# Evaluation of Queueing System Parameters Using Linear Algebraic Queueing Theory - An Implementation

Lester Lipsky, Dilip Tagare and Edward Bigos
Department of Computer Science and Engineering, U-155
University of Connecticut, Storrs, CT 06268

## Abstract

**It is important to study the varied characteristics of queueing networks, in a self-consistent manner and interactive environment. We report here the development of an efficient and practical software package (QST) for the analysis of a particular class of queueing systems, (those described by two subsystems which interact only through the exchange of customers) which can run on an IBM PS/2 Model 70 or equivalent, or with some restrictions, on 80286-based PC's. The package produces data (or graphs) of various performance (including residual behavior) characteristics for these generalized queueing loops. It also generates such transient properties as "Busy Period", "Rush Hour", and "Mean Time to Failure". The equations, which are transparent to the user, come from the "Linear Algebraic Approach to Queueing Theory". For now, the package has been developed within the MATLAB environment. We provide some examples of input and output.**

## Introduction

Queueing theory is used extensively in studying the behavior of any system where contention for resources is significant and service demands are only known to some probabilistic uncertainty. In general a queueing system can be made up of any number of subsystems, but the behavior of even the simplest ones is not easy to predict. Though solutions to a wide variety of systems are known, software packages which can calculate their performance characteristics in a self-consistent manner in an interactive environment are not readily available. The Linear Algebraic approach to Queueing Theory (LAQT) provides a theoretical framework which can handle the diverse equations and distributions required to study various aspects of queueing theory in detail. Since personal computers are now powerful enough to handle intense matrix computations, we have written an efficient and practical software package for the analysis of queueing systems, which can run on a IBM PS/2 Model 70 or an equivalent 80386 PC. A restructured version can run on 80286 PCs.

We are interested in queueing networks which can be partitioned into two sub-networks that interact only by the exchange of customers. The sub-networks are made up of non-exponential and load dependent components. For the present, we have assumed that no blocking occurs for those customers who are already being served, and that no customer can hold two resources simultaneously, although more general networks can be accomodated within our framework. We study not only the steady-state behavior [5, 7], but also transient properties such as "Busy Period" [5], "Rush Hour", "Time to drain" (also called Mean Time to Failure - MTTF) [6].

In this paper we describe a package (QST - Queueing System Tool) developed in the MATLAB environment [2] that calculates various performance characteristics of interest. MATLAB was chosen because of its ease in implementing matrix operations (transparent to the user) and its convenient graphics capabilities.

## Theoretical background

Consider the system shown in Figure 1. It is made up of two sub-systems $S_1$ and $S_2$. The total number of customers in the system is N. At any time, $S_1$ has n customers, while $S_2$ has k customers, where n+k = N. The service time distributions at $S_1$ and $S_2$ define the type of system. The usual approach to queueing theory is to define a distribution b(x) and then find all the parameters. But this is difficult to do if a system as shown in Figure 2 is given.

The LAQT approach allows us to define a sub-stochastic transition matrix P, an entrance vector p, and a completion rate matrix M, and using these as inputs, QST comes up with the distributions of service times and their nth moments $E(x^n)$ . That is, we define the following [3]:

$\rho$ := utilization factor, given by the ratio of average service times of the two servers ( $\bar{x}_1/\bar{x}_2$ ) .

**p** := Entrance vector, whose $i^{th}$ component is the probability that a customer, upon entering $S_1$, will go to server i.

**P** = sub-stochastic transition matrix, whose $ij^{th}$ component is the probability that a customer who has just finished service at i will go to j.

$\varepsilon'$ := [1 1 1 $\cdots$ 1]' where (') denotes transpose.

**M** = completion rate matrix, where $M_{ij} = \mu_i \delta_{ij}$ and $\mu_i$ is the service rate of the $i^{th}$ stage in $S_1$.

$\mathbf{V} = \mathbf{B}^{-1}$ , where $\mathbf{B} = \mathbf{M}(\mathbf{I}-\mathbf{P})$ is the service rate matrix and **V** is the service time matrix.

