

PROGRAM-12

Write a program for congestion control using Leaky bucket algorithm.

Code:

```
# Getting user inputs
storage = int(input("Enter initial packets in the bucket: "))
no_of_queries = int(input("Enter total no. of times bucket content is checked: ")) bucket_size
= int(input("Enter total no. of packets that can be accommodated in the bucket: "))
input_pkt_size = int(input("Enter no. of packets that enters the bucket at a time: "))
output_pkt_size = int(input("Enter no. of packets that exits the bucket at a time: "))

for i in range(no_of_queries): # space left
    size_left = bucket_size - storage
    if input_pkt_size <= size_left: #
        update storage
        storage += input_pkt_size
    else:
        print("Packet loss =", input_pkt_size)

print(f"Buffer size = {storage} out of bucket size = {bucket_size}")

# as packets are sent out into the network, the size of the storage decreases storage
-= output_pkt_size
```

Output

```
Enter initial packets in the bucket: 0
Enter total no. of times bucket content is checked: 4
Enter total no. of packets that can be accommodated in the bucket: 10
Enter no. of packets that enters the bucket at a time: 4
Enter no. of packets that exits the bucket at a time: 1
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
```

17/12/24

9 9-1

Leaky Bucket Algorithm

In the network layer, the network can make Quality of service guarantees, it must know back of traffic is being guaranteed, one of the main causes of congestion is that traffic is often

There are 2 types of traffic shaping:

1. Leaky bucket
2. Token bucket

Ex :- Let $n = 100$

Packet = 200 700 500 450 400 200

Size $n >$ size of the packet at the head of the Queue i.e. $n > 200$

Therefore, $n = 1000 - 200 = 800$

Packet size of 200 is sent into the network
200 700 500 450 400

Now again $n >$ size of the packet at the head of the queue i.e. $n > 400$

Therefore $n = 800 - 400 = 400$

Code :

```
#include <stdio.h>

int main() {
    int incoming, outgoing, bucket_size, n,
        store = 0;
    printf("Enter bucket size, outgoing node and\n"
           "no of i(p);");
    scanf("%d %d %d", &bucket_size, &outgoing, &n);
    while(n != 0) {
```

```

printf("Enter the incoming packet size:");
scanf("%d", &incoming);
printf("Incoming packet size %d\n",
incoming);

if (incoming == (bucket = size = store)) {
    store += incoming;
    printf("Bucket buffer-size %d out of %d\n",
store, bucket-size);
} else {
    printf("Dropped %d no. of bucket packets\n",
incoming - (bucket size) store);
    printf("Bucket Buffer size %d out of %d\n",
store, bucket-size);
    store = bucket-size;
}

store = store - outgoing;
printf("After outgoing %d bytes left out %d
in buffer\n", store, buck-size);
n--;
}
}

```

Output:

```

Enter bucket size, outgoing rate & no. of inputs: 100 20 3
Enter the incoming packet size: 30
Incoming packet size 30
Bucket Buffer size 30 out of 100
After outgoing 10 bytes left out 100 in buffer
Enter the incoming packet size: 10
Incoming packet size 50
Bucket Buffer size 60 out of 100
After outgoing 40 bytes left out 100 in buffer
Enter the incoming packet size: 80

```