*Proceedings of the 1991 Winter Simulation Conference*
*Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)*

# THE ROBUSTNESS OF SEPARABLE QUEUEING NETWORK MODELS

Charles E. Knadler, Jr.

IBM Corporation
9201 Corporate Boulevard
Rockville, Maryland 20850

## ABSTRACT

Discrete event simulation is used to explore the robustness of queueing network models, a set of powerful analytical techniques for evaluating computer system performance. These techniques are shown to give good results even when some of the basic assumptions used to derive them are violated. It is also shown that simulation provides additional insights, into computer system performance, not available when using the mean value analysis characteristic of separable queueing network approaches.

## 1 INTRODUCTION

Queueing network models have proven to be powerful tools to analyze computer systems (Lazowska et al 1984, Lipsky 1977). Buzen and others have shown that much less restrictive assumptions are required to develop an operational based theory of computer system performance than those required to develop Markovian queueing network theory (Buzen 1976, Denning and Buzen 1978, Lazowska et al 1984). However, several widely used solution techniques are predicated upon the queueing networks being separable.

The separable network solutions are based on the following assumptions:

- service center flow balance
- no two jobs change state at exactly the same time
- routing of jobs is independent of service center queue lengths
- service times are independent of the number and placement of jobs at other centers within the network
- the arrival times of jobs external to the network are independent of the number and placement of jobs within the network.

Specific solutions (Lazowska et al 1984) also require load independent exponentially distributed service times.

Computer systems commonly violate certain of these assumptions.

(1) Simultaneous possession of resources violates the assumption: ***no two jobs change state at exactly the same time.*** Examples of simultaneous possession of resources are memory constrained systems (memory and cpu) and modern disk systems (channel, controller, and disk).

(2) Rotational position sensing (rps) disk systems violate the assumption: ***service times are independent of the number and placement of jobs at other centers within the network.*** In rps systems, the disk unit determines if the path to the processor is available or not, when the requested sector is about to rotate under the disk read or write head. If the path is available, it is seized and the data transfer takes place. If the path is busy, the disk rotates one revolution and process is repeated. Thus each time a disk finds the path busy, the data transfer is delayed for one rotational period. Disk service time is therefore a function of the channel utilization which is, in turn, a function of the existence of jobs at other disks (service centers).

(3) Both cpu and disks may violate the assumption: ***service times are load independent and exponentially distributed.*** As discussed above, disk service times, may not be independent and often may not be exponentially distributed. In a multiprocessor, task execution rates are usually a function of tasks running concurrently on other processors as a result of resource contention.

In this paper, we explore the robustness of mean value analysis (MVA) of queueing network models as these assumptions are violated. Substantial analytical work has been done to develop the techniques of queueing network models from operational principles (Buzen 1976, Denning and Buzen 1978) and using the results of Markovian queueing theory (Lazowska et al 1984, Zahorjan 1982). But as far as their usefulness, less work has been published. Case studies, such as those discussed in Lazowska et al (1984), show good agreement with a limited number of measurements, but this author is not aware of a systematic study of their applicability and accuracy. Simulation based, experimental studies are suggested as a technique to investigate these issues.

The suggested approach includes 6 steps:

(1) Determine the system characteristics, which potentially impact the accuracy of the analytical models.

(2) Design an experimental program to determine the sensitivity of the analytical model to these characteristics.

(3) Design and implement a simulation test bed for the experimental study. The results, reported later in this paper, demonstrate that relatively simple simulations can be used to meet the objectives of (2).

(4) Carry out the experimental program.

(5) Evaluate the results, and iterate through steps (2)–(5) as required.

(6) Refine the analytical models to address the sensitivities discovered through this process.

This paper illustrates the application of steps (1)–(5) of this proposed technique with a study of a relatively simple multitasked computer system. Discrete event simulations are constructed which selectively violate the assumptions of separable queueing networks. Then queueing network analysis is used to generate the solutions of these same networks and the accuracy of the models is determined by comparing the two sets of results.

