Efficient Techniques for the Simulation of Computer Communications Networks

William W. LaRue
Victor S. Frost

Telecommunications and Information Sciences Laboratory
University of Kansas
Lawrence, Kansas 66045

Technical Report 6731-7
February 1987

Supported by

National Science Foundation
Presidential Young Investigator Award
and
AT&T Information Systems

TELECOMMUNICATIONS AND INFORMATION SCIENCES LABORATORY
THE UNIVERSITY OF KANSAS CENTER FOR RESEARCH, INC.
2291 Irving Hill Drive - Campus West Lawrence, Kansas 66045

Efficient Techniques for the Simulation of Computer Communications Networks

William W. LaRue
Victor S. Frost

Telecommunications and Information Sciences Laboratory
University of Kansas
Lawrence, Kansas 66045

Technical Report 6731-7
February 1987

Supported by

National Science Foundation
Presidential Young Investigator Award
and
AT&T Information Systems

ACKNOWLEDGEMENTS

I would like to acknowledge the invaluable suggestions, contributions and patience of Dr. Frost during the preparation of this work. This work would not have been possible without the financial support of the National Science Foundation and AT&T Information Systems. I finally wish to thank my parents for their continuous support of all of my academic work.

ABSTRACT

Even with the continuing improvement in the power of modern computers the simulation of computer communications networks still presents significant computational problems. A variety of techniques have been developed over the years to improve the computational efficiency of simulation models of general queueing networks. Some of these approaches have been applied to communications systems. The purpose of this paper is to review the status of efficiency-enhancing techniques as it relates to the modeling of computer communications networks, present some new approaches, and suggest promising areas for future research.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 Introduction

As the number of computers have increased, the need for communication between computers has also increased rapidly. The advantages of networking computers include the convenient use of data that is located remotely, the increase in reliability that results from not being dependent on any single piece of computing hardware, the ease of communication between users, and the savings that accrue from being able to use several smaller processors instead of one very expensive large processor [48].

In order to efficiently design computer communication networks, it is necessary to have estimates or predictions of the systems performance. There are many cases of severe performance degradation that were not anticipated by the network designers [43]. With adequate performance modeling it is possible to anticipate and correct inadequate network performance while avoiding the costs of altering an existing network. Some useful performance measures could be packet delay, buffer requirements, or link utilization statistics.

Techniques for determining performance measures of communications networks are needed for many other applications. One is the comparison of alternative planned network designs. Another is the optimization of network parameters in either an existing or designed communications network.

1.1 Performance Prediction Methods

There are several approaches that can be used to obtain performance estimates of computer networks. One is to actually measure the performance of the network. Another possibility is to use analytical calculation to determine the performance measures of interest. A third possibility is to simulate the network using a computer model [43].

The main advantage of the actual measurement of the network is that the results will, undoubtedly, reflect the performance of the actual network.

Measurements are often expensive to conduct especially if the measurement process requires the network be removed from normal operation. For example, it would be impossible to measure the results of network overloads, link failures, or other unusual conditions without disrupting the normal use of the network. Also it is impossible to measure a network which has yet to be constructed. A related problem is that it is almost impossible to compare alternative topologies, equipment, or link speeds by measurement because of the cost of reconfiguring the network for each desired case.

Performance modeling of a system by analytical formulas has some important advantages. Once an analytical formula has been developed it can be applied at very low cost. Unlike simulation and measurement techniques which give statistical results, analytical formulas provide answers that are deterministic.

Because of the low cost of application and the deterministic nature of the result, analytical modeling is ideal for modeling alternative proposed systems. Analytic modeling can also be applied to the optimization of network parameters.

The disadvantages of analytical models of computer networks arise primarily in the difficulty of developing models that truly represents a real system. There is a limit to the complexity that can be modeled analytically. Priority schemes and finite buffer lengths have proven particularly intractable to analysis.

The simplifying assumptions which are necessary to develop an analytical model often cause a significant loss of accuracy. Often the only means available to determine if the assumption is valid enough for the model is to validate the analytical model with simulation or actual measurements. A further disadvantage of analytical modeling is that it requires the labor of a skilled modeler to develop a sophisticated model [49].

