NUWC-NPT Technical Document 11,418
14 March 2003

QSIM: A Queueing Theory Model with Various Probability Distribution Functions

Keith M. Sullivan
NUWC Division Newport

Ian Grivell
Defence Science and Technology Organisation, Australia

NAVSEA
NEWPORT
Undersea Warfare Center Division

Naval Undersea Warfare Center Division
Newport, Rhode Island

PREFACE

This document was prepared under NUWC Project No. 760B250.

The technical reviewer was David G. Olson (Code 601).

Reviewed and Approved: 14 March 2003

David S. Medeiros
Head, Undersea Warfare Analysis Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 14 March 2003	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

QSIM: A Queueing Theory Model with Various Probability Distribution Functions

5. FUNDING NUMBERS

6. AUTHOR(S)

Keith M. Sullivan
Ian Grivell

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center Division
1176 Howell Street
Newport, RI 02841-1708

8. PERFORMING ORGANIZATION REPORT NUMBER

TD 11,418

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

QSIM is a queueing theory model developed to interpret the various undertakings involved in military operations. The multiple probability distribution functions (PDFs) for arrival, service, and renege rates not only allow QSIM to analyze a variety of specific warfare tasks (especially those characterized by nonexponential PDFs), but they also ensure that the model is applicable to many other situations. Moreover, the relatively efficient run times permit the user to analyze a range of input values quickly.

14. SUBJECT TERMS

USW Modeling — Queueing Modeling — Warfare Task Analysis
QSIM — Queueing Theory

15. NUMBER OF PAGES
56

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

Preceding Page Blank

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF TABLES (Cont'd)

LIST OF ABBREVIATIONS AND ACRONYMS

ASW	Antisubmarine warfare
ASWC	Antisubmarine Warfare Commander
CDF	Cumulative distribution function
COP	Common operating picture
FCFS	First come, first served
GD	General discipline
GUI	Graphical user interface
LCFS	Last come, first served
PDF	Probability distribution function
PMF	Probability mass function
NCW	Network centric warfare
PR	Priority
RSS	Random selection for service
STL	Standard template library
USW	Undersea warfare

QSIM: A QUEUEING THEORY MODEL WITH VARIOUS PROBABILITY DISTRIBUTION FUNCTIONS

INTRODUCTION

Queueing theory can analyze any system characterized by a "demand for service" (or demand for a limited quantity). This demand for service characterizes many warfare operations, with queueing theory having the capability of quantifying the various tasks that would be involved in such events. In particular, queueing theory could be extremely advantageous for anti-air war (including fighter interception and cruise missile defense), strike missions, self-protection, command and control networks, and antisubmarine warfare (ASW). In all these situations, there clearly are "customers" waiting for a "service."

ASW offers many opportunities for the application of queueing theory. One such example involves the sonar detection and classification process, in which the customers are incoming sonar contacts and the servers are the sonar operators who must detect and classify the various contacts. In another instance, the user would be the command and control network inside a submarine or surface ship. Here, the customers are incoming information and reports, and the servers are the decision-makers who must act on the information (anyone from the captain of the ship on down to a squad commander).

Various probability distribution functions (PDFs) describe a queue. Customers arrive at the queue according to some probability distribution, and the time spent receiving service follows another PDF (usually different than the arrival distribution). Customers can leave the system before completing service (renege), with a probability distribution describing the time spent in the system before leaving. In addition, there may be a limit on the length of the queue. Balking occurs if the queue is full when a customer arrives and the customer does not enter the system.

Most analytical results from queueing theory focus on the exponential PDF. To that end, there are numerous closed-form formulas for a variety of queue types and queueing metrics. However, exponential PDFs do not adequately characterize all warfare events or processes (e.g., a high probability of a short decision time could be unrealistic).[1] Because a limited number of formulas exist for nonexponential PDFs (many are loose upper bounds only, not exact solutions), a simulation is needed to analyze the queues described by this type of function. The remainder of the document describes one such model, which is named QSIM.

QSIM is a discrete-event system simulation coded in C++ that uses a Monte Carlo approach to generate various well-behaved statistics. On a given iteration, QSIM generates and processes customers until the global clock exceeds the user-defined simulation time. To ease expansion and testing, QSIM uses an object-oriented (OO) approach, and to facilitate programming, QSIM represents each customer as a C++ class with the following data members:

- *Arrival time* — indicates when the customer entered the system (based on a user-selected PDF).

- *Service time* — describes when the customer started service (determined during execution).

- *Renege time* — indicates how long to wait in the queue/system until reneging (based on a user-selected PDF).

- *Service length* — describes how long to spend in service (based on a user-selected PDF).

- *Priority level* — is employed when the user selects priority queue (if necessary, based on user-input priorities).

The queue and server(s) are stored in a standard template library (STL) vector data structure. The STL vector provides quick access at the ends, along with the capability for sorting the elements.

QUEUEING THEORY NOTATION

This document uses standard queueing theory shorthand notation.[2] Referred to as Kendall notation, a queueing process is described by a series of slashes and symbols such as A/B/X/Y/Z, where A is the PDF of the arrivals, B is the PDF of the services, X is the number of parallel service channels, Y is the maximum system capacity, and Z is the queue discipline. Table 1 presents several of the common symbols.

Table 1. Queueing Theory Shorthand Notation A/B/X/Y/Z

Notation	Explanation	
Arrival time distribution (A) Service time distribution (B)	M D E_k H_k PH G	Exponential Deterministic Erlang type k (k = 1,2,...) Mixture of exponentials Phase type General
Number of parallel servers (X)	1,2,..., ∞	
Restriction on system capacity (Y)	1,2,..., ∞	
Queue discipline (Z)	FCFS LCFS RSS PR GD	First come, first served Last come, first served Random selection for service Priority General discipline

In practice, only the first three symbols (A, B, and X) are typically in use. For this particular study, the FCFS queue discipline is the only one employed (assumed when Z is not present), and the system capacity symbol Y is dropped when $Y = \infty$.

In a sample notation, M/M/2/5 describes a queueing system with exponential input, exponential service, two servers, a system capacity of five (maximum queue length of three), and an FCFS queue discipline. An extension of the Kendall notation handles reneging. For example, A/B/X/Y/Z + C describes a queueing process with reneging, where C is the PDF describing the reneging (C can assume the same values as A and B).

RELATION TO PAST WORK

In the Undersea Warfare (USW) Analysis Department at the Naval Undersea Warfare Center (NUWC) Division in Newport, RI, earlier work with queueing theory focused on analytical results from Ancker and Gafarian.[3] The results in reference 3 are particularly interesting because of the derivation of closed-form formulas for various queueing metrics. However, there are certain limitations, the most significant of which is that the formulas are valid only for exponential distribution functions regarding arrival, service, and renege rates. Formulas for general probability distributions are nonexistent, and thus a simulation is required. Another drawback of the Ancker and Gafarian formulas is their use of cumbersome special functions (e.g., incomplete gamma and beta functions) that require a computer or complex table calculations. Finally, Ancker and Gafarian provide only a short list of metrics, which is often too restrictive to be useful. Table 2 compares their metrics to those that can be computed by QSIM.

Table 2. Queueing Theory Metrics Derived by Ancker and Gafarian Compared to Those Computed by QSIM

Queueing Theory Metric	Ancker and Gafarian	QSIM
Probability of Acquiring Service	X	X
Loss Rate	X	X
Waiting Time in the System/Queue	X	X
Probability of Balking		X
Probability of Reneging		X
Number of Arrivals		X
Number of Services		X
Number of Reneges		X
Cumulative Distribution of Waiting Time (System/Queue)		X

APPLICATION AND INTERPRETATION OF QUEUEING TERMINOLOGY

The relationships between queueing theory terminology and warfare terminology are not necessarily obvious, with table 3 showing one of the many relationships that is possible. It should be noted here that other warfare areas are similarly connected.

Table 3. One of Many Possible Relationships Between Queueing Terminology and Sonar Terminology

Queueing Terminology	Sonar Terminology
Queue	Sonar workstations/watch stations
Server	Sonar operator
Arrival Rate	Rate at which contacts come into detection range
Service Rate	Rate at which contacts are detected/classified by the operator
Renege Rate	Rate at which contacts leave detection range before detection/classification
Balk	Overload of sonar operator resulting in a contact not being seen on the scope

QSIM INPUTS

There are several inputs into QSIM, all of which are unit independent, with the user responsible for ensuring that the units are consistent. Required QSIM inputs are listed below:

- Arrival and service PDFs, along with their associated parameters;
- Number of servers;
- Queue length (can be infinite);
- Simulation time (maximum length of time per iteration);* and
- Number of iterations.

*To ensure accurate results, the simulation time should be greater than or equal to 200,000 and the number of iterations greater than or equal to 1,000. These values, determined through trial and error, were chosen to average out the initial transient period.

The optional QSIM inputs are as follows:

- Renege PDF and associated parameters;
- The choice of reneging out of the queue only or reneging out of both the queue and service;
- Priorities for a priority queue, which are limited to four in the GUI, which is the graphical user interface (with a file, the number of priorities is unlimited);
- File name describing heterogeneous servers (PDF and associated parameters of each server);
- Upper limits on generated PDF of waiting time in the queue and waiting time in the system (default is 100 time units); and
- Threshold value for $P(T > t)$ (computation of tails of generated PDFs) for the waiting time in the queue and the waiting time in the system.

QSIM OUTPUTS

The steady-state QSIM outputs are listed next:

- Mean time in the queue with 95% confidence interval;
- Mean time in the system with 95% confidence interval;
- Probability of acquiring service, balking, and reneging;
- Loss rate;
- Number of arrivals, services, balks, and reneges over all iterations;
- Maximum length that the queue reached during simulation;
- If applicable, $P(T > t)$ for the waiting time in the queue and waiting time in the system;
- If applicable, an Excel chart and graph containing all the above, plus PDFs of waiting time in the queue and waiting time in the system and the user-selected PDFs with associated parameters.

ALGORITHMS IN QSIM

The procedural steps for the primary algorithms involved in the simulation of a QSIM queue are presented in the following subsections.

