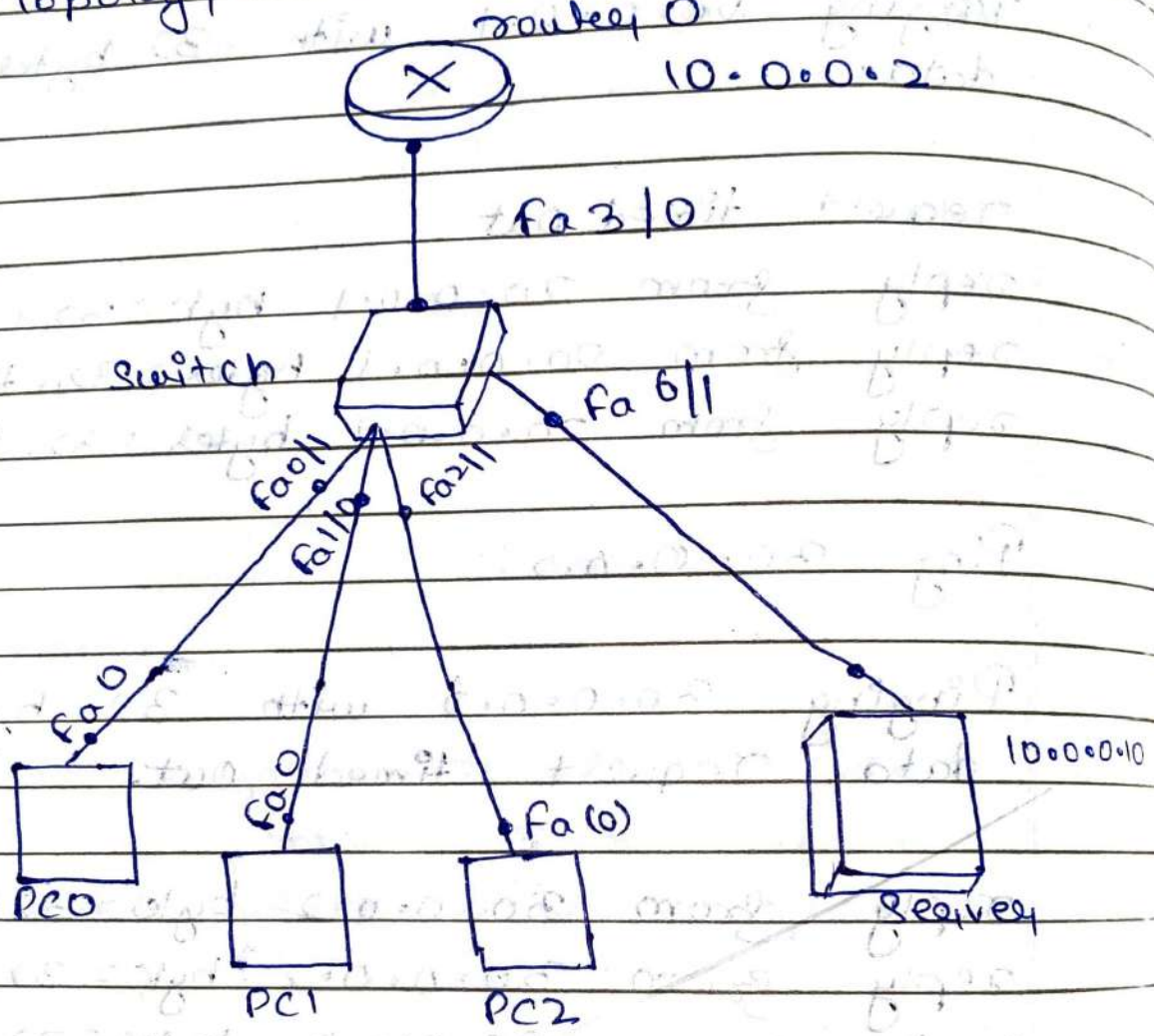
reply from 30.0.0.2 byte = 32 times = 4ms
reply from 30.0.0.2 byte = 32, times = 4ms
reply from 30.0.0.2 bytes = 32, times = 4ms

29/12/22

Lab Work

Aim :- Configuring DHCP within a LAN in a Packet tracer

Topology :-



Procedure :-

* Place 3 PC's, one router, one generic switch, one server and connect everything with copper straight wire
* Place note 10.0.0.2 for router indicating gateway and 10.0.0.1 for server indicating

PAGE NO:
DATE:

* Interface the corresponding fastethenet interface fastetheernet interface for router using CLI by typing commands.

