CECS 622: Simulator Final Report

The Simulation of Car Wash Queuing Model

Phi Bui

University of Louisville

Spring 2014

April 24, 2014

Introduction

Couple of days after this assignment is assigned, I brought my car to have a car wash to wash out the salts that sticks on my car because of the big snow storm few days before. At the car wash station, I were in line and waited about fifteen minutes before I get my car wash service. Based on this real world concept, I thought about this project and it is very interesting to design a simulation queuing model of Car Wash.

The purpose of this project is to simulate car wash queuing model which allows people (or car wash station's owner) to keep track of the system state so they can see the throughput, response time, average response time, average waiting time, and server utilization.

The significance of this queuing model breaks down to three components. First is arrivals (or inputs), second is queue discipline (or waiting in line), and last is the service facility. Based on above three components, we have system of Car Wash, customers are cars, and the server is the car wash facility.

In this car wash model, it can have any number of cars arriving, any number of cars waiting in queue, and number of cars being washed depending on whether if it is single-server or multi-server system. Therefore, cars could change from one state to another. If a car arrives and another car being washed, the arrival car becomes a waiting car in queue and if there is at least one car waiting in queue the car wash facility is available, then a waiting car becomes a being washed state.

Scope of the Simulator

In this project, I will be simulating First-In-First-Out (FIFO) or First-Come-First-Serve (FCFS) for single-server queue model, multi-server single queue model, and multi-server multi queue model.

System state:

- Current time (CLOCK)
- Number of customers in queue(L_Q)
- Number of customers are serving(L_S)

Entities:

- Car (C_i,t_1,t_2) car Ci who arrived at time t_1, begin service at t_2

Event:

- Arrival: (A,t,C_i) Car i arrives at time t
- Departure: (D,t,C_i) Car i departures at time t
- Stop event: (E, T) simulation stops at time T

Activities:

- Inter-arrival time: a*, Uniform Distribution
- Service time: s*, Normal Distribution

Delay:

- Customer (Car) time spent in waiting line.

Case 1: Car Wash Single-Server Queue Model Design

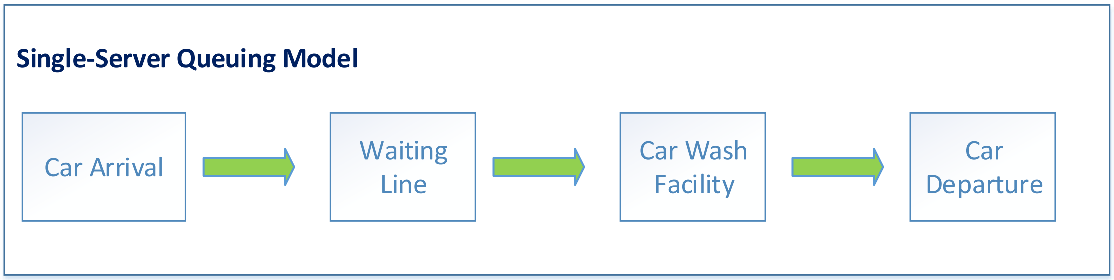
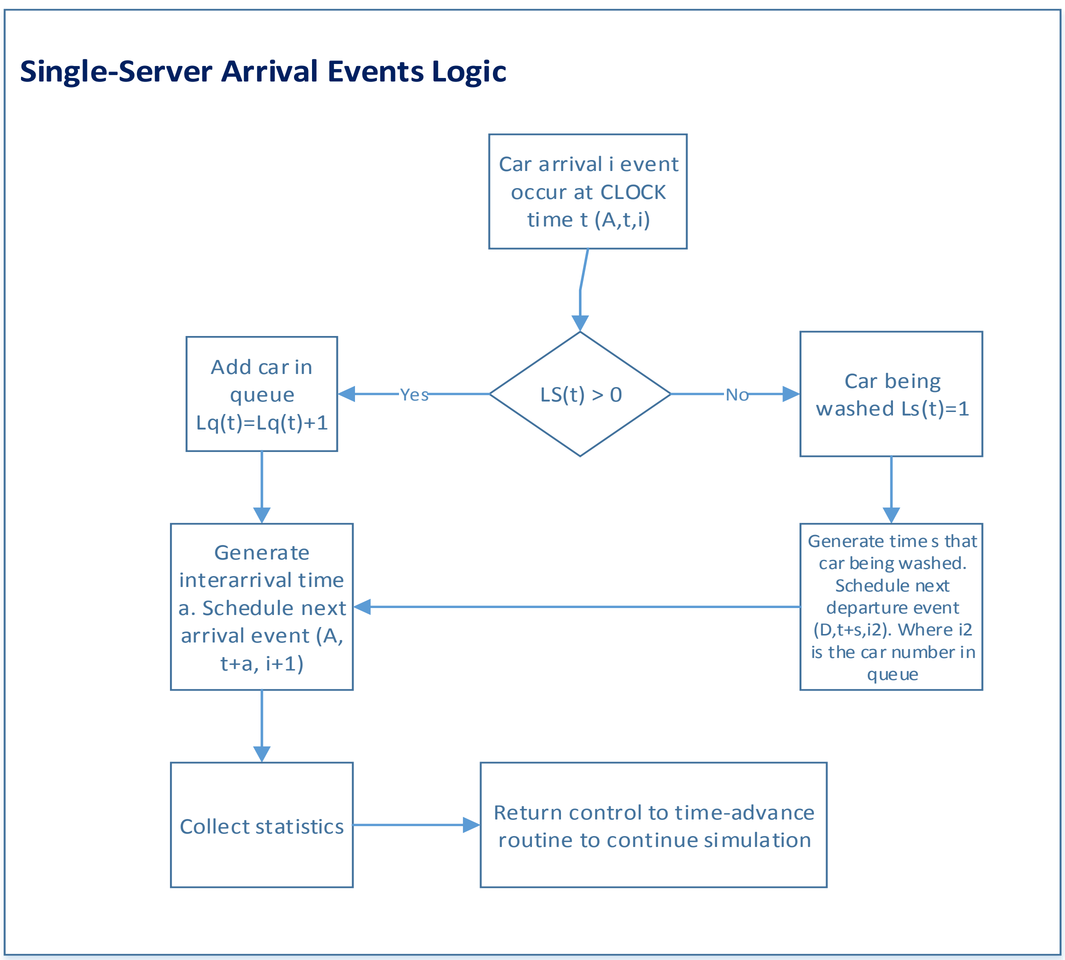
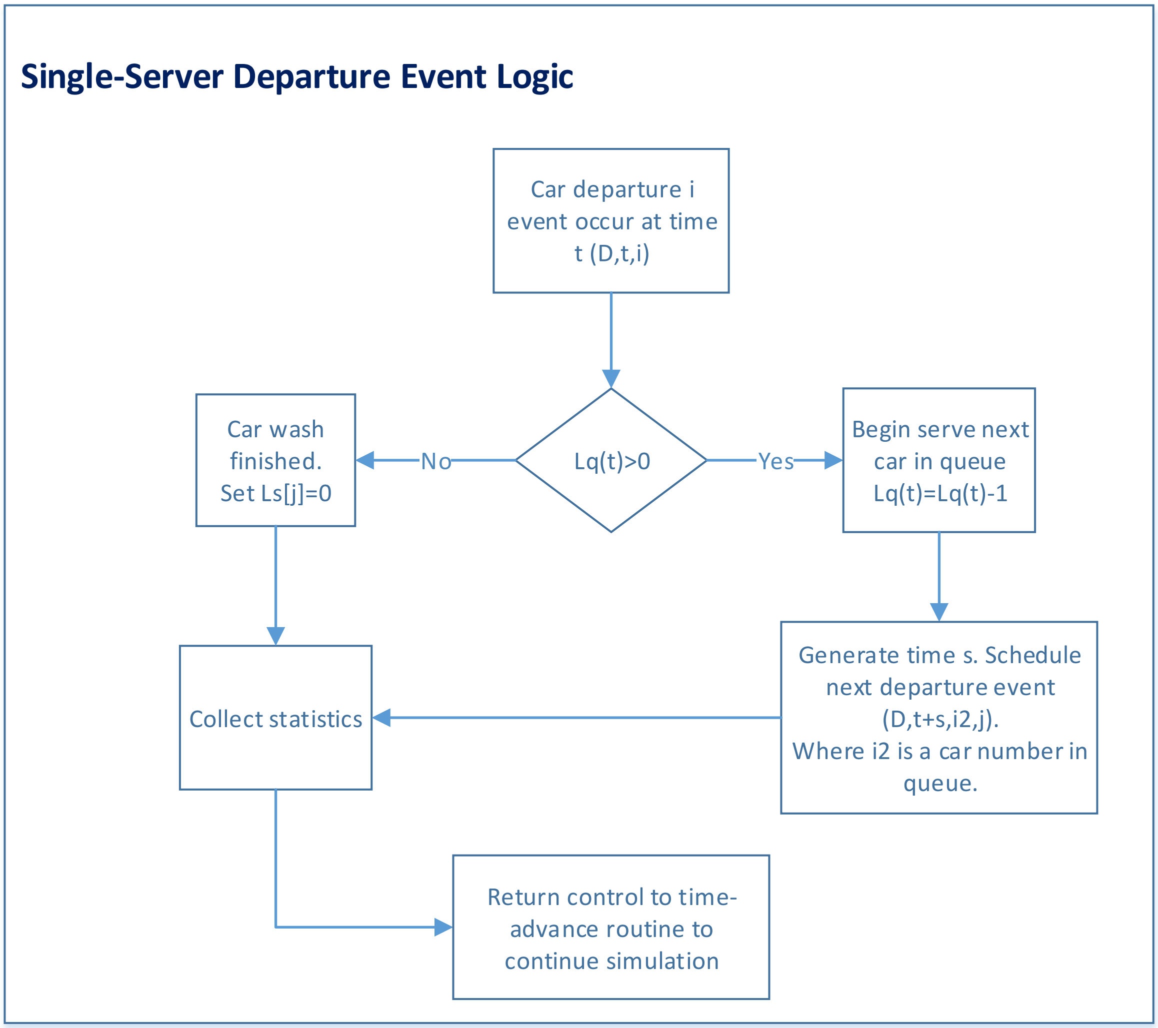


