lab-1

cisco packet tracer

It is an integrated simulation, visualisation, collaboration and assesment environment and it is built by a company named cisco.

* It includes two types of workspace
① logical : It shows the logical network built by user. It contains the device of generic, real and modular and it is a multi user remote network.
② Physical : It shows the physical dimension of the logical network. It contains the device and wiring closet building, city and integrity views and it is a wireless association management.

Hub	Switch	Router
① hub belongs to layer 1 of an OSI model mean a physical layer	① Switch belongs to layer 2 of an OSI model that means it is a data link layer device	① Router belongs to layer 3 of an OSI model that mean it is a network layer
② Hub prefers half duplex transmission	② switch prefer full duplex transmission	② It is full duplex in nature.
③ It is a functional based broadcasting	③ It functions based on Mac address	③ It function based on IP address.
④ Hub is commonly used to link component of a LAN	④ A switch used by LAN	④ A router is utilised by LAN as well as MAN.

Router

IN → CLI

lab-2

Aim: configuring IP address to Router in packet tracer explore following message ping responses Destination unreachable request time out.

Router > enable

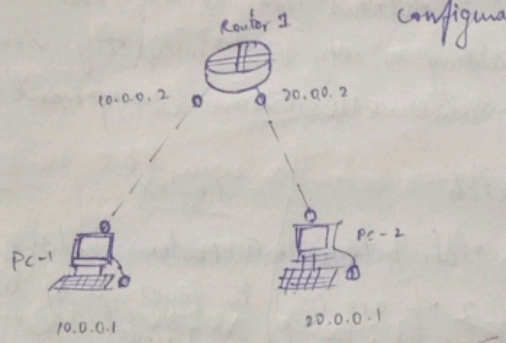
Router# config t

Router(config)# interface FastEthernet 0/0

Router(config-if)# ip address 10.0.0.2 255.0.0.0

Router(config-if)# no shutdown

configuration of Router



PC-1

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 = 1ms TTL = 255

Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 255

Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 255

Reply from 10.0.0.2: byte = 32 time = 0ms TTL = 255

ping statistics for 10.0.0.2:

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

Approximate round trip times in milli-seconds:

minimum = 0ms, maximum = 1ms, Average = 50ms

> Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

reply from 20.0.0.1 : bytes=32 time=0m TTL=127

replay from 20.0.0.1 : bytes=32 time=0m TTL=127

reply from 20.0.0.1 bytes : 32 time=0ms TTL=127

replay from 20.0.0.1 bytes : 32 time=0ms TTL=127

Ping statistics for 20.0.0.1

Packets set = 4 Received = 4 Lost = 0 (0% loss)

Approximate round trip times in ms

Min = 0ms , Max = 0ms , Avg = 0ms

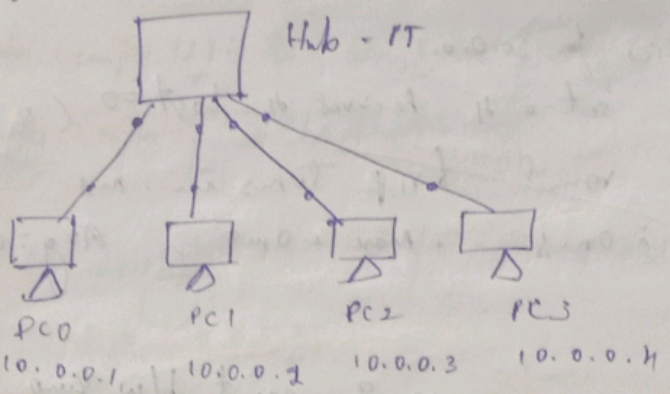
Observation:

- when a simple PDU is sent b/w two devices request time out occurs because the gate ways aren't configured i.e the two networks are isolated
- when the gate way are configured and a simple PDU set is received successfully at the network.

Experiment No: Lab-3

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

Diagram:



> Ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2 bytes = 32 time = 1ms TTL = 128
Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 128

Ping Statistics 10.0.0.2

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

min = 0ms / max = 1ms . Avg = 0ms .

PC> ping 10.0.0.3

Pinging 10.0.0.3 with bytes=32 of data.

