Cisco Packet Tracer — LAB-1

Date 3/11/22
Page 1

Intro

Protocols - LAN: Ethernet (CSMA/CD), PPPoE
Switching
TCP/IP: HTTP, HTTPS, DHCP
Routing, Other, WAN, Security, QoS

Logical Workspace - N/w topo Creation, devices - generic, real mg
Routers, DSL/Cable, devic interfaces

Physical workspace - N/w topo creation, intercity views, Cable, wireless association mgnt.

Realtime Mode - Realtime protocol updates, Confi of routers & switches, Menu based DHCP, DNS, HTTP & NTP servers.

Simulation Mode - Packt animation, global evnt list, OSI model, device table view, User defined multiple packet scenarios

Local Authoring & Sharing - Extensive file sharing options, easily translated GUI, External app, IPC

ping ipaddress

Connecting Two devices / establishing Connection

1. End devices → Generic (for 2 devices)
2. Connection ↯ → ↯ establish Connection until you get green signals at both ends.

3. Go to PC0 → Config → FastEthernet → IP address 10.0.0.1
PC1 → Config → FastEthernet → IP address 10.0.0.2

4. Add Simple PDU(P)  → Place on two devices also.

5. In Simulation mode you will get the idea that how packet is send. Go back to see Successful.

6. Click on PC0 → Desktop → Cmd prompt →
PC > ping ipadd of desti here 10.0.0.2

The result of ping is

Ping statistics for 10.0.0.2:
Packets: Send = 4, Received = 4, Lost = 0
(0% loss)
Approximate round trip in milli-seconds
Minimum = 0ms, Max = 0ms, Avg = 0ms

When you ping for diff IP # Say 10.0.0.3
there will be 100% Loss

Aim: using hub & switch as connecting devices. from source to destination

TTL - Time To Live

Procedure: Hub

1. Hubs → Generic (1)
2. End Devices → Generic (desktop) (4 devices)
3. Connect the device using black wire
4. Configure the IP address as 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4 so on
5. Next in simulation mode add PDU from Src → dest & click on auto capture/play
6. We see that PDU's r being sent to all the devices but only to the correct destination. Only particular device accept it rest all reject it.

Switch

1. From device type selection box select 4 devices by clicking end devices.
2. Connect in the similar manner as you did for the Hub.

Results

>> ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data

Reply from 10.0.0.4 bytes 32 times=0ms TTL

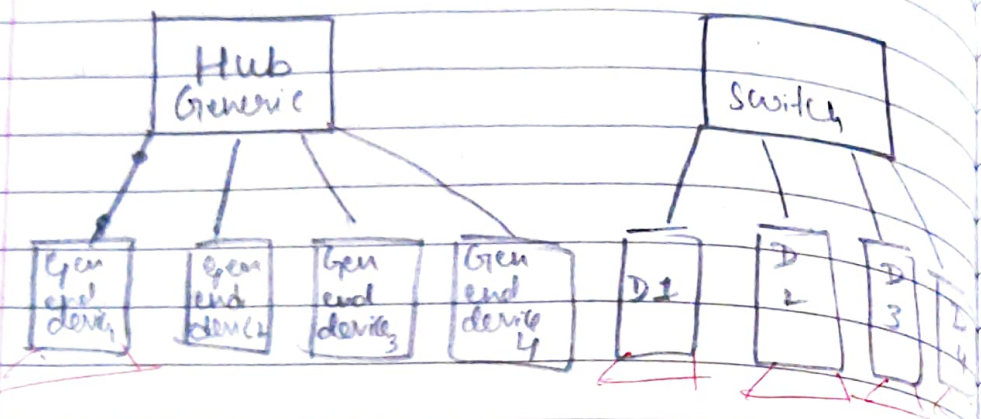
Statistics

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

Round trip times

Minimum = 0ms, Max = 0ms, Avg: 0ms

Topology



Observation

We observed that in Hub, the packets are broadcasted to all end devices connected.

Only for the destination, the packets are received ✓

But in the case of Switch, the PDU are only sent to particular destination

LAB - 3

Date 17/11/22
Page 5

2. **Aim:** Configuring IP address to Routers in Packet Tracer.
Explore the following messages:
1) Ping Responses
2) Destination Unreachable
3) Request timed out
4) Reply.

Procedure

1. Go to Routers → Router Generic → drag
2. Go to End device → Laptop-PT → drag & drop 2 laptops
3. Connections → Automatically Choose Connection type
4. Label the Laptops → Config → ip Laptop as 10.0.0.1

6. Go to Router → CLI type the Commands - No dialog
Router > enable
Router # Configure terminal
Enter Configure Commands, one per line.
Router (Config) # interface fastethernet 0/0
Router (Config-if) # ip address 10.0.0.2 255.0.0.0
Router (Config-if) # no shoutdown

5. Configure for Laptop-PT 1 as 20.0.0.1 & Subnet mask as well.

6 Cont. By clicking on f/o get to know the fast ethernet enter following Commands

interface fastethernet 1/0
ip address 20.0.0.2 255.0.0.0
no shutdown

You will get - Interface FastEthernet 1/0, Changed State to up

7 Later, Laptop0 → Desktop → Command Prompt → Enter as follows

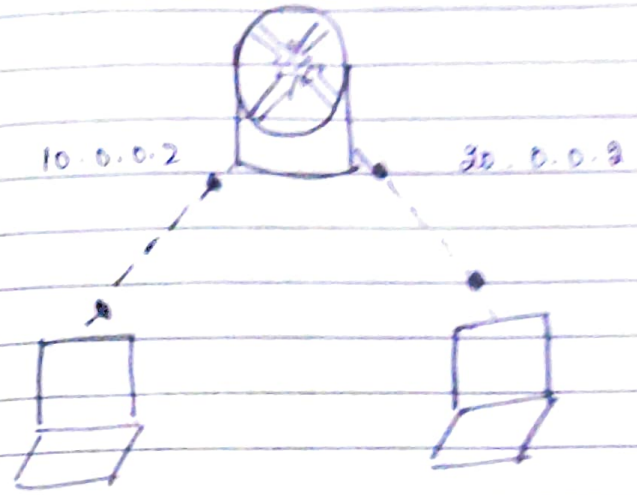
> ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of
Reply from 10.0.0.1: bytes=32 time=6
TTL=128
Reply from 10.0.0.1: bytes=32 time
Ping Statistics for 10.0.0.1:
Packets: Sent=4, Received=4, Lost=0(0
Approximate round trip times in ms:
Minimum=6ms, Maximum=13ms, Avg

Try to ping for 20.0.0.1
Laptop1 → Desktop → Cmd prompt → ping
20.0.0.1
You will get the reply

8. Laptop0 → Config → Gateway 10.0.0.0
Laptop1 → Config → Gateway 20.0.0.0

9. Simulation → Drop PDU Laptop0 to Laptop1 until you get successful → Capture/Forward ✓

Topology



7... You will be getting Request timed out when you try to ping 20.0.0.1 before Configuring other ~~side~~ of the router

PC> Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.