Figure 1: Car Wash Single-Server Queuing Model Design

Single-Server Arrival Events Logic



Single-Server Departure Events Logic



Case 2: Car Wash Multi-Server Single Queue Model Design

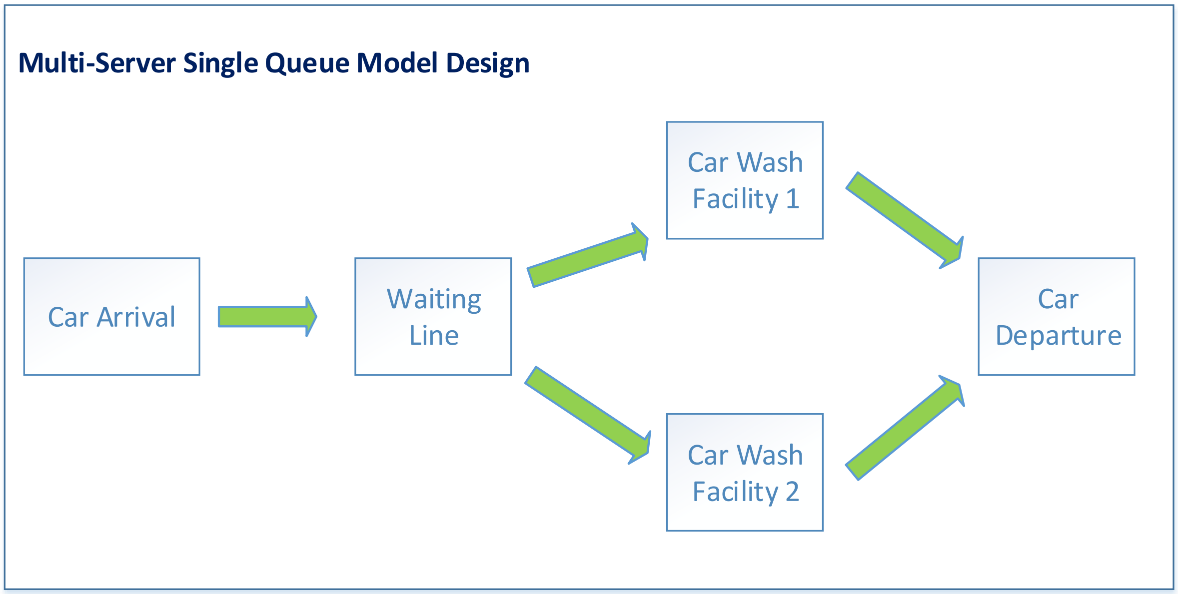


Figure 2 : Car Wash Multi-Server Single Queue *Model Design*

Case 3: Car Wash Multi-Server Multi Queue Model Design

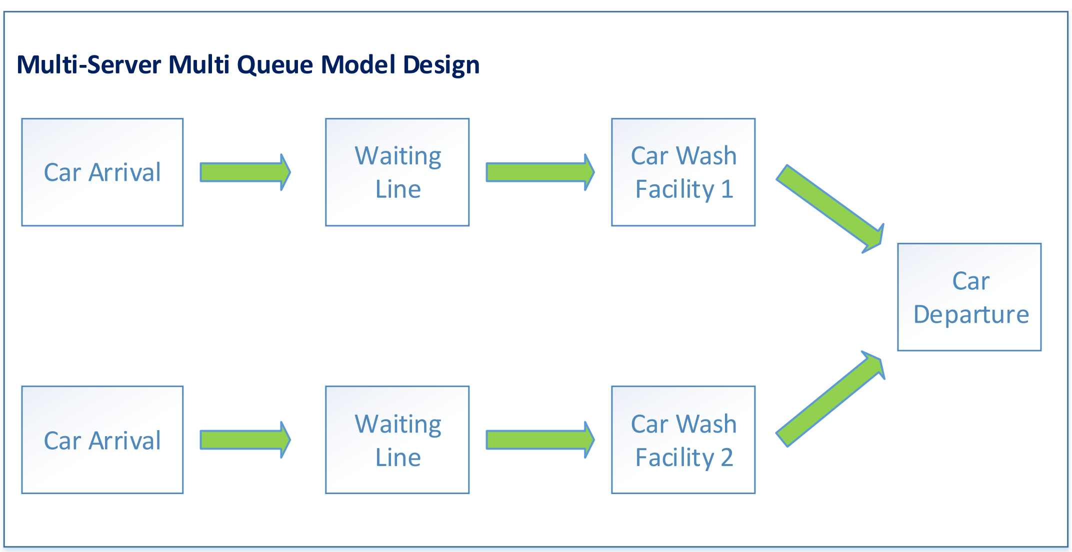
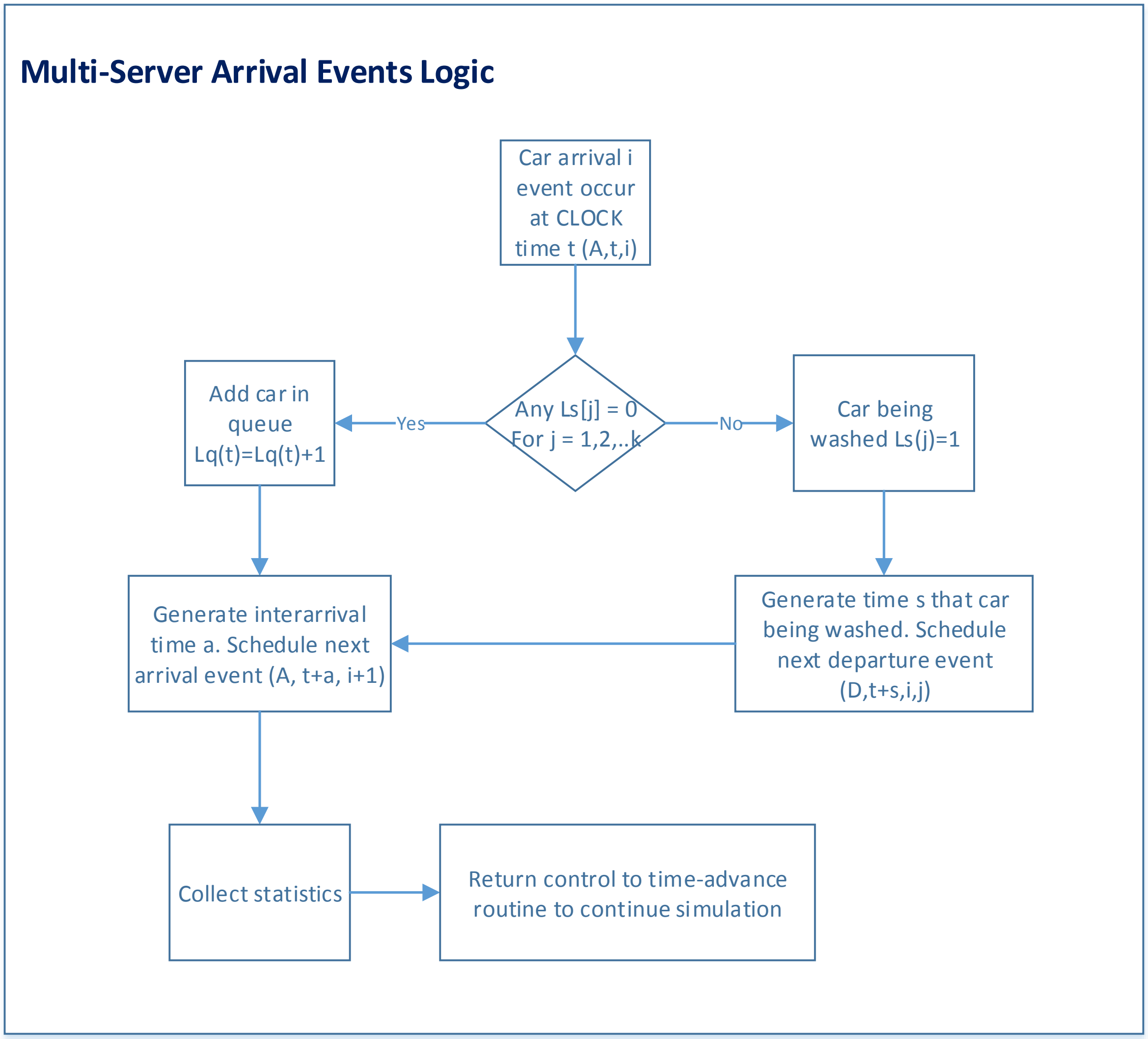
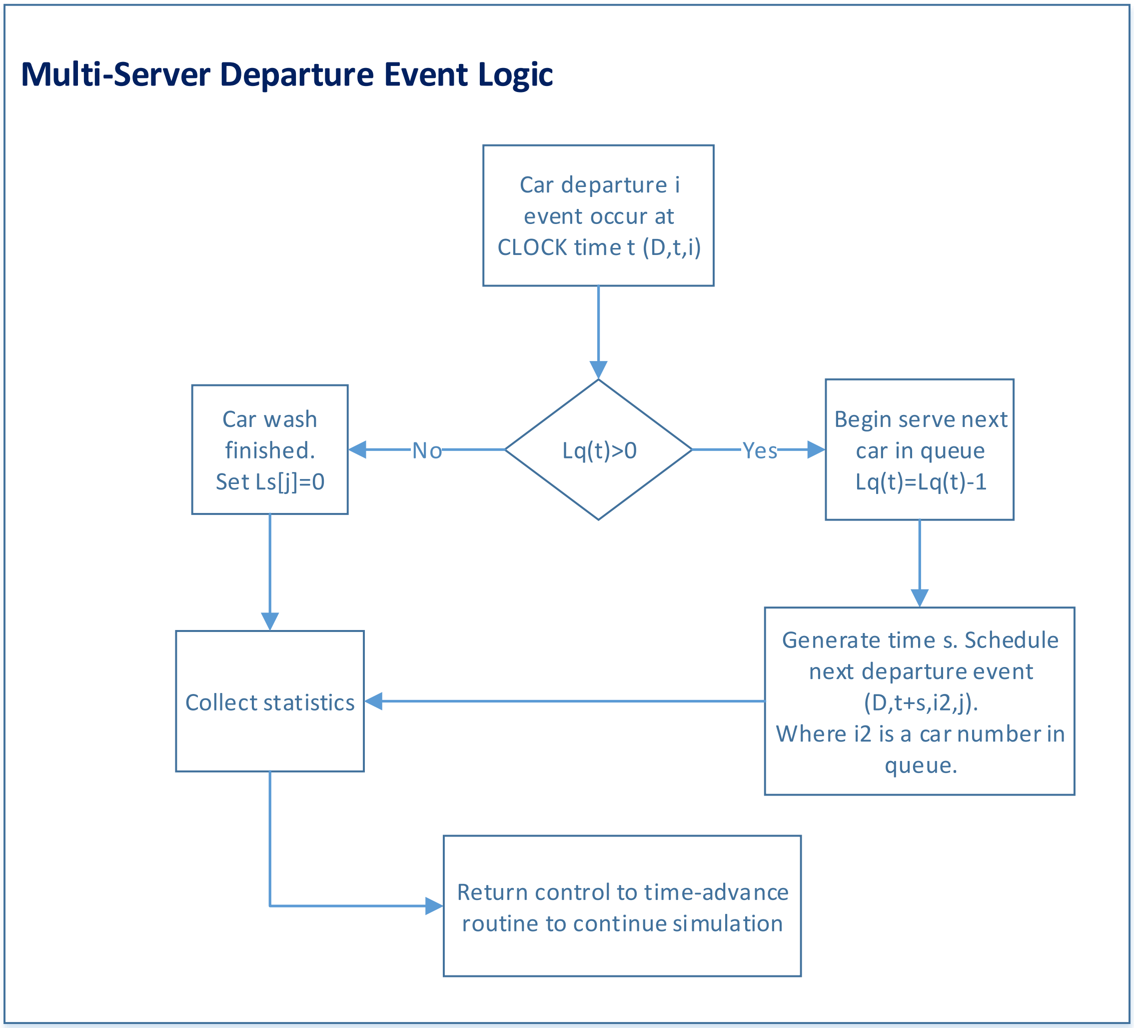


Figure 3: Car Wash Multi-Server Multi Queue *Model Design*

Multi-Server Arrival Events Logic



Multi-Server Departure Events Logic



Assumptions

Car arrival is randomly arrives at any time. In this car wash model, it can have any number of cars arriving, any number of cars waiting in queue, and number of cars being washed depending on whether if it is single-server or multi-server system. Therefore, cars could change from one state to another. If a car arrives and another car being washed, the arrival car becomes a waiting car in queue and if there is at least one car waiting in queue the car wash facility is available, then a waiting car becomes a being washed state.