## 2 THE MODEL

A discrete event simulation of multitasked computer system is used to study the accuracy of network queueing analysis. The system modelled is a uniprocessor with a single channel and four disk drives. The disk subsystem uses rotational position sensing (rps). Tasks are characterized by the number of visits to the cpu and disk subsystems, their service demands, and main memory requirements. The simulation is based on a disk input/output model described in Knadler and May (1990) and the disk system performance was validated by comparing with measurements by Olson (1989).

The two simulated multitasked computer systems are shown in figures 1 and 2. They are a multitasked batch system and an interactive system. The interactive system is a memory constrained central processor attached to a network of interactive users and an input/output system. Processor tasks have exponentially distributed service time per visit. The number of visits per task and task memory requirements are uniformly distributed. Disk data requirements are expressed as a uniformly distributed number of sectors requested per

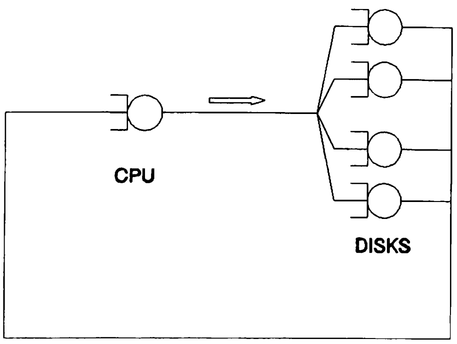


Figure 1: Batch Computer System

access.

A task (either a terminal system request or a new batch job) requests memory when it enters the system and holds it until it completes all visits to both the central processor and the disks. This mechanism matches either the allocation of virtual memory in a paged system (queueing is for virtual memory, not real memory) and the allocation of real memory in a segmented system.

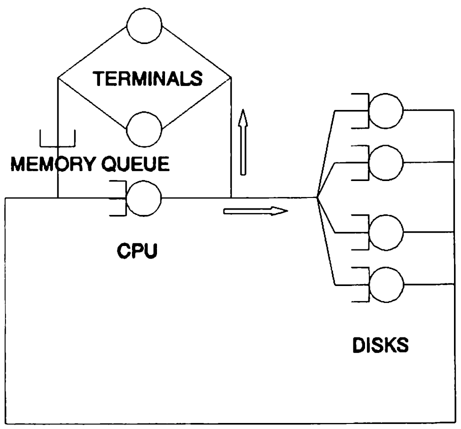


Figure 2: Interactive Computer System

This relatively simple model provides a good benchmark to evaluate some key queueing network model techniques against. A simulation allows very tight control over system characteristics and thus the impact of violating particular assumptions, used to develop the techniques of mean value and queueing network analysis, may be determined.

## 3 SINGLE CLASS MODELS

In single class models, the customers are considered to be indistinguishable from each other. The service time inputs are the average values for all customers and by the same token the outputs are mean values representing all customers. The main advantage of single class models, compared to multiclass models, is the reduced parameterization requirements and the main disadvantage is that results are not available for distinct workload components.

### 3.1 Exact Solution

The exact solution for closed single class models requires the iterative solution of equations (1) for n = 1 to N, where N is the number of jobs in the system and the variables are as defined in Table 1 (Lazowska et al 1984).

$$
\begin{aligned}
R_k(n) &= D_k[1+Q_k(n-1)] \\
X(n) &= \frac{n}{Z+\sum_{k=1}^{K} R_k(n)} \\
Q_k(n) &= X(n)R_k(n)
\end{aligned}
\qquad (1)
$$

Table 1. Single Class Model Variables

| Variable | Definition |
|---|---|
| $R_k(n)$ | mean residence time for queueing center k with a system customer population of n |
| $D_k$ | mean service demand for queueing center k (service time) |
| $Q_k(n)$ | mean queue length for center k when n customers are in the system |
| X(n) | mean system throughput, with n customers in the system |
| K | the number of queueing centers in the network |
| Z | mean think time for interactive (terminal) users; equals 0 for batch systems |
| n | the number of customers in the system |

### 3.2 Comparison with Simulation Results

#### 3.2.1 Batch Processing Systems

A series of simulation experiments were run for the uniprocessor model with a multitasking, batch workload. The simulated system satisfies the basic assumptions of mean value analysis. While the disk subsystem complexity is typical of modern hardware and would violate these assumptions if the input/output service process were subdivided into seek, latency and transfer times (Knadler and May 1990); the queueing network model, equations (1), handles this case with no difficulty since the disk service times include the combined contributions of all components in the subsystem.

