lab 10

Aim:
leaky Bucket problem algorithm for Congestion control

```
# include <bits/stdc++.h>
 using namespace std;
 int bucket size = 500;
void delay (int delay){
int now = time (NULL);
int later = now + delay;
while (now <= later);
        now = time(NULL);
}
void bucketInput (int a, int b){
 if (a > bucketSize){
    cout << "\n\t\t Bucket overflow";
  }
 else {
     delay(1);
     while(a > b){
     cout << "\n\t\t" << b << "bytes outputted";
       a -= b;
      delay(1);
  }
 if (a > 0){
   cout << "\n\t\t last" << a << "bytes sent";
cout << "\n\t\t 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 sending file name & server send back the contents of requested file if present.

[clitcp.py]

```
From socket import *
servername = " DESKTOP_HMPOOFC "
serverport = 12550
clientSocket = socket (AF_INET, SOCK_STREAM)
client Socket = 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_INET, SOCK_DGRAM)
Sentance = Input ("Enter file name")
printf ("From server:', file contents)
    client Socket. close ().
```