Another assumption is this car wash station multiple service types and the total time it takes to serve per car varies and service type will be selected randomly during the simulation.

Design Details

1. Available inputs from the user

- Total Running Time
- Inter-arrival Min time
- Inter-arrival Max time
- Service MU (Exponential distribution with mean mu)
- Service SIGMA (Standard deviation of service time)
- Total Number of Server (Available for Multi-Server simulation)

2. Events explanations based on the flowcharts

- The flowcharts for single-server and multi-server are very much self-explanatory. The customer (Car) arrives at random time, if the car wash facility is not busy then the first car will be serve. If the server is busy then place customer in queue and wait for the next available car wash facility. The total time to finish each car are fixed. Once each car finished car wash, the statistic will be collect at departure and the program will run until it’s complete the simulation.

3. Statistics will be collect during simulation

- Total response time
- Total serve time
- Total waiting time (queue)
- Total Arrival Customers (Cars)
- Total Departure Customers (Cars)

4. Performance parameters will be calculated based on statistics

- **Throughput**: Number of cars serve per unit time = $\frac{\text{(cars served)}}{\text{(Total time)}}$
- **Response time**: Time of a car spends in the system =

$$\text{(wait time } + \text{ serve time)} = \text{(Departure time} - \text{Arrival time)}$$

- **Average response time** = $\frac{\text{total response time}}{\text{(number of departure cars)}}$
- **Average wait time** = $\frac{\text{(total wait time for each car)}}{\text{(total cars)}}$
- **Server Utilization** = $\frac{\text{(Total serve time)}}{\text{(total running time)}}$

5. Outputs display to the user

- Car number
- Clock time
- Average car spends in the system
- Average car wait in queue
- Future event list
- Total response time
- Total wait time
- Departure car
- Number of Server

Manual Simulation

System State

- Clock – Current time
- LQ – Number of customer in queue (waiting line)
- LS – Number of customer in system (car wash facility)

Event Notices

- Arrival: (A,t,C_i) - Customer i arrives at time t

- Departure: (D,t,C_i) - Customer i departures at time t
- Stopping event: (E, T) - Simulation stops at time T

Entities

- Customer (Car): (Ci,t_1,t_2) - Customer C_i who arrived at time t_1, and begin service at t_2

Activities

- Service time: s* and Normal Distribution (service MU, service SIGMA)
- Interarrival time: a* and Uniform Distribution (3, 10)

Delays

- Delay of customer (car) waiting in line (in queue)

Initialization

- Assumed that always at least one customer arrived at time zero (0)

Implementation Details

System State

- Clock
- Statistic (LQ, and LS)

Entity Attributes

- Customer

Future Event List

- FutureEventList

Input Parameters

- TotalRunningTime – Total running time
- InterArrivalMin – Minimum of interarrival time
- InterArrivalMax – Maximum of interarrival time
- ServeMu – Exponential distribution with mean mu
- ServeSigma – Standard deviation

- TotalServer – Number of server to be simulated for multi-server cases.

Statistics

- totalResponseTime – Total response time
- totalServeTime – Total service time
- totalWaitTime – Total time that customer waiting in line (time queue)
- nDepartureCustomer – number of departure customers

Help Functions

- NormalDistribution – Normal distribution helper
- ExponentialDistribution – Exponential distribution helper
- ExponentialDistribution – Exponential distribution helper
- UniformDistribution – Uniform distribution helper
- UniformDescreetDistribution – Uniform discreet distribution helper

General Methods

- genA – Generate customer arrival
- genS – Generate service time
- procArrival – Process customer arrival
- proccessDeparture – process customer departure
- processEvent – process customer event state whether customer arrival or departure
- PrepareRun – Set initialization and run simulation
- PostRun – print statistic after the simulation is completed

List of input details specification for queuing case 1, 2 and 3

Below are available general Input settings for all queuing cases (Descriptions for each of the input parameter are provided in Implement Detail section)

- Total Running Time
- Interarrival Min
- Interarrival Max
- Service MU
- Service SIGMA

Additional input setting for Case 2 and 3

- Total Server

Parameters

Case 1:

Interarrival time: a*, Uniform Distribution (3,10)
Service time: s*, Normal Distribution (mu=5, sigma=1)

Case 2:

TotalServer = 2
Interarrival time: a*, Uniform Distribution (3,10)
Service time: s*, Normal Distribution (mu=5, sigma=1)

Case 3:

TotalServer = 2
Interarrival time: a*, Uniform Distribution (3,10)
Service time: s*, Normal Distribution (mu=5, sigma=1)
Arrival go to total server: j*, Uniform Discreet (1, TotalServer),
j*= floor(Rnd() * TotalServer) + 1

List of outputs displayed for queuing case 1, 2 and 3

Case 1 – Single-server output results

Input Parameters
Total Running Time: 50
Interarrival Min: 3
Interarrival Max: 15
Service MU: 10
Service SIGMA: 1

Case 1: Single-server queuing model simulation results

Clock	LQ	LS	Queue List	Future Event List	R	S	W	Nd
0	0	0		(A,C1,0)(E,C0,50)	0	0	0	0
0	0	1	(C1,0,0)	(A,C2,3)(D,C1,5)(E,C0,50)	0	0	0	0
3	1	1	(C1,0,0)(C2,3,0)	(D,C1,5)(A,C3,6)(E,C0,50)	0	0	0	0
5	0	1	(C2,3,5)	(A,C3,6)(D,C2,12)(E,C0,50)	5	5	0	1
6	1	1	(C2,3,5)(C3,6,0)	(D,C2,12)(A,C4,14)(E,C0,50)	5	5	0	1
12	0	1	(C3,6,12)	(A,C4,14)(D,C3,17)(E,C0,50)	14	12	2	2
14	1	1	(C3,6,12)(C4,14,0)	(D,C3,17)(A,C5,17)(E,C0,50)	14	12	2	2
17	0	1	(C4,14,17)	(A,C5,17)(D,C4,21)(E,C0,50)	25	17	8	3
17	1	1	(C4,14,17)(C5,17,0)	(D,C4,21)(A,C6,25)(E,C0,50)	25	17	8	3
21	0	1	(C5,17,21)	(A,C6,25)(D,C5,25)(E,C0,50)	32	21	11	4
25	1	1	(C5,17,21)(C6,25,0)	(D,C5,25)(A,C7,33)(E,C0,50)	32	21	11	4
25	0	1	(C6,25,25)	(D,C6,29)(A,C7,33)(E,C0,50)	40	25	15	5
29	0	0		(A,C7,33)(E,C0,50)	44	29	15	6
33	0	1	(C7,33,33)	(D,C7,38)(A,C8,42)(E,C0,50)	44	29	15	6
38	0	0		(A,C8,42)(E,C0,50)	49	34	15	7
42	0	1	(C8,42,42)	(D,C8,46)(A,C9,49)(E,C0,50)	49	34	15	7
46	0	0		(A,C9,49)(E,C0,50)	53	38	15	8
49	0	1	(C9,49,49)	(E,C0,50)(D,C9,55)(A,C10,56)	53	38	15	8

Throughput: 0.163265
Average Response Time: 6.625000
Average Wait Time: 1.875000
Server Utilization: 0.775510

