Performance Analysis of a Single Server Queueing System

Md Farhan Ishmam[1] and Md Toshaduc Rahman[2]

[1]Department of Computer Engineering, Islamic University of Technology, Gazipur, Bangladesh
farhanishmam@iut-dhaka.edu

[2]Department of Computer Engineering, Islamic University of Technology, Gazipur, Bangladesh
toshaduc@iut-dhaka.edu

Abstract

Single Server Queueing System is a basic form of a simulation model which can be used to make predictions of a system with a single server serving a single queue. In this paper, we address the performance of this system by making a comparison of the simulated results with the analytical results. We analyze how the output variables differ in the simulation result from the analytical values by graphically representing them. Thereafter, we verify the simulation and estimate the correctness of our simulation results.

Keywords

Single Server Queueing System (SSQS), Simulation, Analytical Result, Arrival Event, Departure Event, Exponential Distribution

1. Introduction

A Single Server Queueing System is the simplest form of simulation modeling system. It consists of a single serving station that can serve only one customer at a time, and a queue of infinite length that will supply customers to the server. However, in practical implementations, the queue is of finite length.

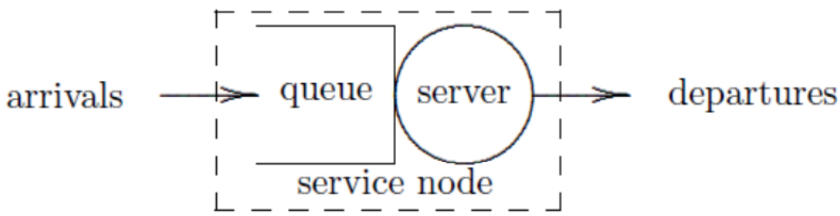
The single server queueing system is essential in conceptually understanding more complex simulation systems such as Multi-Server Queueing System, and Inventory System. The core concept of SSQS revolves around queuing theory which is a set of quantitative mathematical techniques used to construct queuing systems.

In the upcoming section, we will have a look at the description of the SSQS which consists of the diagram of the system, its state variables, events, input, and output variables. Then we will go through the simulation program, its classes, and their relationships. This will be visually presented in the class diagram. Following the class diagram, the flowcharts of some of the major functions of SSQS will be provided. Afterward, we will have a look at the analytical solution by introducing relevant symbols and their corresponding equations. Then, the simulation and analytical solution's data will be presented, and a graphical comparison will be drawn. Finally, we will conclude the graphical comparison of the simulation and analytical results.

2. System Description

The system is analogous to a bank with a single cashier being served by a single queue. The queue follows a FIFO (First In First Out) structure. If a customer enters the system and finds the server busy, then he will join the queue. If he enters the system and finds the server idle, then he will take service. Whenever a customer departs, the first customer from the queue will be given service. If there is no customer in the queue then the server will stay idle. The length of the queue can vary from zero to infinity but since infinite numbers can not be stored in a program, we take the maximum queue length as a very large positive number. The server can provide service to only one customer at a time. Hence, it can have two states only - busy and idle.

The simulation will be performed for a fixed duration. Two types of events are possible - Arrival Event and Departure Event. The Arrival Event deals with an item entering the system and possibly joining the queue. The Departure event deals with an item after taking the service and would thereafter, leave the system. The given simulation program will serve exactly 100 items after which no event will be scheduled. After the simulation, a trace file will be generated which will contain a list of the events, the time of occurrence, and the state of system variables when that event occurred. Another report file will be generated which will consist of the output statistical variables of the simulation.



Single Server Queueing System

The model will take a set of arrival and departure times of the items. The interarrival and inter-departure periods follow the exponential distribution generated by the computer using Inverse Transform Method. The output of the model will be the average number of customers or items in the queue, the average number of customers in the system, the average queueing delay, the average server utilization time, and the average time spent in the system. All the output statistical variables will be generated in the report file for different values of traffic intensity. Traffic intensity is the ratio of arrival and departure rates. The rates will be provided to the system as the mean of inter-arrival and inter-departure times.

3. Simulation Program Description

The program will consist of four major classes - **the event class, the server class, the scheduler class, and the queue class**.

The event class will represent an arrival or departure event. It will have functions to create events and handle them. The server class will consist of the system variables, the queue, and the output statistical variables. Handling an event involves updating the statistical variables and creating changes to the system variables.

