**Simulation Project – 3rd Assignment**

*Please note that any exchange of code with another student is discouraged as it constitutes cheating*. *We will run Moss to verify that there has not been code sharing.*

**1. Project description**

The objective of this assignment is to estimate the time it takes a packet to be delivered to the client and played out. This is the time elapsed from the moment a packet begins its service at the server to the moment its service is completed at the client. We shall refer to this time as the *end-to-end delay* *W*. Specifically, we are interested in estimating the 95th percentile of *W.* Below, there is a brief description of how to calculate the 95th percentile and its confidence interval.

*The 95th percentile of the end-to-end delay*

For each packet *i*, mark the time *ti* at which it starts its service at the server. When the packet departs from the client, calculate its end-to-end delay *wi* by subtracting its start time *ti* from its departure time given by the clock CL-cq. Store this value in an array and continue this process until you have collected a predefined number of values.

Based on the observations stored in the array calculate the 95th percentile as follows. The 95th percentile of the end-to-end delay is a value *T* such that Prob[*W ≤ T*] = 0.95. Now, suppose that the total number of observations is *n*, i.e., *w1*, *w2*, …, *wn*. To calculate the percentile *T*, you have to sort them out in ascending order. Let *y1≤y2≤ …≤yn* be the sorted observations. Then, the 95th percentile *T* is the value *yk* where *k=ceiling*(0.95*n*), where *ceiling(x)* is the ceiling function that maps the real number *x* to the smallest integer not less than *x*. For instance, if *n* = 50, then *k* = 48, and the percentile is the value *y48*.

*The confidence interval of the 95th percentile of the end-to-end delay*

A confidence interval provides an indication of the error associated with our estimate, which in this case is the 95th percentile. To calculate the confidence interval you will have to obtain about 30 different percentiles of the end-to-end delay for the same input values. An easy way to do this is to use the *batch method*. Run your simulation for a large number of departures, and then group your results as follows. Let us say that you run the simulation for 90,000 departures, and that you divide all departures into batches of 3000. Therefore, there are 30 batches. For the first 3000 departures in batch 1, calculate the 95th percentile of the end-to-end delay and store this number in an array. Continue the simulation for another 3000 departures without changing anything in your simulation, and so on until you have obtained 30 different percentiles. Now, let *T1, T2, T3, ..., Tn* be the calculated percentiles for *n=*30 batches. Then, the mean of the percentiles is:

,

and the standard deviation *s* is:



The confidence interval at 95% confidence is given by:



The confidence interval tells us that the true 95th percentile lies within the interval 95% of the time. That is, if we repeat the above simulation 100 times, 95% of these times on the average the true 95th percentile will be within the interval. Accurate simulation results require that the confidence interval is very small. That is, the error 1.96(s/) ~ 0.01*Tmean*. If the error is not small enough, increase the batch size from 3000 arrivals to 6000, and run the simulation again for 30 batches. Keep increasing the batch size until you are satisfied that the error is within the limits.

**2. Deliverables**

* Suppress the output your program produced in assignment 2 each time an event occurred.
* Use the same initial conditions as before, i.e., at time zero the server starts transmitting a packet at the high transmission speed, and the infinite server queue and the client queue are both empty. Modify your code so that the time to transmit a packet at the server at the high or low rate is exponentially distributed.
* Use the same prompts as in assignment 2 to provide the input parameters to the simulation, including an additional prompt for the batch size. That is:
* N: Number of packets that departed from the client queue.
* DH: Mean time to transmit a packet from the server at high rate.
* DL: Mean time to transmit a packet from the server at low rate.
* 1/μd: Mean service time in the infinite server queue.
* 1/μq: Mean service time in the client queue.
* TL and TH: Low and high threshold level in the server queue.
* Batch size

Use the following vales: DH = 1, DL = 2, 1/μd = 10, 1/μq = 1.5, TL = 3, TH = 6, N=90,000, batch size=3,000. Modify the batch size as needed per discussion above.

* *Warm-up period:* Before you start collecting the observations for the 30 batches, you need to run the simulation through a warm-up period in order to get away from the initial conditions and the seed used in the pseudo-random number generator. Run the simulation for 500 departures. Discard the observations collected and continue to simulate without changing anything else in your simulation in order to obtain the required observations for the batch means method.

*What to obtain*

Calculate the 95th percentile of the end-to-end delay and its confidence interval. Also, plot the histogram of all the end-to-end delays using the entire sample.

*Validation*

If you set DH = DL = 2, then it is possible to estimate the mean end-to-end delay using queueing theory. In particular, this mean is equal to DH + 1/μd + the mean time in an M/M/1 queue with λ=1/DH and μ = μq.

*What to submit*

Submit your program and a report containing the 95th percentile of the end-to-end delay and its confidence interval, and the validation result with a short discussion. Make sure that your program runs and compiles on eos machines.