The parameters of the queueing network model are taken from the first of five (5) simulation runs made for each data point. This method of parameterization was chosen, because of its analogy to a frequently used queueing network model parameterization technique. Namely, make measurements of a real system to determine average values for central processor and input/output service times to obtain the required service demands. The simulation and analytical throughput and residence time results are shown in figure 3. Figure 3 is drawn with dual y–axes. Throughput values, X: Sim and X: MVA, are plotted versus the left hand axis and the residence time values, R: Sim and R: MVA, are plotted versus the right hand axis. The simulation results are the sample means for the series of five simulation runs for each task population.

We see that there is good agreement between the simulation and analytical results. The analytical results for residence time are within 15% of the simulation results and the throughput results are within 11% of the simulation results. Interestingly the analytical results are slightly closer to the sample means than to the particular runs used for their parameterization. These results are an indication that queueing network models have sufficient accuracy to perform tuning and sensitivity studies of batch processing systems.

#### 3.2.2 Interactive processing systems

The batch system simulation is easily modified to model an interactive system. A new simulation event is added at task completion to simulate the time a terminal user spends thinking about his/her next system request.

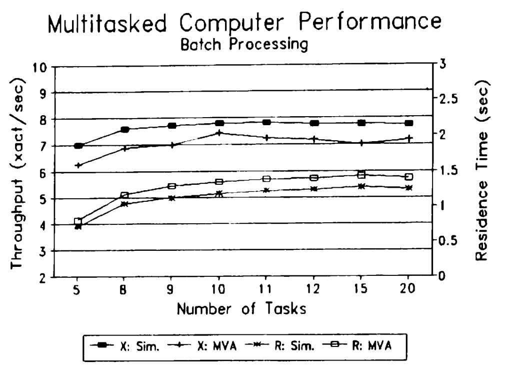


Figure 3: Relative Performance of Simulation and Analytical Models of a Batch Computer System

Think time is modelled as a uniformly distributed random variate with specified mean. The mean think time is varied to produce the results shown in figure 4.

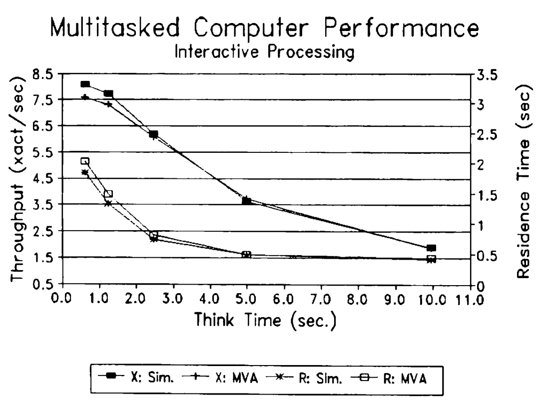


Figure 4: Relative Performance of Simulation and Analytical Models of an Interactive Computer System

The queueing network model provides even better agreement for the interactive cases than for the batch cases. Possible explanations for the better interactive results are the dominance of think time in the throughput equation and the increased randomness in the interactive workload intensity compared to the batch workload. These conjectures are supported by the observation that the agreement between analytical and simulation results improves with increasingly large think times.

Mean think time is a difficult parameter to measure for real systems and is often calculated with the MVA throughput equation, using measured values of throughput and response time. An inaccuracy in estimating Z will potentially cause the performance measures calculated either from simulation results or from analytical models to be incorrect. Experience suggests two techniques to solve problems of this type. First, perform sensitivity studies and vary the unknown parameters over the range of possible values to determine the sensitivity of results. If the results are relatively insensitive to these variations, good estimates of the performance measures are possible, else a range of possible values is found.

Second, if only one parameter is unknown, either or both simulation and MVA analysis may be used to estimate it. Vary the parameter in the simulation or analytical model to get the closest possible agreement with the measurements of performance measures: throughput, residence times, queue lengths and/or utilizations. (The use of this technique to estimate the service demands for a disk subsystem is illustrated in the **Disk System Analysis** section of this paper.)