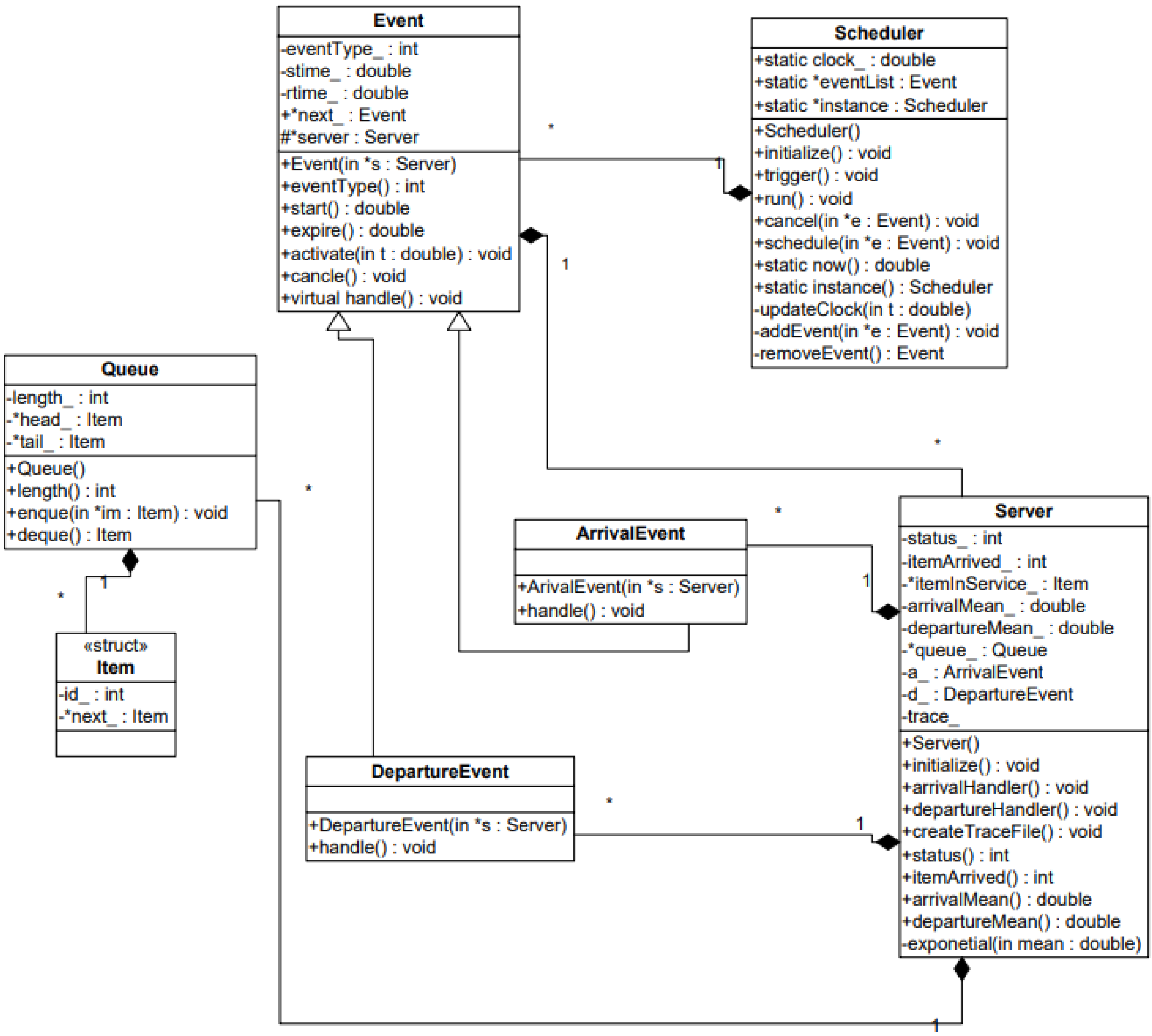
The server mainly consists of the system variables, output statistics, and the queue. The server will initialize all these variables at the beginning of the simulation with appropriate values. It will generate random variates based on the arrival and departure mean, and assign them as the arrival and departure times for an event. Furthermore, this class will have the functions to make changes to system variables and output statistics whenever an event occurs. These functions will be called by the arrival or departure event handler of an event class. The server will also generate a trace file and a report file at the end of the simulation.

The queue class is simply a FIFO queue where each element is represented by a structure called Item. The queue will have basic functions to enqueue, dequeue and show its length.

Finally, we have the scheduler class which mainly consists of the event list and system clock. The scheduler class will have a run function that will remove an event from the event list and execute it. The function will do so until the event list is empty. This class will also schedule new events at the server-generated time. The clock will always be updated to the time of occurrence of a new event.

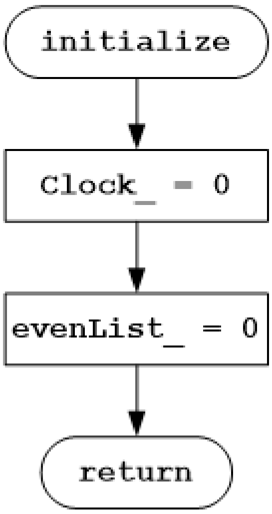
To summarize, the simulation program will take the means of inter-arrival and inter-departure times as inputs, generate exponential random variates with that mean, and provide those numbers as arrival and departure times of the customers to the system. The system will then run, by scheduling events and executing them. Afterward, it will produce a trace file which will contain the record of all the events that occurred in the simulation and a report file which will contain the values of the output statistics.