Case 2 – Multi-server single queue output results

Input Parameters

Total Running Time: 50

Interarrival Min: 3

Interarrival Max: 15

Service MU: 10

Service SIGMA: 1

Total Server: 2

```
Case 2: Multi-server single queuing model simulation results
==================================================================================================================================
Clock  LQ   LS1     LS2   |   SL1     ||   SL2      ||    Queue   |||            Future Event List                     ||||  R  W Nd S1 S2
----------------------------------------------------------------------------------------------------------------------------------
 0     0    0       0     |           ||            ||            |||                           (A,C1,0,0)(E,C0,50,0)  ||||  0  0  0  0  0
 0     0    1       0     | (C1,0)    ||            ||            |||                 (D,C1,10,1)(A,C2,12,0)(E,C0,50,0)  ||||  0  0  0  0  0
10     0    0       0     |           ||            ||            |||                           (A,C2,12,0)(E,C0,50,0)  |||| 10  0  1 10  0
12     0    1       0     | (C2,12)   ||            ||            |||                 (A,C3,15,0)(D,C2,21,1)(E,C0,50,0)  |||| 10  0  1 10  0
15     0    1       1     | (C2,12)   ||  (C3,15)   ||            |||       (A,C4,18,0)(D,C2,21,1)(D,C3,25,2)(E,C0,50,0)  |||| 10  0  1 10  0
18     1    1       1     | (C2,12)   ||  (C3,15)   || (C4,18)    |||       (D,C2,21,1)(A,C5,25,0)(D,C3,25,2)(E,C0,50,0)  |||| 10  0  1 10  0
21     0    1       1     | (C4,18)   ||  (C3,15)   ||            |||       (A,C5,25,0)(D,C3,25,2)(D,C4,30,1)(E,C0,50,0)  |||| 19  9  2 10  0
25     1    1       1     | (C4,18)   ||  (C3,15)   || (C5,25)    |||       (D,C3,25,2)(D,C4,30,1)(A,C6,30,0)(E,C0,50,0)  |||| 19  9  2 10  0
25     0    1       1     | (C4,18)   ||  (C5,25)   ||            |||       (A,C6,30,0)(D,C4,30,1)(D,C5,34,2)(E,C0,50,0)  |||| 29 19  3 10  0
30     1    1       1     | (C4,18)   ||  (C5,25)   || (C6,30)    |||       (D,C4,30,1)(A,C7,33,0)(D,C5,34,2)(E,C0,50,0)  |||| 29 19  3 10  0
30     0    1       1     | (C6,30)   ||  (C5,25)   ||            |||       (A,C7,33,0)(D,C5,34,2)(D,C6,40,1)(E,C0,50,0)  |||| 41 31  4 10  0
33     1    1       1     | (C6,30)   ||  (C5,25)   || (C7,33     |||       (D,C5,34,2)(D,C6,40,1)(A,C8,46,0)(E,C0,50,0)  |||| 41 31  4 10  0
34     0    1       1     | (C6,30)   ||  (C7,33)   ||            |||       (D,C6,40,1)(D,C7,44,2)(A,C8,46,0)(E,C0,50,0)  |||| 50 40  5 10  0
40     0    0       1     |           ||  (C7,33)   ||            |||                 (D,C7,44,2)(A,C8,46,0)(E,C0,50,0)  |||| 60 40  6 20  0
44     0    0       0     |           ||            ||            |||                           (A,C8,46,0)(E,C0,50,0)  |||| 71 41  7 20 10
46     0    1       0     | (C8,46)   ||            ||            |||                 (E,C0,50,0)(D,C8,55,1)(A,C9,58,0)  |||| 71 41  7 20 10

==================================================================================================================================

Throughput: 0.152174
Average Response Time: 10.142857
Average Wait Time: 5.857143
Server Utilization (1): 0.434783
Server Utilization (2): 0.217391
```

Case 3 – Multi-sever multi queue output results

Input Parameters

Total Running Time: 50

Interarrival Min: 3

Interarrival Max: 15

Service MU: 20

Service SIGMA: 1

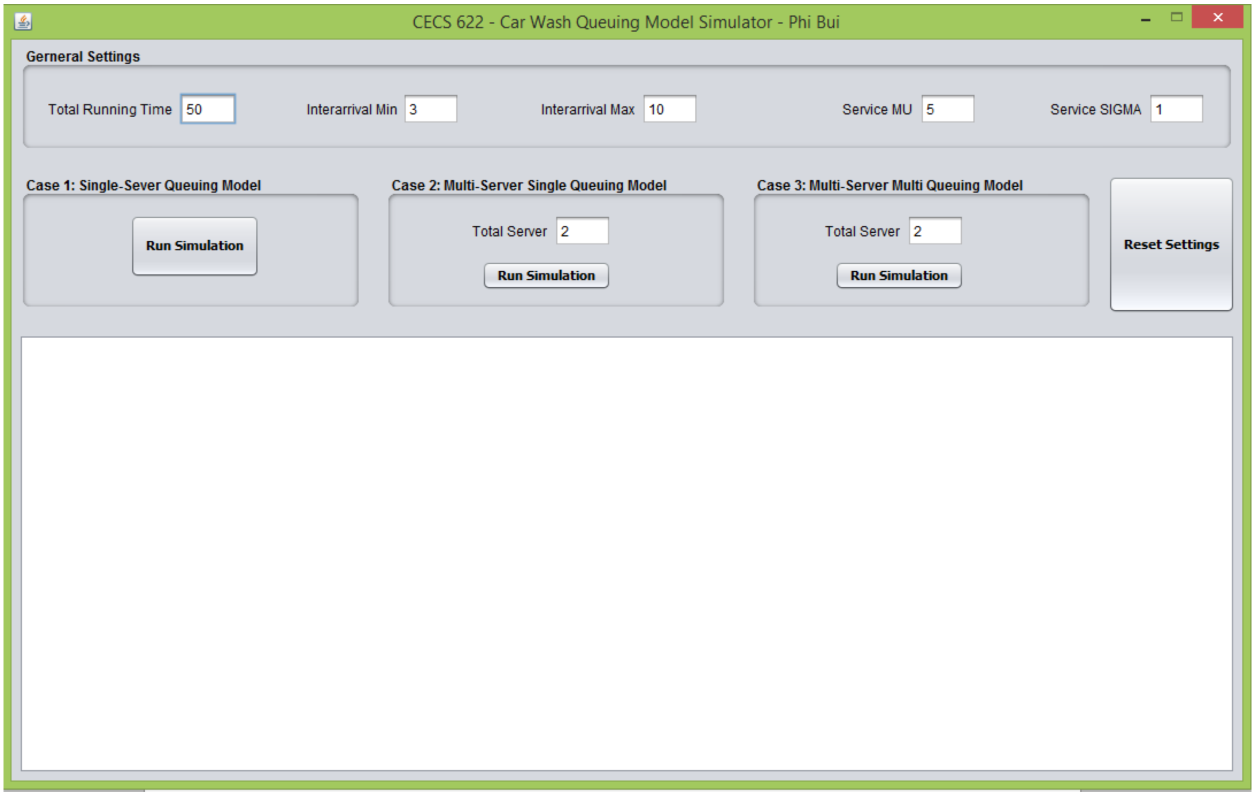
Total Server: 2

```
Case 3: Multi-server multi queuing model simulation results
=====================================================================================================================================================
Clock   LQ1   LQ2   LS1   LS2 |   SL1    |||  SL2    |||  Q1     ||#||  Q2    ||#||             Future Event List                         ||||| R  W  Nd  S1 S2
-----------------------------------------------------------------------------------------------------------------------------------------------------
 0      0     0     0     0   |          |||         |||         ||#||        ||#||                     (A,C1,0,0)(E,C0,50,0)           ||||| 0  0  0   0  0
 0      0     0     1     0   | (C1,0)   |||         |||         ||#||        ||#||           (A,C2,9,0)(D,C1,18,1)(E,C0,50,0)          ||||| 0  0  0   0  0
 9      0     0     1     1   | (C1,0)   ||| (C2,9)  |||         ||#||        ||#||(D,C1,18,1)(A,C3,22,0)(D,C2,29,2)(E,C0,50,0)     ||||| 0  0  0   0  0
18      0     0     0     1   |          ||| (C2,9)  |||         ||#||        ||#||           (A,C3,22,0)(D,C2,29,2)(E,C0,50,0)     ||||| 18 0  1   18 0
22      0     1     0     1   |          ||| (C2,9)  |||         ||#||(C3,22)||#||           (A,C4,25,0)(D,C2,29,2)(E,C0,50,0)     ||||| 18 0  1   18 0
25      0     1     1     1   | (C4,25)  ||| (C2,9)  |||         ||#||(C3,22)||#||(D,C2,29,2)(A,C5,36,0)(D,C4,44,1)(E,C0,50,0)     ||||| 18 0  1   18 0
29      0     0     1     1   | (C4,25)  ||| (C3,22)|||          ||#||        ||#||(A,C5,36,0)(D,C4,44,1)(D,C3,49,2)(E,C0,50,0)     ||||| 38 20 2   18 0
36      1     0     1     1   | (C4,25)  ||| (C3,22)|||(C5,36)   ||#||        ||#||(D,C4,44,1)(A,C6,47,0)(D,C3,49,2)(E,C0,50,0)     ||||| 38 20 2   18 0
44      0     0     1     1   | (C5,36)  ||| (C3,22)|||          ||#||        ||#||(A,C6,47,0)(D,C3,49,2)(E,C0,50,0)(D,C5,63,1)     ||||| 57 39 3   18 0
47      1     0     1     1   | (C5,36)  ||| (C3,22)|||(C6,47)   ||#||        ||#||(D,C3,49,2)(E,C0,50,0)(A,C7,52,0)(D,C5,63,1)     ||||| 57 39 3   18 0
49      1     0     1     0   | (C5,36)  |||         |||(C6,47)   ||#||        ||#||           (E,C0,50,0)(A,C7,52,0)(D,C5,63,1)     ||||| 84 46 4   18 20
=====================================================================================================================================================

Throughput: 0.081633
Average Response Time: 21.000000
Average Wait Time: 11.500000
Server Utilization (1): 0.367347
Server Utilization (2): 0.408163
```