MAIN ALGORITHM

Step 1: If the user selects reneging, then call the Renege Algorithm.

Step 2: Handle the current event.

Step 2(a): If the current event is an arrival, call the Arrival Algorithm.

Step 2(b): If the current event is a service, call the Service Algorithm.

Step 3: Call the Global Clock Algorithm.

Step 4: If the user selects a priority queue, then order the queue according to priorities.

Step 5: Order the service queue by time remaining to complete service.

Step 6: If the global clock exceeds the simulation time, then compute statistics and proceed to the next iteration.

Step 7: If the number of iterations is less than the user-input maximum number of iterations, then return to step 1. Otherwise, display the results to the user.

ARRIVAL ALGORITHM

Step 1: Increase the arrival count by one.

Step 2: Check to determine if there is space in the queue for another customer. If not, increase the balk count by one and return.

Step 3: Generate a new customer based on user-input parameters and PDFs.

Step 4: Based on the user-selected distributions, determine how long the customer should wait in service, and if the user selected reneging, determine how long the customer should wait until reneging (both are draws from supplied probability distributions and associated parameters).

Step 5: If the user selected a priority queue, then determine the priority of the customer using the Priority Algorithm.

Step 6: Place the new customer at the end of the queue.

Step 7: If a server is available, call the Service Algorithm.

SERVICE ALGORITHM

Step 1: If no customer is currently in service, place the head of the queue into service and return.

Step 2: Compute how long the head of the service queue was in service and update the histogram of waiting time in the system. Do the same for the waiting time in the queue.

Step 3: Increase the service count by one.

Step 4: Delete the front of the service queue.

Step 5: If there are customers in the queue, place the head of the queue into service.

RENEGE ALGORITHM

Step 1: If the queue is empty, return.

Step 2: Scan the entire queue, recording which customers have been in the queue longer than their renege time.

Step 3: Delete any customers from the queue who have been in the queue longer than the computed renege time and increase the renege count by one for every customer deleted.

Step 4: If the user selected reneging from service, then repeat steps 1-3 for the service queue.

PRIORITY QUEUE ALGORITHM

Step 1: Determine if the user-input priorities are from a dialog box or from a file.

Step 2: If from a dialog box, then use the following format to save the priorities into an array:

x[1] = p1
x[2] = p1 + p2
x[3] = p1 + p2 + p3
x[4] = p1 + p2 + p3 + p4
Go to step 5.

Step 3: If the user-input priorities are from a file, then open the file.

Step 4: While reading the file one line at a time, compute a running total of the priorities and store this running total after each line is read.

Step 5: Generate a uniform (0,1) U using the RANROT W routine (described later).

Step 6: For some i, determine $x[i] < U < x[i + 1]$ and then assign the customer priority i.

Step 7: Once QSIM is running, order the queue according to priorities on every time step.

ALGORITHM FOR ADVANCING THE GLOBAL CLOCK AND DETERMINING THE NEXT EVENT

Step 1: Determine when the arrival will occur and when the next service will occur.

Step 2: Set the next event to the smaller of the two times from step 1.

Step 3: Advance the global clock by the smaller of the two times from step 1.

HETEROGENEOUS SERVER ALGORITHM

Step 1: Read the input file and create an array containing the PDF and associated parameters for each server.

Step 2: To assign a server to a customer, randomly assign one of the free servers.

Step 3: Compute how long the customer should stay in service based on the parameters of the assigned server.

PROBABILITY DISTRIBUTION FUNCTIONS

There are eight PDFs presently available in QSIM:

- Exponential
- Poisson
- Log normal
- Inverse Gaussian
- Gamma
- Erlang
- Weibull
- Uniform

Appendix A provides the PDF, the cumulative distribution function (CDF), parameter descriptions, and, as shown in table 4 below, a typical interpretation of PDFs in the sonar world.

Table 4. Typical Interpretation of PDFs in Sonar Applications

Distribution Function	Typical Interpretation in Sonar Applications
Exponential	Time until first sonar contact
Poisson	Number of sonar contacts in a given time
Log Normal	Clutter density in signal processing
Inverse Gaussian	Time delays in ASW processes characterized by random walk with positive drift
Gamma	Waiting times between each sonar contact in a given time period
Erlang	Service times in an *N* server queue with homogeneous server times
Weibull	Time until failure of first sonar unit
Uniform	When arrival or service times are known but nothing is known about the distributions

QSIM can parameterize all the probability distributions, thus allowing the user to analyze ranges of input values. The user chooses the range of the parameters for the distribution and the step size. At this point in its evolution, QSIM can only parameterize one of the arrival, service, or renege events at a time.

In addition to the eight PDFs in table 4, the arrival distribution has two additional options: a deterministic arrival rate and a multiple arrival rate. The deterministic option allows the user to provide a file containing the times when arrivals will occur (e.g., arrivals occur at times 1, 3, 5, 7, 9, 11). Such deterministic arrivals arise in a variety of situations, as can be seen from an analysis of the logs that provide the data relating to the detection of new contacts. These logs are typically used to study the different configurations of sonar operators and sensors. The multiple arrival rates option allows the arrival of various types of customers with different PDFs and parameters. For instance, benign and hostile sonar contacts arrive at different rates and follow different PDFs. QSIM picks the arrival of the next customer by computing the time until the next arrival for each of the distributions and chooses the smallest of these times.

COMPUTATION OF STATISTICS

A Monte Carlo approach is used to compute various queueing metrics. Over all iterations, QSIM stores running totals of the number of arrivals, completed services, reneges, and balks. Equation (1) lists formulas for the various probabilities calculated with these counts:

$$
\begin{aligned}
P(Acquiring\ Service) &= \frac{\#\,\text{completed services}}{\#\,\text{arrivals}}, \\
P(renege) &= \frac{\#\,\text{reneges}}{\#\,\text{arrivals}}, \\
P(balk) &= \frac{\#\,\text{balks}}{\#\,\text{arrivals}}, \\
Loss\ Rate &= \frac{\#\,\text{balk} + \#\,\text{renege}}{\text{simulation time } * \text{ number of iterations}}.
\end{aligned}
\tag{1}
$$

To compute the average waiting time in the queue and the average waiting time in the system, QSIM saves the time for every arrival. The following tasks are performed after the customer completes service:

- Compute the time in the queue and in the system.
- Keep a running total for both waiting times.
- Compute the averages upon completion of the given iteration.
- Compute the averages over all the iterations at program completion.

The result is the sample mean for the waiting time in the queue and the sample mean for the waiting time in the system (average waiting times exclude customers who renege, but include customers who enter service immediately). This is the point where the long simulation time and large number of iterations are important. If the sample mean contains a disproportionate number of values from the initial transient period, then the steady-state averages will not be correct. The large simulation time helps to average out the times from the transient period.

Equations (2) and (3) show how QSIM computes confidence intervals. First, from the sample means above, the standard deviation is computed as

$$\sigma = \sqrt{\sum_{i=1}^{\#\,\text{iterations-1}} \left(x_i - \bar{x}\right)^2}\ . \tag{2}$$

QSIM then computes the 95% confidence intervals based on the normal distribution as

$$\bar{x} \pm \frac{1.96 * \sigma}{\sqrt{\#\,\text{iterations}}}\ . \tag{3}$$

When a customer completes service, QSIM updates a histogram of the waiting time in the queue and the waiting time in the system. The intervals for the histogram are one time unit apart and initially bounded above by 100 (this upper bound can be changed). As each customer completes service, QSIM updates the number of occurrences for every interval. Upon program completion, the number of occurrences in each interval is divided by the total number of occurrences to obtain the final PDF.

GENERATION AND VALIDATION OF RANDOM VARIATES

To create arrival, service, and renege times, QSIM requires a random variate based on user input. QSIM generates this random variate by drawing from the distribution (see appendix B).

All the routines from appendix B depend on the generation of a uniform (0,1) variate, which can be accomplished by the RANROT W algorithm. Developed by Agner Fog,[4] RANROT W has been successfully run through the DIEHARD suite of tests developed by George Marsaglia.[5] Other tests, conducted at the 0.05 significance level, are listed below:

- Kolmogorov-Smirnov — for uniformity;
- Chi-squared — for uniformity;
- Runs up and runs down — for independence;
- Runs above and below mean — for independence.

For every test, 100 streams of $2^{23} = 8{,}388{,}608$ numbers were generated with different seeds (time in milliseconds) used in each stream. Four runs failed the Kolmogorov-Smirnov test, and one run failed the chi-squared test. Seven runs failed the runs-up and runs-down test, and six failed the runs above and below mean test. On average, five runs should fail during each of the four tests. Based on these results, it was determined that RANROT W passed the tests for independence and uniformity.

ADVANCED PROPERTIES

Based on user input, QSIM can compute several advanced properties. The first computation involves limiting the PDFs of the waiting time in the queue and the waiting time in the system. If users know to truncate the PDFs *a priori*, they can prompt the program as to where the cutoff should be. For the waiting time in the queue and the waiting time in the system, QSIM can compute $P(T > t)$ for any t (i.e., the tail end of the PDF).

The length of the warmup period (usually set to zero) is used to reduce the effect of the initial transient period and can be changed by the user.

Finally, the user can set the seed that initializes the RANROT W routine. If this seed is left at zero, then QSIM uses the time in milliseconds as the initial seed.

QSIM VERIFICATION

The metrics for verification include waiting time in the queue, waiting time in the system, and the probability of acquisition. Results from QSIM were compared to closed-form analytical formulas (see appendix C for formula derivation), with the following inputs used in each run:

- Exponential PDF for arrival, service, and, if necessary, renege;
- Simulation time of 200,000;
- 2,000 iterations; and
- Infinite queue length.

Tables 5 through 8 summarize the comparison of QSIM to the analytical formulas, and appendix D provides detailed verification results.

The slight differences between the values calculated by QSIM and those calculated by the formulas are due to round-off errors and the inclusion of times from the transient period. It is speculated that increasing both the simulation time and the number of iterations will improve accuracy, with the only effect being an increased run time.