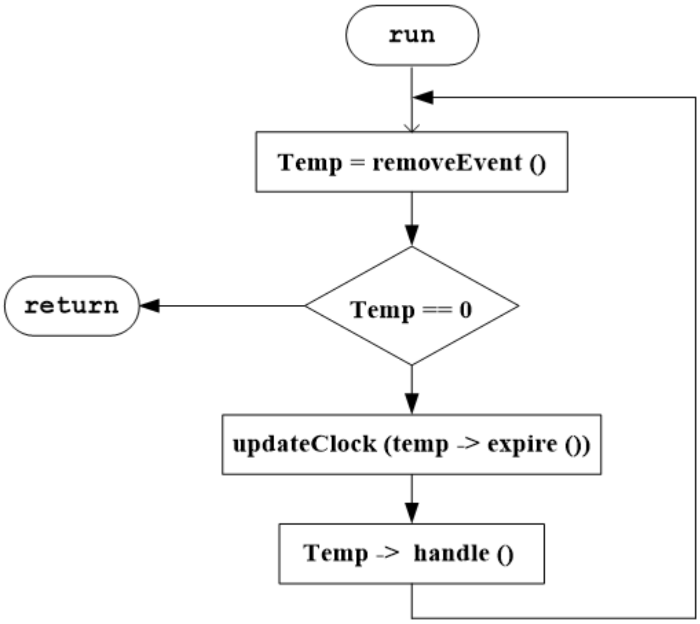
4. Class Diagram

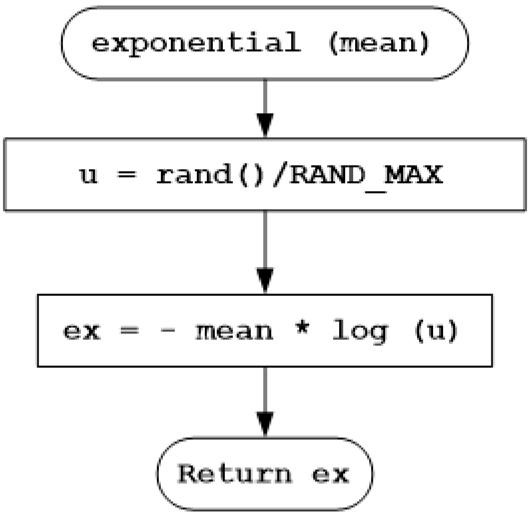


5. Flowcharts

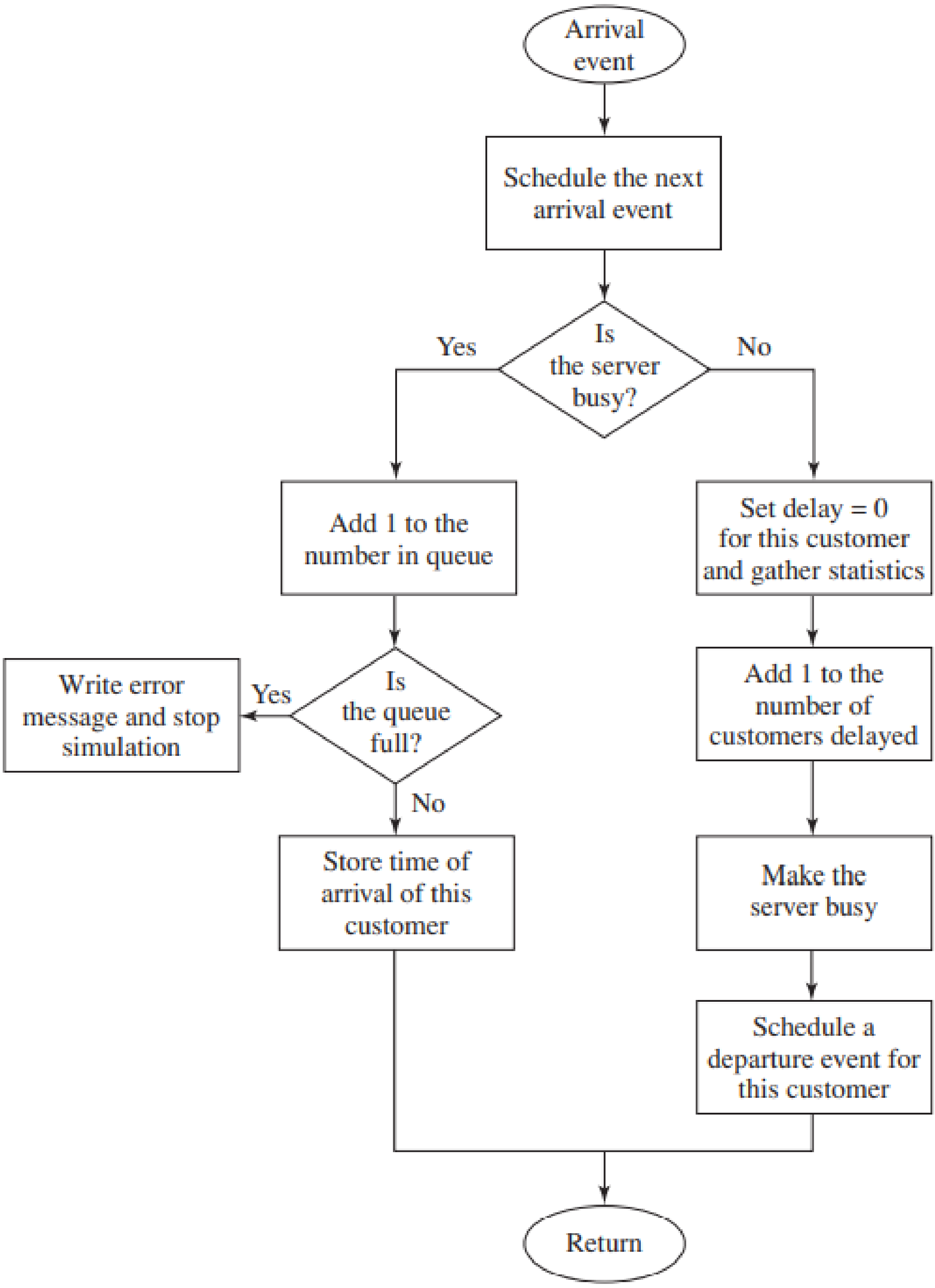
Initialize, Run, and Exponential



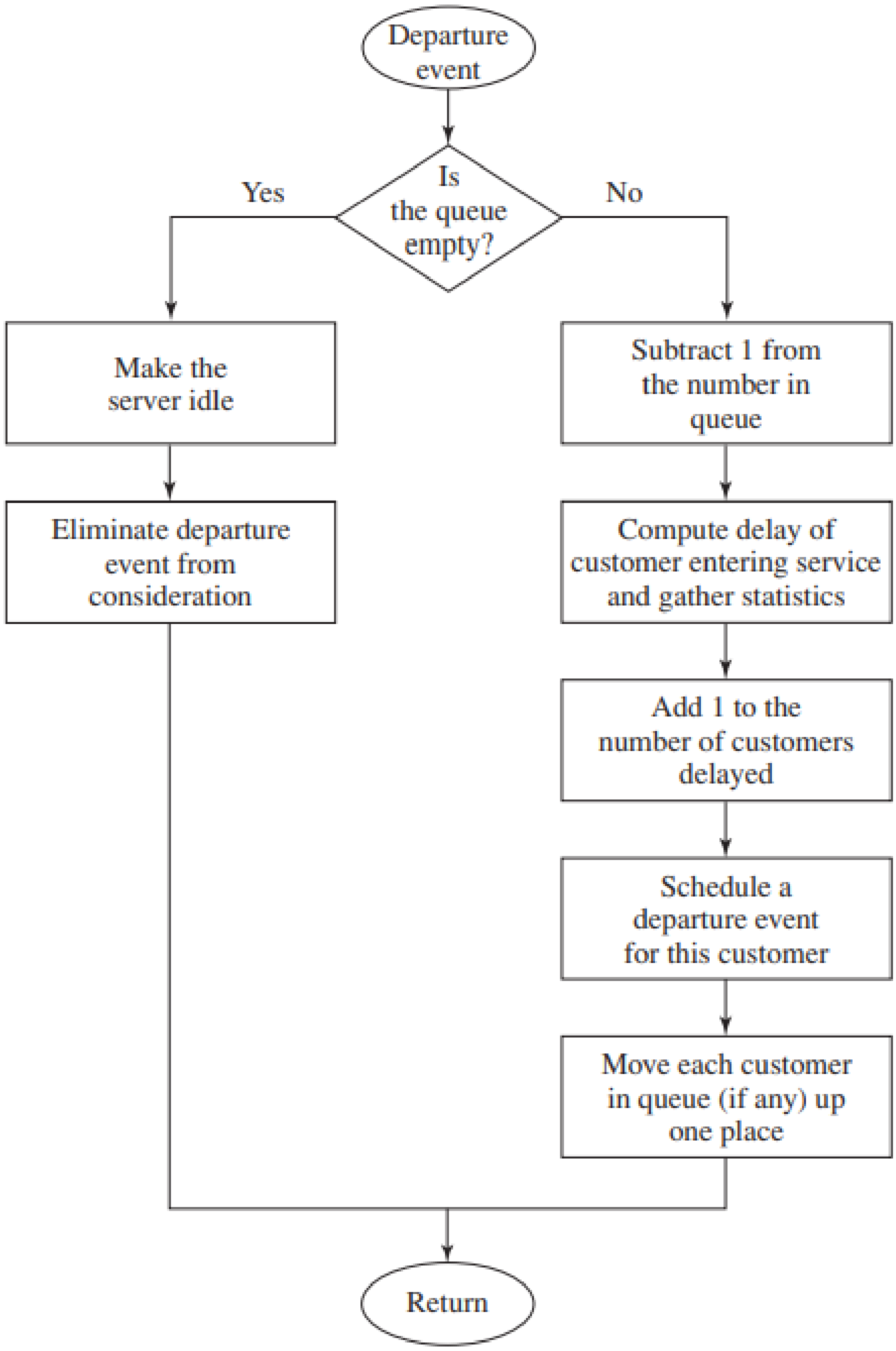




Arrival Event



Department Event



6. Analytical Solution

Following queueing theory, we can analytically derive the values of output statistics of our simulation. For this, we will introduce some new terms and symbols:

λ = the arrival rate of the customers
μ = the departure rate of the customers
$\rho = \lambda/\mu$ = traffic intensity which is the ratio of arrival and departure rate
L_q = time-averaged number of customers in the queue
L_u = time-averaged server utilization
