ECE 358: Computer Networks

Winter 2015

Project 1: Queue simulation

Date submitted: Thursday, February 5, 2015

Submitted by: Haodong Huang

Student ID: 20442592

Student name: Haodong Huang

Email: [h53huang@uwaterloo.ca](mailto:h53huang@uwaterloo.ca)

Marks received:

Marked by

M/M/1 Queue

# Question 1:

int numbersTotal = 1000;

double Random::generateRanNum()

{

return (rand() / (double)(RAND\_MAX));

}

double Random::generateExponentialRanVar(double lambda)

{

return ((double)(-1.0)/lambda)\*log(1.0-generateRanNum());

}

void Random::variableTest()

{

double number, total, total2, mean, variance;

total = 0;

for(int i = 0; i < numbersTotal; i++)

{

number = generateExponentialRanVar(75.0);

total += number;

generatedNumbers[i] = number;

}

mean = total / numbersTotal;

for(int i = 0; i < numbersTotal; i++)

{

total2 += (generatedNumbers[i] - mean) \* (generatedNumbers[i] - mean);

}

variance = total2 / numbersTotal;

std::cout<< "Mean: " << mean << "\n";

std::cout<< "Variance: " << variance;

}

int main()

{

srand(time(NULL));

Random \*random = new Random();

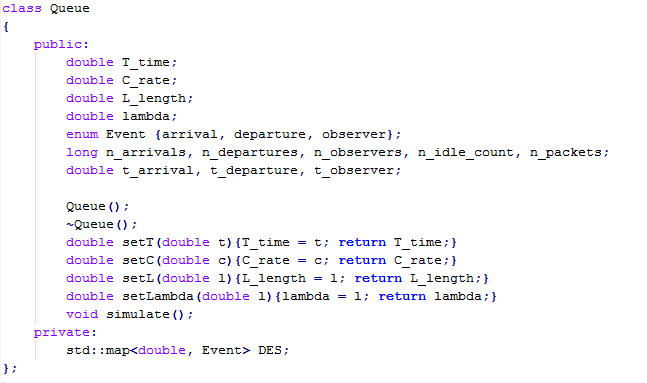
std::cout<< random->generateExponentialRanVar(75)<< std::endl;

delete random;

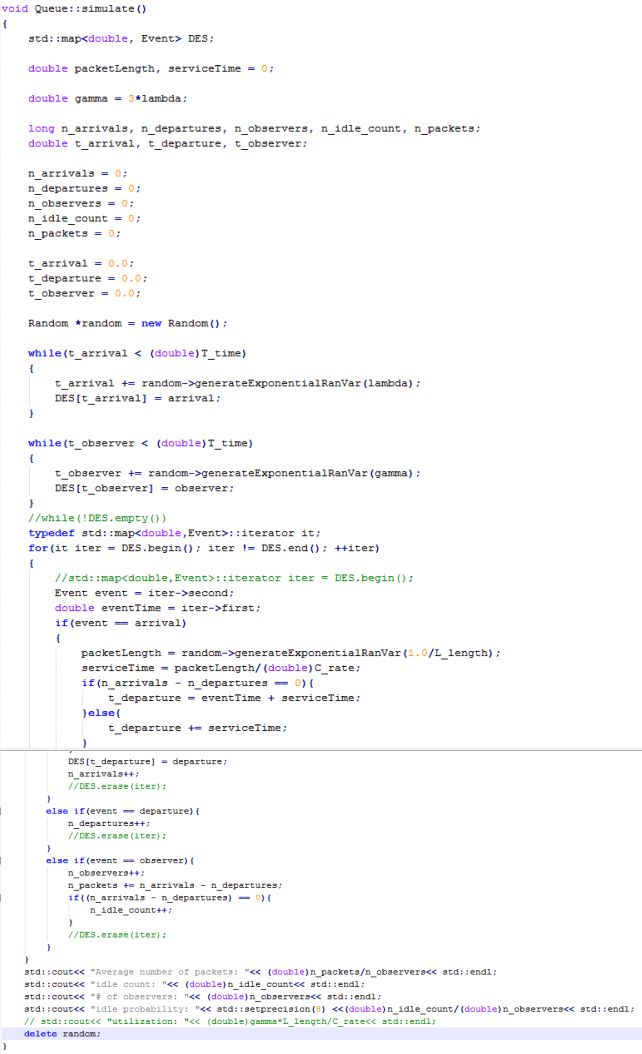
}

The expected value (0.013276) and variance (0.000179143) are close to 1/75 and 1/75^2 as expected.

# Question 2:



The variables n\_arrivals, n\_departures and n\_observers are used to count their respective events. n\_idle\_count is to keep track of how often the queue is empty, and n\_packets is used to count the number of packets in the queue at a given observation (this number needs to be averaged). T\_time is the length of the simulation in seconds.



The random number generator and exponential random variable generator are used to generate incoming packet lengths and times. The random number is uniform between 0 and 1. The exponential variable is given by var = -1/λ \* ln(1- U(0,1)) where U(0,1) is our random number.

First, we initialize some variables such as the simulation time (T), bit rate (C) and the average bit length (L) and lambda (ρ\*C/L). We let the simulation run for T seconds and generate random arrivals and observers based on the exponential generator using lambda as the parameter for arrivals and 3\*lambda for observers. These events are put in our DES and ordered in an ordered map based on their times. We then check these events in order. If it’s an arrival, then a departure time is created, based on the previous departure time plus service time. We increment our counter for arrival. If it’s a departure event that’s first in the DES, we increment our count for departures. If it’s an observer, we increment our counter for observers. We make note of how many number of packets are in the DES for later. If the buffer is empty, i.e. arrival counter = departure counter, then we increment our idle counter.

# Question 3:

|  |  |  |
| --- | --- | --- |
| Ρ (utilization) | E[N] | P\_idle |
| 0.25 | 0.333608 | 0.749799 |
| 0.35 | 0.535664 | 0.650774 |
| 0.45 | 0.820634 | 0.549453 |
| 0.55 | 1.22489 | 0.448976 |
| 0.65 | 1.8273 | 0.352883 |
| 0.75 | 2.99545 | 0.250727 |
| 0.85 | 5.76602 | 0.148064 |
| 0.95 | 18.5054 | 0.0505985 |

The code in question 2 was used to find E[N] and P\_idle. A for loop was used to run through each size of rho. The value of T is 10000.

The average number of packets (E[N]) was calculated by dividing the sum of the number of packets observed every time divided by the number of observations. As ρ increases, the average number of packets in the system goes up because more packets are being generated in between observations without being serviced because service time is fixed. Idle time goes down because you are unlikely to have all the packets serviced at the time of the observation.

# Question 4:

|  |  |  |
| --- | --- | --- |
| ρ | E[N] | P\_idle |
| 1.2 | 83437.4 | 3.3360678e-06 |

As ρ increased to 1.2, the number of packets increased by a significant amount and idle time dropped to almost zero. The service time has not changed but the number of packets that get generated keep going up. The service cannot keep up with the number of packets being generated so more and more packets fill up the DES. The average number of packets is expected to increase as time goes on.

M/M/1/K Queue

# Question 5: