CN - LAB

VIVEK RAJEEV

PROGRAM 8 : LEAKY BUCKET ALGORITHM

CODE

```
import os
clear = lambda : os . system('clear')

class Client :
  ~~def __in~~
  def _init_(self, rate=int ,data=[]):
      self.rate =rate
      self. data =data

  def _str_(self):
      return str ([str(self.rate),
                   str (self . data)])

class Buffer :

  def _init_(self, buffer_size=int, buffer=[]
      self. buffer-size = buffer_size
      self. buffer= buffer

  def checkstate (self) :
      if len (self. buffer)==0:
          return True

  def _str_(self):
      return str([str (self. buffer_size),
                  str(self. buffer)]);

basestate = True
sec=1
```

```
buffer = Buffer(int(input("enter buffer size")))
client = Client(int(input("enter client acceptance
                          rate in bps")))
data-to-send = str

while basestate :
    data_to_send = input("enter a string ")
    count=0
    if buffer.checkstate():
        for i in range(0, len(data_to_send)):
~~else~~    if i < client.rate :
                Client.data.append(data_to_send[i])
            else:
                if count< buffer.buffer_size:
                    buffer.buffer.append(~~data_to_send~~
                                        (data_to_send[i])
                    count = len(buffer.buffer)
                else:
                    print("Data loss "+data_to_send[i])
    else :
        j=0
        for i in range(0,len(data_to_send)+
                          len(buffer.buffer)):
            if i< client.rate:
                if len(buffer.buffer):
                    client.data.append(buffer.buffer[0])
                    del buffer.buffer[0]
                else:
                    client.data.append(data_to_send[j])
                    j+=1
            else:
                ~~if len(buffer.buffer)<buffer.buffer~~
```

```
if (len(buffer.buffer) <= buffer.buffer_size)
    if j< len(data_to_send):
        buffer.buffer.append(data_to_send[j])
        j+ = 1
else
    if j < len(data_to_send):
        print("Data loss"+ data_to_send[j])
        j+ = 1;
```