GUI Screenshots

Initial Program Screen



A figure above is the initial GUI when the program is first launched. The top section is the general settings for all three cases (single-server, multi-server single queue, and multi-sever multi queue). The general settings contains the following user input parameters:

- Total Running Time – 50 by default
- Interarrival Min – 3 by default
- Interarrival Max – 10 by default
- Service MU – 5 by default
- Service SIGMA – 1 by default

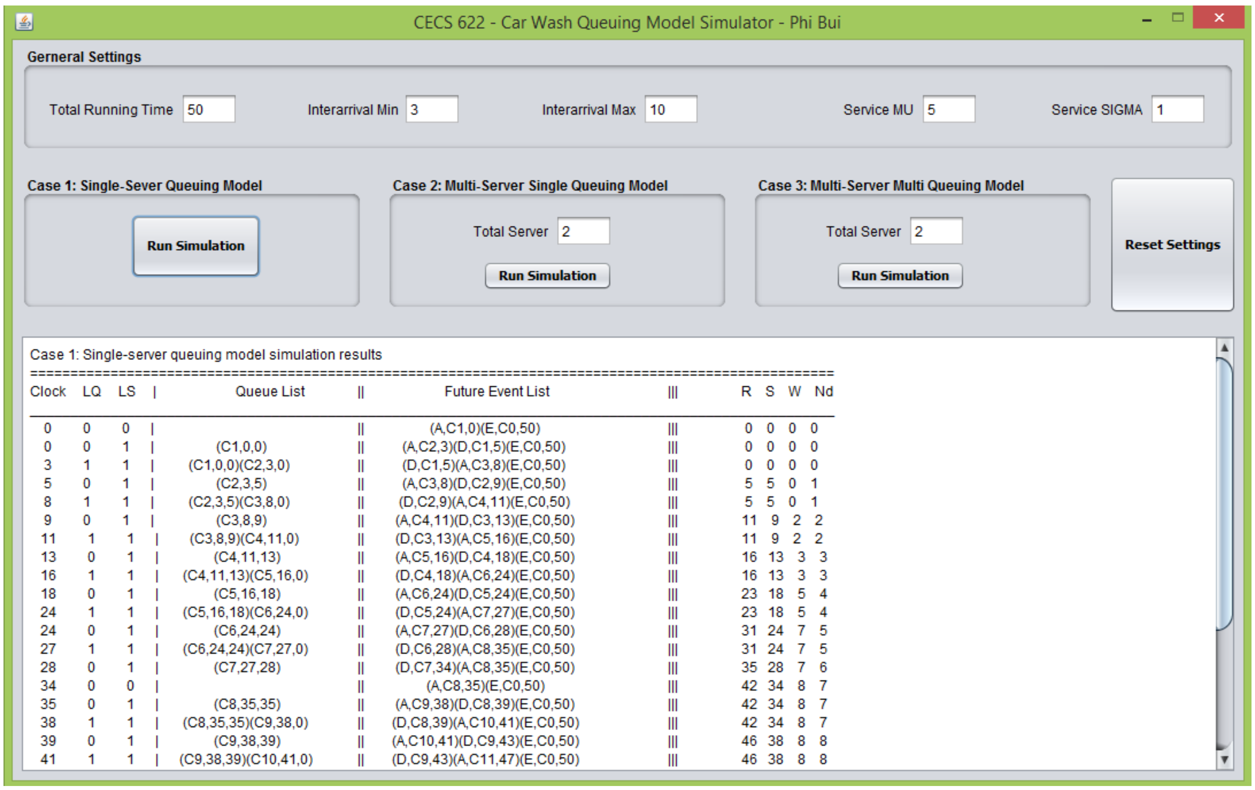
Below general settings section, I separated three different group for case 1, 2, and 3. Within each cases, it contains "Run Simulation" button which allows user to begin the simulation based on the input parameters.

Total server user input setting is additionally for case 2 and 3 which allows user to enter the number of server(s) to be simulated. 2 servers is by default.

Next to case 3 is “Reset Settings” button which allows user to reset the program to default settings and clear the result panel.