### 3.2.3 Simultaneous Resource Possession

Memory constrained computer systems may not be separable, because of a violation of the assumption that there is no simultaneous resource possession. If the memory constraint is sometimes, but not always reached the model will not be separable. There are three cases to be considered for an interactive computer system. First, there is sufficient memory (real or virtual) for all active users and the interactive model with nonzero think time is indicated. Second, the memory demand always exceeds the available memory and the system performs like a batch system with average multiprogramming level, $n_0$, less than the number on interactive users and should be modelled as a batch system with $z = 0$ and $n = n_0$. Third, the memory constraint is some times reached, but not always and neither the batch nor interactive model is valid.

In order to determine the sensitivity of the accuracy of the analytical models, an interactive constrained memory system is modelled with variable think time to produce a series of cases in which the frequency with which the computer system memory constraint was reached on task arrival ranges from 0.0% to 68.0%. The analytical and simulation throughput and residence time results are plotted versus the fraction of time that an arriving customer has to wait for memory to become available. The performance measures are shown in figure 5.

The analytical throughput results are within 4.3% for all cases, but the residence time results are seen to diverge with increasing memory queueing with differences of 2.4%, 9.2%, 11.3%, 17.0%, 21.3%, 20.6%, 35.5% and 62.9% respectively. The divergence in residence time results is easily explained by the observation that the analytical model projects a larger computer system population, than is the case since some

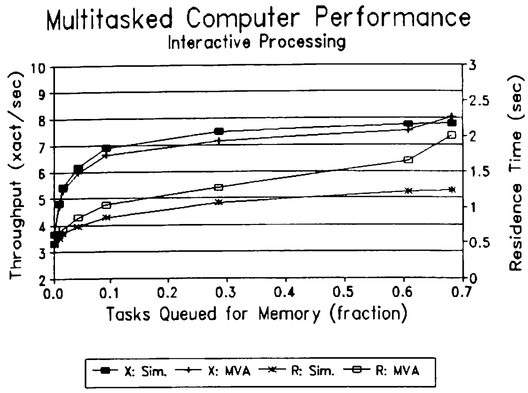


Figure 5: Relative Performance of Simulation and Analytical Models of a Memory Constrained, Interactive Computer System

users are queued for memory.

Also the excellent agreement between throughput results should not be surprising, the memory constraint is met only when there are sufficient tasks in the computer system to effectively utilize both the central processing unit and disks.

These results show that the analytical models are quite robust, as the residence time results are within 21% of the simulation results with up to 29% of the arriving tasks queueing for memory.

An extension to these analytical techniques incorporates load dependent service centers (Lazowska et al 1984). This extension is a hierarchical approach in which the memory constrained processor and the disks are replaced, in the analytical model by a flow equivalent service center (FESC) which has the same mean throughput and residence time. The load dependent throughput of the flow equivalent service center is found by solving equations (1) for all possible populations of the central processor system.

Throughput and residence time are shown in figure 6 as functions of think time for the simulation, equation (1) analytical results and the FESC extension to equation (1). The FESC results do not agree as well with the simulation results as do the basic interactive queueing network results. The poorer agreement of residence times can be explained as follows: First, the FESC service demand is equal to the sum of the mean simulation tasks' cpu and disk service demands, thus an arriving task must wait longer for its initial service than if the tasks ahead of it had the shorter service times characteristic of the visits to the cpu and disks. Second, the FESC includes the time spent waiting for memory in the residence time, as contrasted to the simulation and interactive queueing network models which only include the time spent in the cpu and disk service centers and their queues. Thus the FESC residence time will increase sharply for shorter think times reflecting the saturation of main memory.

By Little's Law (Gross and Harris 1985, Kleinrock 1975, Lazowska et al 1984) this error in residence time induces a corresponding error in throughput. The impact, of this residence time error on throughput, should decrease with increasing think time as is seen from the throughput equation in equations (1). Z will begin to dominate the denominator as it grows. This effect is observed in figure 6, where throughput agreement improves as think time increases.

