

CN LAB

LAB 10

Write a program for congestion control using Leaky bucket algorithm

Program :

```
#include <stdio.h>
```

```
int main() {
```

```
    int bucket_size = 10;
```

```
    int output_rate = 1;    // packets per second
```

```
    int times = 5;
```

```
    int arrival[5] = {6,4,8,1,0};
```

```
    int t;
```

```
    int bucket = 0;        // current packets in bucket (queued)
```

```
    int sent = 0;
```

```
    int dropped = 0;
```

```
    printf("Time | Arrival | BeforeQueue | AfterArrival | Sent | InQueue | DroppedCumul\n");
```

```
    printf("-----+-----+-----+-----+-----+-----+-----\n");
```

```
    for (t = 0; t < times; t++) {
```

```
        int beforeQueue = bucket;
```

```
        // 1. Incoming packets:
```

```
        if (arrival[t] + bucket <= bucket_size) {
```

```
            bucket += arrival[t];
```

```
        } else {
```

```
            int canAccept = bucket_size - bucket;
```

```

        if (canAccept < 0) canAccept = 0;
        dropped += (arrival[t] - canAccept);
        bucket = bucket_size;
    }
    int afterArrival = bucket;

    // 2. Output (leak) at fixed rate:
    int actualSent;
    if (bucket >= output_rate) {
        bucket -= output_rate;
        actualSent = output_rate;
    } else {
        actualSent = bucket;
        bucket = 0;
    }
    sent += actualSent;

    // 3. Print tabular row:
    printf("%4d | %7d | %11d | %12d | %4d | %7d | %13d\n",
        t, arrival[t], beforeQueue, afterArrival, actualSent, bucket, dropped);
}

printf("\nTotal packets sent = %d (at %d/sec)\n", sent, output_rate);
printf("Total packets dropped = %d\n", dropped);

return 0;
}

```

OUTPUT:

Time	Arrival	BeforeQueue	AfterArrival	Sent	InQueue	DroppedCumul
0	6	0	6	1	5	0
1	4	5	9	1	8	0
2	8	8	10	1	9	6
3	1	9	10	1	9	6
4	0	9	9	1	8	6

Total packets sent = 5 (at 1/sec)

Total packets dropped = 6