Assignment 4

Sanjana S Prabhu

September 17, 2019

1 Introduction

The problem is to simulate a network node. Packets arrive randomly at the switch, and are sent out every Δt seconds. . If too many packets arrive, they are queued. A maximum of N packets can be queued. To model the arrival of packets at the node, we use the following function nextTime() which gives the next time a packet arrives for a Poisson process. If the number of packets at the node is larger than N, then these packets are dropped.

2 Mathematical model of the problem

We use queues to solve this problem. To make the implementation easier, we use circular queues. The functions used for implementing the circular queue are given below:

```
struct Queue{
int length;
int front;
int rear;
int *array;
};

struct Queue* MakeNull(int length){
struct Queue* queue=(struct Queue*)malloc(sizeof(struct Queue));
queue->length=length;
queue->front=0;
queue->rear=-1;
queue->array=(int *)malloc(sizeof(int));
return queue;
}

int isEmpty(struct Queue* q){
return q->rear<q->front;
}

int number_elements(struct Queue* q){
return q->rear-q->front+1;
}

int checkFull(struct Queue* q){
if(q->length==number_elements(q))
return 1;
return 0;
}

void enqueue(struct Queue* q, int x){
q->array[(++q->rear)%(q->length)] = x;
}
int dequeue(struct Queue* q,int N){
if(isEmpty(q)){
return -1;
}
int a=q->array[((q->front)++%N)];
return a;
}
```

3 Simulation of node

The node is simulated in the following fashion:

- Firstly , we allocate some memory to a packet. Now we have to run a while loop some N times so as to capture the queueing process.
- The time of arrival of the packet is noted and is compared with $1/\mu$, which is time at which packets are sent or forwarded. This is only for the initial iteration.
- Henceforth, the time of arrival is compared with the forward time, which is the time at which the previous packet has been forwarded.
- If the time of arrival is greater than the forward time by some factors of μ, those many packets are forwarded, or the dequeue-ing process occurs.
- Also the delay for each packet is computed by subtracting its forward time from its time of arrival.

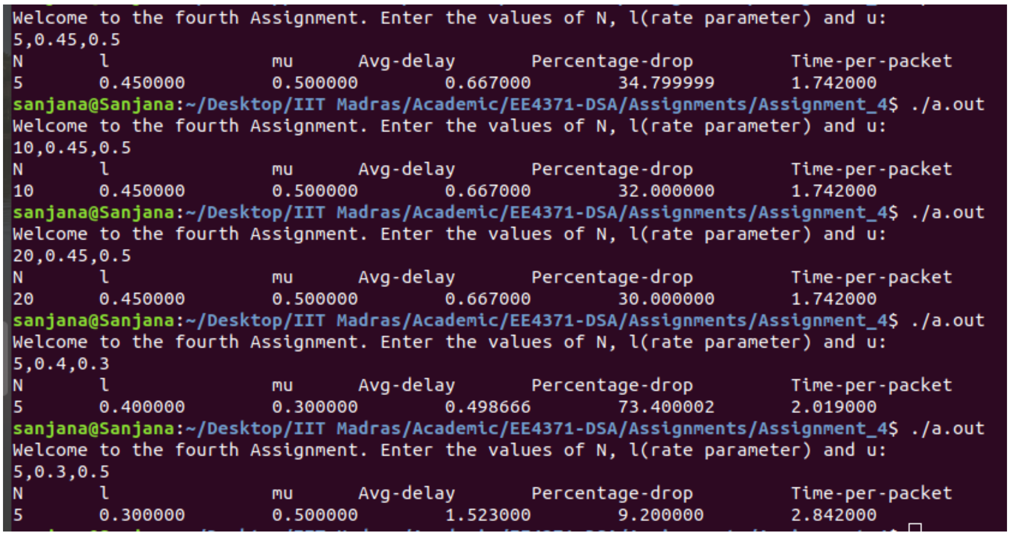
- Finally at each iteration, is the queue is full, the packets are dropped and if not the packets are inserted or queued. The code to simulate the node is as follows:

```
    int packet_simulation(int N, double l, double mu){

PACKET *p;
p = (PACKET *)malloc(sizeof(PACKET));
struct Queue* q = MakeNull(N);
int i=0;
int drop=0;
int t_arrival;
float forward_time=1/mu;
do{
p->id=i;
t_arrival=nextTime(l);
if(t_arrival>forward_time){
delay+=t_arrival-forward_time;
}
int j=1;
while(t_arrival-j*forward_time>=0){
if(number_elements(q)>0){
dequeue(q,N);
}
else
break;
j++;

}
total_time+=t_arrival;
p->t0=total_time;
if(number_elements(q)<N){
enqueue(q,p->id);
}
else{
drop=drop+1;
}
}while(++p!=NULL&&++i<1000);
return drop;
}
```

4 Example output



```
Welcome to the fourth Assignment. Enter the values of N, l(rate parameter) and u:
5,0.45,0.5
N       l               mu      Avg-delay       Percentage-drop         Time-per-packet
5       0.450000        0.500000        0.667000        34.799999       1.742000
sanjana@Sanjana:~/Desktop/IIT Madras/Academic/EE4371-DSA/Assignments/Assignment_4$ ./a.out
Welcome to the fourth Assignment. Enter the values of N, l(rate parameter) and u:
10,0.45,0.5
N       l               mu      Avg-delay       Percentage-drop         Time-per-packet
10      0.450000        0.500000        0.667000        32.000000       1.742000
sanjana@Sanjana:~/Desktop/IIT Madras/Academic/EE4371-DSA/Assignments/Assignment_4$ ./a.out
Welcome to the fourth Assignment. Enter the values of N, l(rate parameter) and u:
20,0.45,0.5
N       l               mu      Avg-delay       Percentage-drop         Time-per-packet
20      0.450000        0.500000        0.667000        30.000000       1.742000
sanjana@Sanjana:~/Desktop/IIT Madras/Academic/EE4371-DSA/Assignments/Assignment_4$ ./a.out
Welcome to the fourth Assignment. Enter the values of N, l(rate parameter) and u:
5,0.4,0.3
N       l               mu      Avg-delay       Percentage-drop         Time-per-packet
5       0.400000        0.300000        0.498666        73.400002       2.019000
sanjana@Sanjana:~/Desktop/IIT Madras/Academic/EE4371-DSA/Assignments/Assignment_4$ ./a.out
Welcome to the fourth Assignment. Enter the values of N, l(rate parameter) and u:
5,0.3,0.5
N       l               mu      Avg-delay       Percentage-drop         Time-per-packet
5       0.300000        0.500000        1.523000        9.200000        2.842000
```

Figure 1: Screenshot of output

The following screenshot is a table depicting the different parameters from different values of N, rate parameter and μ: As noted, the percentage drop varies as follows:

Decreases when μ is greater than λ.

Decreases for larger N. The time per packet increases when μ and λ differ by a greater amount. But is independent of N.

5 Conclusions

The time complexity is manage the queue per packet is O(1) because for every operation, irrespective of whether it is adding elements or removing elements from the queue, we need to perform a constant number of operations, that is changing the values of front and rear and add an element is needed. Hence unlike an array implementation of queue which needs order of n operations for any one operation, the circular array implementation is of O(1) complexity.