Request timed out.

:

Ping Statistics for 20.0.0.1:

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

~~Result~~	0.000	-	PC0	ICMP	☐
Observation	31.964		Router0	CDP	☐
	31.944		Router0	CDP	☐
	31.945	Router0	PC0	CDP	☐
	31.945	Router0	PC1	CDP	☐

Result: PC0 PC1 ICMP 0.000 N

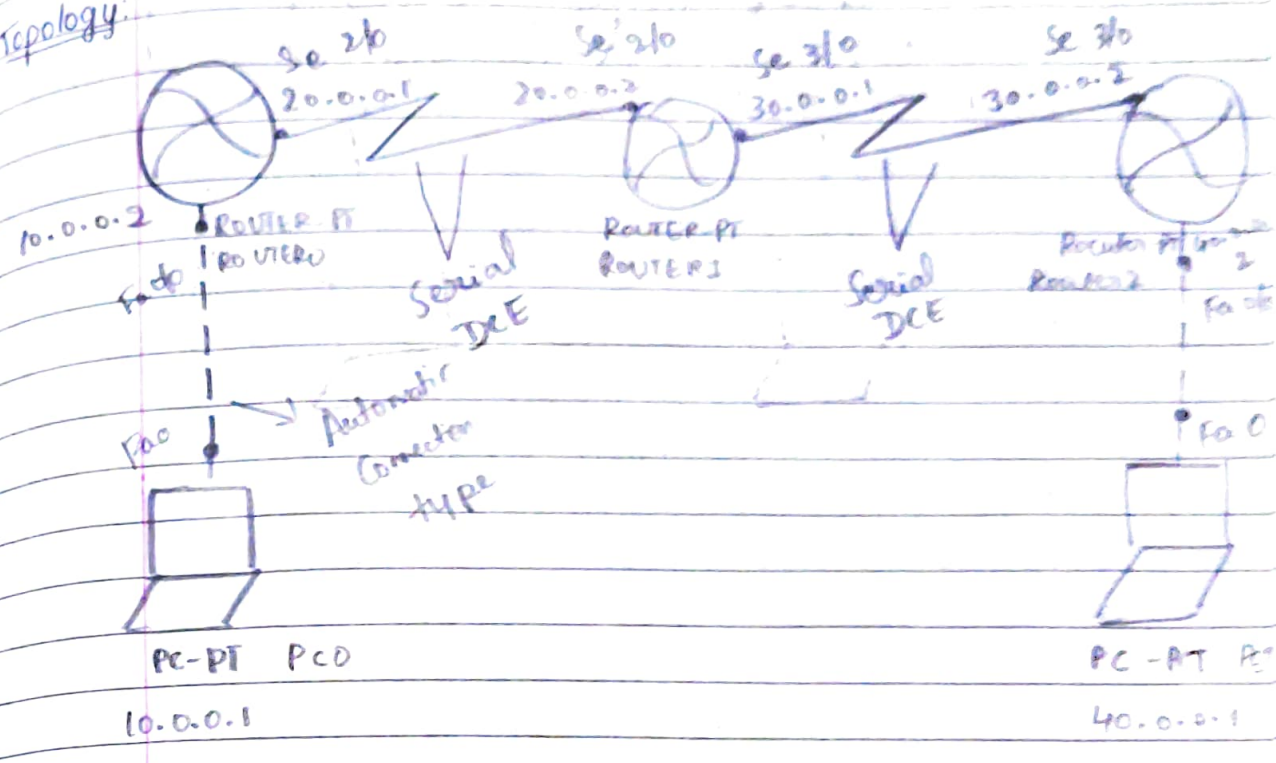
LAB - 1

3

24/11/22

Aim: Configuring ~~default~~ Static route to the router

Topology:



Procedure:

* Routers → Router-PT → drag and drop 3 of it on screen.
* End Devices → PC-PT → Place 2 PC-PT towards Router 0 & 1.
* Connect Router → PC with (connections → Automatically connection type)
* Connect Router → Router with (connections → Serial DCE)
* PC-PT 0 → Give the ip address for the following 10.0.0.1
* PC-PT 1 → ip 40.0.0.1

ip route destination address subnet mask next hop address

Router-0 Configuration

Continue with configuration dialog [Yes/No]? N
Router > enable
Router # config terminal
Router (config) # interface FastEthernet 0/0
Router (config) # IP address 10.0.0.2 255.0.0.0
Router (Config) # no Shutdown
Router (Config) # exit

Router (Config) # interface Serial 2/0
Router (Config) # ip address 20.0.0.1 255.0.0.0
Router (Config-if) # No Shutdown Router (Config) # ex
Router (Config) # ip route 30.0.0.0 255.0.0.0 20.0.0.2
ip route 40.0.0.0 255.0.0.0 20.0.0.2

Note: Syntax ip route destination address Subnet mask next-hop address

Router # Show ip route
C 10.0.0.0/8 is directly Connected, FastEthernet
C 20.0.0.0/8 is directly Connected, Serial 2/0
S 30.0.00/8 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2

Router-1 Configuration

Continue with configuration dialog [Yes/No]?
Router > enable
Router # Config terminal
Router (Config) # interface ~~fastEthernet~~ Serial 1/0
Router (Config-if) # ip address 30.0.0.2 255.0.0.0
Router (Config-if) # no Shutdown

Router	(config)#	exit
Router	(config-if)#	interface Serial 3/0
Router	(config-if)#	ip address 30.0.0.2 255.0.0.0
Router	(config-if)#	no Shutdown
Router	(config-if)#	exit
Router	(config)#	ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router	(config)#	ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router #		Show ip route