L_s = time-averaged number of customers in the system
W_q = job-averaged queueing delay
W_s = job-averaged system delay
P_n = probability of n customers in the queue

Both the inter-arrival and inter-departure time follows the exponential distribution.

Hence, $\rho = \frac{\lambda}{\mu}$

The probability of the system having 0 customers in the queue is $P_0 = 1 - \rho = 1 - \frac{\lambda}{\mu}$

The probability of the system having n customers in the queue is $P_n = \left(\frac{\lambda}{\mu}\right)^n P_0$

Expected number of customers in the system $L_s = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$

Expected number of customers in the queue $L_q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu(\mu-\lambda)}$

Expected server utilization time $L_u = \rho$

The average waiting time in the system $W_s = \frac{1}{\mu-\lambda}$

The average waiting time in the queue $W_q = \frac{\lambda}{\mu(\mu-\lambda)}$

7. Simulation and Analytical Results

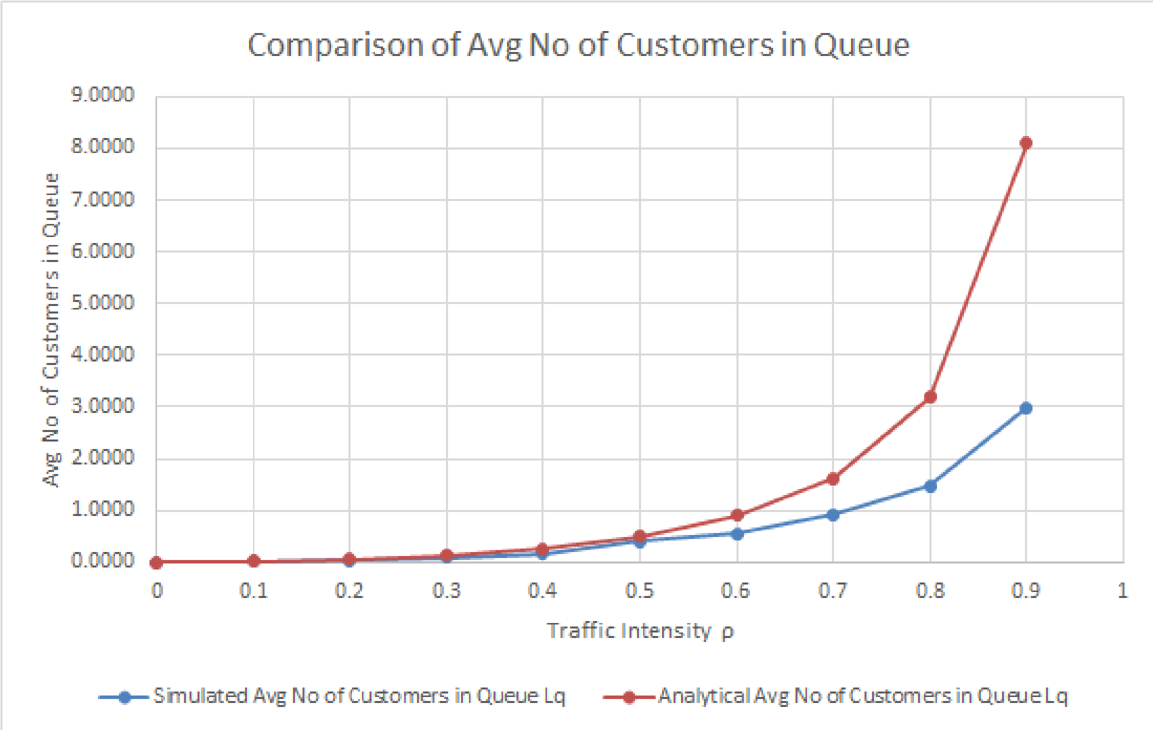
The simulation results were taken by running the simulation 10 times for different values of traffic intensity ρ. The departure rate was fixed to 10 and by changing the arrival rate, we got different values of traffic intensity. The traffic intensity ranges from 0 to 0.9 with a 0.1 increase in each step. The same values of traffic intensity were used to find the analytical results. We used the formulas from the Analytical Solution section to determine the values of the five output statistics.