→ Config t
→ interface fastetheernet 3/0
→ ip address 10.0.0.2 25.0.0.0
→ NO shut

* Now Config Server by cliking service tab in Server1
→ Set default gateway
→ Set DNS Server
→ Set Starting ip address as 10.0.0.3

* Observation

Learning Outcome :-

→ Clicking on IP Configuration on and selecting DHCP mode automatically fetches.

ip address, gateway from Server for which we had configured already

Result :-

Ping 10.0.0.4

Ping 10.0.0.5 with 32 bytes of data.

reply from 10.0.0.5 bytes = 32 time = 4ms

reply from 10.0.0.5 : 5 bytes = 32 time = 4ms

reply from 10.0.0.5 bytes = 32, time = 4ms

Ping statistics for 10.0.0.5

Packets sent = 4, received = 4

lost = 0

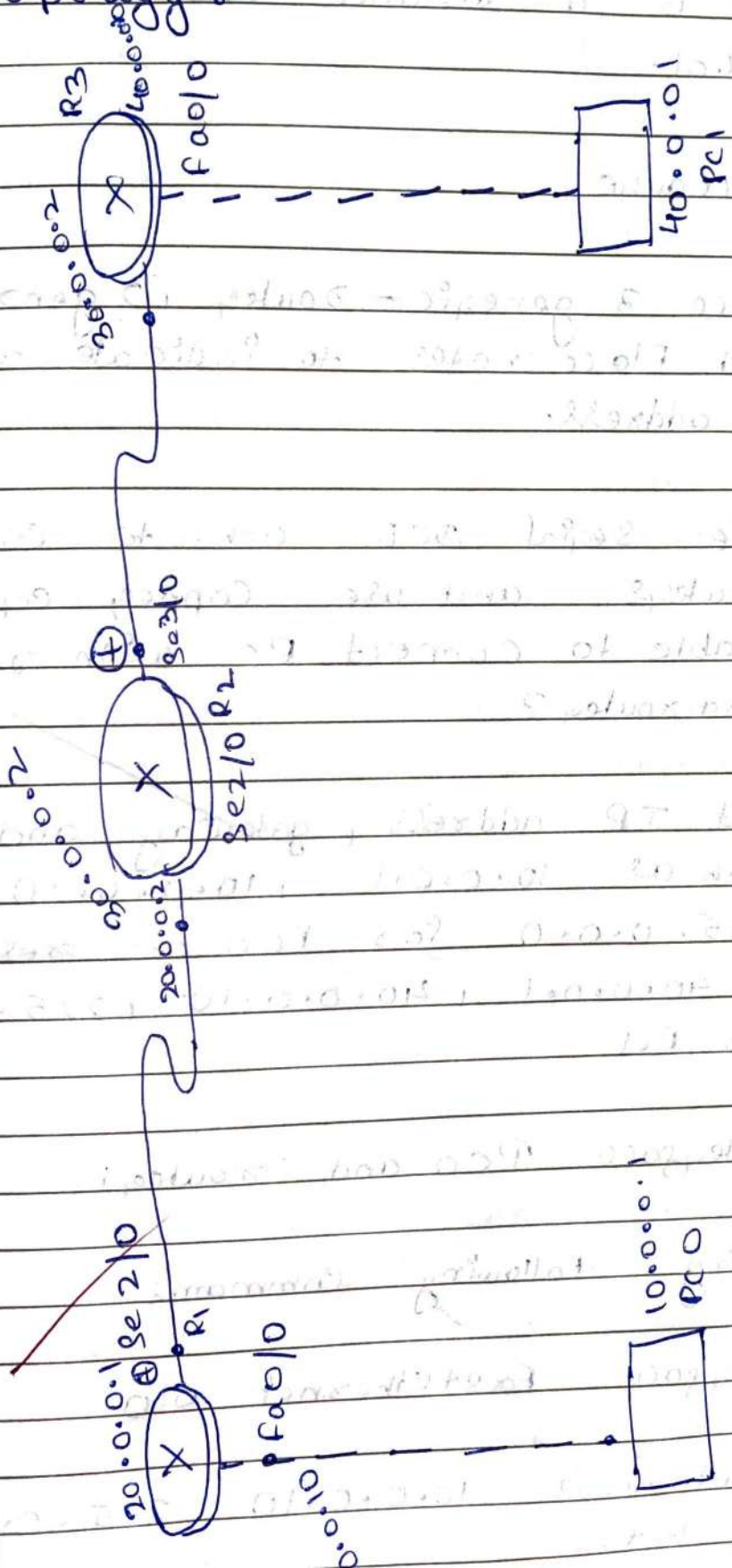
29/12/22

PAGE NO:
DATE:

Lab - 05

Aim : Configuring RIP routing Protocol in routers.