Router-2 Configuration

Continue with Configuration dialog [Yes/No]? No

Router	>	enable
Router	#	Config terminal
Router	(config)#	interface Serial 3/0
Router	(config-if)#	ip address 30.0.0.2 255.0.0.0
Router	(config-if)#	no Shutdown
Router	(config-if)#	exit
Router	(config)#	interface fast ethernet 0/0
Router	(config-if)#	ip address 40.0.0.2 255.0.0.0
Router	(config-if)#	no Shutdown
Router	(config-if)#	exit
		ip route
Router	>	enable
Router	#	Config 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 25.00.0 30.0.0.1
Router	(config)#	exit
Router	(config)#	Show ip route

PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of dat
Reply from 40.0.0.1 : bytes = 32 time = 9ms TTL
Reply from 40.0.0.1 : bytes = 32 time = 6ms TTL=
Reply from 40.0.0.1 : bytes = 32 time = 6ms TTL
Reply from 40.0.0.1 : bytes = 32 time = 9ms TTL=

Ping Stastistics

Packets : Sent = 4 , Received = 4, Lost = 0 (0 % L

Approximate round trip times in milliseconds

Minimum = 3 ms , Maximum = 9 ms , Average = 4

PDU from Source end device (10.0.0.1)
Sent Successfully to destination end device (40.0

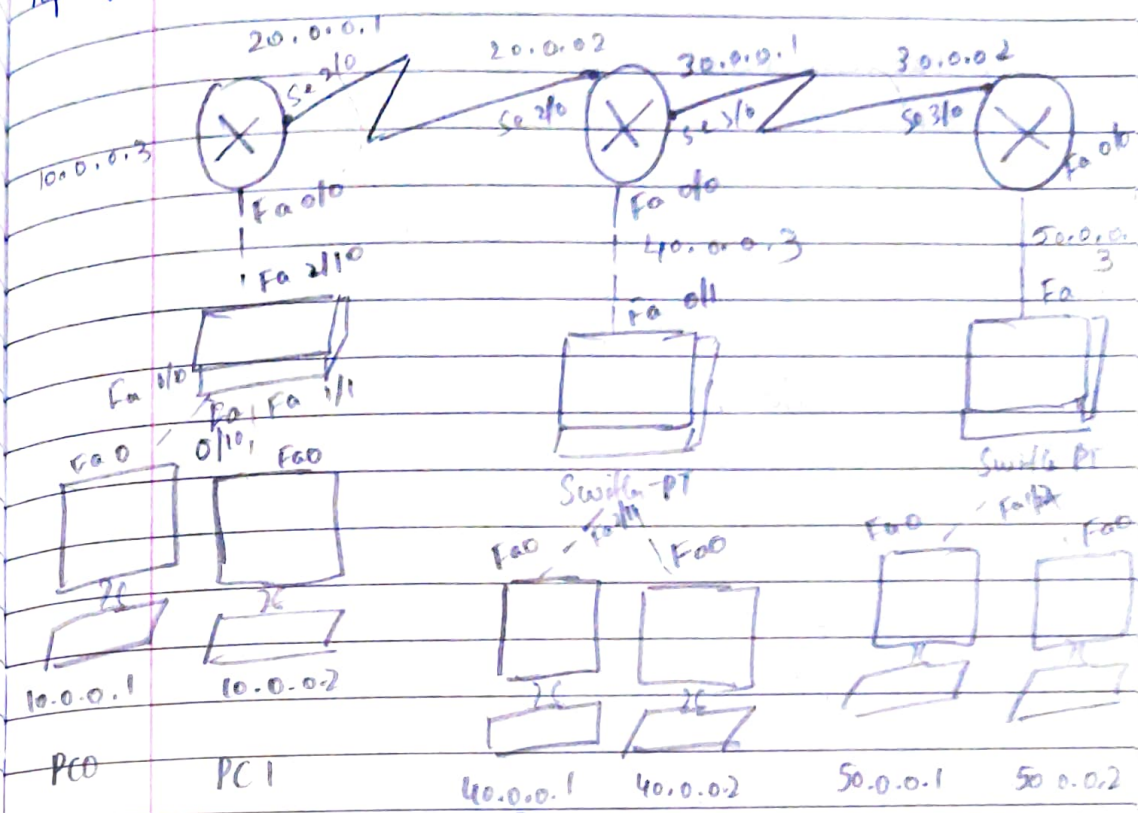
Observation : Packet sent from 10.0.
to 40.0.0.1 has received success
Using the static routing experiment

LAB - 5

Date 1/12/30
Page 13

Aim: Configuration of default route.

Topology:



Setting up Components →

1) End Routers → Router PT [3 nos]
2) Switches → Switch-PT [3 no's]
3) End devices → PC-PT [6 no's]

Connections →

1) Connections → Automatically choose Connection type
   [b/w router → switches → PC]
2) Connections → Serial DCE
   [b/w the routers]

Naming →

1)	Routers -	Router-PT Router1	Router-PT Router2	Router-PT Router3
2)	Switches -	Switch-PT Switch0	Switch-PT Switch1	Switch-PT Switch2

PC-PT PC-PT PC-PT PC-PT PC-PT

3) PC - PC0 PC1 PC2 PC3 PC4

Assigning IP to PC's →

Click on the particular PC

1) FastEthernet0 → IPV4 Address as 10.0.0.1
SubnetMask 255.0.0.0
for PC0, SubnetMask remains same.
Similarly do it for all PC's.

2) PC1 → IPV4 10.0.0.2
PC2 → IPV4 40.0.0.1
PC3 → IPV4 40.0.0.2
PC4 → IPV4 50.0.0.1
PC5 → IPV4 50.0.0.2

Router 0 Configuration →

would you like to enter initial Config dia [Yes/No]:

Router > enable
Router # Config terminal
Router (Config)# interface fastEthernet 0/0
Router (Config-if)# ip address 10.0.0.3 255.0.0.0
Router (Config-if)# no Shutdown

Alternatively →

Select partialar Router → Choose the type of interface → IPv4 address type it & Selet the Subnet mask.