$\psi[\ ]$ is a linear operator where for any matrix **X**,

$$\psi[\mathbf{X}] := p\ \mathbf{X}\ \varepsilon'.$$

Then,

$$b(x) = \mathbf{p}\ \mathbf{B}\ exp(-\mathbf{B}x)\ \varepsilon' = \psi[\mathbf{B}\ \exp(-\mathbf{B}x)],$$

and

$$E(x^n) = \int_0^\infty x^n\ b(x)\ dx = n!\ \psi[\mathbf{V}^n]$$

From this, the user can ask e.g., for steady state queue length distributions, r(n), for arrival rate $\lambda$ to an M/G/1 queue. Then [4, 10],

$$r(n) = (1-\rho)\ \psi[\mathbf{U}^n]$$

where $\mathbf{A} = \mathbf{I} + \frac{1}{\lambda}\mathbf{B} - \mathbf{Q}$ , $\mathbf{U} = \mathbf{A}^{-1}$, and $\mathbf{Q} = \varepsilon'\ \mathbf{p}$

Similar formulae exist for the evaluation of numerous other system characteristics, all dependent on the same matrices (See [1] for full details).

## Overview of the Software

QST consists of a collection of utilities in an interactive environment which is provided by the MATLAB matrix evaluation software. MATLAB keyboard commands allow variables to be created, displayed, and modified. The interactive nature of the package encourages experimentation. MATLAB was chosen to quickly produce software prototypes and explore new methods, but, the finished product will be translated to C. Utilities are constructed as MATLAB "m-files" which are disk resident scripts of MATLAB commands. Utilities are provided to enter the system description matrices, evaluate the system, and to plot the results. Users can create their own utilities and expand the system.

## System Characterization

A queueing system is characterized by a consistent set of matrices. The evaluation utility calculates the system parameters from this basis. Matrices which describe the queueing system can be entered in a number of ways. Initially the user is prompted for the number of stages in $S_1$. The utility then prompts the user for the correct number of row and column values for **M** , **P** , and **p**. Alternately, the matrices for common distributions can be created. For instance, an Erlangian-k distribution can be characterized by entering the number of internal stages then calling the Erlangian-k matrix building utility. Since numeric data entry is not practical for large matrices, the matrix building utilities are useful. For example to create the matrices for an Erlangian- 60 server the user would create a variable "STAGES", set it to sixty, then call the Erlangian-k utility. The utility creates the 60 X 60 **M** and **P** matrices and the 1 X 60 entrance vector **p**. Advanced users can bypass the data entry utility or write their own for specialized applications, but more detailed knowledge of the MATLAB system is required.

## Calculating System Parameters

QST calculates the system parameters for a single run or performs repetitive calculations. Input to the evaluation utility is a list of $\rho$ values and a list of N values. At start up the primary memory is checked for the existence of the parameter lists. If they do not exist they are created and set to default values. The user is then given the opportunity to retain the current input parameter list or create a new list. On repetitive runs the user can retain the same parameter lists for any number of queues. When the lists contain a single parameter only a single evaluation of the queueing system is performed. When a list contains more than one parameter the evaluation utility calculates the queueing system equations for each value in the list. The two lists are scanned in a nested loop structure with the scan of the N list as the inner-most loop. This allows the system to avoid unnecessary calculations by retaining some partial results. The selection of $\rho$ is used to calculate a set of parameters used in the processing of the N list and void any previous partial results. The user has the choice of using the input data verbatim or selecting automatic cycle time normalization. The value of $\bar{x}$ is calculated from the **M** , **P**, and **p** input data. $\lambda$ is calculated from the definition of $\rho$. The list is processed with the assumption that the previous calculations are valid for the entire list. Vector operations are used in the place of matrix operations whenever possible to lower the order of

complexity of the calculations.

Information on the queueing system can be displayed or collected in a summary. The default is to display the information to the screen. The user can also collect the data in summary form within primary memory. The summaries are useful for long run times or the comparison of similar systems. The user selects parameters of interest then runs the evaluation utility like a batch job. The summary information can be examined after the run, and screen and hardcopy plots can be generated from the summary. Alternately all the calculated information can be saved in a disk file. The data is converted to text form by MATLAB then written to a disk file. This option can dramatically increase the run time, but, it is useful when all the system parameters are of interest or when primary memory becomes a limitation. The increase in run time is due to the conversion process and the disk accesses.

In the MATLAB implementation of QST, variables and executing m-files compete for free memory. MATLAB '386 runs in 80386 protected mode. It uses as much memory as is installed in the machine. Virtual memory which pages to disk is not available. MATLAB for the 80286 and 8086 machines uses memory up to the 640k limit. Large matrices can cause the system to abort an evaluation due to insufficient memory. Test runs were made using MATLAB '386 in a machine with 640k of memory and in a machine with 4 MB of memory. With 640K of memory and N, the number of customers fixed at 20 the evaluation software can process a queueing system specified by 65 X 65 element matrices. When the matrix size is fixed at 20 X 20 the number of customers in the loop can be increased to 250. When the memory is increased to 4MB and N, the number of customers fixed at 20 the evaluation software can process a queueing system specified by 200 X 200 element matrices. When the matrix size is fixed at 20 X 20 the number of customers in the loop can be increased to 3000. During the testing no summaries were retained in primary memory.