Reply from 10.0.0.3. bytes=32 time=0ms TTL=128.

Reply fm 10.0.0.3 bytes=32 time=0ms TTL=128

Reply from 10.0.0.3 byte=32 time=0m TTC=128

Reply fm 10.0.0.3 byte=32 time=0ms TTL=128

Ping statistics 10.0.0.3

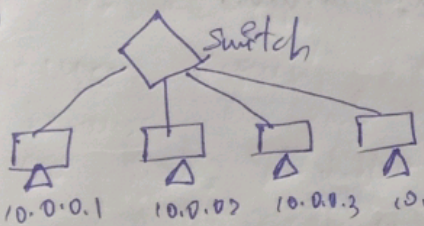
packet: Sent=4, received=4 lost=0

(0% loss)

Approx round trip in ms:

Min=0ms Max=0ms Avg=0ms.

2)



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=1ms TTL=128

reply fm 10.0.0.2 bytes=32 time=1m TTL=128

reply fm 10.0.0.2 bytes=32 time 1m TTL=128

reply fm 10.0.0.2 bytes=32 time=1ms TTL=128

Ping statistics 10.0.0.2

packet Sent=4 Received=4 Lost=0

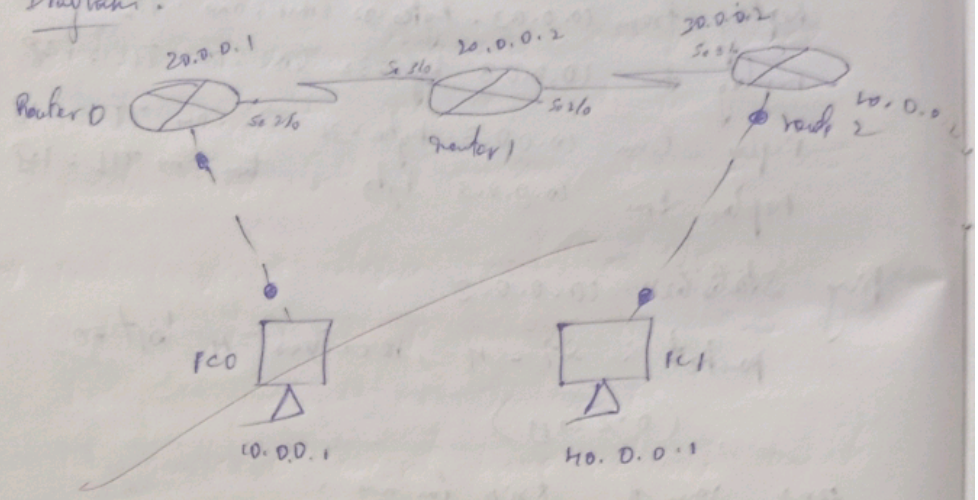
(0% Loss)

min =0 ms Max =1ms Avg =0ms

Lab 4

Configuring default routes to the routers

Diagram:



given IP address to fc0 as 10.0.0.1

given IP address to fc1 as 40.0.0.1

> Click router 0, CLI

Router > enable

Router # configure terminal

Router (Config) # interface Fastethernet 0/0

Router (Config-if) # ip address 10.0.0.2 255.0.0.0

Router (Config-if) # no shutdown

Router (Config-if) #

% Link 5 - changed : Interface fastEthernet 0/0

Router (Config-if) # exit

Router

* IP address serial 2/0 20.0.0.1

Router (config)# interface serial 2/0

Router (config-if)# ip address 20.0.0.1 255.0.0.0

Router (config-if)# No shutdown

%link 5 changed : Interface Serial 2/0

Router 1 Serial 3/0 20.0.0.2

Router> enable

Router # Configure terminal

Router (config)# interface serial 3/0

Router (config-if)# IP address 20.0.0.2 255.0.0.0

Router (config-if)# exit

Router 1 serial 2/0 30.0.0.1

Router (config)# interface serial 2/0

Router (config-if)# ip address 30.0.0.1 255.0.0.0

Router (config-if)# no shutdown.

%link-5 changed : Interface serial 2/0

changed state to down.