Because of the object-oriented nature of the program, the verification results show the validity of QSIM independent of the arrival, service, and renege PDFs. Thus, the logic of moving customers through the system is correct, as is the computation of statistics.

Table 5. Comparison of QSIM and M/M/1 Queue Without Reneging

Arrival Rate (per hour)	Service Rate (per hour)	Average Waiting Time in the System (minutes)	
		QSIM	Formula
4	8	15.00	15
5	6	59.90	60
5	7.5	24.00	24

Table 6. Comparison of QSIM and M/M/1 Queue with Renege Rate of 2 per Hour

Arrival Rate (per hour)	Service Rate (per hour)	Average Waiting Time in the Queue (minutes)		Probability of Acquisition	
		QSIM	Formula	QSIM	Formula
4	8	2.96	2.96	0.883767	0.883799
5	6	6.05	6.06	0.768012	0.768097
5	7.5	4.31	4.32	0.833837	0.833855

*Table 7. Comparison of QSIM and M/M/k Queue Without Reneging**

Arrival Rate (per hour)	Service Rate (per hour)	Number of servers	Average Waiting Time in the System (minutes)	
			QSIM	Formula
4	8	2	8.00	8.02
4	8	3	7.54	7.54
8	4	3	21.64	21.66
6	3	3	28.84	28.9
100	4	50	14.99	15

*The last line shows QSIM can handle large numbers of servers and rapid arrival rates.

Table 8. Comparison of QSIM and M/M/k Queue with Renege Rate of 2 per Hour

Arrival Rate (per hour)	Service Rate (per hour)	Number of Servers	Average Waiting Time in the Queue (minutes)		Probability of Acquisition	
			QSIM	Formula	QSIM	Formula
4	8	2	0.339	0.340148	0.987110	0.987120
4	8	3	0.0359	0.036064	0.998687	0.998685
8	4	3	2.079	2.081446	0.919322	0.919322
6	3	3	2.287	2.287861	0.907835	0.907787

QSIM EXAMPLE

The following example shows how QSIM works and, more specifically, how its output is used to analyze a warfare task. Although this particular illustration is not a complex one, it should be noted that QSIM is capable of addressing problems that are far more complicated. To begin, the following assumptions regarding the example are made:

- A blue ASW force is searching for a red submarine in a cluttered environment.
- The Antisubmarine Warfare Commander (ASWC) decides to search with helicopters only.
- Each helicopter searches with a dipping sonar.
- Multiple helicopters are operating in a network centric warfare (NCW) environment (i.e., each helicopter shares a real-time common operating picture (COP)).

The question to be answered is how much improvement is gained (or lost) by having multiple helicopters search for the red submarine. The two metrics examined are the probability of acquiring service (i.e., the probability that a given sonar contact is detected and classified) and the mean waiting time in the queue (i.e., how long the contact was on the sonar scope before the sonar operator began the classification process). This example addresses neither the question of correct classification nor the question of contact prioritization based on target behavior and/or operator experience.

Each helicopter is a server while incoming sonar contacts are the customers. To examine how the helicopters fare against varying threat levels, the arrival rate (rate of incoming sonar contacts) is parameterized from 0.25/hour to 5/hour (1 every 4 hours to one every 12 minutes) in 0.25-hour increments, with an exponential PDF. To account for the red submarine possibly moving out of the detection range before classification, a renege rate of 0.33/hour was used with an exponential PDF (reneging occurs only from the queue). It is assumed that all helicopters are equal and that each has a service rate (rate at which sonar operator can classify contacts) of 0.5/hour, with an exponential PDF. Each helicopter can handle a finite number of contacts, such that the maximum number of contacts will vary as a function of the number of helicopters. However, for simplicity, it was assumed that the system of helicopters can handle only 20 contacts at a time (i.e., the queue length is 20).

The other inputs into QSIM are the simulation time and the number of iterations. In this case, the simulation time is 100,000 with 1,000 iterations. To generate the graphs shown in figures 1 and 2, three runs are required (one each for 1, 3, and 5 servers). Merging the data from all three runs creates the single data set seen in the plots.

The figures illustrate the effect of NCW on the probability of acquisition. As the number of searching helicopters increases, the number of contacts seen by each helicopter becomes less; thus, the effective arrival rate decreases. Hence, the mean time a contact spends on the scope before detection and classification decreases, and the probability that a helicopter will detect and classify a given contact increases (note that false contacts and misclassification are not addressed). In addition, NCW allows assets to share information, thus avoiding investigation of contacts that are not of interest. Consequently, the arrival rate of new contacts to each helicopter is reduced (note the NCW effect as shown by the arrows on the two figures).

SUMMARY AND RECOMMENDATIONS

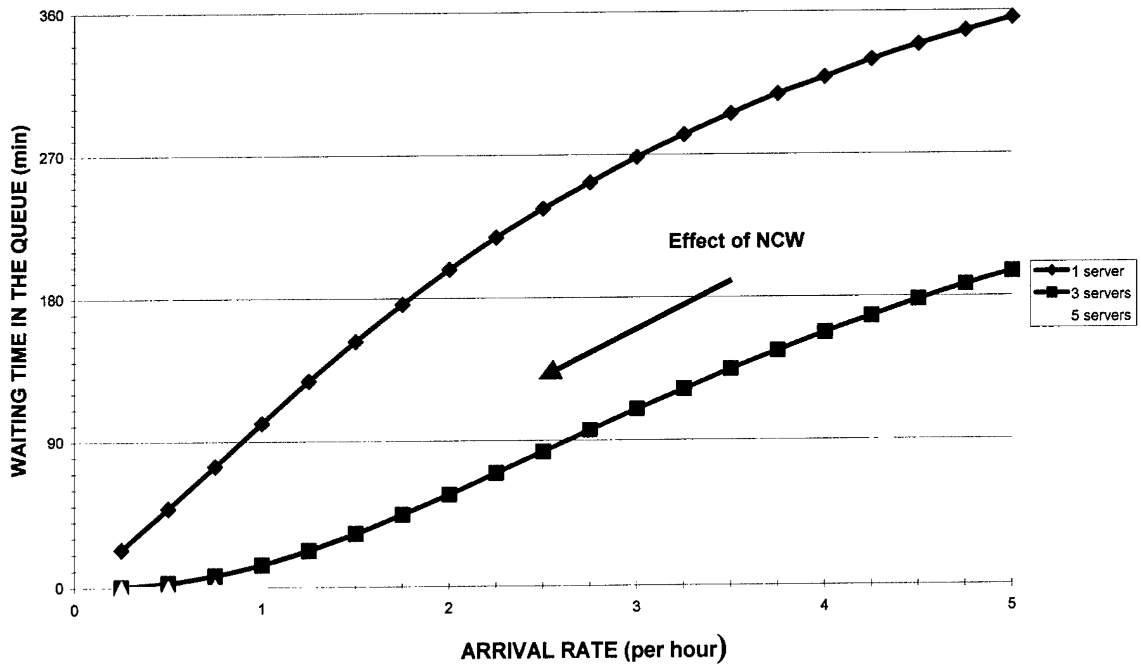
QSIM is a queueing theory model developed to interpret the various undertakings involved in military operations.* The multiple probability distribution functions (PDFs) for arrival, service, and renege rates not only allow QSIM to analyze a variety of specific warfare tasks (especially those characterized by nonexponential PDFs), but they also ensure that the model is applicable to many other situations. Moreover, the relatively efficient run times permit the user to analyze a range of input values quickly.

As with all models, there is room for improvement, with the current list of recommendations for QSIM as follows:

- *Add the capability for the networking of queues.* Currently, there is no way to analyze systems where the customers leaving one queue are the incoming customers to another queue. Situations of networked queues occur all the time (e.g., detection and classification of sonar contacts), and the capability to simulate/analyze these networks would be highly useful.

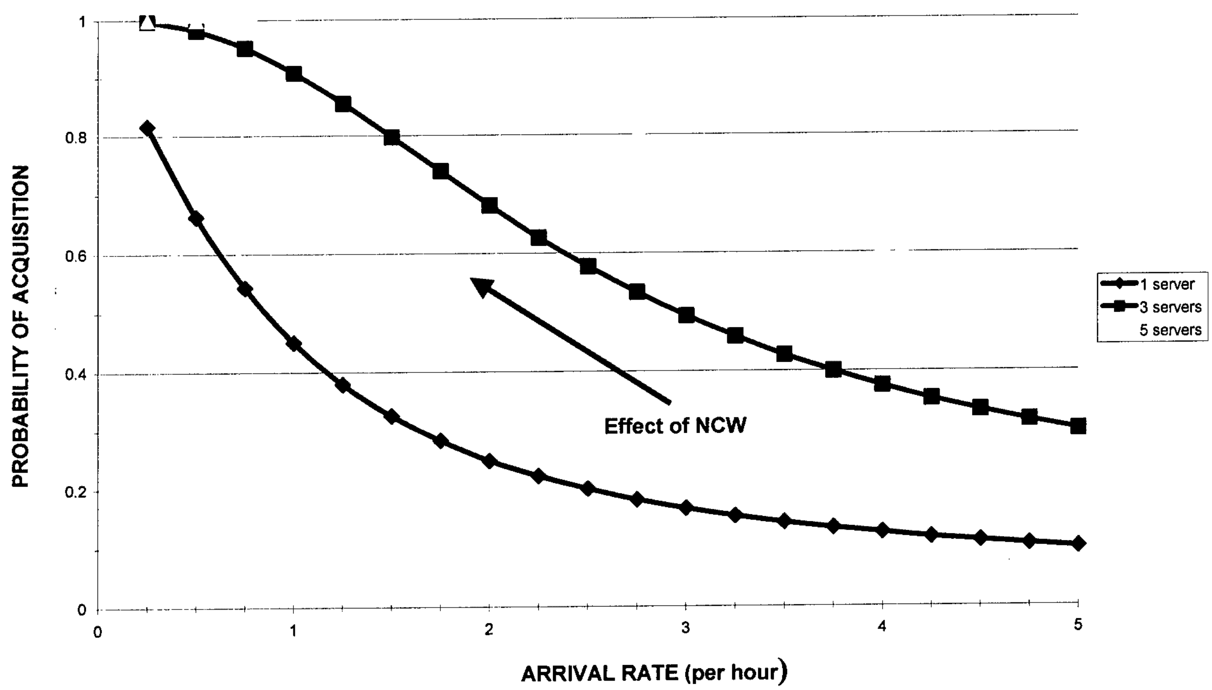
- *Link other mathematical constructs to QSIM.* To accurately model the complex nature of warfare, QSIM must have the capability to link with various mathematical constructs.