Even five years ago such calculations required a mainframe or super mini-computer to perform. The advent of personal computers with the performance of some mini-computers, high resolution graphics, large memories, and an efficient computational method makes these investigations practical on a personal computer. The one major deficiency in the personal computer software is the lack of paged, virtual memory. Problem size is limited to the physical memory installed in the machine.

**Plotting Output**

QST provides basic plotting functions. The plot utility was developed to simplify the production of standard plots and to minimize the work necessary to produce additional plots. Once the plotting variables are defined the user can produce similar plots without new setup information. Setups for common problems can be saved in a plot specification file for future use. The plot utility is supplied with a vector which specifies the x axis values and one or more domain vectors. The domain vectors may be packed into a single matrix. Summary information is often packed into a matrix during the evaluation process. One or more of the columns of the summary can be plotted against the common x axis. Plots can be made on the screen or hardcopies can be generated. More than one plot can be displayed on the same screen or hardcopy.

**An Example**

The software package has been set up where server $S_2$ has exponentially distributed service times and server $S_1$ can be made to have any distribution of service times within the class of "matrix exponential" functions. As an example, we chose the distribution

$$b(x) = 0.9\,x\,e^{-x} + 0.1\,(0.1\,e^{-0.1x})$$

represented by $S_1$ as shown in Figure 3. This distribution has a mean service time of $\bar{x}_1 = 0.9\,(2) + 0.1(10) = 2.8$ . QST requires a matrix description of the distribution. The matrix description of this system is given by:

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{p} = [0.9 \;\; 0 \;\; 0.1], \quad \text{and} \quad \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

We specified a list of several values for ρ and N. Once specified, the evaluation tool proceeded to calculate the system characteristics. Typical output for the steady-state queue length probabilities, residual times, system distribution, and throughput has been plotted in Figures 4 through 7.

**Summary**

QST currently evaluates the steady state queue length probabilities, system throughput, queueing time, system time (sum of queueing time and service time), relaxation (transient) time, and the residual times for the M/G/1 and G/M/1 queues. In the near future, we will implement software to evaluate distribution of interdeparture times, busy periods, time of queue-length growth, and multiple server queues. The long term goal of this research is to evaluate characteristics of M/ME/C//N and ME/ME/1//N queues and even generalized ME/ME/C//N queues. The last class of queues is equivalent to all Quasi-Birth-Death processes.

## References


[1] Lipsky, Lester, "A Linear Algebraic Approach to Queueing Theory", University of Connecticut Technical Report BRC/CSE-TR-89-22, 1989

[2] **PC - MATLAB User's Guide** The Math Works Inc., 1987

[3] M. Neuts, **Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach**, Johns Hopkins University Press, 1981

[4] M. Neuts, "Explicit Steady-State Solutions to Some Elementary Queueing Models", **Operations Research** 30, pp 480-489, 1982

[5] Leonard Kleinrock, **Queueing Systems, Volume 1: Computer Applications**, John Wiley, New York, 1975

[6] Kishor Trivedi, **Probability & Statistics with Reliability, Queueing, and Computer Science Applications**, Prentice-Hall 1982

[7] A. Tehranipour, L.Lipsky, A. van de Liefvoort, "Residual Lifetimes as a Function of Queue Length for M/G/1//N Loops",**Joint ACM-IEEE Workshop on Applied Computing, 1989**

[8] A. van de Liefvoort, L. Lipsky, "An Explicit Matrix Algebraic Solution to Steady-State G/G/1//N Type Loops", **Journal of the ACM,** 33, pp 207-223, January 1986.

[9] L. Lipsky, "Explicit Solutions to M/G/C//N Type Queueing Loops with Generalizations", **OPerations Research**, 33, pp 911-927, July 1985

[10] J. L. Carroll, L. Lipsky, A. van de Liefvoort, "Solutions of M/G/1//N Type Loops, with Extensions to M/G/1 and GI/M/1 Queues", **Operations Research**, 30, pp.490-514, May 1982.


