Genba Sopanrao Moze College of Engineering, Balewadi, Pune-45

Department of Electronics And Telecommunication

Academic year 2022_23

Roll No:	**Subject: Control System Lab**
Date :	**Staff Sign:**

Experiment no:-5

AIM: Write a program to simulate leaky bucket/token bucket

REQUIREMENT:

WINDOWS 10 with LAN Connectivity.

Software : C/C++ compiler

Theory:

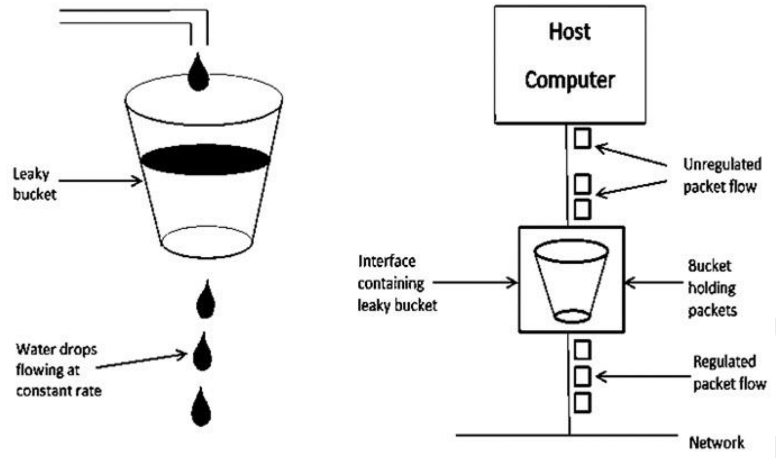
When too many packets are present in the network it causes packet delay and loss of packet which degrades the performance of the system. This situation is called congestion.

The network layer and transport layer share the responsibility for handling congestions. One of the most effective ways to control congestion is trying to reduce the load that transport layer is placing on the network. To maintain this, the network and transport layers have to work together.

With too much traffic, performance drops sharply.

There are two types of Congestion control algorithms, which are as follows −

- Leaky Bucket Algorithm
- Token Bucket Algorithm



Leaky Bucket Algorithm mainly controls the total amount and the rate of the traffic sent to the network.

Step 1 − Let us imagine a bucket with a small hole at the bottom where the rate at which water is poured into the bucket is not constant and can vary but it leaks from the bucket at a constant rate.

Step 2 − So (up to water is present in the bucket), the rate at which the water leaks does not depend on the rate at which the water is input to the bucket.

Step 3 − If the bucket is full, additional water that enters into the bucket that spills over the sides and is lost.

Step 4 − Thus the same concept applied to packets in the network. Consider that data is coming from the source at variable speeds. Suppose that a source sends data at 10 Mbps for 4 seconds. Then there is no data for 3 seconds. The source again transmits data at a rate of 8 Mbps for 2 seconds. Thus, in a time span of 8 seconds, 68 Mb data has been transmitted.

That’s why if a leaky bucket algorithm is used, the data flow would be 8 Mbps for 9 seconds. Thus, the constant flow is maintained.