Topology :-



DATE :

* Rip is routing information Protocol which find shortest Path between source and destination network.

It is a distance vector routing Portcol.

Procedure

* Place 3 generic - routers, 2 genric - Pc and Place notes to indicate respective IP address.

* use Serial DCE cable to connect routers and use Copper cross over Cable to connect PC with router1 and router 3.

* Set IP address, gateway and Subnet mask as 10.0.0.1 , 10.0.0.10 , 225.0.0.0 for PC0 and respectively set 40.0.0.1 , 40.0.0.10 , 225.0.0.0 for PC1

* Interface PC0 and router1 using following Commands

→ Interface fastethernet 0/0

→ Ipaddress 10.0.0.10 225.0.0.0

→ no shut

PAGE NO:
DATE:

* For interfacing serial 2/0 of router 1 use following commands.

→ Interface Serial 2/0
→ ip address 20.0.0.1 225.0.0.0
→ encappulation PPP
→ Clock rate 64000
→ no shut

* Use above commands for interfacing router which has clock symbol in cable may to it and for other interfacing of routers use same above command except "clock rate 64000" command

* once all the green lights are visible, follow the commands below for each router.

→ router rip
→ network 10.0.0.0
→ network 20.0.0.0
→ exit

* Repeat above command's for router 2 and router 3 with respective network address

→ Observation

* Instead of using Static IP routing for all routers, by using using Rip, routing becomes easy when large numbers of routers are present.

Result :-

Pinging 10.0.0.1 with 32 bytes of data.

reply from 10.0.0.1 : byte = 32
reply from 10.0.0.1 bytes = 32
reply from 10.0.0.1 bytes = 32
reply from 10.0.0.1 bytes = 32

Ping statistics for 10.0.0.1
Packets : Sent = 4, received = 4,
lost = 0.

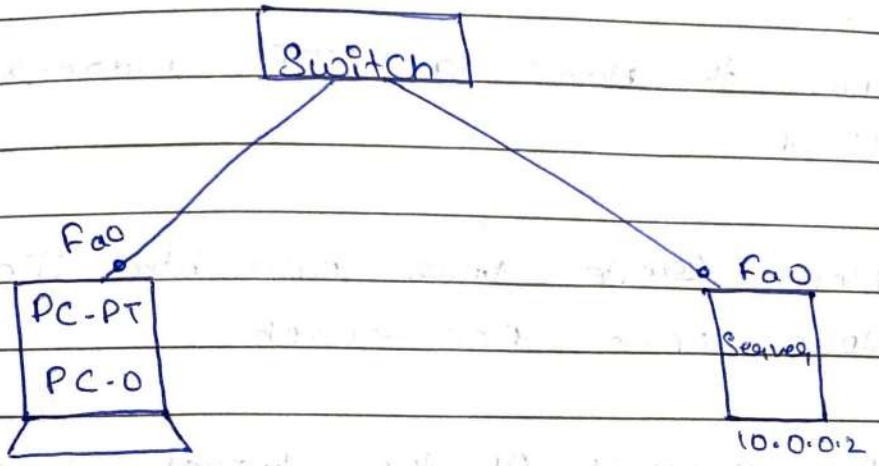
29/12/22

15/12/22

Lab-6

Aim:- Demonstration of WEB3 Server and DNS using Packet Tracer

Topology:-



Procedure

* Place a Pc, Switch and Server on the workspace and connect them.
* Set the IPA Address and Subnet mask of the PC as 10.0.0.1 and 255.0.0.0 respectively.
* Set the IP address and Subnet mask of Server as 10.0.0.2 and 255.0.0.0
* Open PC0 → Desktop → webbrowser → give IP address of Server (10.0.0.2)
* Open Server - service → HTTP → The http windows open click on the edit option of index.html and change the contents

Domain Naming System:-

* To activate DNS Open Server → Services → DNS → On → enter the name of the resource record and IP address of the Server and click on Add.

* Now the Name and IP address is fixed.

* Now Create your own html file.
* Now open PC0 → web

* The contents of the HTML file created is shown

Result:-

HTML File created :- <html>
<head> Hellow </head>
<body>
Name: Himanshu
branch: CSE
</body>
</html>

Output:- Hello
Name: Himanshu
branch: CSE

CV
Table
form.