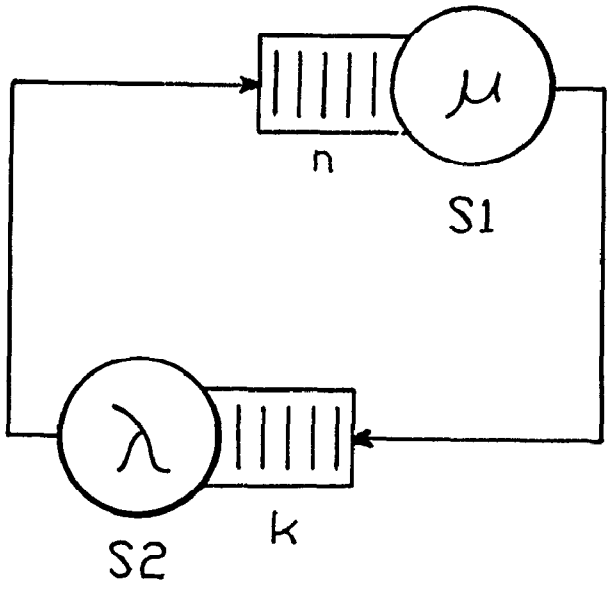


**Figure 1.** A closed loop with 2 subsystems, $S_1$ and $S_2$. Only one customer at each subsystem can be in service at one time.

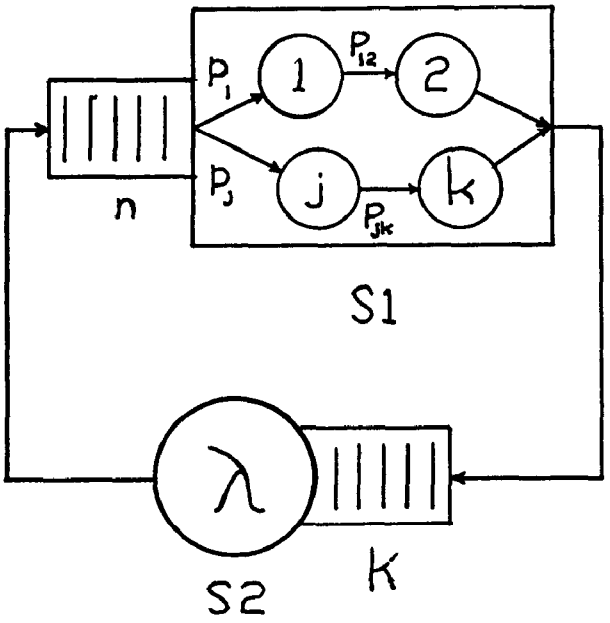


**Figure 2.** A closed loop with two subsystems. $S_2$ is a pure exponential server with service rate λ, and $S_1$ is a general server represented by a matrix exponential distribution. There are n customers at $S_1$ with the active customer at [internal] server i. There are k customers at $S_2$. The total number of customers in the system is N = n + k.

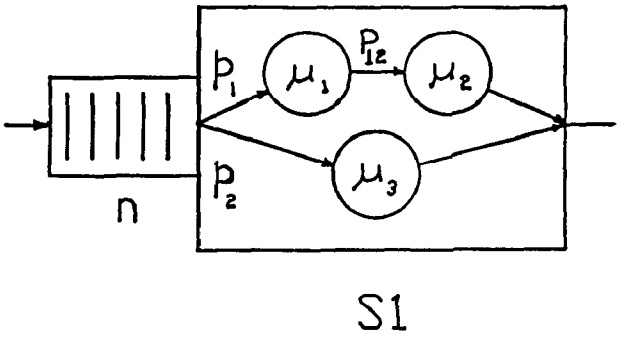


**Figure 3.** An internal representation of $S_1$. Upon entry a customer can be served by the 2-stage Erlangian path with probability .9 or is served by the single stage with probability .1. Each Erlangian stage has a service rate of $\mu_1 = \mu_2 = 1.0$. The single stage path has a service rate $\mu_3 = 0.1$.