Interface FastEthernet 0/0, Changed State to up
Router (Config-if)# exit
Router (Config)# interface Serial 2/0
Router (Config-if)# ip address 20.0.0.1 255.0.0.0
Router (Config-if)# no Shutdown

To check for Router 0 [enabled routes] →

Router > enable
Router # Config terminal
Router (Config)# ip route 0.0.00 0.0.00 20.0.0.2
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

Router 2 Configuration →

Router >enable
Router # Config terminal
Router (config)# interface Serial 2/0
Router (Config-if)# ip address 20.0.0.2 255.0.0.0
Router (config-if)# no Shoutdown
Interface Serial 2/0, Changed State to up
Router # exit
Router (Config)# interface Serial 3/0
Router (Config-if)# ip address 30.0.0.1 255.0.0.0
Router (config-if)# no Shutdown
Interface Serial 3/0, Changed State to down
Router # exit
Router (Config)# interface fast Ethernet 0/0
Router (Config-if)# ip address 40.0.0.3 255.0.0.0
" # no Shutdown
Interface FastEthernet 0/0, Changed State to up
exit

Router> enable
Config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.2
exit
Show ip route

Set the
Say for PC0 - Gateway 10.0.0.3

C 20.0.0.0/8 is dir Conn, Serial 2/0
C 30.0.0.0/8 " " " " 3/0
C 40.0.0.0/8 " " " FastEthernet 0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.2

> enable
Config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.1
exit
Show ip route
C 20.0.0.0/8 is dir Conn, Serial 2/0
C 30.0.0.0/8 " " " " 3/0
C 40.0.0.0/8 " " " FastEthernet 0/0
S* 0.0.0.0/0 [1/0] via 20.0.0.1

Router 3 Configuration →
Router> enable
Configure terminal
interface Serial 3/0
ip address 30.0.0.2 255.0.0.0
~~exit~~ no Shutdown
Interface Serial 3/0, Changed State to up
interface fastethernet 0/0
ip address 50.0.0.3 255.0.0.0
no Shutdown
Interface FastEthernet 0/0, Changed State to up
exit
> enable
Config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1
ip route 0.0.0.0 0.0.0.0 40.0.0.3
exit
Router # Show ip route

C 30.0.0.0/8 is dir conn Serial3/0
C 50.0.0.0/8 is " " FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Pinging →

PC0 →

ping 50.0.0.2

pinging 50.0.0.2 with 32 bytes of data:
Reply from 50.0.0.2: bytes=32 time=33ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125

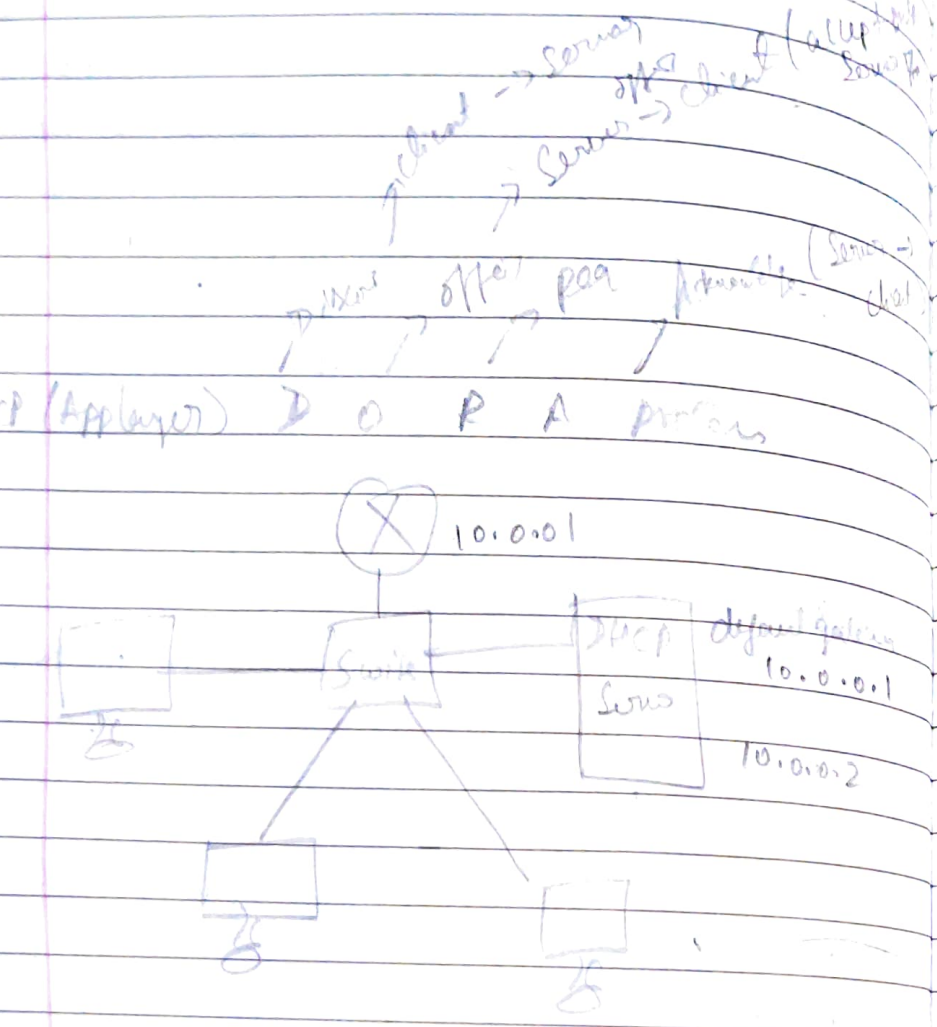
Ping Stastistics for 50.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum = 2 ms, Maximum = 33ms, Average = 9ms

PC3 →

i) ping 40.0.0.3
Pinging 40.0.0.3 with 32 bytes of data:
Reply from 40.0.0.3 : bytes=32 time=33ms TTL=127
Reply " 40.0.0.3 : bytes=32 time=2ms TTL=127
Reply " 40.0.0.3 : bytes=32 time=2ms TTL=127
Reply " 40.0.0.3 : bytes=32 time=2ms TTL=127
ping Stastics for 40.0.0.3
Packets Sent = 4, Received = 4, Lost = 0 (0% Loss)

ii) ping 50.0.0.3
Pinging 50.0.0.3 with 32 bytes of data:
Reply from 50.0.0.3 : bytes=32 time=33ms TTL=125
Reply from " : bytes=32 time=33ms TTL=125
Reply from " : bytes=32 time=33ms TTL=125
Reply from " : bytes=32 time=33ms TTL=125