Lab-7

PAGE NO:
DATE: 29/12/22

```
#include <stdio.h>
#include <string.h>
#define N strlen(gen-Poly)
char data[28];
char check-value[28];
char gen Poly[10];
int data length i,j;
void XOR() {
for(j=1; j<n; j++)
check value[j] = ((check-value[j]==gen Poly
                    ? '0' : '1'); }
void crc(){
for(i=0;i<n;i++)
   check value[i] = data[i];
do {
      if(check value[0]=='1')
         XOR();
      for(j=0; j<N-1; j++)
check_value[j] = data[i+1];
} while(i<=data-length+N-1);
}

void receiver(){
Printf("Entered the received data:");
Scanf("%s", data);
Printf("Data received: %s", data);
crc()
for(i=0; i<n-1)&& check_value[i]!='1') i++;
if(i<N-1)
Printf("\n Error detected");
```

```
else
Printf ("\n No error detecte");
}
int main()
{
    Printf ("Enter Codeword");
    Scanf ("%s", data);
    Printf ("Enter Generating Polynomial");
    Scanf ("%s", gen-Poly);
    data_length = strlen (data);
for(i = data_length; i < data_length + N-1; i++)
    data[i]='0'
Printf ("\n Data with zeros: %s", data)
    calc();
Printf ("CRC value is: %s", check_value);
for (i = data_length; i < data_length N+1
        i++);
Printf ("Find data Sent is %s", data)
receiver();
return 0;
}
```

out put:

Enter codeword: 10110101010
Enter Generating Polynomial : 1010 — CRC-16
Data with zeros: 1011010101010000
CRC value : 000
Final data to be sent is:- 1011010101010000
Enter Data Received:- 1011010101010000
error detected

Lab-8

PAGE NO:
DATE: 5/01/23

Write a program for congestion control using leakly bucket Algorithm

```
#include <iostream>
using namespace std;
int main()

Count << " Enter bucket size " << end;
int bucket size;
int filled = 0;
int output rate;
int input rate;
int choice
cin >> bucket size;
Count << " Enter output rate " << end;
Cin >> output rate;
do {
    Cout << " Enter packet size " << end;
    Cin >> input packet;
    if ( input packet <= bucket size ) {
    if ( filled + input packet > bucket size)
{
  Cout << " Packet too big for bucket " <<
    endl;
}
else {
filled = filled + input bucket;
}
else
{
Cout << packet is too big for bucket " << endl
}
```

```
else {
if (filled <= output rate)
{
    filled = 0;
}
else {
    filled = filled - output rate ; }
cout << " Amount of bucket filled " << filled;
cout << " Do you want to enter another
        Packet (9 for yes, 8 for no) " << end
cin >> choice
}
while (choice == 9);
}
```

Output:

```
Enter Bucket size :- 500
Enter output rate :- 50
Enter packet size :- 700
Packet too big for bucket
Do you want to enter packet (9 Yes & No) : 9
Packet filled :- 1
Enter Packet size = 200
Packet filled :- 150
Do you want to Enter packet (9 yes, 8 No) = 9
Enter packet size : 250
bucket Size :- 350
Do you want to enter packet (9 yes & No) : 9
Enter packet size : 250
Packet too big for bucket
Amount of bucket filled 300.
```

12/1/23

Lab - 9 Bellman - Ford

PAGE NO :
DATE :

```
#include <stdio.h>

int Bellman_ford (int G[20][20], int v, int E
int edge [20][20])
{
 int i, u, v, k, distance [20], Parent [20], s, flag
                                              =1;
for (i=0; i<v ; i++)
      distance [i] = 1000, Parent [i] = -1;
 Printf ("Enter Source:");
 Scanf ("%d", &s)
 distance [s-1] = 0;
 for (i=0; i<v-1; i++)
   {
      for (k=0 ; k<E ; k++)
   {
      u = edge [k][0] , v = edge [k][1];
   if ( distance [u] + G[u][v] < distance[v])
   distance [v] = distance[u] + G[u][v],
   Parent [v] = u;
   }
}
For (k=0; k<E; k++)
        {
            u = edge [k][0] , v = Edge [k][1];
   if (distance [u] + G [u][v] < distance [v])
distance [v] = dist
        flag = 0;
}
int (flag)
   for (i=0; i<v ; i++)
        Printf ("vertex %d → cost = %d
                Parent = %d \n" i+1,
```

```
distance[i], Parent[i]+1);
return flag;

int main()
{
    int v, edge[20][2], G[20][20], i, j, k=0;
    Printf("Bellman Ford \n");
    Printf("Enter no of vertices: ");
    Scanf("%d", &v);
    Printf("Enter graph in matrix form: \n");
    for(i=0; i<v; i++);
        for(j=0; j<v; j++)
    {
        Scanf("%d", &G[i][j]);
        if(G[i][j]!=0)
            edge[k][0]=i, edge[k++][1]=j;
    }
if(Bellman_ford(G, v, k, edge)
    Printf("\n NO Negative weight cycle\n");
else
    Printf("\n Negative weight cycle exist\n");
return 0;
}
```