*An executable of QSIM is available upon request from the authors of this document.



Note: As the helicopters work together, the number of contacts that each helicopter must attempt to detect and classify has decreased. Hence, the time each contact is on the scope before an operator begins to detect and classify also decreases. This effect increases as the number of servers (helicopters) increases.

Figure 1. Plot Showing Mean Waiting Time Decreasing as Arrival Rate Decreases



Note: As the helicopters work together, the number of contacts each helicopter must attempt to detect and classify has decreased. Hence, the probability of detection and classification has increased. This effect increases as the number of servers (helicopters) increases.

Figure 2. Plot Showing Probability of Acquisition Increasing as Arrival Rate Decreases

In particular, the use of stochastic colored Petri nets and information theory to model the flow of information through a command and control network would allow for the generation of accurate arrival, service, and renege PDFs. Furthermore, linking multiple constructs would allow for the creation of an integrated family of simulations to analyze complex warfare tasks.

- *Add the capability to parameterize more than one distribution at a time.* Parameterization of multiple distributions simultaneously would allow the user to more efficiently explore various parameter combinations, thus reducing the time required to perform a study

APPENDIX A — DISTRIBUTIONS USED BY QSIM

QSIM employs eight PDFs, with tables A-1 through A-8 providing the PDF, CDF, parameters, and a typical interpretation of the distribution in a sonar application.

Table A-1. Exponential Distribution

PDF	$f(x)=\begin{cases}\lambda e^{-\lambda x} & x\geq 0\\ 0 & x<0\end{cases}$
CDF	$F(x)=\begin{cases}1-e^{-\lambda x} & x\geq 0\\ 0 & x<0\end{cases}$
Parameter	$\lambda>0$ is the shape parameter
Typical Interpretation	Time until first sonar contact

Table A-2. Poisson Distribution

PMF (Probability Mass Function)	$f(x)=\begin{cases}\dfrac{e^{-\alpha}\alpha^{x}}{x!} & x=0,1,2,\ldots\\ 0 & \text{otherwise}\end{cases}$
CDF	$F(x)=\sum_{i=0}^{x}\dfrac{e^{-\alpha}\alpha^{i}}{i!}$
Parameter	$\alpha>0$
Typical Interpretation	Number of sonar contacts in a given time

Table A-3. Log-Normal Distribution

PDF	$f(x)=\begin{cases}\frac{1}{x\sqrt{2\pi\sigma^2}}e^{\frac{-(\ln x-\mu)^2}{2\sigma^2}} & x>0\\ 0 & \text{otherwise}\end{cases}$
CDF	$F(x)=\Phi\left(\frac{\ln x-\mu}{\sigma}\right)$, where Φ is the standard normal distribution function
Parameters	Shape parameter $\sigma > 0$, scale parameter $\mu \subset (-\infty,\infty)$
Typical Interpretation	Clutter density in signal processing

Table A-4. Inverse Gaussian Distribution

PDF	$f(x)=\begin{cases}\sqrt{\frac{\lambda}{2\pi}}x^{-3/2}e^{\frac{-\lambda(x-\mu)^2}{2\mu^2 x}} & x>0\\ 0 & \text{otherwise}\end{cases}$
CDF	$F(x)=\Phi\left[\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu}-1\right)\right]+e^{2\lambda/\mu}\Phi\left[-\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu}+1\right)\right]$, where Φ is the standard normal distribution function
Parameters	$\mu > 0$ is the mean, $\lambda > 0$ is the scale parameter
Typical Interpretation	Time delays in ASW processes characterized by random walk with positive drift

Table A-5. Gamma Distribution

PDF	$f(x)=\begin{cases}\dfrac{\beta^{-\alpha}x^{\alpha-1}e^{-\frac{x}{\beta}}}{\Gamma(\alpha)} & x>0\\ 0 & \text{otherwise}\end{cases}$
CDF (assuming $\alpha > 0$, an integer. If α is not an integer, there is no closed form)	$F(x)=\begin{cases}1-e^{-\frac{x}{\beta}}\sum_{j=0}^{\alpha-1}\dfrac{\left(\frac{x}{\beta}\right)^{j}}{j!} & x>0\\ 0 & \text{otherwise}\end{cases}$
Parameters	$\alpha > 0$ is the shape parameter, $\beta > 0$ is the scale parameter
Typical Interpretation	Waiting times between each sonar contact in a given time period

Table A-6. Erlang Distribution

PDF	$f(x)=\begin{cases}\dfrac{\alpha^{r}}{\Gamma(r)}x^{r-1}e^{-\alpha x} & x>0\\ 0 & \text{otherwise}\end{cases}$
CDF	$F(x)=1-e^{-\alpha x}\sum_{j=0}^{r-1}\dfrac{(x\alpha)^{j}}{j!}$
Parameter	$r > 0, r = 1,2,\ldots$ shape parameter, $\alpha > 0$ scale parameter
Typical Interpretation	Service time in an n server queue with homogeneous service times

Table A-7. Weibull Distribution

PDF	$f(x)=\begin{cases}\alpha\beta^{-\alpha}x^{\alpha-1}e^{-\left(\frac{x}{\beta}\right)^{\alpha}} & x>0\\ 0 & \text{otherwise}\end{cases}$
CDF	$F(x)=\begin{cases}1-e^{-\left(\frac{x}{\beta}\right)^{\alpha}} & x>0\\ 0 & \text{otherwise}\end{cases}$
Parameters	$\alpha > 0$ is the shape parameter, $\beta > 0$ is the scale parameter
Typical Interpretation	Time until failure of first sonar unit

Table A-8. Uniform Distribution

PDF	$f(x)=\begin{cases}\dfrac{1}{b-a} & a\le x\le b\\ 0 & \text{otherwise}\end{cases}$
CDF	$F(x)=\begin{cases}0 & x<a\\ \dfrac{x-a}{b-a} & a\le x\le b\\ 1 & x>b\end{cases}$
Parameters	a and b
Typical Interpretation	When arrival or service times are known but nothing is known about the distributions

APPENDIX B — GENERATION OF RANDOM VARIATES

This appendix describes how QSIM generates random variates, with the generation of these variates dependent on a uniform (0,1).

GENERATION OF UNIFORM (0,1)

Due to its ease of implementation, randomness, uniformity, number of digits, and speed of execution, the free RANROT W routine by Anger Fog[4] was chosen to generate uniform (0,1) variates. The RANROT W generator is similar to the additive or lagged Fibonacci generators, but with extra rotation or swapping of bits.

Implementation of RANROT W simply requires insertion of a `#include` statement. For speed, QSIM uses the assembly language version. Randomness and uniformity are discussed in the section entitled Generation and Validation of Random Variates in the main portion of this document. The RANROT W routine requires less than 50 clock ticks per execution and generates numbers with 63 bits.

EXPONENTIAL DISTRIBUTION

For the exponential distribution, the inverse transform technique is used:[6]

$$\begin{aligned} 1 - e^{-\lambda X} &= U , \\ e^{-\lambda X} &= 1 - U , \\ -\lambda X &= \ln(1-U) , \\ X &= -\frac{1}{\lambda}\ln(1-U) . \end{aligned} \tag{B-1}$$

Step 1: Generate U.

Step 2: Return $-(1/\lambda) * \ln(1 - U)$.

WEIBULL DISTRIBUTION

An inverse transform technique is again used:[6]

$$\begin{aligned} 1 - e^{-\left(\frac{X}{\alpha}\right)^{\beta}} &= U , \\ -\left(\frac{X}{\alpha}\right)^{\beta} &= 1 - U , \\ X &= \alpha\left(-\ln(1-U)\right)^{1/\beta} . \end{aligned} \tag{B-2}$$

Step 1: Generate U.

Step 2: Return $\alpha(-\ln(1 - U))^{(1/\beta)}$.

ERLANG DISTRIBUTION

The convolution method is used for the Erlang distribution.[6] However, first note that an Erlang random variable with parameters (K, θ) is the sum of K-independent exponential random variables, each with mean $1/K\theta$. Since it is known how to generate exponential random variables, the following expression is obtained:

$$X = \sum_{i=1}^{K} -\frac{1}{K\theta}\ln U_i \, ,$$
$$= -\frac{1}{K\theta}\ln\left(\prod_{i=1}^{K} U_i\right). \qquad \text{(B-3)}$$

As can be seen, converting to a product results in computational efficiency.

Step 1: Compute Prod = $U_1 * U_2 * \ldots U_K$.

Step 2: Return $X = (-1/\lambda) * \ln(\text{Prod})$.

POISSON DISTRIBUTION

Beginning with

$$N + 1 = \min\left\{ n : \prod_{i=1}^{n} U_i < e^{-\lambda} \right\} \qquad \text{(B-4)}$$

shows that N has a Poisson distribution,[7] as seen in the following expression:

$$N = \max\left\{ n : \sum_{i=1}^{n} -\log U_i < \lambda \right\} . \qquad \text{(B-5)}$$

But $-\log U_i$ is exponential with rate 1, and so if $-\log U_i$ is interpreted as the interarrival times of a Poisson process having rate 1, then $N = N(\lambda)$ would equal the number of events by time λ. Hence, N is Poisson with mean λ.

Step 1: Compute Prod = Prod * U_i, where i is the count of the number of U's generated.

Step 2: If Prod $< e^{-\lambda}$, return (count -1); otherwise, repeat step 1.

GAMMA DISTRIBUTION

The generation of a gamma variate is based on the generation of a normal variate[8] as follows:

Step 1: Set $D = \alpha - 1/3$, $C = 1/\text{sqrt}(9D)$.

Step 2: Generate $v = (1 + C * X)^3$ where X is standard normal.

