**Ex No: 10 Date:**

**WRITE A PROGRAM FOR CONGESTION CONTROL IN A NETWORK USING LEAKY BUCKET ALGORITHM.**

**Aim:**

To simulate network congestion control using the Leaky Bucket algorithm.

**Theory:**

After the completion of this experiment, student will be able to

* Understand the Leaky Bucket algorithm and its role in network congestion control.
* Simulate the process of packet regulation using the Leaky Bucket mechanism.
* Analyze how packet loss occurs when the bucket capacity is exceeded.
* Interpret the relationship between input packet size, bucket size, and output packet size in maintaining steady network transmission.
* Understand the importance of traffic shaping techniques in managing network bandwidth and preventing congestion.

A simple leaky bucket algorithm can be implemented using FIFO queue. A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.

**Algorithm:**

Step-1: Initialize a counter to n at the tick of the clock.

Step-2: Repeat until n is smaller than the packet size of the packet at the head of the queue.

* Pop a packet out of the head of the queue, say P.
* Send the packet P, into the network
* Decrement the counter by the size of packet P.

Step-3: Reset the counter and go to step 1.

**Example:** Let n=1000

Packet=



Since 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.



Now, again n > size of the packet at the head of the Queue, i.e. n > 400

Therefore, n = 800-400 = 400

Packet size of 400 is sent into the network.



Since, n < size of the packet at the head of the Queue, i.e. n < 450 Therefore, the procedure is stopped.

Initialise n = 1000 on another tick of the clock.

This procedure is repeated until all the packets are sent into the network.

**Program:**

**Sample Output:**

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

**Screenshot of output:**

import java.io.\*;

import java.util.\*;

class LEAKYB {

public static void main(String[] args) {

int no\_of\_queries, storage, output\_pkt\_size;

int input\_pkt\_size, bucket\_size, size\_left;

storage = 0;

no\_of\_queries = 4;

bucket\_size = 10;

input\_pkt\_size = 4;

output\_pkt\_size = 1;

for (int i = 0; i < no\_of\_queries; i++) {

size\_left = bucket\_size - storage;

if (input\_pkt\_size <= size\_left) {

storage += input\_pkt\_size;

} else {

System.out.println("Packet loss = " + input\_pkt\_size);

}

System.out.println("Buffer size= " + storage + " out of bucket size= " + bucket\_size);

storage -= output\_pkt\_size;

if (storage < 0) {

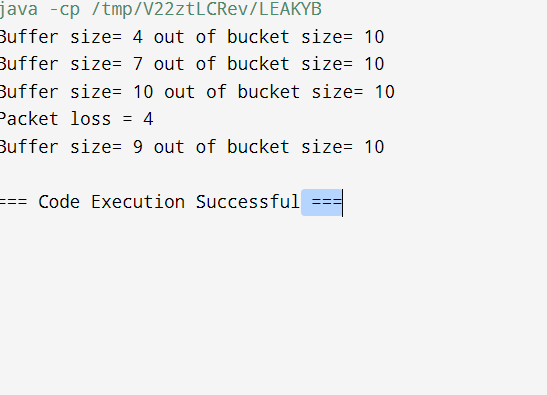
storage = 0;

}

}

}

}



**Result:**

Thus the program for network congestion control using the Leaky Bucket algorithm has been simulated successfully, by implementing java code.