Observation: Packets are sent to any unknown destination to a single next hop address



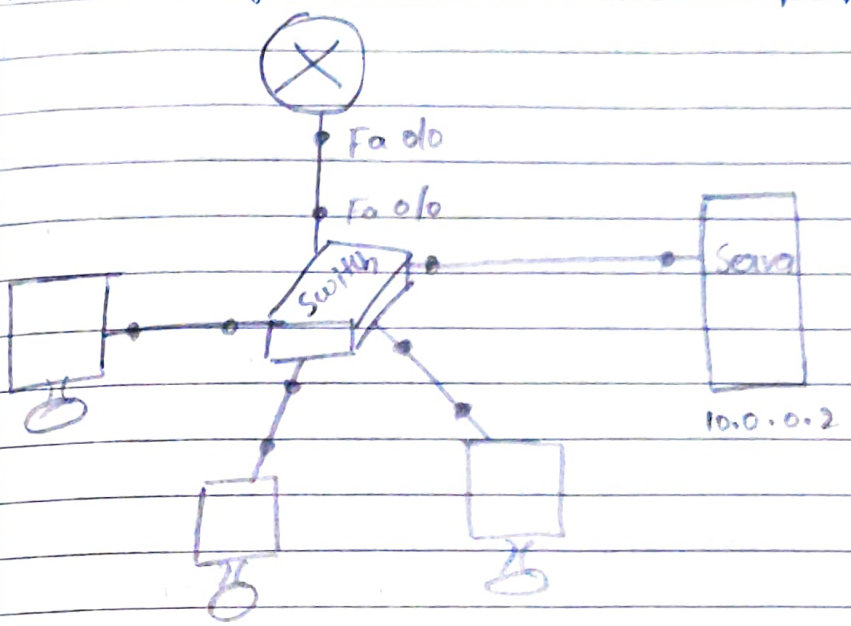
LAB-6

Aim: Configuring DHCP within a LAN in packet Tracer

Date: 8/12/22
Page: 19

Aim: Students will design n/w based on topology of nodes & req given 10.0.0.1 & Simulate the topology.

Topology:



Setting up Components

1) Routers → Router-PT [1 no]
2) Switches → Switch-PT [1 no]
3) End Devices → Generic [3 no's]
4) End Devices → Server-PT [1 no's]

Connections

1) Connections → Automatically Choose Connection Type [for all the devices]

IP addresses & Default Gateway of Server

1) Server → Config → Gateway as 10.0.0.1
   Fast Ethernet0 → IP Address 10.0.0.2
   Subnet Mask 255.0.0.0

