

## Leaky Bucket Algorithm for Congestion Control.

```
#include <iostream.h>
```

```
#include <climits.h>
```

```
#include <stdlib.h>
```

```
#define bucketSize 512.
```

```
void bucketInput (int a, int b) {
```

```
    if (a > bucketSize)
```

```
        cout << "\n\t\t Bucket Overflow";
```

```
    else {
```

```
        delay (500);
```

```
        while (a > b) {
```

```
            cout << "\n\t\t" << b << " bytes outputted ";
```

```
            a -= b
```

```
            delay (500); }
```

```
        if (a > 0)
```

```
            cout << "\n\t\t last " << a << " bytes sent \t";
```

```
            cout << "\n\t\t Bucket output successful"; }
```

```
void main () {
```

```
    int op, pktSize;
```

```
    randomize (1);
```

```
    cout << "Enter output rate:";
```

```
    cin >> op;
```

```
    for (int i = 1; i <= 5; i++) {
```

```
        delay (random (1000));
```

```
        pktSize = random (1000);
```

```
        cout << "\n Packet no " << i << "\t Packet size = " << pktSize;
```

```
        bucketInput (pktSize, op); }
```