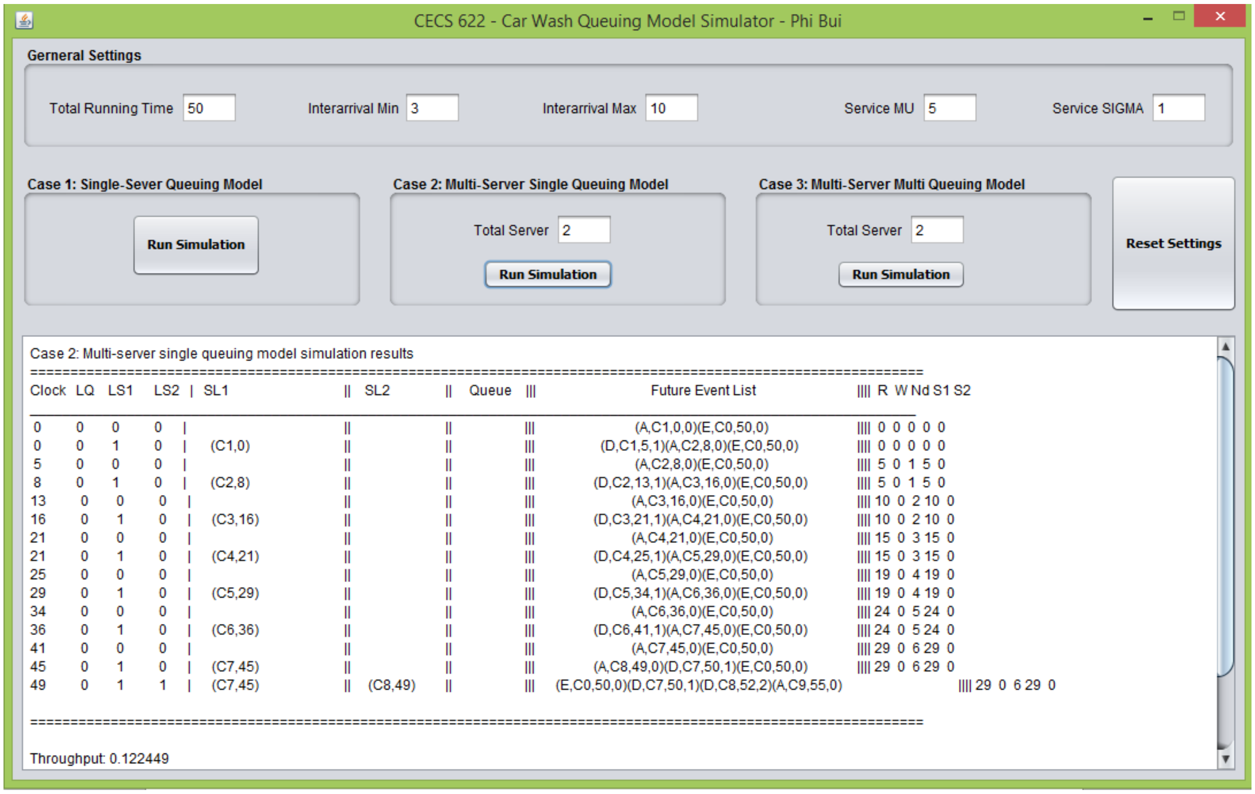
Finally, the white box at the bottom of the program will be used to display the results for the simulation.

Case 1 Simulation Results Screen



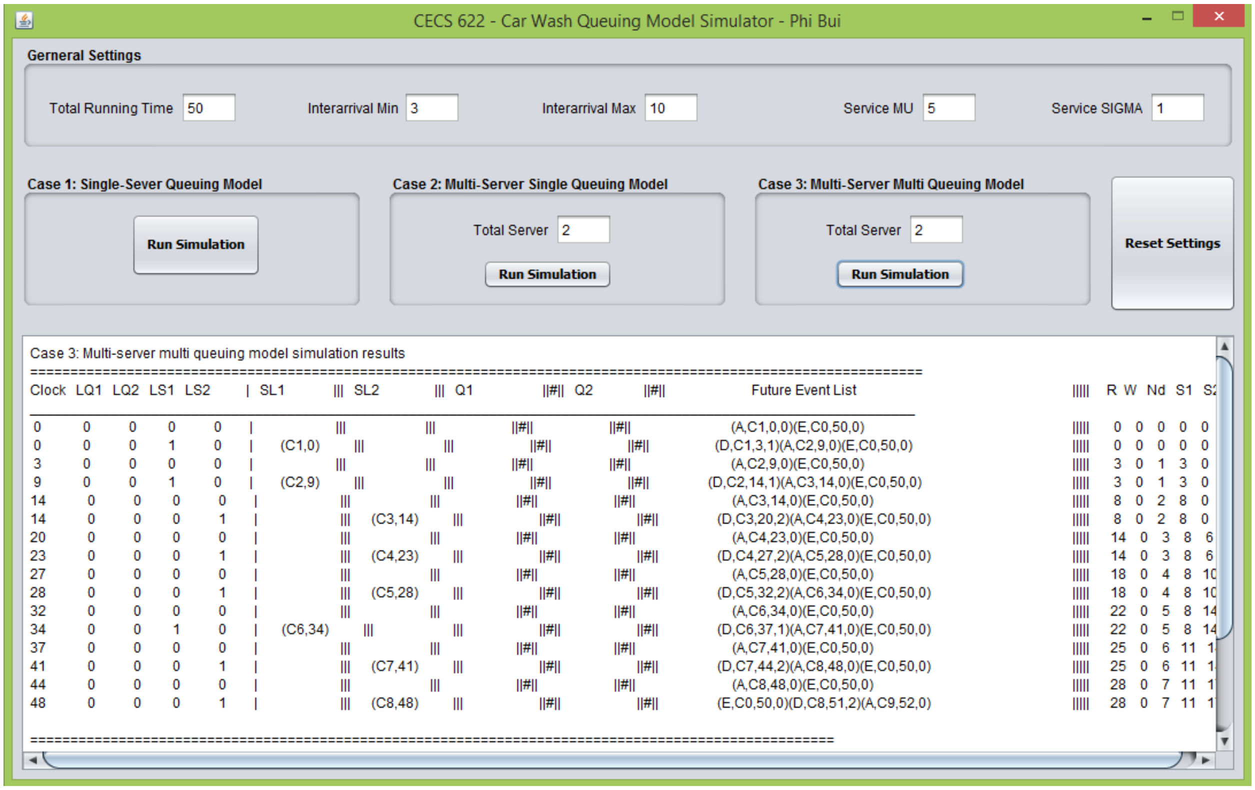
A figure above shown the results of simulation for a single-server. The default input parameters are used for this illustration. Total running time is 50, Interarrival Min is 3, Interarrival is 10, Service MU is 5, and Service SIGMA is 1.

Case 2 Simulation Results Screen



A figure above shown the results of simulation for multi-server single queue. The default input parameters are used for this illustration. Total running time is 50, Interarrival Min is 3, Interarrival is 10, Service MU is 5, Service SIGMA is 1, and total server is 2.