The data from simulation and analytical results are given below:

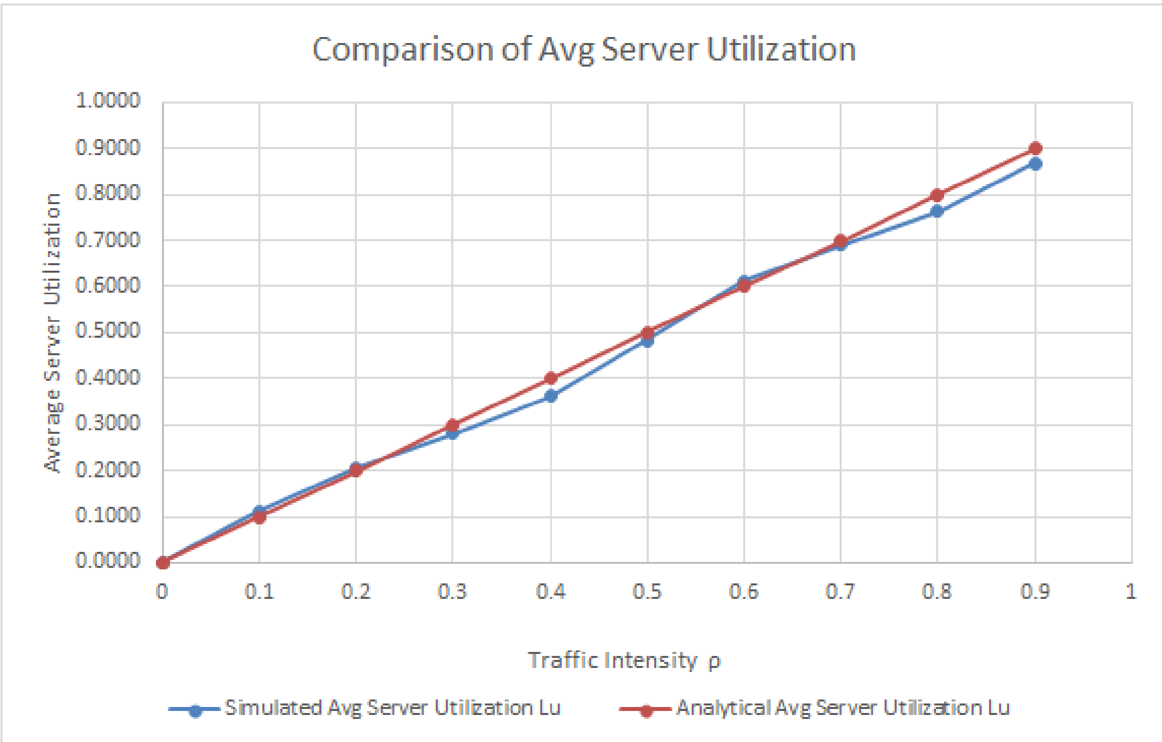
Name	Symbol, Eq	1	2	3	4	5	6	7	8	9	10
Arrival Rate	λ	0	1	2	3	4	5	6	7	8	9
Departure Rate	μ	10	10	10	10	10	10	10	10	10	10
Traffic Intensity	$\rho = \lambda/\mu$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Simulation Results											
Avg No of Customers in Queue	Lq	0	0.0085	0.0432	0.0809	0.158	0.3944	0.5427	0.9852	1.4513	3.0015
Avg Server Utilization	Lu	0	0.9042	0.8571	0.8432	0.8233	0.8404	0.8394	0.8541	0.8622	0.9425
Avg No of Customers in System	Ls	0	0.9127	0.9002	0.9241	0.9814	1.2349	1.3821	1.8393	2.3135	3.9439
Average Queueing Delay	Wq	0	0.0117	0.0227	0.0342	0.0492	0.0897	0.0964	0.1403	0.2033	0.372
Average System Delay	Ws	0	0.1238	0.1301	0.1369	0.1499	0.1948	0.2036	0.2459	0.3084	0.48
Analytical Results											
Avg No of Customers in Queue	Lq	0	0.0111	0.05	0.1286	0.2667	0.5	0.9	1.6333	3.2	8.1
Avg Server Utilization	Lu	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Avg No of Customers in System	Ls	0	0.1111	0.25	0.4286	0.6667	1	1.5	2.3333	4	9
Average Queueing Delay	Wq	0	0.0111	0.025	0.0429	0.0667	0.1	0.15	0.2333	0.4	0.9
Average System Delay	Ws	0.1	0.1111	0.125	0.1429	0.1667	0.2	0.25	0.3333	0.5	1