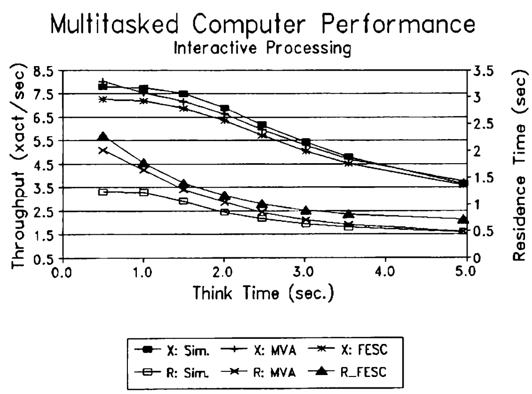


Figure 6: Relative Performance of Simulation, Analytical, and Flow Equivalent Service Center Models of a Memory Constrained, Interactive Computer System

While the results of a single test case should not be taken as an indication that this technique is without value, these results suggest that simple FESC's should be used with caution, because of the inherently different environment seen by arriving customers to a single service center versus the environment seen by an arriving customer in a network.

### 3.2.4 Disk System Analysis

If sufficiently accurate measurements or estimates of disk subsystem service times are not available then a more detailed disk subsystem model is required to obtain them. Lazowska et al (1984) describe the following iterative approach for rps disk systems

(1) Define the queueing network of the system, representing the input/output system only as disks and assume the system throughput, X, equals 0.

(2) Iterate

2.1 For each disk k, model its contribution to the total channel utilization as

$U_{ch}(k) = X \cdot V_k \cdot transfer_k$

2.2 Calculate channel utilization

$U_{ch} = \Sigma U_{ch}(k)$

2.3 For each disk, set

$retries_k = (U_{ch} - U_{ch}(k))/(1 - U_{ch})$ and

$D_k = V_k\ (seek_k + latency_k + transfer_k + (retries_k + rotation_k))$

2.4 Solve equations (1) using these $D_k$'s for the disks

Repeat step 2 until the successive values calculated for X are sufficiently close, say 0.1%

(3) Use the performance measures from the last iteration.

The variables in the algorithm not previously defined are as follows: $seek_k$, $latency_k$, $transfer_k$, and $rotation_k$ are respectively the average seek, latency, transfer, rotation times for the $k^{th}$ disk and $retries_k$ is the mean number of tries that the $k^{th}$ disk must make before it finds the path to the cpu available. $V_k$ is the average number of times that a job accesses $disk_k$ and $U_{ch}$ and $U_{ch}(k)$ are respectively the channel (bus) utilization and $k^{th}$ disk's contribution to it.

Applying this technique to the same nonmemory-constrained, interactive computer system modelled above provides the results shown in figures 7 and 8, where simulation results are labeled with "Sim.", standard MVA results are labeled with "MVA", and the iterative results with "rps".

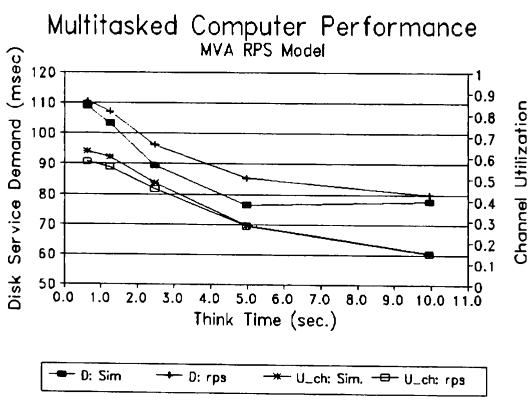


Figure 7: Analytical Model of Disk Service Demand and Channel Utilization

The iterative approach is shown to produce excellent estimates of channel utilization and good estimates of disk service demand. These estimates of basic disk subsystem performance measures, then produce the excellent agreement with both the simulation and equation (1) MVA results, seen in figure 8. This iterative approach has the additional very real advantage of not requiring measurements of disk service demand. This parameter is very load dependent, as is seen in figure 7, and therefore difficult to measure and estimate.

