Simulation project
Task 3 – Statistical estimation of performance metrics

Objectives

Given the following input variable:

1. Arrival process of new requests: Poisson distributed with mean λ.
   We recall that 1,000 IoT devices attempt to be certified randomly over a 5-hour period. We assume that new requests arrive in a Possion manner at the rate of λ=1,000/5x60x60=0.0556/sec. That is, the inter-arrival time of successive new requests is assumed to be exponentially distributed with a mean 1/λ=1/0.0556=17.98 sec.
2. Arrival process of retransmitted requests: Poisson distributed with parameter *d.*
3. Service time: Constant equal to *s*.
4. Buffer size: *B*.

We want to estimate the following performance metrics:

1. The mean and 95th percentile and their confidence interval of the time *T* it takes to certify an IoT device.
2. The mean and 95th percentile and their confidence interval of the time *D* it takes a new request to enter the queue if it is rejected the first time it arrived.
3. The mean and its confidence interval of the time *P* it takes to process all 1,000 IoT devices.

This will be done using the method of repetitions in order to simulate at least 30 5-hour periods.

You can use any high-level language of your choice. You can develop your program on your personal computer, but it has to run and compile on eos. Programming elegance is not required! What is important is that your program runs correctly. For this, you are *strongly* advised to follow the instructions in this handout!

Remember that any exchange of code with another student is forbidden as it constitutes cheating. Keep the code to yourselves! We will run Moss at the end of this assignment to verify that there has not been any code sharing. Offenders will receive an automatic zero for the entire project.

Additions to the basic code

Delete all print statements in your code, which print one line of output each time the simulation advances to the next event.

Your program should read the following input parameters using prompts from the console: mean inter-arrival time 1/λ*,* service time s, mean retransmission time *1/d,* buffer size *B,* and number of repetitions *N*

1. Statistics collection

Modify the basic code you developed so far to collect data on a) T, the total amount of time it takes for a request to be processed from the moment it arrives to the queue to the moment it departs from the server, including possible retransmissions, b) D, the amount of time elapsing from the moment a request is rejected to the moment it enters successfully the buffer, and c) P, the total time to process all 1,000 IoT devices. For a) you will have a total of 1,000 observations, whereas for b) less than 1,000, and for c) just a single observation.

At the end of the simulation run (i.e., when all 1,000 IoT devices have been processed), calculate the following.

1. The mean and 95^{th} percentile of the T.
2. The mean and 95^{th} percentile of the D.

Store this information along with P in an array of size N (see below).

2. Accounting for the initial conditions

In the real system, we do start with an empty buffer and an idle server. In the simulation, the only initial assumption we make is regarding the time of the first arrival. Consequently, it is highly unlikely that this will seriously affect the results. That is, if we used a different arrival time for the first request, we will not get different values for T, D, and P. In view of this, we will not account for the starting conditions. That is, we will use all the results in our statistical calculations and we will not ignore some of the early results, as is normally done in other simulations.

3. Calculation of the 99^{th} percentile.

Given a random variable, T (or D), its 95^{th} percentile is a value $t_{0.95}$ such that only 5% of the values of the random variable are greater than itself, that is, Prob[$T \leq t_{0.95}$] = 0.95. The percentile gives us an idea of how long the tail is of the distribution. This is calculated as follows.

Let t_1, $t_{2, \ldots,}$ t_{1000} be the sample of the T values obtained after we run the simulation. To calculate the percentile, you have to sort them out in ascending order. Let $y_1 \leq y_2 \leq \ldots \leq y_{1000}$ be the sorted observations. Then, the 95^{th} percentile T is the value y_k where k=ceiling(0.95x1000), where ceiling(x) is the ceiling function that maps the real number x to the smallest integer not less than x. For a sample of 1,000 observation, k = 950, and the percentile is the value y_{950}. For D apply the same procedure, only remember that the sample size will typically be less than a 1,000.

4. Estimation by repetition: Run the above code N times.

Rerun your code with the above additions N times. Each time you start on a new repetition, you zero all counters, and start again from the beginning with the same initial condition, but use a different seed for the pseudo-random number generator, so that to generate a different sequence of random numbers. (Otherwise, you will get the same sample path as in previous repetition!)

5. Final statistical calculation

After N repetitions you should have populated a table with N entries, where the ith entry contains

1. The mean and 95th percentile of the T for the ith repetition
2. The mean and 95th percentile of the D for the ith repetition
3. Total time P it took to process all 1,000 IoT devices

As shown in table below.

Repetition	Mean T	95th percentile of T	Mean D	95th percentile of D	P
1	$\bar{t}_1$	$\bar{t}_{1,0.95}$	$\bar{d}_1$	$\bar{d}_{1,0.95}$	p_1
2	$\bar{t}_2$	$\bar{t}_{2,0.95}$	$\bar{d}_2$	$\bar{d}_{2,0.95}$	P_2
		. . .			
N	$\bar{t}_N$	$\bar{t}_{N,0.95}$	$\bar{d}_N$	$\bar{d}_{N,0.95}$	p_N

An important observation is that all the N entries for each of these values are i.i.d! This means that you can calculate the grand mean and confidence interval of of T using the values $\bar{t}_1, \bar{t}_2, \ldots \bar{t}_N$. Likewise for the 95th percentile of T, mean D, and 95th percentile of D.

For P, we can only compute the mean and its confidence interval. We can also compute the 95th percentile, but it will not be very accurate because of the small sample.

Deliverables

Submit your code, and a text file named "output.txt" with all your results and comments. Use the following input data. (Remember, that the input values should be entered using prompts from the console.)

Mean inter-arrival time	$1/\lambda$ = 17.98 sec
Service time	s
Mean retransmission time	$1/d$ = 10 sec
Buffer size	B = 5
Number of repetitions	N = 50

Task 1:

Vary s as follows: 11, 12, 13, 14, 15, 16, 17. For each value obtain the statistics:

1. Mean and 95th percentile and their confidence interval of T.
2. Mean and 95th percentile and their confidence interval of D.
3. Mean and confidence interval of P

Provide the results in a table and also give graphs for the results in 1 and 2. Comment on your results.

Task 2:

Investigate the effect of varying *B* on all the above statistics for *s* =16. Provide numerical results in tables and/or curves and comments on your findings.

Grading

- 60 points if the code compiles and runs on eos correctly
- 30 points if the code gives correct results on test data designed by TA
- 10 points for summarizing your results.