8. Graphical Comparison of Simulation and Analytical Results

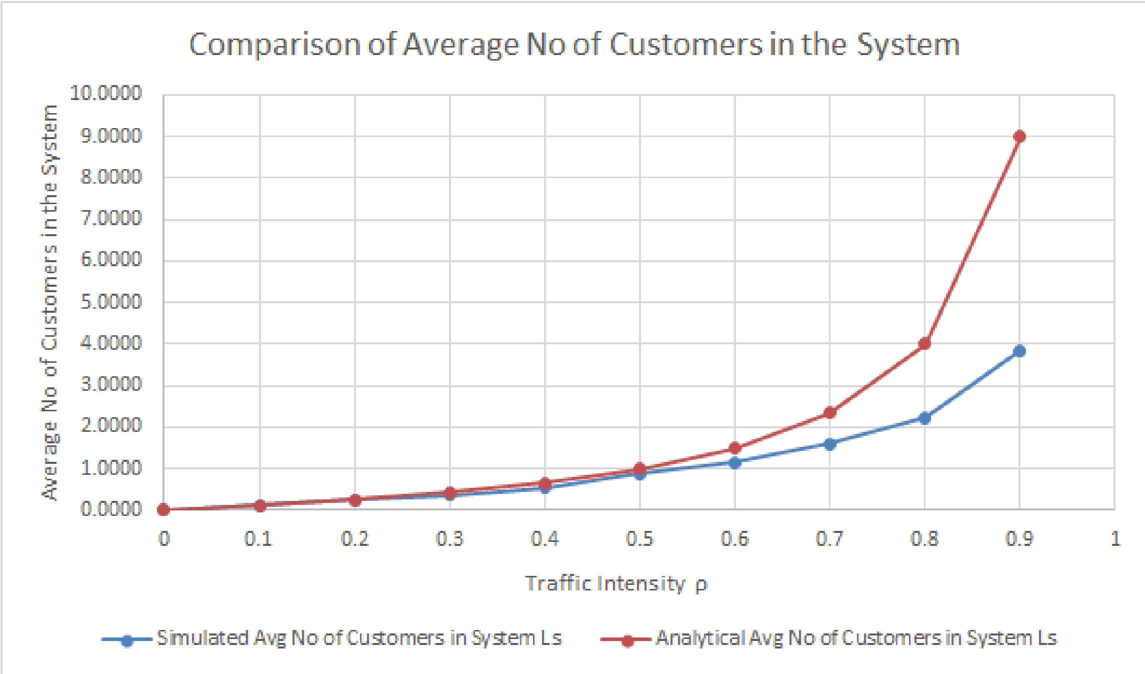
Time Averaged Number of Customers in the Queue: The simulation and analytical results were similar up to 0.5 traffic intensity but the analytical values increased sharply afterward.



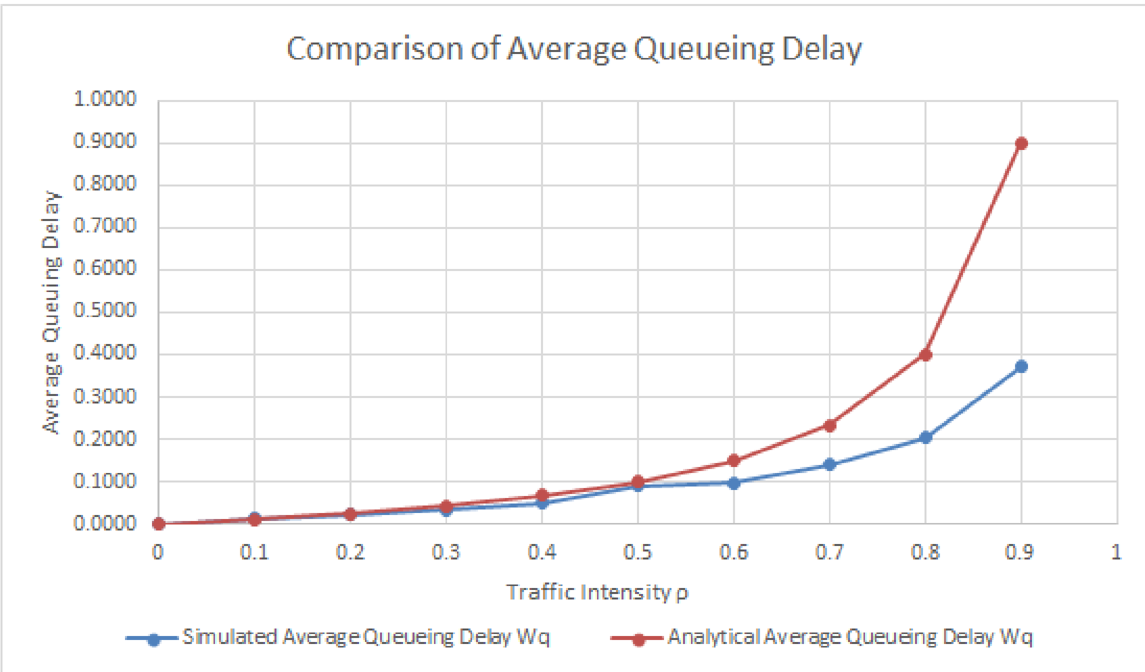
Time Averaged Server Utilization: The server utilization of the simulation and analytical results were similar for the various values of traffic intensity.