Step 3: Generate U.

Step 4: If $U < 1 - 0.0331 * X^4$, then return $D * v$.

Step 5: If $\log(U) < 0.5 * X^2 + D * (1 - v * \log(v))$, then return $D * v$.

Step 6: Return to step 2.

If $\alpha < 1$, then generate a gamma variate as above, but use $\alpha = 1 + \alpha$. Then, return $G * U^{(1/\alpha)}$, where G is a gamma variate with $\alpha = 1 + \alpha$.

LOG-NORMAL DISTRIBUTION

QSIM uses a property of the normal distribution to generate a log-normal variate. Namely, if $Y = N(\mu, \sigma^2)$, then $X = e^Y = \text{LN}(\mu, \sigma^2)$, resulting in the following algorithm:[6]

Step 1: Generate $Z = N(0,1) = (-2\ln U_1)^{1/2}\cos(2\pi U_2)$.

Step 2: Let $W = \mu + \sigma Z$.

Step 3: Return $X = e^W$.

INVERSE GAUSSIAN DISTRIBUTION

A transformation[9] of the following form $v = g(x)$ is considered; in particular,

$$V = g(x) = \frac{\lambda(X-\mu)^2}{X\mu^2} \sim \chi_1^2 = v_0 \ . \tag{B-6}$$

For each chi-square variate, v_0, the above transformation is solved for X to obtain a corresponding observation from an inverse Gaussian distribution. The square of a standard normal is the symbol v_0. For any v_0, there are two solutions:

$$\begin{aligned} x_1 &= \mu + \frac{\mu^2 v_0}{2\lambda} - \frac{\mu}{2\lambda}\sqrt{4\mu\lambda v_0 + \mu^2 v_0^2} \ , \\ x_2 &= \frac{\mu^2}{x_1} . \end{aligned} \tag{B-7}$$

A uniform (0,1) U is generated, and the root is chosen by comparing U to

$$p(v_0) = \frac{\mu}{\mu + x_1} . \tag{B-8}$$

If $U > p(v_0)$, then return x_2. Otherwise, return x_1.

Step 1: Generate U_1, U_2.

Step 2: Generate $Z = N(0,1) = (-2 * \ln U_1)^{1/2} * \cos(2 * U_2 * \pi)$.

Step 3: $Z = Z^2$.

Step 4: $X_1 = \mu + Z/(2 * \lambda) - \mu/(2 * \lambda) * \text{sqrt}(4 * \mu * \lambda * Z + \mu^2 * Z^2)$.

Step 5: $X_2 = \mu^2/X_1$.

Step 6: $p = \mu/(\mu + X_1)$.

Step 7: Generate U_3.

Step 8: If $(U_3 < p)$ return X_1. Otherwise, return X_2.

EXPLICIT DISTRIBUTION

This method assumes that the data are from a continuous distribution.[6] The user provides a histogram in the form x_{i-1}, x_i, o_i, where (x_{i-1}, x_i) is the i^{th} interval and o_i is the number of occurrences in the i^{th} interval. The end points of each interval are stored, and the cumulative frequency for each interval is computed. Then, U is generated. Next determined is $c_{i-1} < U \leq c_i$, where c_i is the cumulative frequency of the first i intervals. Now,

$$a_i = \frac{x_i - x_{i-1}}{c_i - c_{i-1}} \tag{B-9}$$

is computed, resulting in

$$X = x_{i-1} + a_i(U - c_{i-1}) . \tag{B-10}$$

Step 1: Read input data and generate cumulative distribution.

Step 2: Generate U and determine (c_{i-1}, c_i).

Step 3: Compute a_i.

Step 4: Return $x_{i-1} + a_i(U_2 - c_{i-1})$.

UNIFORM DISTRIBUTION

The inverse transform technique is used as follows:[6]

$$\frac{X-a}{b-a}=U\ ,$$

$$X=a+(b-a)*U\ . \qquad \text{(B-11)}$$

Step 1: Generate *U*.

Step 2: Return $a + U * (b - a)$.

APPENDIX C — DERIVATION OF VERIFICATION FORMULAS

This appendix presents a derivation of all the formulas used to verify QSIM. These derivations follow from the work of Gross and Harris[2] and Prabhu.[10]

PRELIMINARIES

First considered is a general birth-death process, with states labeled by i, $i = 0,1, \ldots, \infty$. If it is assumed that λ_n is the rate at which transitions from state i to $i + 1$ occur, μ_n is the rate at which transitions from state i to i - 1 occur, and p_k is the steady-state probability that there are k customers in the system, then, at steady state, the queue will satisfy the following differential-difference equations for the birth-death process:

$$
\begin{aligned}
0 &= -(\lambda_j + \mu_j)p_j + \lambda_{j-1}p_{j-1} + \mu_{j+1}p_{j+1} \qquad (j \geq 1)\,, \\
0 &= -\lambda_0 p_0 + \mu_1 p_1 \,.
\end{aligned}
\tag{C-1}
$$

Rewriting yields

$$
\begin{aligned}
p_{j+1} &= \frac{\lambda_j + \mu_j}{\mu_{j+1}} p_j - \frac{\lambda_{j-1}}{\mu_{j+1}} p_{j-1} \qquad (j \geq 1)\,, \\
p_1 &= \frac{\lambda_0}{\mu_1} p_0 \,.
\end{aligned}
\tag{C-2}
$$

Thus, p_i is computed as

$$
\begin{aligned}
p_2 &= \frac{\lambda_1 + \mu_1}{\mu_2} p_1 - \frac{\lambda_0}{\mu_2} p_0 \,, \\
&= \frac{\lambda_1 + \mu_1}{\mu_2} \frac{\lambda_0}{\mu_1} p_0 - \frac{\lambda_0}{\mu_2} p_0 \,, \\
&= \frac{\lambda_1 \lambda_0}{\mu_2 \mu_1} p_0 \,; \\
p_3 &= \frac{\lambda_2 + \mu_2}{\mu_3} p_2 - \frac{\lambda_1}{\mu_3} p_1 \,, \\
&= \frac{\lambda_2 + \mu_2}{\mu_3} \frac{\lambda_1 \lambda_2}{\mu_2 \mu_1} p_0 - \frac{\lambda_1}{\mu_3} \frac{\lambda_0}{\mu_1} p_0 \,, \\
&= \frac{\lambda_2 \lambda_1 \lambda_0}{\mu_3 \mu_2 \mu_1} p_0 \\
&\;\;\vdots
\end{aligned}
\tag{C-3}
$$

The pattern that appears to be emerging is

$$p_k = \frac{\lambda_{k-1}\lambda_{k-2}\cdots\lambda_0}{\mu_k\mu_{k-1}\cdots\mu_1} p_0 ,$$

$$= p_0 \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i} . \tag{C-4}$$

Induction verifies this formula.[2] From a queueing theory perspective, λ_n is the mean rate at which customers enter the system and μ_n is the mean rate at which customers leave the system.

M/M/1

Because the M/M/1 system is a birth-death process with constant birth and death rates, $\lambda_n = \lambda$ and $\mu_n = \mu$. Thus,

$$p_k = p_0 \prod_{i=1}^{k} \frac{\lambda}{\mu} = p_0 \left(\frac{\lambda}{\mu}\right)^k . \tag{C-5}$$

Next, $\rho = \lambda/\mu$. For the existence of a steady-state solution, ρ must be less than 1, or, equivalently, λ must be less than μ. This reasoning makes sense because if $\lambda > \mu$, the server will lag further and further behind, resulting in a queue length that grows without bound. Also, if $\lambda = \mu$, then the server does not have the opportunity to reduce the queue size as more customers arrive at the system. Computation of p_0 uses the fact that the probabilities must sum to 1, along with the well-known solution to the geometric series, as follows:

$$1 = \sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k p_0 \quad \Rightarrow \quad p_0 = \frac{1}{\sum_{k=0}^{\infty} \rho^k} = \frac{1}{\frac{1}{1-\rho}} = 1 - \rho . \tag{C-6}$$

Thus,

$$p_k = \rho^k (1-\rho) . \tag{C-7}$$

Letting N be the random variable denoting the number in the system at steady state and letting L be its expected value yields

$$L = E[N] = \sum_{k=0}^{\infty} kp_k = (1-\rho)\sum_{k=0}^{\infty} k\rho^k = \rho(1-\rho)\sum_{k=1}^{\infty} k\rho^{k-1} ,$$

$$= \rho(1-\rho)\frac{d}{d\rho}\left(\frac{1}{1-\rho}\right),$$

$$= \frac{\rho(1-\rho)}{(1-\rho)^2},$$

$$= \frac{\lambda}{\mu-\lambda}. \qquad \text{(C-8)}$$

If N_Q is the random variable denoting the number in the queue at steady state and L_Q is its expected value, then

$$L_Q = E[N_Q] = \sum_{k=1}^{\infty}(k-1)p_k = \sum_{k=1}^{\infty} kp_k - \sum_{k=1}^{\infty} p_k = L - (1-p_0) = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu(\mu-\lambda)} . \qquad \text{(C-9)}$$

Now, applying Little's formulas to obtain mean waiting time in the system W and mean waiting time in the queue W_Q results in

$$W = \frac{L}{\lambda} = \frac{1}{\mu-\lambda},$$

$$W_Q = \frac{L_Q}{\lambda} = \frac{\lambda}{\mu(\mu-\lambda)}. \qquad \text{(C-10)}$$

M/M/S

In the M/M/S system, each server has an independent and identically distributed exponential service time, and the arrival times also follow an exponential distribution. Thus, M/M/S is a birth-death process, such that $\lambda_n = \lambda$ and a determination of μ_n must be made.

