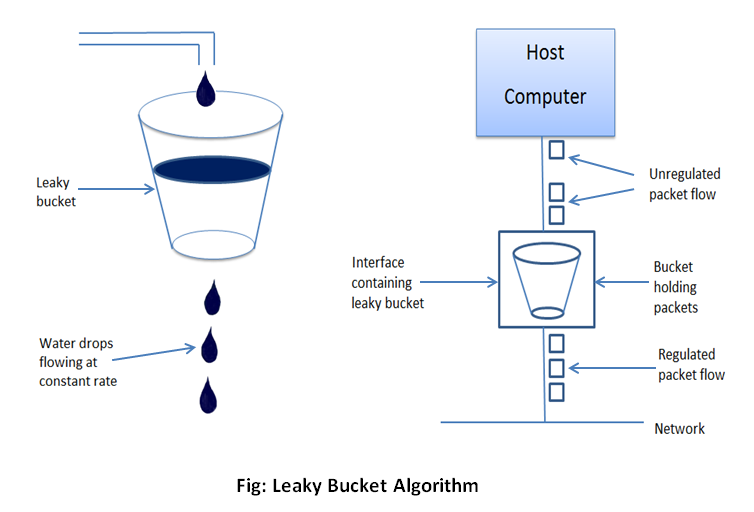
**DCN Lab Program**

Write a program for congestion control using Leaky Bucket Algorithm.



What is Leaky Bucket

* There is a bucket with a hole at the bottom.
* Water will be pouring into the bucket from the top.
* Water flows out at a constant rate irrespective of input of water into the bucket.

Lets apply the same concept to our network bucket.

* + Tap 🡪 Host Computer
  + Water flow 🡪 unregulated packet flow.
  + Bucket 🡪 Interface

Result:

Packet Size beyond capacity 🡪 Dropped

Packet Size within the capacity 🡪Accept

Example:

Initially the bucket is empty : Remaining = 0

Bucket Capacity = 4

Rate = 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i | A[i] | Accept by Bucket | Sent | Remaining |
| 1 | 2 | 2 | 2 | 0 |
| 2 | 4 | 4 | 3 | 1 |
| 3 | 1 | 1 | 2 | 0 |
| 4 | 5 | Dropped | 0 | 0 |
| 5 | 3 | 3 | 3 | 0 |

Code:

import java.util.Scanner;

import java.lang.\*;

public class LeakyBucket

{ public static void main(String[] args)

{ int i;

int a[]=new int[20];

int buck\_rem=0,buck\_cap=4,rate=3,sent,recv;

Scanner in = new Scanner(System.in);

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

int n = in.nextInt();

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

for(i=1;i<=n;i++)

a[i]= in.nextInt();

System.out.println("Clock \t packet size \t accept \t sent \t remaining");

for(i=1;i<=n;i++)

{ if(a[i]!=0)

{ if(buck\_rem+a[i]>buck\_cap)

recv=-1;

else

{ recv=a[i]; buck\_rem+=a[i];

}

}

else recv=0;

if(buck\_rem!=0)

{ if(buck\_rem<rate)

{ sent=buck\_rem; buck\_rem=0;

}

else

{ sent=rate;

buck\_rem=buck\_rem-rate;

}

}

else

sent=0;

if(recv==-1)

System.out.println(+i+ "\t\t" +a[i]+ "\t dropped \t" + sent +"\t" +buck\_rem); else System.out.println(+i+ "\t\t" +a[i] +"\t\t" +recv +"\t" +sent + "\t" +buck\_rem);

}

}

}