DATE :

Lab-10 Dijkstras

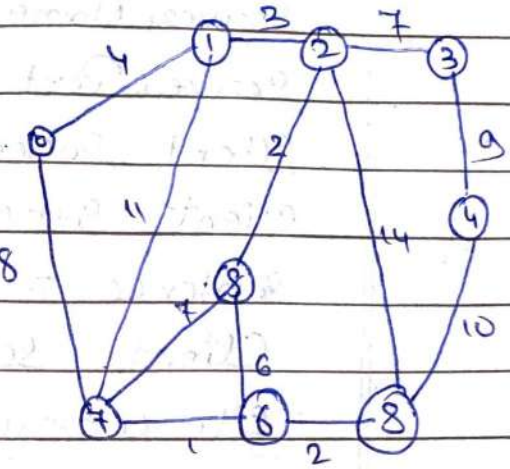
```
#include <bits/stdc++.h>
#include <limits.h>
#include <stdio.h>
    using namespace std;

#define V 4

int minDistance (int dist[], bool sptset[])
{
    int min = Int_Max, min_index;
for ( int v=0; v<V; v++)
        int (sptset[v]==false && dis[v]
                <=min)
        min = dist[v], min_index=v;
return min_index;
}
void PrintSolution (int dist[])
    {
        Printf ("Vertex \t\t Distance from Source\n");
for (int i=0; i<V; i++).
Print ("%d \t\t %d\n", i, dist[i]);
}
void dijkstra (int graph[V][V], int src)
{
  int dist[V];
  bool sptset[V];
  for (int i=0; i<V; i++)
      dist[i] = Int_Max, sptset[i]=False;
dist[src]=0;
    for (int count = 0; count<V-1; count++)
  {
```

```
int u = minDistance(dist, sptSet);

    sptSet[u] = true;

    for (int v = 0; v < V; v++)

        if (!sptSet[v] && graph[u][v] &&
            dist[u] != INT_MAX && dist[u] +
          graph[u][v] < dist[v])
        dist[v] = dist[u] + graph[u][v];
}
    printSolution(dist);
}
int main()
{
    int graph[V][V];
Cout << "Enter the graph " << endl;
for (int i = 0; i < V; i++)
  {
      for (int j = 0; j < V; j++)
          cin >> graph[i][j];
  }
    dijkstra(graph, 0);
    return 0;
}
```

PAGE NO :
DATE :

Output

Vertex	Distance
0	0
1	4
2	12
3	19
4	21
5	11
6	9
7	8
8	14



Lab-11

Socket Programming

clitcp.py

```
from socket import *
Seavey Name = 'Desktop- HMPODEC'
Seavey Port = 12530
Client Socket = Socket(AF-INET, SOCK-STREAM)
client Socket . Connect (Seavey Name, Seavey Port)
Sentence = input ("Enter file name")
Client Socket . Send (Sentence . encode())
file contents = Client Socket . rev(1024).
                decode()
Print ( ' From Seavey : ', file contents)
clinet Socket . close()
from Socket import *
  Seavey Name = " 127.0.0.1"
  Seavey port = 12000
Client Socket . Send to (bytes ( Sentence, 1, utf
(Seavey-name, Seavey port)
file content ( Seavey Address = client Socket
                recv from (2048)
Print ('From Seavey : ' file contents)
    Client Socket . close()
```

Seaup dp.Py

```
from Socket import *
  Seavey pocket = 12000
Seavey Socket . blind (" 127.0.0.1" (Seavey
                       Port)
Print (" The Seavey is ready to receive)
while 1
```

PAGE NO :
DATE :

```
Sentence  clientAddress = ServerSocket.recvfr
                                (2048)
file = open (sentence, "r")
   d = file.read(2048)

ServerSocket.sendto(byte(1,"utf-8"),client
                                Add(res)
Print ("Sent back to client",1)
    file.close()
while 1:
   connectionSocket, addr = ServerSocket.accept
                                ()

sentence = connectionSocket.recv(1024).decode()
   file = open (sentence,"r")
     d = file.read(1024)
   connectionSocket.send(1.encode())
   file.close()
   connectionSocket.close()
```

Output :-

```
Server started
waiting for a client....
connected
    Server started
       waiting for a client:
       client Accepted.
```