If there are more than S customers in the system, then each of the S servers is outputting at mean rate μ, with the mean system output being $S\mu$. If there are fewer than S customers, say $n < S$, then the system is outputting at mean rate $n\mu$. Hence,

$$\mu_n = \begin{cases} n\mu & 1 \le n < S , \\ S\mu & n \ge S . \end{cases} \qquad \text{(C-11)}$$

Next, p_k is computed as

$$p_k = \begin{cases} \dfrac{\lambda^k}{k!\,\mu^k} p_0 & 1 \le k < S\,, \\ \dfrac{\lambda^k}{S^{k-S} S!\,\mu^k} p_0 & k \ge S\,. \end{cases} \quad \text{(C-12)}$$

To find p_0, the fact that the probabilities must sum to 1 is again used:

$$p_0 = \left(\sum_{k=0}^{S-1} \frac{\lambda^k}{k!\,\mu^k} + \sum_{k=S}^{\infty} \frac{\lambda^k}{S^{S-k} S!\,\mu^k} \right)^{-1} . \quad \text{(C-13)}$$

Now L_Q can be computed by letting $r = \lambda/\mu$ and $\rho = r/S = \lambda/S\mu$:

$$\begin{aligned} L_Q &= \sum_{k=S+1}^{\infty} (k-S) p_k = \sum_{k=S+1}^{\infty} (k-S) \frac{\lambda^k}{S^{k-S} S!\,\mu^k} p_0 = \frac{r^S p_0}{S!} \sum_{m=1}^{\infty} m\rho^k\,, \\ &= \frac{r^S \rho p_0}{S!} \sum_{m=1}^{\infty} m\rho^{m-1} = \frac{r^S \rho p_0}{S!} \frac{d}{d\rho} \sum_{m=1}^{\infty} \rho^m = \frac{r^S \rho p_0}{S!} \frac{d}{d\rho} \left(\frac{1}{1-\rho} - 1 \right), \\ &= \frac{r^S \rho p_0}{S!(1-\rho)^2} . \end{aligned} \quad \text{(C-14)}$$

Next, Little's formulas are used to calculate W and W_Q as

$$\begin{aligned} W_Q &= \frac{L_Q}{\lambda} = \left(\frac{r^S}{S!(S\mu)(1-\rho)^2} \right) p_0\,, \\ W &= \frac{1}{\mu} + W_Q = \frac{1}{\mu} + \left(\frac{r^S}{S!(S\mu)(1-\rho)^2} \right) p_0\,. \end{aligned} \quad \text{(C-15)}$$

M/M/S + M

Results from this subsection address the single and multiple server case (homogeneous servers). Customers renege from the queue at rate α (i.e., mean time between reneges is α^{-1}), and once service begins, it continues until completion. The time in the queue and the time in the system are for customers who acquire service.

The M/M/S + M system is a birth-death process, such that $\lambda_n = \lambda$ and a determination of μ_n must be made. To reflect the death process under reneging requires that (1) a customer be able to leave the system from any of the S servers and (2) a customer be able to renege from the queue once all the S servers are busy. Hence,

$$\mu_n = \begin{cases} n\mu & 1 \le n \le S\,, \\ S\mu + (n-S)\alpha & n > S\,. \end{cases} \tag{C-16}$$

Now p_k is computed as

$$p_k = p_0 \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} = \begin{cases} p_0 \left(\dfrac{\lambda}{\mu}\right)^k \dfrac{1}{k!} & 1 \le k \le S\,, \\ p_0 \left(\dfrac{\lambda}{\mu}\right)^S \dfrac{1}{S!} \left(\dfrac{\lambda}{\alpha}\right)^{k-S} \dfrac{1}{\beta_{k-S}} & k > S\,, \end{cases} \tag{C-17}$$

where

$$\beta_k = \prod_{m=1}^{k} \left(m + \frac{S\mu}{\alpha} \right). \tag{C-18}$$

Again, the fact that the probabilities must sum to 1 to compute p_0 is used:

$$1 = \sum_{k=1}^{\infty} p_k = \sum_{k=1}^{S} p_k + \sum_{k=S+1}^{\infty} p_k = p_0 \sum_{k=1}^{S} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} + p_0 \left(\frac{\lambda}{\mu}\right)^S \frac{1}{S!} \sum_{k=S+1}^{\infty} \prod_{i=1}^{k-S} \frac{\lambda}{S\mu + i\alpha}\,,$$

$$\Rightarrow p_0 = \left(\sum_{k=1}^{S} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} + \left(\frac{\lambda}{\mu}\right)^S \frac{1}{S!} \sum_{k=1}^{\infty} \left(\frac{\lambda}{\alpha}\right)^k \frac{1}{\beta_k} \right)^{-1}. \tag{C-19}$$

Now, the CDF of the waiting time in the queue must be derived. In particular, it is desired that

$$F_q(x) = \Pr\{W(t) < x \mid \text{customer is served}\},$$

$$= \frac{\Pr\{\text{customer is served and } W(t) < x\}}{P_{acq}}, \tag{C-20}$$

where P_{acq} is the probability of acquisition (i.e., the probability that an arriving customer will acquire service) and $W(t)$ is the time a customer waits in the queue if arriving at time t. Whether a customer is served or not is determined by a race between the time taken waiting for a server to become available (W_0) and the time the customer is prepared to wait before reneging (D). The customer is served if $W_0 < D$. The two times are independent, resulting in

$$F_q(x) = \frac{1}{P_{acq}} \int_0^x \int_y^\infty f_W(y) \alpha e^{-\alpha z} dz dy,$$
$$= \frac{1}{P_{acq}} \int_0^x f_W(y) e^{-\alpha y} dy, \tag{C-21}$$

where $f_W(x)$ is the density of W_0. If a customer arrives at time t to find n customers already in the system and then decides to wait until they are served, then the new customer would wait in the queue for time $W_0(t)$, expressed as

$$W_0(t) = \begin{cases} 0 & n < S, \\ D_1 + D_2 + \ldots + D_{n-S+1} & n \geq S, \end{cases} \tag{C-22}$$

where D_1, D_2, . . . are the intervals of time between successive departures, due either to reneging or to service completion. The intervals D_1, D_2, . . . are therefore independent random variables with D_m having an exponential distribution with parameter $S\mu + (n - S - m + 1)\alpha$. Thus, $X = D_1 + D_2 + \ldots D_{n-S+1}$ is said to be a hypoexponential random variable and has the following density:

$$f(t) = \sum_{k=0}^{n-S} \frac{(-1)^k S\mu\beta_{n-S}}{k!(n-S-k)!} e^{-(S\mu+k\alpha)t} . \tag{C-23}$$

Hence,

$$\Pr\{W_0(t) > x\} = \sum_{n=S}^{\infty} \Pr\{Q(t) = n\} \sum_{k=0}^{n-S} \frac{(-1)^k S\mu\beta_{n-S}}{k!(n-S-k)!} \frac{e^{-(S\mu+k\alpha)t}}{S\mu + k\alpha}, \tag{C-24}$$

where $Q(t)$ is the number of customers in the system at time t. Now, the density function $f_W(y)$ is obtained from equation (C-24) as

$$f_W(t) = \sum_{n=0}^{S-1} p_n \delta(t) + \sum_{n=S}^{\infty} p_n \sum_{k=0}^{n-S} \frac{(-1)^k S\mu\beta_{n-S}}{k!(n-S-k)!} e^{-(S\mu+k\alpha)t},$$

$$= \sum_{n=0}^{S-1} p_n \delta(t) + S\mu p_S \sum_{n=S}^{\infty} \left(\frac{\lambda}{\alpha}\right)^{n-S} \sum_{k=0}^{n-S} \frac{(-1)^k}{k!(n-S-k)!} e^{-(S\mu+k\alpha)t},$$

$$= \sum_{n=0}^{S-1} p_n \delta(t) + S\mu p_S e^{\lambda/\alpha} \sum_{k=0}^{\infty} \frac{e^{-(S\mu+k\alpha)t}}{k!} \left(\frac{-\lambda}{\alpha}\right)^k . \tag{C-25}$$

The PDF $f_q(t)$ is therefore

$$f_q(t)=\frac{e^{-\alpha t}}{P_{acq}}\left[\sum_{n=0}^{S-1}p_n\delta(t)+S\mu p_S e^{\lambda/\alpha}\sum_{k=0}^{\infty}\frac{e^{-(S\mu+k\alpha)t}}{k!}\left(\frac{-\lambda}{\alpha}\right)^k\right], \qquad \text{(C-26)}$$

from which the mean waiting time in the queue of customers who acquire service is

$$W_Q=\int_0^{\infty}f_q(x)x\,dx,$$

$$=\frac{S\mu p_S e^{\lambda/\alpha}}{P_{acq}}\sum_{k=0}^{\infty}\left(\frac{-\lambda}{\alpha}\right)^k\frac{1}{k!}\frac{1}{[S\mu+(k+1)\alpha]^2}. \qquad \text{(C-27)}$$

Taking the mean of the sum gives the mean waiting time in the system as

$$W=W_Q+\frac{1}{\mu}. \qquad \text{(C-28)}$$

PROBABILITY OF ACQUISITION

Clearly, in the absence of balking and reneging, the probability of acquisition is unity. A time interval during which N_S customers are served is considered, with the length of this interval shown as

$$T=T_0+\sum_{i=1}^{N_S}s_i+\sum_{n=2}^{S}(n-1)T_n, \qquad \text{(C-29)}$$

where T_n is the time during which there are n customers being served and s_i is the time taken to serve the i^{th} customer. In the limit of $N_S\rightarrow\infty$ is found

$$\sum_{i=1}^{N_S}s_i\rightarrow\frac{N_s}{\mu}, \qquad \text{(C-30)}$$

$$T_n\rightarrow\begin{cases}p_nT & n<S,\\ \left(1-\sum_{k=0}^{S-1}p_k\right)T & n=S,\end{cases} \qquad \text{(C-31)}$$

and so

$$T \to \frac{N_S/\mu}{1 - p_0 + \sum_{n=1}^{S-1}(n-1)p_n + (S-1)\left(1 - \sum_{k=0}^{S-1} p_k\right)},$$

$$= \frac{N_S/\mu}{\sum_{n=1}^{S-1} np_n + S\left(1 - \sum_{k=0}^{S-1} p_k\right)}. \qquad \text{(C-32)}$$

Letting N_a be the number of arrivals during the interval T results in

$$N_a \to \lambda T. \qquad \text{(C-33)}$$

Thus, the probability of acquiring service is

$$P = \sum_{n=1}^{S-1} \frac{n\mu p_n}{\lambda} + \frac{S\mu}{\lambda}\left(1 - \sum_{k=0}^{S-1} p_k\right). \qquad \text{(C-34)}$$

APPENDIX D — DETAILED VERIFICATION OF QSIM

QSIM was validated against closed-form analytical formulas (see appendix C for the derivation of the formulas).