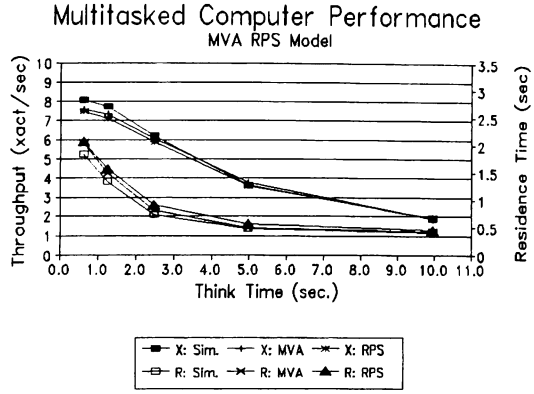


Figure 8: Relative Performance of Simulation, Analytical, and Iterative Models of an Interactive Computer System

### 3.2.5 Nonexponentially Distributed Services Times

The ***independent exponentially distributed service time*** assumption is used to derive an estimate for the remaining service time for the job in service at the instant that a new job arrives. The significance of the impact of nonexponential service time distributions can be determined by rerunning one of the test cases with various service times distributions. System throughput, X, and residence time, R, are shown in figure 9 for seven simulation test cases made with the same mean cpu service time, but with different standard deviations. Each data point represents the average of five simulation runs and is plotted versus the cpu service time coefficient of variation. Where the coefficient of variation is defined to be the ratio of the standard deviation to the mean.

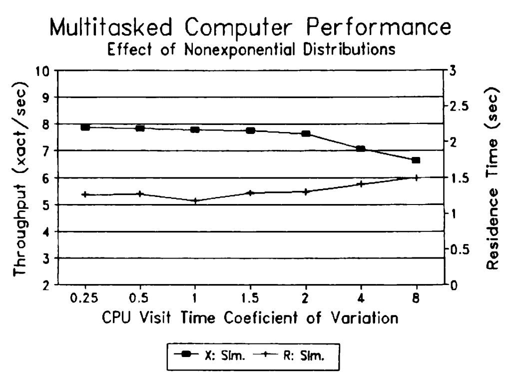


Figure 9: Variation of Batch Computer Throughput and Residence Time with the CPU's Visit Time Coefficient of Variation

Throughput results varied by 15% and residence time by 27% as the coefficient of variation ranged between 0.25 and 8.0, recall that the exponential distribution's coefficient of variation is 1.0. With coefficients of variation between 0.25 and 2.0, both throughput and residence time vary by only 11%. Again we see that the accuracy of queueing network results is relatively insensitive to the assumptions required to derive the analytical model.

## 4 CONCLUSIONS

The analytical models have been shown to be very robust for the simple computer systems considered by this paper.

(1) Simultaneous resource possession. The analytical models provided good estimates of throughput and residence time with as many as 25% of all interactive jobs queueing for memory (i.e., violating the assumption).

(2) Load dependent service demand. The mean value analysis provided excellent results for rps disk systems over a broad range of channel utilizations, using a simple iterative extension. The simple flow equivalent service center (FESC) approach was shown to provide poorer results, than simply ignoring the lack of separability and using the basic MVA model.

(3) Independently distributed service times. Clearly the disk service times are not independent of each other, as a result of channel contention. Yet excellent results were obtained. The analytical models are not necessarily equally amenable to other forms of dependence.

(4) Exponentially distributed service times. Simulation results showed performance measures to be relatively constant (within 15%) as a function of the coefficient of variation of the cpu service times, but with greater sensitivity for coefficients of variation larger than 1, than for those less than 1.

This paper does not present a final evaluation of network queueing models, but rather indicates a marked robustness. A simulation based methodology is suggested and demonstrated for the evaluation of the suitability of queueing network analysis for particular problem sets.

### 4.1 Queueing Network Model versus Simulation

There is an ongoing debate between the use of simulation and queueing network models, possibly missing an important point. They can be used to supplement each other. Based on the author's experience, the major cost of a queueing network study is parameterization. With a properly designed effort, this parameterization could be a virtually free result of a simulation effort. The analytical models could then be used both to help verify the simulation model and for the many "what-if" studies desired, but often too expensive to perform using simulation.

### *4.2 Programming effort*

If queueing network algorithm software is available, the effort required to develop the queueing network models used to obtain the results for this analysis is negligible. Less than 0.5 hours were required to obtain any of the analytical results and less than a labor week is required to program and test the implementation of the queueing network algorithms given in Lazowska et al (1984). The simulation model required approximately two labor months of development.