* Router >

Router > enable

Router # Configure terminal

enter Configuration commands one per line,

Router (Config)#. interface serial 3/0

Router (Config-if)# ip address 30.0.0.2 255.0.0.0

* Router (Config)# interface FastEthernet 1/0

Router Config-if) # ip address 40.0.0.2 255.0.0.0

Router Config-if) # no shutdown

⇒ The router 0 doesn't have network with 30.0.0 and 40.0.0.0

To see connection of router

CLI > Router > Show ip route

PC > ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data

reply from 10.0.0.2 destination host unreachable

reply from 10.0.0.2 destination host unreable

Request time out

Reply from 10.0.0.2 destination host unreachable.

Packets Sent = 4 received = 0 lost = 4

(100% loss)

To the connection

Router > show ip route

C 10.0.0.0/8 is directly connected
Fastethernet 0/0

C 20.0.0.18 is directly connected
serial 2/0

=> Router # config t

Router (config) # ip route 20.0.0.0 255.0.0.0 20.0.0.2

Router (config) # ip route 40.0.0.0 255.0.0.0 20.0.0.2

exit

Router # show ip route

C 10.0.0.0 Fastethernet 0/0

C 20.0.0.0 Serial 2/0

S 30.0.0.0 Via 20.0.0.2

S 40.0.0.0 via 20.0.0.2

Route 1

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
30.0.0.1

Router # show ip route

S 10.0.0.0/8 Via 20.0.0.1

C 20.0.0.0 Serial 2/0

C 30.0.0.0 Serial 2/0

S 40.0.0.0 Via 30.0.0.2

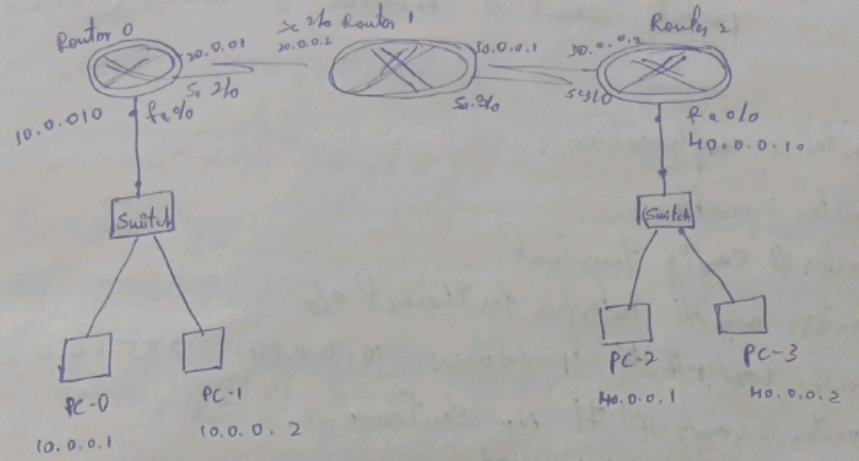
observations:

Observation :

Since the static route was not set destination host was unreachable as we have sent static route now we get reply.

Default Routing

Lab - 5



1) Select 4 end devices and 2 switches
2) connect 2 end devices to each switch
3) Add 3 routers to workspace and connect it as shown in topology
4) connect two two switches to two routers
5) Configure ip address 1st two end devices as 10.0.0.1 and 10.0.0.2 and next two devices as 40.0.0.1 and 40.0.0.2
6) Send packets without routers

Observation

for PC0

PC > ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data

Request time out
Request time out
Request time out
Request time out

Ping statistics for 10.0.0.1
packets: sent=4 Received=50, lost=0 (100% loss)

Router 0 configuration:

Router > enable
Router # Config terminal
Router(config)# Interface fastethernet 0/0
Router(config-if)# ip address 10.0.0.10 255.0.0.0
Router(config-if)# No shutdown
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 sol shutdown.
Router(config-if)# exit

Router > show ip route
C 20.0.0.0/8 is directly connected, serial 2/0
C 10.0.0.0/8 is directly connected, Fastethernet 0/0

Router > config t
Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router(config)# exit

Router 2 configuration

Router > enable
Router # config t
Router(config) # interface serial 2/0
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 fastethernet 0/0
Router(config-if) # ip address 40.0.0.10 255.0.0.0
Router(config-if)# no shutdown.
Router(config-if) # exit

Router # show ip route
C 30.0.0.0/8 is directly connected serial 2/0
C 40.0.0.0/8 is directly connected, fastethernet 0/0

Router # config terminal
Router(config) # ip route 0.0.0.0 0.0.0.0 30.0.0.