Tables D-1 through D-8 in this appendix compare QSIM runs to the formulas, with the runs in each table consisting of unlimited queue length, a maximum time of 150,000 per iteration, and a total number of 1,000 iterations. It is observed that increasing the maximum time per iteration will improve accuracy, although it should be noted that this approach will result in longer execution times.

Table D-1. Comparison of QSIM with Gross and Harris Formulas for Mean Waiting Time in the Queue and Mean Waiting Time in the System for M/M/1 Queue*

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)
7.5	111.742	112.5	119.242	120
4.61538	10.1902	10.2273	17.6877	17.7273
3.33333	5.35592	5.35714	12.855	12.8571
2.6087	3.62338	3.62903	11.1233	11.129
2.14286	2.74213	2.7439	10.2436	10.2439
1.81818	2.20527	2.20588	9.7049	9.70588
1.57895	1.84525	1.84426	9.34825	9.34426
1.39535	1.58114	1.58451	9.07848	9.08451
1.25	1.3883	1.38889	8.88936	8.88889
1.13208	1.23342	1.23626	8.73374	8.73626
1.03448	1.11418	1.11386	8.61254	8.61386
0.952381	1.01138	1.01351	8.50789	8.51351
0.882353	0.928446	0.929752	8.42667	8.42975
0.821918	0.859047	0.858779	8.36157	8.35878
0.769231	0.795339	0.797872	8.29462	8.29787
0.722892	0.746091	0.745033	8.24631	8.24503
0.681818	0.699816	0.698758	8.20136	8.19876
0.645161	0.658415	0.657895	8.15887	8.15789
0.612245	0.619698	0.621547	8.12009	8.12155
0.582524	0.590375	0.589005	8.09186	8.08901
0.555556	0.558507	0.559701	8.05764	8.0597

*Service rate is 8 per hour

*Table D-2. Comparison of QSIM with Gross and Harris Formulas for Mean Waiting Time in the Queue and Mean Waiting Time in the System for M/M/2 Queue**

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)
7.5	2.11211	1.62562	9.61347	9.12562
4.61538	0.680194	0.602852	8.18076	8.10285
3.33333	0.339253	0.317483	7.83553	7.81748
2.6087	0.204298	0.195865	7.70354	7.69586
2.14286	0.136932	0.132783	7.63369	7.63278
1.81818	0.0981868	0.0958923	7.59944	7.59589
1.57895	0.0735183	0.0724728	7.57272	7.57247
1.39535	0.0571676	0.056685	7.55757	7.55669
1.25	0.0461385	0.0455418	7.54408	7.54554
1.13208	0.0378717	0.037386	7.53757	7.53739
1.03448	0.0316581	0.0312385	7.5324	7.53124
0.952381	0.0266479	0.0264904	7.5242	7.52649
0.882353	0.0228983	0.0227474	7.527	7.52275
0.821918	0.0199794	0.0197447	7.51823	7.51974
0.769231	0.0172348	0.0172993	7.51215	7.5173
0.722892	0.0154225	0.0152814	7.51687	7.51528
0.681818	0.0136631	0.0135968	7.51065	7.5136
0.645161	0.0124371	0.0121761	7.5127	7.51218
0.612245	0.0111801	0.0109669	7.51166	7.51097
0.582524	0.00994147	0.00992924	7.50896	7.50993
0.555556	0.0090545	0.00903211	7.5079	7.50903

*Service rate is 8 per hour

*Table D-3. Comparison of QSIM with Gross and Harris Formulas for Mean Waiting Time in the Queue and Mean Waiting Time in the System for M/M/3 Queue**

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)
7.5	0.282253	0.244059	7.77985	7.74406
4.61538	0.0683808	0.0658599	7.56852	7.56586
3.33333	0.026817	0.0263011	7.52914	7.5263
2.6087	0.0131066	0.013002	7.51201	7.513
2.14286	0.00741906	0.00734591	7.50613	7.50735
1.81818	0.00454836	0.00454651	7.50258	7.50455
1.57895	0.00305491	0.00300629	7.50631	7.50301
1.39535	0.00207056	0.00209004	7.50373	7.50209
1.25	0.00146832	0.00151129	7.50493	7.50151
1.13208	0.00108488	0.00112793	7.50155	7.50113
1.03448	0.000866945	0.000863994	7.50089	7.50086
0.952381	0.000675803	0.000676383	7.50156	7.50068
0.882353	0.000543176	0.000539386	7.50101	7.50054
0.821918	0.000431764	0.000437024	7.50142	7.50044
0.769231	0.00034423	0.000359008	7.50091	7.50036
0.722892	0.000287073	0.00029851	7.49695	7.5003
0.681818	0.000266051	0.000250876	7.50256	7.50025
0.645161	0.000194209	0.000212861	7.5016	7.50021
0.612245	0.000195603	0.000182155	7.49996	7.50018
0.582524	0.000151597	0.000157082	7.49811	7.50016
0.555556	0.000120998	0.000136407	7.50316	7.50014

*Service rate is 8 per hour

*Table D-4. Comparison of QSIM with Gross and Harris Formulas for Mean Waiting Time in the Queue and Mean Waiting Time in the System for M/M/4 Queue**

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)
7.5	0.039694	0.0383807	7.53853	7.53838
4.61538	0.00664145	0.00657687	7.50561	7.50658
3.33333	0.00193033	0.00192896	7.50019	7.50193
2.6087	0.000767387	0.000754889	7.49868	7.50075
2.14286	0.000336912	0.000353222	7.50178	7.50035
1.81818	0.000167889	0.000186622	7.50219	7.50019
1.57895	0.000110045	0.000107663	7.49985	7.50011
1.39535	6.79396E-05	6.64E-05	7.50211	7.50007
1.25	4.23307E-05	4.31E-05	7.50125	7.50004
1.13208	2.62157E-05	2.92E-05	7.50323	7.50003
1.03448	1.91543E-05	2.05E-05	7.49713	7.50002
0.952381	1.62049E-05	1.48E-05	7.49723	7.50001
0.882353	1.11597E-05	1.09E-05	7.49981	7.50001
0.821918	8.15201E-06	8.27E-06	7.50031	7.50001
0.769231	7.3799E-06	6.37E-06	7.50049	7.50001
0.722892	4.38335E-06	4.98E-06	7.49737	7.5
0.681818	3.50833E-06	3.95E-06	7.50215	7.5
0.645161	1.91686E-06	3.18E-06	7.50525	7.5
0.612245	2.10691E-06	2.58E-06	7.50455	7.5
0.582524	1.83171E-06	2.12E-06	7.49806	7.5
0.555556	9.16891E-07	1.76E-06	7.50087	7.5

*Service rate is 8 per hour

*Table D-5. Comparison of QSIM with Prabhu Formulas for Mean Waiting Time in the Queue, Mean Waiting Time in the System, and Probability of Acquisition for M/M/1 Queue with Renege Rate of 2 per Hour**

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System		Probability of Acquisition	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)	QSIM	Formula
[illegible]	6.58665	6.59195	14.085	14.092	0.761837	0.761688
4.61538	3.52435	3.52744	11.0233	11.0274	0.863421	0.863331
3.33333	2.38806	2.38464	9.88889	9.88464	0.905237	0.905287
2.6087	1.79584	1.7967	9.29802	9.2967	0.927769	0.927749
2.14286	1.44022	1.44006	8.9398	8.94006	0.941622	0.941663
1.81818	1.20024	1.20108	8.69927	8.70108	0.951096	0.951107
1.57895	1.03024	1.02992	8.53028	8.52992	0.957931	0.95793
1.39535	0.901174	0.901361	8.39918	8.40136	0.96309	0.963087
1.25	0.799359	0.801278	8.3002	8.30128	0.967145	0.967121
1.13208	0.721994	0.721167	8.22363	8.22117	0.970346	0.970362
1.03448	0.655689	0.6556	8.15644	8.1556	0.973016	0.973022
0.952381	0.600801	0.60095	8.10308	8.10095	0.975286	0.975244
0.882353	0.554539	0.554701	8.05002	8.0547	0.977172	0.977129
0.821918	0.514868	0.515057	8.01661	8.01506	0.978689	0.978747
0.769231	0.481243	0.480698	7.98247	7.9807	0.980183	0.980152
0.722892	0.4513	0.450633	7.95201	7.95063	0.981443	0.981383
0.681818	0.423846	0.424106	7.92445	7.92411	0.98246	0.98247
0.645161	0.401747	0.400526	7.90441	7.90053	0.983434	0.983437
0.612245	0.379685	0.37943	7.88035	7.87943	0.984324	0.984303
0.582524	0.360966	0.360444	7.86132	7.86044	0.985091	0.985083
0.555556	0.343308	0.343266	7.84038	7.84327	0.985776	0.985789

*Service rate is 8 per hour

*Table D-6. Comparison of QSIM with Prabhu Formulas for Mean Waiting Time in the Queue, Mean Waiting Time in the System, and Probability of Acquisition for M/M/2 Queue with Renege Rate of 2 per Hour**