Case 3 Simulation Results Screen



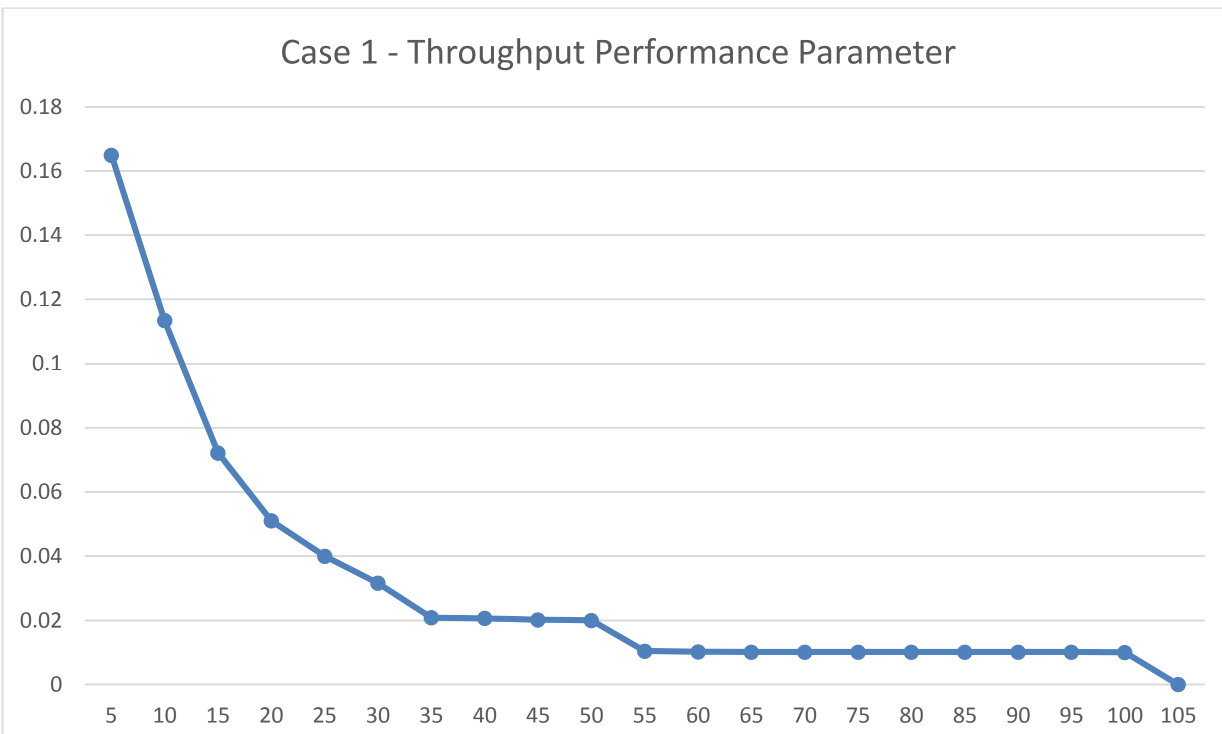
A figure above shown the results of simulation for multi-server multi queue. The default input parameters are used for this illustration. Total running time is 50, Interarrival Min is 3, Interarrival is 10, Service MU is 5, Service SIGMA is 1, and total server is 2.

Performance parameter curves/graphs for queuing case 1, 2 and 3

Input parameters (for all queuing cases):

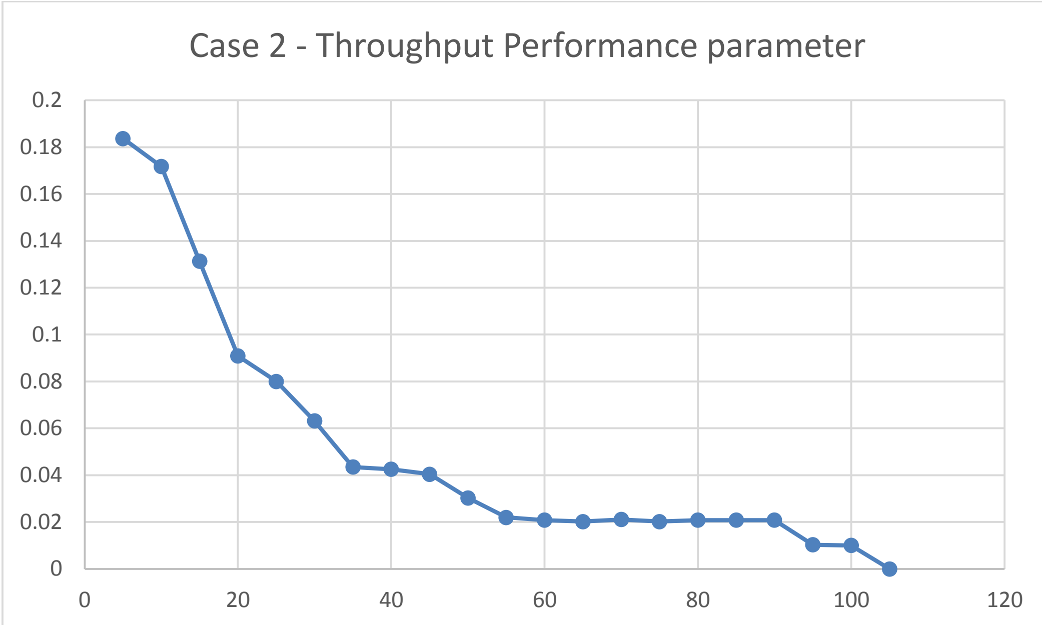
- Total Running Time: 100
- Interarrival Min: 3
- Interarrival Max: 10
- **Service MU: 5 – 105 (increment by 5)**
- Service SIGMA: 1

Case 1 - Throughput performance graph



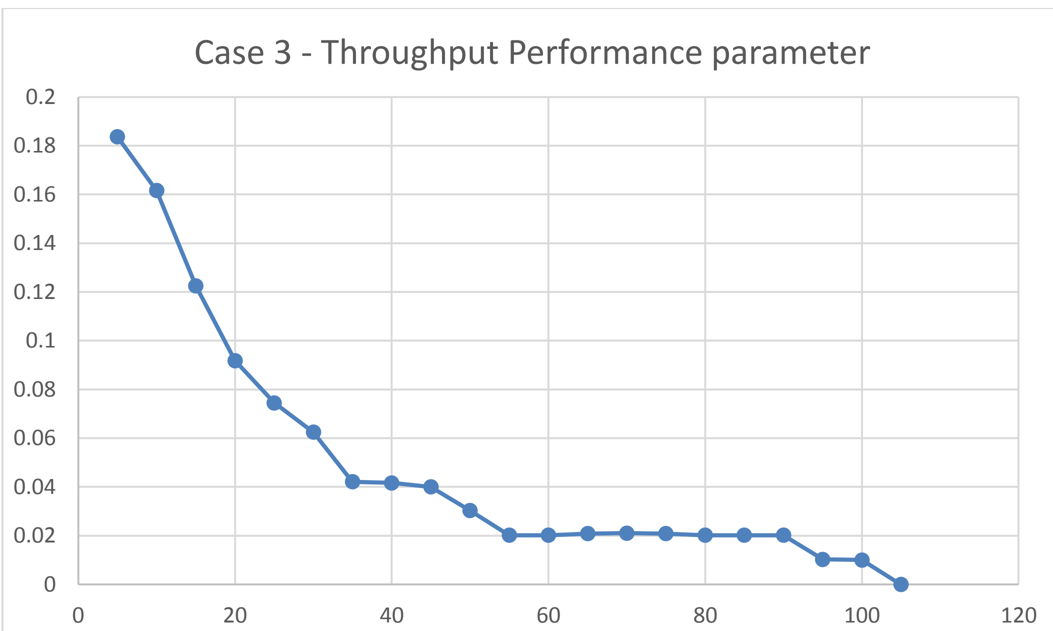
Case 1: Service MU vs Throughput graph

Case 2 - Throughput performance graph



Case 2: Service MU vs Throughput graph

Case 3 - Throughput performance graph



Case 3: Service MU vs Throughput graph

The throughput performance graphs for each of the queuing cases above shown how the throughput change as changing the service mu (or exponential distribution with mean mu). Based on the trend of all three cases, the greater number of service mu, the throughput will becomes smaller and zero out as gets to the maximum number of total running time.

Lastly, to understand all of the parameters, notation, and detail explanations about each columns from the simulation results, they can be found from the book "Discrete-Event System Simulation" written by Banks, Nelson, and Nicol (see at the references section below).

References

[1] J. Banks, J. Carson II, B. Nelson and D. Nicol, Discrete-Event System Simulation, Pearson, 2010.

Source code (*Form Java, Java, JDK 8.0*)

- **Carwashmodel**
  - Main.form
  - Main.java

  - **Class**
    - MultiServerMultiQueueSystem.java
    - MultiServerSingleQueueSystem.java
    - RandomGeneratorHelper.java
    - SingleJobServerSystem.java
    - SingleServerQueueSystem.java