Command prompt : for Router2 Observation:

PC : ping 10.0.0.1

pinging with 32 bytes of data

Reply from 10.0.0.1 bytes=32 time=15ms TTL=[illegible]
Reply from 10.0.0.1 bytes=32 time=4ms TTL=[illegible]
Reply from 10.0.0.1 bytes=32 time=11ms TTL=[illegible]
Reply from 10.0.0.1 bytes=32 time 6 ms TTL=[illegible]

Ping statistics for 10.0.0.1

packet Sent =4 Received =4, Lost=0

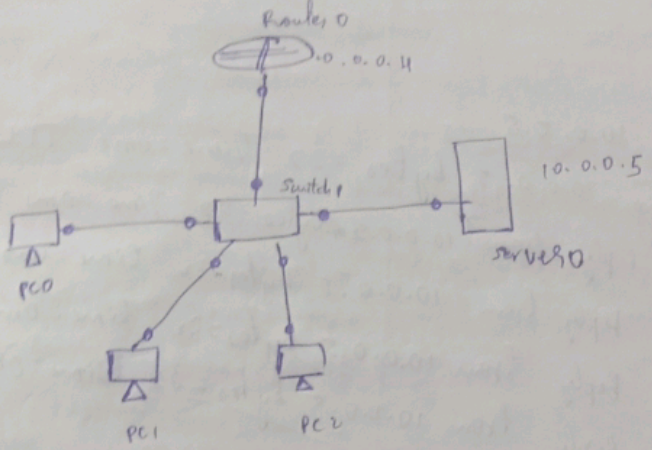
(0% loss)

Approximate round trip times in milliseconds

minimum = 4ms Maximum=15ms Avg =4ms

lab-6

AIM: Configuring DHCP within a LAN in packet tracer



Configure routers

Router > enable

Router # config t

Router (config) # interface fastethernet 0/0

Router (config-if) # ip address 10.0.0.4 255.0.0.0

Router (config-if) # no shutdown

Configure server

servies → DHCP → on service

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 server 10.0.0.2

Observation: After performing these steps, it will automatically allocates IP address to all end devices

Result:

> ping 10.0.0.5

pinging 10.0.0.5 bytes = 32 time = 0ms TTL = 128

Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128

Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128

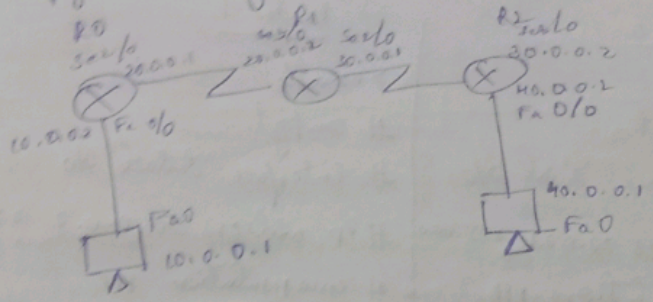
Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128

Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128

ping statistics for 10.0.0.5

packets: Sent = 4, Received = 4 lost = 0 (0% loss)

Aim: Configure Routing Information Protocol



Procedure

- set up network using 3 router and 2 end devices by connecting the router to router via serial DCE cable and router to end devices using copper cross-cable.
→ configure ip address of end devices
→ configure ip address of fastethernet interface of router

Command (Router)

>enable
config t
interface FastEthernet 0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
exit

Router 2

>enable
config t
interface FastEthernet 0/0
ip address 40.0.0.2 255.0.0.0
no shutdown
exit

→ Config ip address of router

Router 0

```
enable
# config t
# Interface serial 2/0
# ip address 20.0.0.1 255.0.0.0
# encapsulation ppp
# clock rate 64000
# no shutdown
# exit
```