2) Router → 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
```

Interface FastEthernet 0/0, changed State

Note: while connecting to multiple ports you might not get, go to Switch-PT → physical → turn off Switch → at bottom add the Switch, drag & drop on any window.

Server → Services → DHCP → Set the

Default Gateway : 10.0.0.1
DNS Server : 10.0.0.2
Start IP Address : 10 0 0 3
Subnet Mask : 255 0 0 0
TFTP : 10.0.0.

and Save it

Turn on the Switch, to turn the links green.

More About DHCP →

Dynamic Host Configuration Protocol is a client/server protocol that automatically provides an IP host with its IP # & other related Configuration information such as the subnet mask and default gateway.

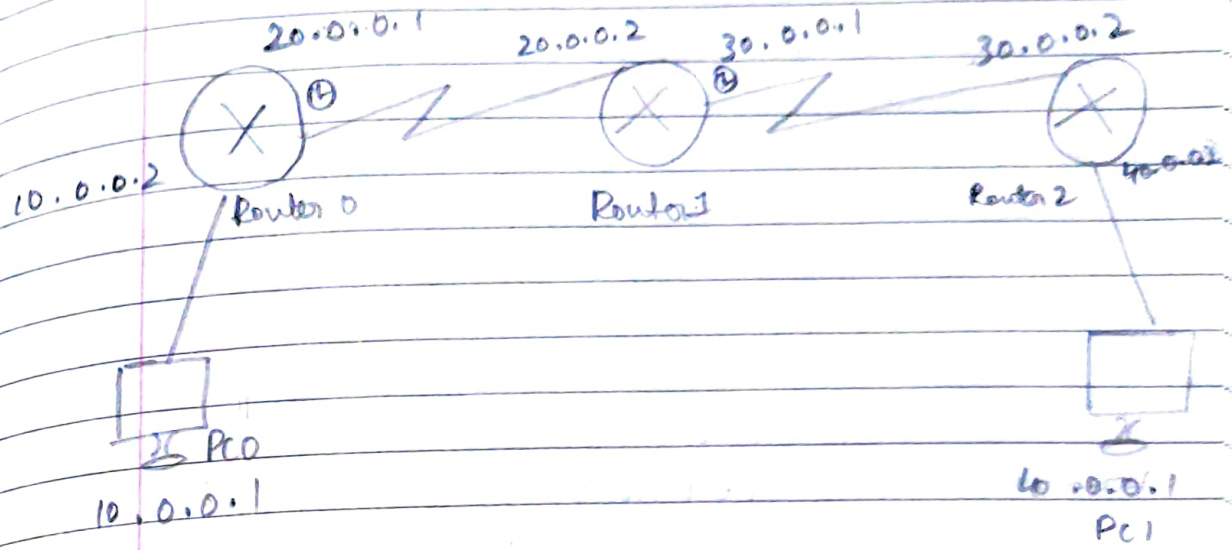
Observation: PC'S get their IP Configuration from DHCP Server. IP assignment is done automatically

LAB - 7

Aim: 6. Configuring RIP Routing Protocol in Routers

Date 15/12/22
Page 21

Aim: Should design a n/w, apply the learnt prwp tocols and justify the usage



Setting up components →

1) Routers → Router-PT [3 no's]
2) End Devices → Generic-PT [2 no's]

Connections →

1) Blw Routers & PC, Select Automatic Connection type.
2) Blw the Routers, Select Serial DCE

Gateways for PC0's →

1) PC0 → Config → Gateway 10.0.0.2
2) PC1 → Config → Gateway 40.0.0.2

IP address for PC's →

1) PC0 → config → FastEthernet0 → IP 10.0.0.1
   Subnet 255.0.0.0
2) PC1 → Config → FastEthernet0 → IP 40.0.0.1
   Subnet 255.0.0.0

Configuring Routers for FastEthernet →
Router0

[Yes/no] : no
> enable
Config terminal
interface fastethernet 0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
Interface FastEthernet 0/0, Changed State to up

[Yes/no] : no
> enable
Config terminal
interface fastethernet 0/0
ip address 40.0.0.2 255.0.0.0
no shutdown
Interface FastEthernet 0/0, Changed State to up

Configuring Router for Serial
NOTE: For Serial Conn with (L) we need Clock rate 64000, no need for other end.
Router0
interface Serial 2/0
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
Clock rate 64000
no shutdown
Interface Serial 2/0, Changed State to up

Router1
[Yes/NO] : no
>enable
Config terminal
interface Serial 2/0
ip address 20.0.0.2 255.0.0.0

encapsulation PPP
no shutdown
Interface Serial 2/0, Changed State to up

Router 1

interface Serial 3/0
ip address 30.0.0.1 255.0.0.0
no shutdown encapsulation PPP
Clock rate 64000
no shutdown
exit

Router 2

interface Serial 3/0
ip address 30.0.0.2 255
encapsulation ppp
no shutdown
exit

Router 0 →

> # enable
Config terminal
router rip
network 10.0.0.0
network 20.0.0.0
exit

Router 1

router rip
network 20.0.0.0
network 30.0.0.0
exit

router rip
network 30.0.0.0
network 40.0.0.0
exit

Ping> PC0>

Ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data
Request timed out.
Reply from 40.0.0.1: bytes=32 time=1ms
Reply from 40.0.0.1: bytes=32 time=2ms TT
Reply from 40.0.0.1: bytes=32 time=21ms TT

Ping Statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (

PC> ping 40.0.0.1
Reply from 40.0.0.1 bytes=32 times=2
Reply from 40.0.0.1 bytes=32 time=1
Reply from 40.0.0.1 bytes=32 time=14
Reply from 40.0.0.1 bytes=32 time=27

Ping Statistics for 40.0.0.1
Packets: Sent = 4, Received = 4, Lost = 0 (0
Approximate round trip times in milli-seconds
Minimum = 14ms, Maximum = 27ms, Averag

PC 1 >

PC> Ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes:
Reply from 10.0.0.1 bytes=32 Time=25
Reply from 10.0.0.1 bytes=32 time=9m
Reply from 10.0.0.1 bytes=32 time=9

Reply from 10.0.0.1. bytes=32 time=2ms TTL=125

Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

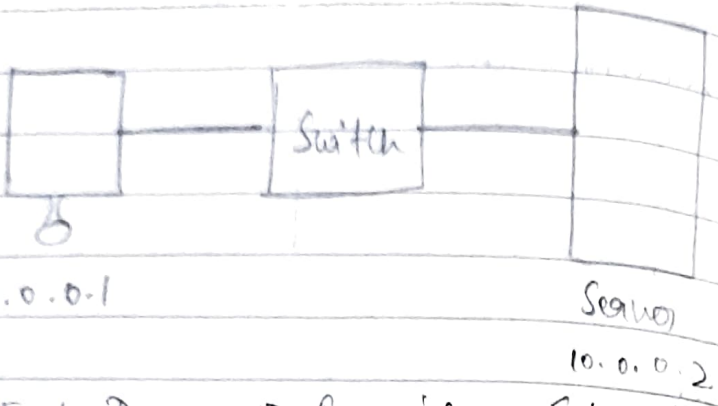
Observation: Ping message is sent from PC0 → PC1, eventhough hop address is not set

Aim

Demonstration of WEB Server and DNS using Packet Tracer

7

Aim: Students should design a network based the topology & Simulate the topology.



1) End Devices → Generic [1 no's]
2) Switches → Generic [1 no]
3) End Devices → Server-PT → Generic [1

IP to PC0 →

1) PC0 → Config → FastEthernet0 →
IP address 10.0.0.1
Subnet Mask 255.0.0.0

IP to Server0 →

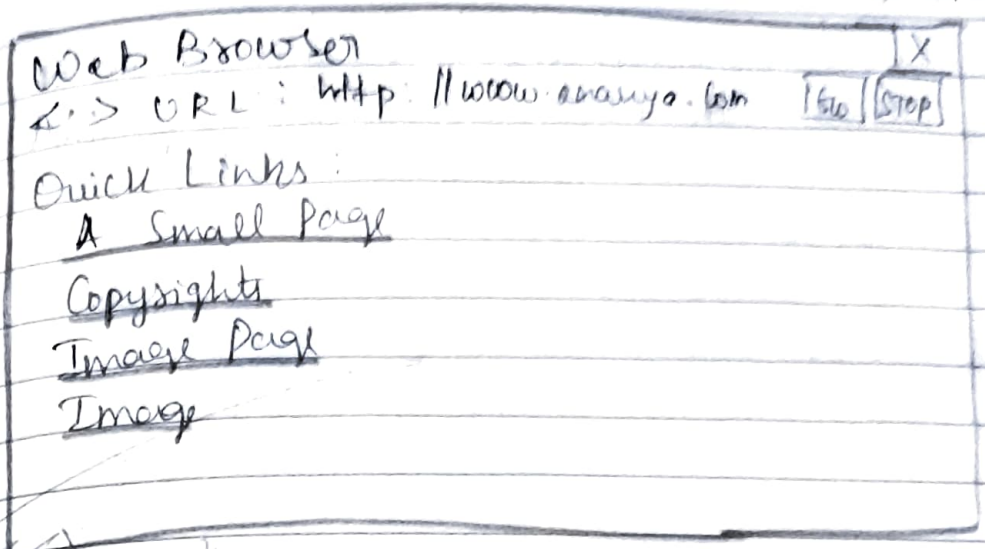
1) Server0 → FastEthernet0 → IP 10.0.0.2
Subnet 255.0.0.0

1) Server0 → Services → HTTP ⊙ on [Should
Check with DNS also,
2) DNS → DNS Service ⊙ on
3) Name: www.ananya.com
Address: 10.0.0.2 [Server IP]
4) Add → Save
5) PC0 → Desktop → Web Browser →
URL www.ananya.com → Go

links will be displayed.

Observation: Links are generated for the created URL.

Web Browser



Date 15/12/2022

Exp 8

Aim: Write a program for error detecting code using CRC-CCITT (16-bit)

Concept: Error Detection : CRC

i) Data Message : 1001101 (16 bit div)

ii) Generator poly x^3+x+5 → 1011 (k) $x^{16}+x^{12}+x^5+1$

append k-1 0's i.e here 4-1 = 3 0's to data msg.

1001101000 +16 0's