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System		Probability of Acquisition	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)	QSIM	Formula
	1.15924	1.15923	8.66064	8.65923	0.956415	0.956409
4.61538	0.449371	0.449299	7.94853	7.9493	0.982974	0.982991
3.33333	0.238894	0.23857	7.73971	7.73857	0.990999	0.990969
2.6087	0.147738	0.147961	7.64845	7.64796	0.994404	0.994404
2.14286	0.10039	0.100747	7.59961	7.60075	0.996195	0.996193
1.81818	0.073312	0.0730281	7.57343	7.57303	0.997254	0.997243
1.57895	0.0552606	0.0553679	7.55329	7.55537	0.997918	0.997911
1.39535	0.0434785	0.043424	7.54662	7.54342	0.99836	0.998362
1.25	0.0349282	0.0349693	7.5338	7.53497	0.998682	0.998682
1.13208	0.0286495	0.0287651	7.52623	7.52877	0.998926	0.998916
1.03448	0.0239739	0.0240777	7.52373	7.52408	0.999102	0.999093
0.952381	0.0204971	0.0204498	7.52049	7.52045	0.999236	0.99923
0.882353	0.0177909	0.0175846	7.51993	7.51758	0.999339	0.999338
0.821918	0.0152476	0.0152821	7.51572	7.51528	0.999421	0.999425
0.769231	0.0133731	0.0134041	7.51294	7.5134	0.9995	0.999496
0.722892	0.0118411	0.0118523	7.5113	7.51185	0.999562	0.999554
0.681818	0.0105949	0.0105552	7.51089	7.51056	0.9996	0.999603
0.645161	0.00943042	0.00946005	7.5075	7.50946	0.999643	0.999644
0.612245	0.00846306	0.00852692	7.5095	7.50853	0.999679	0.999679
0.582524	0.00767938	0.00772537	7.50946	7.50773	0.999712	0.99971
0.555556	0.007035	0.00703178	7.50782	7.50703	0.999738	0.999736

*Service rate is 8 per hour

Table D-7. Comparison of QSIM with Prabhu Formulas for Mean Waiting Time in the Queue, Mean Waiting Time in the System, and Probability of Acquisition for M/M/3 Queue with Renege Rate of 2 per Hour

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System		Probability of Acquisition	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)	QSIM	Formula
7.5	0.204564	0.204332	7.70132	7.70433	0.992528	0.992504
4.61538	0.0537731	0.053835	7.55541	7.55383	0.998044	0.998034
3.33333	0.0215666	0.021551	7.52143	7.52155	0.99921	0.999215
2.6087	0.0107505	0.010709	7.51406	7.51071	0.999611	0.999611
2.14286	0.00605784	0.006078	7.50746	7.50608	0.999783	0.999779
1.81818	0.0038534	0.003776	7.50302	7.50378	0.99986	0.999863
1.57895	0.00251681	0.002504	7.50159	7.5025	0.99991	0.999909
1.39535	0.00176683	0.001745	7.50115	7.50175	0.999938	0.999937
1.25	0.00125628	0.001265	7.49978	7.50126	0.999955	0.999954
1.13208	0.000917958	0.000945	7.50017	7.50095	0.999965	0.999966
1.03448	0.000732567	0.000725	7.50301	7.50073	0.999973	0.999974
0.952381	0.000560954	0.000568	7.50251	7.50057	0.99998	0.999979
0.882353	0.000440237	0.000454	7.50089	7.50045	0.999984	0.999984
0.821918	0.00037727	0.000368	7.49938	7.50037	0.999987	0.999987
0.769231	0.000298345	0.000302	7.49891	7.5003	0.99999	0.999989
0.722892	0.000245161	0.000252	7.50007	7.50025	0.999991	0.999991
0.681818	0.00021755	0.000212	7.5006	7.50021	0.999992	0.999992
0.645161	0.000173681	0.00018	7.50263	7.50018	0.999993	0.999994
0.612245	0.000154152	0.000154	7.5001	7.50015	0.999995	0.999994
0.582524	0.000134588	0.000133	7.50209	7.50013	0.999996	0.999995
0.555556	0.000121532	0.000115	7.50282	7.50012	0.999996	0.999996

*Service rate is 8 per hour

*Table D-8. Comparison of QSIM with Prabhu Formulas for Mean Waiting Time in the Queue, Mean Waiting Time in the System, and Probability of Acquisition for M/M/4 Queue with Renege Rate of 2 per Hour**

Lambda (per hour)	Average Waiting Time in Queue		Average Waiting Time in System		Probability of Acquisition	
	QSIM (minute)	Formula (minute)	QSIM (minute)	Formula (minute)	QSIM	Formula
7.5	0.0326052	0.032631	7.53139	7.53263	0.998833	0.99883
4.61538	0.00565102	0.005603	7.50216	7.5056	0.999801	0.9998
3.33333	0.00166301	0.001658	7.49841	7.50166	0.999941	0.999941
2.6087	0.000638069	0.000653	7.50067	7.50065	0.999978	0.999977
2.14286	0.000308724	0.000307	7.50402	7.50031	0.999988	0.999989
1.81818	0.000159394	0.000163	7.49994	7.50016	0.999995	0.999994
1.57895	9.52E-05	9.40E-05	7.50041	7.50009	0.999996	0.999997
1.39535	5.48E-05	5.81E-05	7.50335	7.50006	0.999998	0.999998
1.25	3.89E-05	3.78E-05	7.49941	7.50004	0.999998	0.999999
1.13208	2.69E-05	2.56E-05	7.50142	7.50003	0.999999	0.999999
1.03448	2.00E-05	1.80E-05	7.49902	7.50002	0.999999	0.999999
0.952381	1.36E-05	1.30E-05	7.49913	7.50001	1	1
0.882353	8.78E-06	9.62E-06	7.49978	7.50001	1	1
0.821918	6.36E-06	7.28E-06	7.50124	7.50001	1	1
0.769231	7.42E-06	5.60E-06	7.49878	7.50001	1	1
0.722892	7.22E-06	4.38E-06	7.50011	7.5	1	1
0.681818	2.26E-06	3.48E-06	7.50188	7.5	1	1
0.645161	3.01E-06	2.80E-06	7.49796	7.5	1	1
0.612245	2.75E-06	2.27E-06	7.49913	7.5	1	1
0.582524	2.01E-06	1.87E-06	7.50058	7.5	1	1
0.555556	9.54E-07	1.55E-06	7.50103	7.5	1	1

*Service rate is 8 per hour

REFERENCES

1. Ralph S. Klingbeil and Keith M. Sullivan, "An Analysis of Decision Making Time Delay," NUWC-NPT Technical Memo 03-028, Naval Undersea Warfare Center Division, Newport, RI, 10 February 2003.

2. Donald Gross and Carl M. Harris, *Fundamentals of Queueing Theory,* 3rd edition, John Wiley and Sons Inc., New York, 1998.

3. C. J. Ancker and A. V. Gafarian, "Queuing with Reneging and Multiple Heterogeneous Servers," *Naval Research Logistics Quarterly,* vol. 10, 1963, pp. 125–149.

4. Agner Fog, "Pseudo Random Number Generators," http://www.agner.org/random/.

5. George Marsaglia, DIEHARD, http://stat.fsu.edu/~geo/diehard.html.

6. Jerry Banks, John Carson II, Barry Nelson, and David Nicol, *Discrete Event System Simulation,* 3rd edition, Prentice Hall, Upper Saddle River, NJ, 2001.

7. Sheldon M. Ross, *Introduction to Probability Models,* 7th edition, Harcourt Academic Press, San Diego, CA, 2000.

8. George Marsaglia and Wai Wan Tsang, "A Simple Method for Generating Gamma Variables," *ACM Transactions on Mathematical Software,* vol. 26, no. 3, September 2000, pp. 363–372.

9. John R. Michael, William R. Schucany, and Roy W. Haas, "Generating Random Variates Using Transformations with Multiple Roots," *The American Statistician,* vol. 30, no. 2, May 1976, pp. 88-90.

10. N. U. Prabhu, *Foundations of Queueing Theory, International Series in Operations Research and Management Science*, vol. 7, Kluwer Academic Publishers, Boston, MA, 1997.

BIBLIOGRAPHY

Allen, A. O., *Probability, Statistics, and Queueing Theory with Computer Science Applications,* 2nd edition, Academic Press Inc., Boston, MA, 1990.

Ancker, C. J., and A. V. Gafarian, "Queueing with Impatient Customers Who Leave at Random," *The Journal of Industrial Engineering,* vol. 13, 1962, pp. 84-90.

Ancker, C. J., and A. V. Gafarian, "Some Queuing Problems with Balking and Reneging I," *Operations Research,* vol. 11, 1963, pp. 88-100.

Ancker, C. J., and A. V. Gafarian, "Some Queuing Problems with Balking and Reneging II," *Operations Research,* vol. 11, 1963, pp. 928-937.

Ayyub, B. M., and Richard H. McCuen, *Probability, Statistics, & Reliability for Engineers,* CRC Press, New York, 1997.

Chhikara, R. S., and J. Leroy Folks, *The Inverse Gaussian Distribution: Theory, Methodology, and Applications*, Marcel Dekker Inc., New York, 1989.

Cormen, Thomas, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms,* 2nd edition, The MIT Press, Cambridge, MA, 2001.

Hoel, Paul G., Sidney C. Port, and Charles J. Stone, *Introduction to Probability Theory,* Houghton Mifflin Company, Boston, MA, 1971.

Horton, I., *Beginning Visual C++,* Wrox Press Ltd., Birmingham, UK, 2001.

Johnson, N. L., S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions, Volume 1,* 2nd edition, John Wiley & Sons Inc., New York, NY, 1994.

Law, Averill, and David Kelton, *Simulation Modeling and Analysis,* 3rd edition, McGraw-Hill, Boston, MA, 2000.

Prosise, J., *Programming Windows with MFC,* 2nd edition, Microsoft Press, Redmond, WA, 1999.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Defence Science and Technology Organisation, Rockingham, Western Australia (C. Davis)	1
Defence Science and Technology Organisation, Ministry of Defence, Edinburgh, Australia (M. Fewel, I. Grivell (5))	6
Defence Research and Development Canada, Dartmouth, Nova Scotia, Canada (M. Hazen)	1
Maritime Operational Research Team National Defence Headquarters, Ottawa, Ontario, Canada (R. Burton)	1
Maritime Systems Group, Defense Technology Agency, Her Majesty's New Zealand Naval Base, Auckland, New Zealand (S. McMillian)	2
Defence Science and Technology Laboratory, Portland, United Kingdom (P. Marland)	1
Space and Naval Warfare Systems Command, SPAWAR Systems Center, San Diego, CA (G. Galdorisi)	1
Defense Technical Information Agency	2
Center for Naval Analyses	1