Router 1

```
> enable
# config t
# Interface serial 3/0
# ip address 20.0.0.2 255.0.0.0
# encapsulation ppp
# no shutdown
# exit
# Interface serial 2/0
# ip address 30.0.0.1 255.0.0.0
# encapsulation ppp
# clock rate 64000
# no shutdown
# exit
```

Router 2

```
> enable
# config t
# Interface serial 3/0
# ip address 30.0.0.2 255.0.0.0
# no shutdown
# exit
```

→ connect the networks

Router 0

```
> enable
# Config t
# router rip
network 10.0.0.0
network 20.0.0.0
exit
```

Router 1

```
> enable
# Config t
# router rip
# network 20.0.0.0
# network 30.0.0.0
# exit
```

Router 2

```
> enable
# config t
# router rip
# network 30.0.0.0
# network 10.0.0.0
# exit
```

Observation: RIP is a dynamic routing protocol that uses hop count as a routing matrix to find the best path between the source and destination.

PC > ping 10.0.0.1

pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1 bytes=32 time=23ms TTL=125
Reply from 10.0.0.1 bytes=32 Time=11ms TTL=125
Reply from 10.0.0.1 bytes=32 Time=2ms TTL=125
Reply from 10.0.0.1 bytes=32 Time=15ms TTL=125

ping statistics -

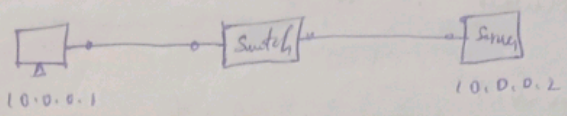
packets: Sent=4, Received=4, Lost=0 (0% loss)

Approx round trip time

min=2ms max=23ms Avg=14ms

Aim : Demonstrate of WEE server and DNS 15/12/22

Topology :



Procedure :

→ Set up connection between end device, switch and server using genral connections

→ open server → services → enable http and dNS

→ add a domain name [eg. www.abc.com]

address = [server ip] [10.0.0.2]

→ add and save

→ go to web browser in end device and seach for the added domain name

observation.

DNS domain name mapped to IP address which is easy to remember, through which we can access web content of server.

Web page :

Cisco packet Tracer

Quick links :

A small page

Copy writes

Image

Aim: Error detection using CRC (16 bit)

Program:

```
import java.util.*;
class Main{
  int n;
  public void main (String[] args)
{
  Scanner in = new Scanner(System.in);
      Main ob = new Main();
String code, copy, rec, zero = "0000000000000000";
System.out.println("Enter message");
code = in.nextLine();
n = code.length();
  copy = code;
  code += zero;
  code = ob.divide(code);
  System.out.println("Message =" + copy);
  copy = copy.substring(0, n) + code.substring(n);
  System.out.println("CRC=");
  System.out.println(code.substring(n));
  System.out.println("transmitted frame is" + copy);
  System.out.println("Enter received data");
  rec = in.nextLine();
  if (zero.equals(ob.divide(rec).substring(n)))
    System.out.println("Correct bits received");
  else
    System.out.println("Received frame contain error");
  in.close();
}
```

```
public String divide(String s)
{ int i,j;
  char k;
  String div = "10001000000100001";
  for(i=0; i<n; i++)
  { k = s.charAt(i);
    for(j=0; j<17; j++)
    {
      if(k == '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;
  }
}
```

Output:

Enter message

10001000100011

Message = 10001000100011

CRC = 00010010000010000

Transmitted frame is 10001000100011000100100000010000

Enter received data

10001000100011000100100000010000

Correct bits recieved

lab 8 :

Aim : Implement Dijkstras algorithm to compute the shortest path for a given topology.

```
#include <iostream>
#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++)
    if ( sptSet[v] == false && dist[v] <= min)
      min = dist[v], min_index = v;
  return min_index;
}
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
                < dist[v])

            dist[v] = dist[u] + graph[u][v];
}

printSolution(dist);
}

int main(){

}

void printSolution(int dist[])
{
    cout << "Vertex \t\t Distance from Source\n";
    for (int i=0; i<V; i++)
        cout << i << "\t\t" << dist[i];
}
```