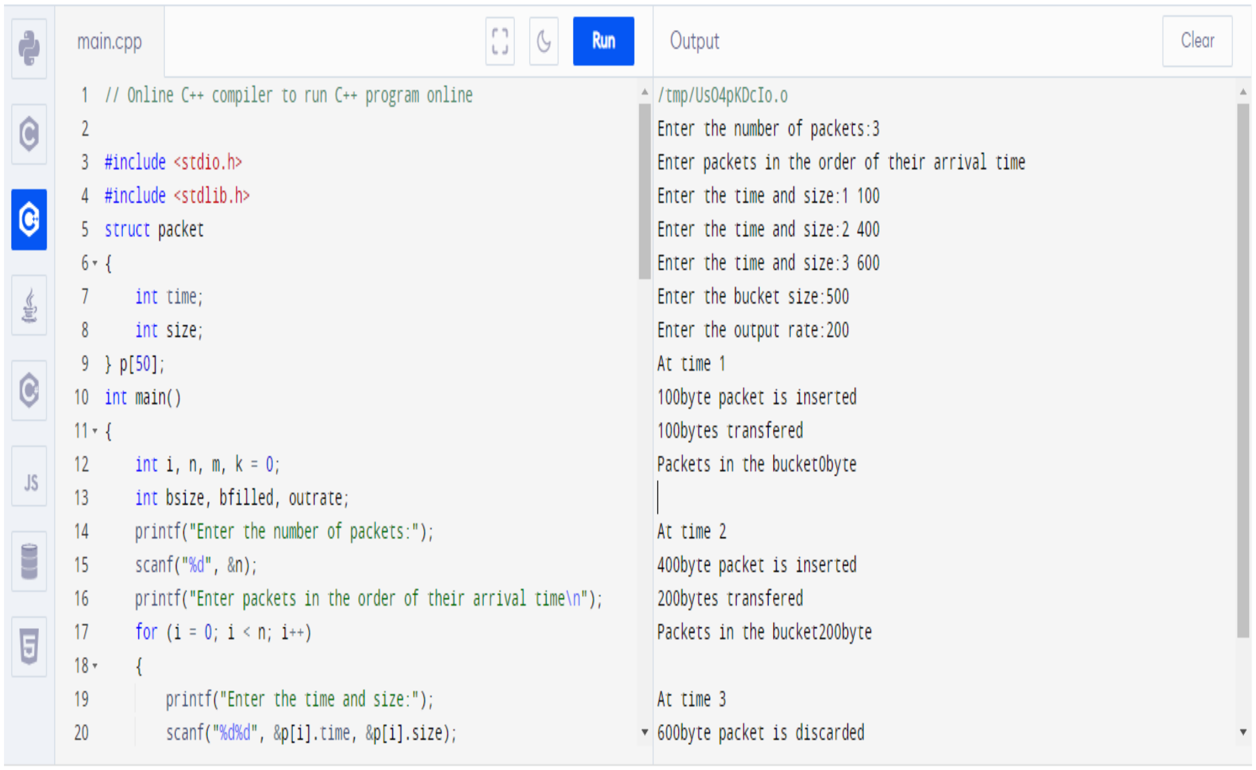
Code:

```
#include <stdio.h>
#include <stdlib.h>
struct packet
{
    int time;
    int size;
} p[50];
int main()
{
    int i, n, m, k = 0;
    int bsize, bfilled, outrate;
    printf("Enter the number of packets:");
    scanf("%d", &n);
```

```
printf("Enter packets in the order of their arrival time\n");
for (i = 0; i < n; i++)
{
   printf("Enter the time and size:");
   scanf("%d%d", &p[i].time, &p[i].size);
}
printf("Enter the bucket size:");
scanf("%d", &bsize);
printf("Enter the output rate:");
scanf("%d", &outrate);

m = p[n - 1].time;
i = 1;
k = 0;
bfilled = 0;
while (i <= m || bfilled != 0)
{
   printf("\n\nAt time %d", i);

   if (p[k].time == i)
   {
      if (bsize >= bfilled + p[k].size)
      {
         bfilled = bfilled + p[k].size;
         printf("\n%dbyte packet is inserted", p[k].size);
         k = k + 1;
      }
      else
      {
         printf("\n%dbyte packet is discarded", p[k].size);
         k = k + 1;
      }

   }

if (bfilled == 0)
{
   printf("\nNo packets to transmitte");
}
else if (bfilled >= outrate)
{
   bfilled = bfilled - outrate;
   printf("\n%dbytes transfered", outrate);
}

else
```

```
    {
       printf("\n%dbytes transfered", bfilled);
       bfilled = 0;
    }
    printf("\nPackets in the bucket%dbyte", bfilled);
    i++;
    }
    return 0;
}
```

Result Printouts:



main.cpp | Run | Output | Clear

```
// Online C++ compiler to run C++ program online

#include <stdio.h>
#include <stdlib.h>
struct packet
{
    int time;
    int size;
} p[50];
int main()
{
    int i, n, m, k = 0;
    int bsize, bfilled, outrate;
    printf("Enter the number of packets:");
    scanf("%d", &n);
    printf("Enter packets in the order of their arrival time\n");
    for (i = 0; i < n; i++)
    {
        printf("Enter the time and size:");
        scanf("%d%d", &p[i].time, &p[i].size);
```

```
/tmp/Us04pKDcIo.o
Enter the number of packets:3
Enter packets in the order of their arrival time
Enter the time and size:1 100
Enter the time and size:2 400
Enter the time and size:3 600
Enter the bucket size:500
Enter the output rate:200
At time 1
100byte packet is inserted
100bytes transfered
Packets in the bucket0byte

At time 2
400byte packet is inserted
200bytes transfered
Packets in the bucket200byte

At time 3
600byte packet is discarded
```

CONCLUSION: