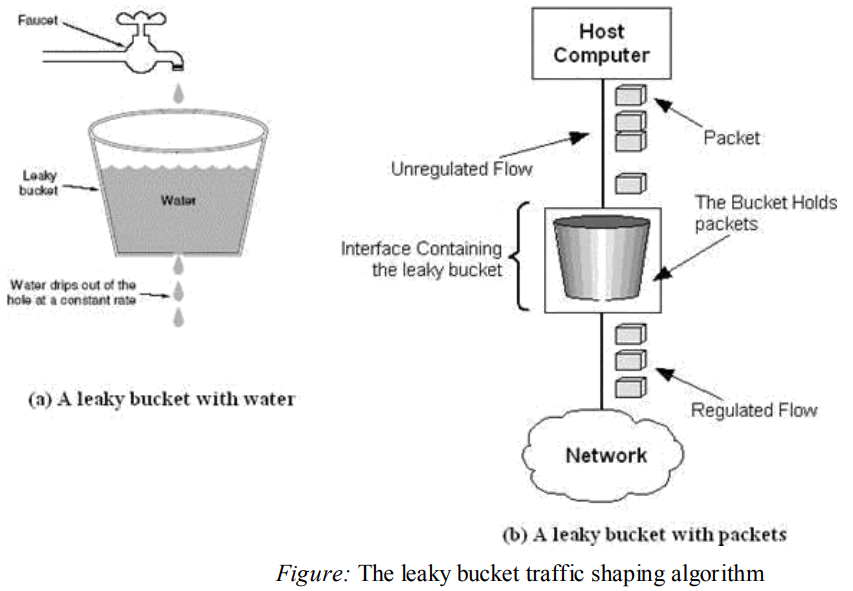
**Lab Program 12:**

**Write a program for congestion control using a leaky bucket algorithm.**



**Code:**

import java.util.Scanner;

public class bucket {

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

int bucket=0;

int op\_rate,i,n,bsize;

System.out.println("Enter the number of packets");

n=sc.nextInt();

System.out.println("Enter the output rate of the bucket");

op\_rate=sc.nextInt();

System.out.println("Enter the bucket size");

bsize=sc.nextInt();

System.out.println("Enter the arriving packets(size)");

int pkt[]=new int[n];

for(i=0;i<n;i++)

{

pkt[i]=sc.nextInt();

}

System.out.println("\nSec\tpsize\tBucket\tAccept/Reject\tpkt\_send");

System.out.println("----------------------------------------------------");

for(i=0;i<n;i++)

{

System.out.print(i+1+"\t"+pkt[i]+"\t");

if(bucket+pkt[i]<=bsize)

{

bucket+=pkt[i];

System.out.print(bucket+"\tAccept\t\t"+min(bucket,op\_rate)+"\n" +"");

bucket=sub(bucket,op\_rate);

}

else

{

int reject=(bucket+pkt[i]-bsize);

bucket=bsize;

System.out.print(bucket+"\tReject "+reject+"\t"+min(bucket,op\_rate)+"\n");

bucket=sub(bucket,op\_rate);

}

}

while(bucket!=0)

{

System.out.print((++i)+"\t0\t"+bucket+"\tAccept\t\t"+min(bucket,op\_rate)+"\t");

bucket=sub(bucket,op\_rate);

}

}

static int min(int a,int b)

{

return ((a<b)?a:b);

}

static int sub(int a,int b)

{

return (a-b)>0?(a-b):0;

}

}

**Output:**

[root@localhost 21cs52]# javac bucket.java

[root@localhost 21cs52]# java bucket

Enter the number of packets

4

Enter the output rate of the bucket

7

Enter the bucket size

8

Enter the arriving packets(size)

6 8 9 5

Sec psize Bucket Accept/Reject pkt\_send

----------------------------------------------------

1 6 6 Accept 6

2 8 8 Accept 7

3 9 8 Reject 2 7

4 5 6 Accept 6

**Output2:**

[root@localhost 21cs52]# javac bucket.java

[root@localhost 21cs52]# java bucket

Enter the number of packets

4

Enter the output rate of the bucket

6

Enter the bucket size

8

Enter the arriving packets(size)

4 5 6 10

Sec psize Bucket Accept/Reject pkt\_send

----------------------------------------------------

1 4 4 Accept 4

2 5 5 Accept 5

3 6 6 Accept 6

4 10 8 Reject 2 6

5 0 2 Accept 2

This code prompts the user to input the number of packets, the output rate of the bucket, and the bucket size. Then, it simulates the behavior of the leaky bucket algorithm by processing each packet one by one and determining whether to accept or reject it based on the current state of the bucket.