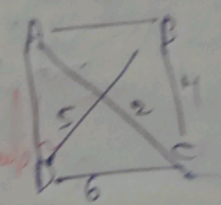
```
int main()
{
    int graph[V][V]
    cout << "Enter the graph" << endl;

    for (int i=0 ; i<V ; i++)
    {
        for (int j=0 ; j<V ; j++)
            cin >> graph[i][j];
    }
    dijkstra(graph, 0);
    return 0;
}
```

Output :

Enter the graph

0	1	2	3
1	0	4	5
2	4	0	6
3	5	6	0



Vertex	Distance from source
0	0
1	1
2	2
3	3

Lab 9 Distance vector algorithm to find suitable path for transmission.

```
Distance_vector_Routing (){
    D[myself] = 0
    for y = 1 to n {
        if y is neighbour
            D[y] = c[myself][y]
        else D[y] = ∞ }
    Send vector { D[1], D[2] ... D[N] } to all neighbours
    repeat (forever) {
        wait vector Dw for a neighbour w or any
            change in a link
        for y = 1 to n {
            D[y] = min { D[y], (c[myself][w] + Dw[y]) }
        if any change in vector {
            send vector { D[1], D[2] ... D[N] } to
                all neighbours
        }
    }
}
```

O/P

Enter the no. of nodes required (less than 10 pls): 4

Enter adjacency matrix:

0	3	2	99
3	0	1	4
2	1	0	99
99	4	99	0

routing table for node 1 is :

1	2	3	4
1	2	2	7

routing table for node 2 is

1	2	3	4
3	0	1	4

routing table for node 3 is

1	2	3	4
2	1	0	5

routing table for node 4 is

1	2	3	4
7	4	5	0

Enter the nodes btw which shortest path is to found

A D

Shortest path is 7

Lab 10

Aim: Leaky Bucket problem algorithm for congestion control

```
#include <bits/stdc++.h>
using namespace std;
int bucket size = 500;
void delay (int dela){
int now = time (NULL);
int later = now + dela;
while (now <= later);
    now = time(NULL);
}
void bucketInput (int a, int b){
if (a > bucketsize){
  cout << "In |+|+| Bucket overflow";
}
else{
  delay (1);
  while(a > b){
  cout << "In |+|+|+" << b << "bytes outputted";
    a -= b;
    delay (1);
}
if (a > 0){
  cout << "In |+|+| Last" << a << "bytes sent";
cout << "In |+|+| Bucket output successful";
}
}
```

Output

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

Socket Programming : using TCP sockets, write client server program make client send file name & server send back the contents of requested file if present.

[client.py]

```
From socket import *
serverName = "DESKTOP-HMPOOFC"
serverport = 12550
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket = connect((serverName, serverPort))
Sentance = input("Enter Filename");
client.socket.send(sentence.encode())
file contents = clientsocket.recv(1024).decode()
print('From Server:', file content)
client socket.close().
```

```
From socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INFP, SOCK_DGRAM)
Sentance = input("Enter file name")
print("From server:", file contents)
client Socket.close().
```

Output

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

Socket Programming : using TCP sockets, write client server program make client send file name & server send back the contents of requested file if present.

[client.py]

```
From socket import *
serverName = "DESKTOP-HMPOOFC"
serverport = 12550
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket = connect((serverName, serverPort))
Sentance = input("Enter Filename");
client.socket.send(sentence.encode())
file contents = clientsocket.recv(1024).decode()
print('From Server:', file content)
client socket.close().
```

```
From socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INFP, SOCK_DGRAM)
Sentance = input("Enter file name")
print("From server:", file contents)
client Socket.close().
```

[servudp.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)
serverSocket.sendto(bytes(l,"utf-8"), clientAddress)
print("sent back to client", l)
file.close()
```

```
from socket import *
serverName = "DESKTOP-HMPODEC"
serverport = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(("", serverport))
serverSocket.listen(1)
print("The server is ready to receive")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(2048).decode()
    file = open(sentence, "r")
    l = file.read(2048)
    connectionSocket.send(l.encode())
file.close()
connectionSocket.close()
```

Output:

Server started
waiting for a client
connected
Server started
Waiting for a client
Client Accepted.

1/3/2023