```
1011 ) 1001101000 (
       1011
       -----
        0010 1
           0000
           ----
           0101 1
           1011
           ----
           0000 1
              0000
              ----
              0001 0
```

$x^{16}+x^{12}+x^5+1$

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0

+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 of 16 0's

Code:

```
import java.util.*;
public class crc
{
    public static int n;
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        Crc ob = new crc();
        String Code, Copy, rec, Zero = "0000000000000000";
        System.out.print("Enter poly: ");
        Code = in.nextLine();
        System.out.println("Generating poly nominal:
            10001000000100001");
        n = Code.length();
        Copy = Code;
        Code += Zero;
        System.out.println("Modified poly: "+Code);
```

```
code = ob divide (code);
System out. println ("CheckSum : "+code.substring(n).
copy = copy.substring (0, n)+ code. Substring (n);
System. out. println ("Final codeword : " + copy);
System. out print (" Test error detection 0 (yes)
    1 (no)? : ");
Int choice = in. nextInt ();
if (choice == 0)
{
    System. out. print ("Enter position on error. ");
    Int errorPos = in. nextInt ();
    if (copy. charAt (errorPos) == '1')
      copy = copy. substring (0, errorPos)+ "0" +
      copy. substring (errorPos +1);
    else
      copy = copy. substring (0, errorPos) + "1"
      + copy. substring (errorPos +1);
    System. out. println ("Error recons data: " +
        copy);
    S.o.p ("Error detected");
}
else
    S.o.p ("No error detected");
}
000";
public String divide (String s)
{
  Int i, j;
  Char x;
  String div = "10001 0000 00 10001";
  for (i=0; i<n; i++)
  {
    x = s.charAt (i);
    for (j = 0; j < 17; j++)
```

```
        }
        if (x == '1')
        {
            if (S.charAt(i+j) != div.charAt(j))
                S = S.substring(0, i+j) + "1" + S.substring(i+j+1);
            else
                S = S.substring(0, i+j) + "0" + S.substring(i+j+1);
        }
    }
    return S;
```

O/P

```
Enter Poly : 1011101
10000000000000000
Generating polynomial
1000100000010000
Modified Poly : 10111010000000000000000000000000
Test error detection 0(yes) 1(No)
0
Enter the position of error
3
Erroneous data : 10101000000000000000
Error detected
Test error detection 0(yes) 1(No)
23
Enter the position of error
No error detected
```

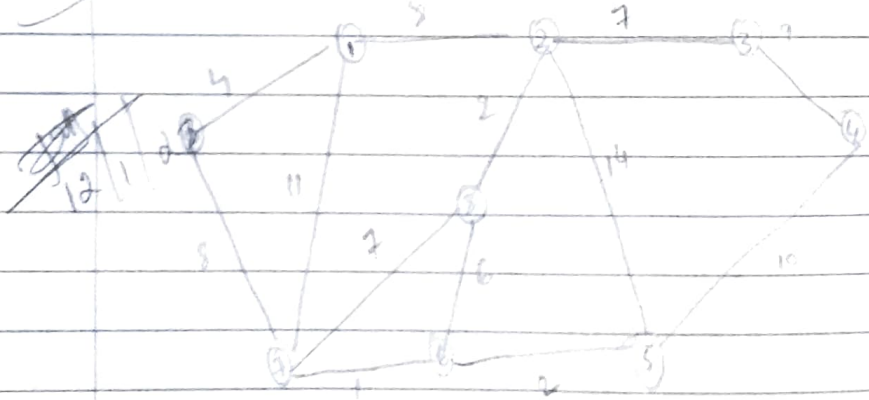
Aim
Implement Dijkstra's algorithm to compute the shortest path for a given topology

```
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>
#define V 9
int minDistance(int dist[], bool sptSet[])
{
  int min = INT_MAX, min_index;
  for (int v = 0; v < V; v++)
    if (sptSet[v] == false && dist[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++)
    printf("%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[v] > dist[u] + graph[u][v]
```

```
int main()
{
  int graph[V][V] = {{0,4,0,0,0,0,8,0},
                     {4,0,8,0,0,0,11,0},
                     {0,8,0,7,0,4,0,2},
                     {0,0,7,0,9,14,0,0},
                     {0,0,0,9,0,10,0,0},
                     {0,0,4,14,10,0,2,0},
                     {0,0,0,0,0,2,0,1,6},
                     {8,11,0,0,0,0,1,0,7},
                     {0,0,2,0,0,0,6,7,0}};
  dijkstra(graph,0);
  return 0;
}
```

O/P

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



Write a program for ... shortest paths for ... routing ...

Algorithm:-

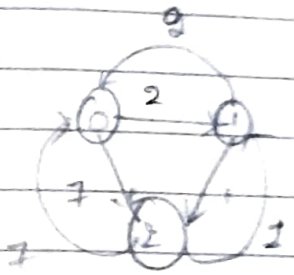
```
Distance vector Routing()
{ // Initialize (create initial vectors for the node)
  D(myself) = 0
  for (y = 1 to N)
  {
    if (y is a neighbor)
      D[y] = C[myself][y]
    else
      D[y] = ∞
  }
  send vector [D[1], D[2] ... D[N]] to all neighbors
  // update (improve the vector with the vector received from
  // a neighbor)
  repeat (forever)
  { wait (till a vector Dw from a neighbor w or any
    change in the link)
    for (y = 1 to N)
    {
      D[y] = min[D[y], (C[myself][w] + Dw[y])]
      // Bellman-Ford equation
    }
    if (any changes in the vector)
      send vector [D[1], D[2] ... D[N]] to all neighbors
  }
```