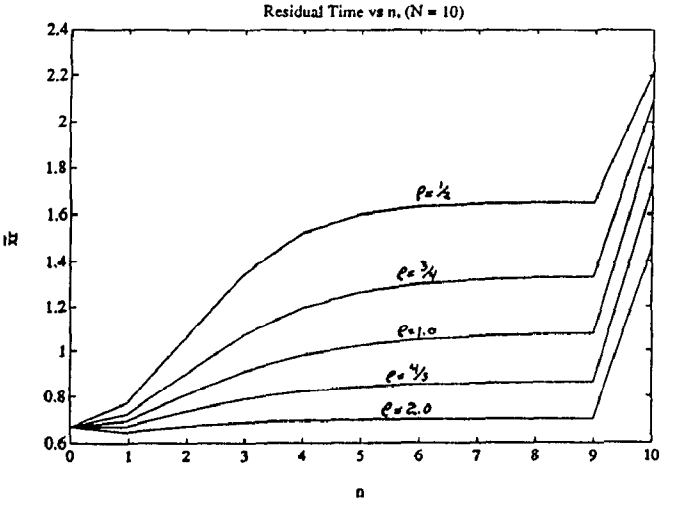


**Figure 5.** Plot of the residual times, $\bar{x}_r$, vs n for $\rho = 1/2$, 3/4, 1, 4/3, 2 and $S_1$ as shown in Figure 3.

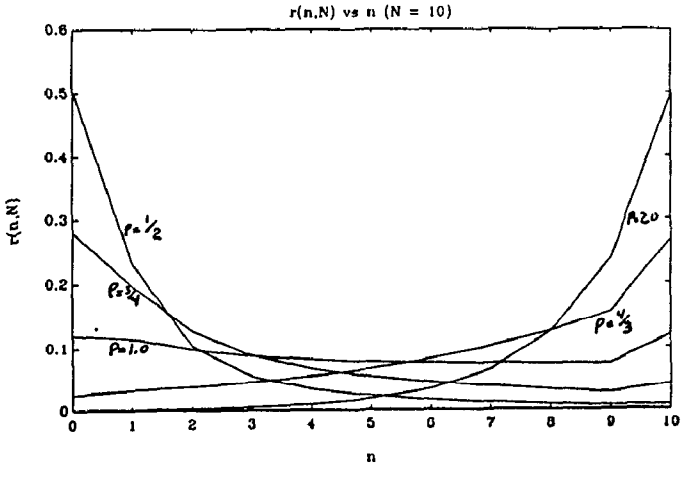


**Figure 4.** Plot of the steady-state queue length probabilities, r(n,N), vs n for $\rho = 1/2$, 3/4, 1, 4/3, 2 and $S_1$ as shown in Figure 3.

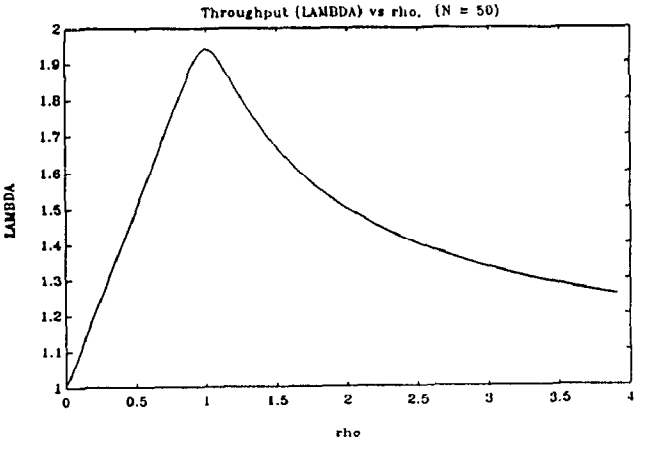


**Figure 6.** Plot of the system throughput, $\Lambda$, vs $\rho$ for N = 50 and $S_1$ as shown in Figure 3.

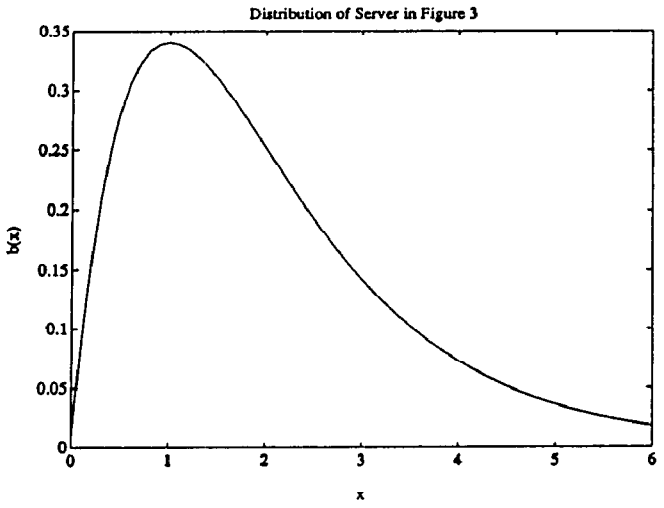


**Figure 7.** Plot of the system distribution, b(x), vs x for the $S_1$ shown in Figure 3.