### *4.3 Value*

Queueing network analysis is probably unequalled for its ability to provide mean value performance measures accurate to on the order of 10 to 20% for little effort other than model parameterization. This type of analysis has been used successfully to select, configure and tune computer systems (Lazowska et al 1984, Lipsky and Church 1977, Zahorjan et al 1982). It is particularly useful for sensitivity analysis. However, its strengths are also its weaknesses. The algorithms discussed provide mean values with little insight into variation of performance measures. If it is important to study maximum, minimum as well as mean values of performance measures, then the use of simulation is indicated. The simulation runs produced a wealth of statistics for all aspects of system performance, in addition to the mean values of throughput, residence time, queue lengths, and service center utilization provided by the analytical models.

One of the negatives of the use of simulation, development cost, can be mitigated by good methodology. Simulations should be developed only after a good understanding is gained of what questions are to be answered (Balci 1990). The level of detail used in a simulation should be chosen to be sufficient to answer these questions and avoid unnecessary effort. Experience indicates that the simulation value to cost ratio should be expected to be low for simulations developed without a careful examination of the questions to be answered and the best techniques to be used to arrive at these answers. While the simulation value to cost ratio can be quite high when techniques suggested by Balci (1990) are used. Relatively simple simulations have provided good and sufficient insights into system performance; e.g. Mitchell et al (1974) and Knadler and

May (1990). However to get statistically meaningful results, multiple simulation runs and statistical analysis of results are required (Law 1990, Law and Kelton 1991).

## 5 FURTHER WORK

Several areas of future work are suggested by these results: investigate FESC's more exhaustively, study the robustness of both preemptive and nonpreemptive multiclass models, and study the robustness of open class models.

## REFERENCES


Balci, O. 1990. Guidelines for Successful Simulation Studies. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R. P. Saddowski, and R. E. Nance. 25–32. Association for Computing Machinery. New Orleans, Louisiana.

Buzen, J. P. 1976. Fundamental Operational Laws of Computer System Performance. *Acta Infomatica*, No. 7. Springer–Verlag. 167–182.

Denning, P. J. and J. P. Buzen. 1978. The Operational Analysis of Queueing Network Models. *Computing Surveys*, Vol. 10, No. 3. 225–261.

Gross, D. and C. M. Harris. 1985. *Fundamentals of Queueing Theory, Second Edition.* 79. New York: John Wiley & Sons.

Kleinrock, L. 1975. *Queueing Systems Volume 1: Theory.* 17, 240. New York: John Wiley & Sons.

Knadler, C. E. and R. May. 1990. Disk I/O, a Study in Shifting Bottlenecks. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R. P. Saddowski, and R. E. Nance. 826–830. Association for Computing Machinery. New Orleans, Louisiana.

Law, A. M. 1990. Design and Analysis of Simulation Experiments for Manufacturing Applications. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R. P. Saddowski, and R. E. Nance. 33–37. Association for Computing Machinery. New Orleans, Louisiana.

Law, A. M. and W. D. Kelton. 1991. *Simulation Modeling and Analysis, Second Edition.* New York: McGraw–Hill.

Lazowska, E. D. et al. 1984. *Quantitative System Performance.* Englewood Cliffs, New Jersey: Prentice Hall.

Lipsky, L. and J. D. Church. 1977. Applications of a Queueing Network Model for a Computer System. *Computing Surveys.* Vol. 9, No. 3. 205–221.

Mitchell, J. et al. 1974. Multiprocessor performance analysis. In *Proceedings of the 1974 National Computer Conference.* 399–403.

Olson, T. M. 1989. Disk Array Performance in a Random IO Environment. *Computer Architecture News*, Vol. 17. No. 5, ACM Press. 71–77.

Zahorjan, J. et al. 1982. Balanced Job Bound Analysis of Queueing Networks. *Communications of the ACM.* Vol. 25, No. 2. 134–141.


## AUTHOR BIOGRAPHY

**CHARLES E. KNADLER, JR.** is a Senior Engineer in the IBM Corporation's Federal Sector Division and an Associate Professorial Lecturer in the Department of Electrical Engineering and Computer Science at The George Washington University. His research interests include computer architecture, computer system simulation, and the use of personal computers in computer science research.