```
#include<stdio.h>
struct node
{
  unsigned dist[20];
  unsigned from[20];
} rt[10];
int main()
{
  int costmat[20][20];
  int nodes, i, j, k, count = 0;
  printf("\n Enter the number of nodes: ");
  scanf("%d", &nodes);
  printf("\n Enter the cost matrix: \n");
  for(i=0; i<nodes; i++)
  {
    for(j=0; j<nodes; j++)
    {
      scanf("%d", &costmat[i][j]);
      costmat[i][i]=0;
      rt[i].dist[j]=costmat[i][j];
      rt[i].from[j]=j;
    }
  }
  do
  {
    count=0;
    for(i=0; i<nodes; i++)
    for(j=0; j<nodes; j++)
      for(k=0; k<nodes; k++)
        if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
        {
          rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
          rt[i].from[j]=k;
          count++;
```

```
} while (count!=0);
for (i=0; i<nodes; i++)
}
printf("\n\n Router %d\n", i+1);
for(j=0; j<nodes; j++)
{
printf("\t N Node %d via %d Distance %d",
j+1, rt[i].from[j]+1, rt[i].dist[j]);
}
}
printf("\n\n");
getch();
}
```

O/P

```
Enter the number of nodes: 3
Enter the Cost matrix:
0 2 7
2 0 1
7 1 0
For router 1
Node 1 Via 1 Distance 0
Node 2 Via 2 Distance 2
Node 3 Via 2 Distance 3
For router 2
Node 1 Via 1 Distance 2
Node 2 Via Distance 0
Node 3 Via Distance 1
For router 3
Node 1 Via 2 Distance 3
Node 2 Via 2 Distance 1
Node 3 Via 3 Distance 0
```



4 Write a program for Congestion Control using Leaky bucket algorithm.

```
#include <bits/stdc++.h>
#include <unistd.h>
using namespace std;
#define bucketsize 500
void bucketInput(int a, int b)
{
 if (a > bucketSize)
    cout << "\n\t Bucket overflow";
 else
   {
    sleep(5);
    while(a > b)
    {
     cout << "\n\t\t" << b << "bytes outputte
     a -= b;
     sleep(5);
    }
    if (a > 0)
       cout << "\n\t\t Last " << a << " byte
       Sent \t";
    cout << "\n\t\t Bucket output Successfu
   }
}
int main()
{
 int op, pktsize;
 cout << "Enter Output rate : ";
 cin >> op;
 for(int i = 1; i <= 5; i++)
 {
  sleep(rand() % 10);
  pktsize = rand() % 700;
```

```
Size = " << pktSize;
bucketInput (pktSize, op);
}
cout << endl;
return 0;
}
```

O/P

Enter output rate: 50

Packet no 1 Pkt Size = 186
50 bytes outputted
"
"
Last 36 bytes sent
Bucket output successful

Packet no 2 Packet Size = 215
50 bytes outputted
"
"
"
Last 15 bytes sent
Bucket output successful

Packet no 3 Packet Size = 535
Bucket overflow

Packet no 4 Packet Size = 492
50 bytes outputted
"
"
"
"
"
"
"
"
Last 42 bytes sent
Bucket output successful

Packet no 5 Packet Size = 521
Bucket overflow

5. Using TCP/IP Sockets, write a client-server pro to make client sending the file name and the server to send back the contents of req file if present

Date 28/1/25
Page 38

⇒ tcp_ip_server.py

```
import socket
serverName = '127.0.0.1'
serverPort = 12345
#create
server_socket = socket.socket(socket.AF_INET,
 -ckt.SOCK_STREAM)
# bind
Server_socket.listen(5)
while True:
    print("Server waiting for connection")
    client_socket, addr = server_socket.accept()
    print("client_socket.recv(1024).decode()
    file=open(sentence in ")
    l = file.read(1024)
    client_socket.send(l.encode())
    file.close()
    client_socket.close()
```

⇒ tcp_ip_client.py

```
Import socket
serverName = '127.0.0.1'
serverPort = 12345
client_socket = socket.socket(socket.AF_INET,
        socket.SOCK_STREAM)
client_socket.connect((serverName, serverPort))
sentence = input("Enter file name: ")
```

client_socket.close()

⇒ tcp-ip-example.txt

This is eg of TCP/IP pgm

Server & Client windows : OUTPUT

```
C:\Users\Ananya>d:
D:\>cnlab
D:\cnlab>python tcp-ip-client.py
Enter the filename: tcp-ip-example.txt
From Server: This is eg of TCP/IP pgm
```

```
D:\cnlab>python tcp-ip-server.py
Server waiting for connection
Client connected from ('127.0.0.1', 56450)
Server waiting for connection.
```

Observation: Using TCP/IP sockets, communication between the client-server established successfully.

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

⇒ udp-Server.py

```
Import socket
ServerName = '127.0.0.1'
ServerPort = 12345
#Create a datagram socket
UDPServerSocket = socket.socket(socket.AF_INET,
Socket.SOCK_DGRAM)
#Bind to address and ip
UDPServerSocket.bind((ServerName, ServerPort))
print("UDP Server up and listening")
# Listen for incoming datagrams
while (True):
    Sentence, ClientAddress = UDPServerSocket.re
    -cvfrom(20148)
    file = open(Sentence, "r")
    l = file.read(20148)
    UDPServerSocket.sendto(bytes(1, "utf-8"),
    ClientAddress)
    print("Sent back to client:", 1)
file.close()
```

⇒ upd-Client.py

```
import socket
ServerName = '127.0.0.1'
ServerPort = 12345
# Create a UDP socket at client side
```

```
UDPClientSocket = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
# Send to server using created UDP socket
sentence = input("Enter file name:")
UDPClientSocket.sendto(bytes(sentence,"utf-8"),
    (serverName, serverPort))
fileContents, serverAddress = UDPClientSocket.recei
-vefrom(2048)
print('From Server:',fileContents)
UDPClientSocket.close()
```

=> upd-example.txt

This is an eg for UDP program

Output Server.py

```
D:\cn\labs>udp-server.py
UDP Server up and listening
Sent back to Client: This is an example for
        UDP pgm.
```

client.py

```
D:\cn\labs>python udp-client.py
Enter file name: udp-example.txt
From Server: b 'This is an example file for
  UDP program.'
```

Observation: UDP Server-Client connection established successfully.