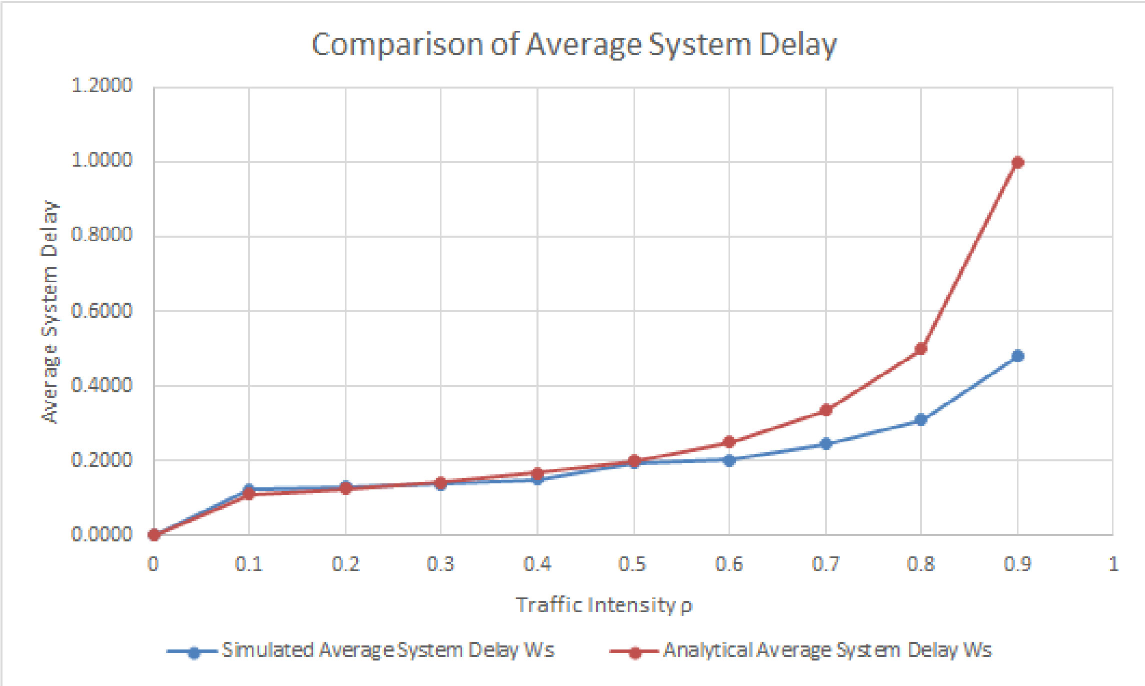
Time Averaged No of Customers in the System: Similar to the comparison of the number of customers in the queue, the simulation and analytical results were similar up to 0.5 traffic intensity and analytical values increase sharply afterward.



Job Averaged Queueing Delay: The queueing delay is also similar for simulation and analytical values up to 0.5 traffic intensity but analytical values increase sharply with higher values of traffic intensity.



Job Averaged System Delay: Just like the previous graphs, the simulation and analytical results were identical up to 0.5 traffic intensity but higher values of traffic intensity produced relatively higher values of delay in the analytical solution.



9. Conclusion:

The performance of the simulation was almost identical to the values we found from the analytical solution. The server utilization graph for both the simulation and analytical solution was identical but the other graphs deviated for traffic intensities higher than 0.5. For such higher values of traffic intensity, the analytical solution was producing higher values of the output statistics. The simulation ran for a finite number of items for a finite time. But, the analytical solution doesn't consider such constraints. This is why the analytical statistical values tend towards infinity as traffic intensity gets close to 1. On the contrary, the simulation produces higher values for traffic intensity = 1 but the value isn't infinity. That's because the simulation ran for 100 customers, so the produced delay for higher values of traffic intensity will be finite. Overall, the simulation succeeded in producing expected results and the performance of the SSQS was satisfactory.

References:

[1] Simulation Modeling and Analysis by Averill Law, 4th Edition

[2] Discrete Event Simulation: A First Course by Lawrence Leemis, December 2004 Revision