Another alternative technique for obtaining performance estimates is the computer simulation of the system. In this paper, simulation will refer to the use of a digital computer program which attempts to duplicate the physical system. The computer simulation is driven with random inputs supplied by random-number generators, and produces results which are stochastic. An advantage of using computer simulation for determining the performance of a computer network is that any amount of complexity can be modeled and that it can be applied to systems not yet built. Also in a computer simulation we have much better control over the experimental conditions than in a measurement [34]. Essentially simulation combines the detailed modeling available with measurement with the ability to evaluate and compare unbuilt systems that is available with analytical modeling.

The major disadvantages of simulation are the time required to write and debug a sophisticated simulation, the effort necessary to validate the simulation, and the analysis required of the statistical simulation output.

The simulation modeling of a sophisticated computer network requires a sophisticated computer program which is very expensive and time-consuming to develop [34]. A simulation model must also be validated to ensure that it is an accurate representation of the system.

The statistical nature of the output causes several difficulties. One difficulty arises from the correlation between output samples which make it difficult to form a confidence interval for the estimated performance measures. The statistical nature of the output makes it difficult to use simulation models for the optimization of network parameters. Another problem resulting from the statistical nature of simulation is the very large amounts of computation required to accurately characterize the performance measures.

Much work has been done of efficient simulation techniques for uses such as operations research and computer performance modeling, but little has been applied to computer communications network simulation. This paper will examine the techniques that have been used in other applications and will also evaluate the effectiveness of each technique for computer communication network simulation.

1.2 The Need For Efficient Simulation Techniques

This paper is concerned with the use of techniques for enhancing the computational efficiency of computer communications network simulations. These methods can decrease the cost and increase the accuracy of a network simulation. More computing power cannot replace the néed for these techniques since there will be a requirement to model ever more complex networks and protocols. Sometimes computationally efficient techniques can make the difference between an impossibly expensive simulation project and a frugal one [33].

In section 2 of this paper background material is presented which provides the basis for understanding the common efficiency enhancing techniques. Section 3 defines some of these approaches relative to network simulation. Sections 4 and 5 describe these approaches in detail, and provide evidence of successful applications in the literature. These sections discuss the relevance of each technique to computer communications network performance analysis.

2.0 Background

Some background material is necessary to understand the computationally efficient techniques that are presented in the rest of the paper. The concepts of queueing systems, confidence intervals, and regenerative cycles are necessary to the understand many of the computationally efficient techniques presented.

2.1 Queueing Systems

A queueing system consists of one or more servers and a queue or waiting line where entities, such as packets, wait for their turn to be serviced. In order to apply efficient techniques to the solution of queueing systems, it is necessary to be able to classify them. Queues can be classified according to the distribution of the arrival of entities, the distribution of service times required by the entities, the number of servers available to serve the entities, and the service discipline used by the servers.

The notation used to denote a queueing system is of the form GI/G/n. The first expression, GI, stands for an interarrival distribution which is general but independent of the service time distribution. The second G stands for a general service time distribution. Another symbol for interarrival and service time distributions is M which is used for exponential distributions [34]. The n stands for the number of servers for that queue. The service discipline, which is the order in which the entities arriving at the queue are serviced, could be FIFO, where the entities are service in order of their arrival, or a different discipline such as a priority scheme.

When describing a queueing system λ is often used to denote the mean interarrival rate, and μ denotes the mean service time. The traffic intensity for the queue is the mean utilization of the servers. It is denoted by ρ which equals $\lambda * \mu /n$.

2.2 Confidence Intervals

In a computer communications network simulation, there is a tradeoff between the computational costs of performing the simulation and the reliability of the results. Therefore when performing a simulation study, it is important to not only estimate the expected value of the performance measure but also determine the reliability of the measurement. Such a determination is often based on confidence intervals. A confidence interval is an interval which contains the point estimate of the parameter, and has a given probability of containing the true value of the parameter. Thus a 95% confidence interval would be an interval which based on the statistical data available has a 95% chance of containing the value of the parameter. Obviously a small confidence interval width is desirable because this will pin down the performance measure more precisely. A confidence interval can be calculated from data by using the central limit theorem to approximate the sum of a large number of independent identically distributed sample as having a normal distribution.

The width of the confidence interval is proportional to the sample standard deviation. Since the sample size is inversely proportional to the sample variance, the confidence interval width will be inversely proportional to the square root of the sample size. Therefore to reduce the confidence interval width by a factor of 10 will require about 100 times the simulation effort.

2.3 Calculating Confidence Intervals for Performance Measures

A problem in computing confidence intervals for the network performance measures of interest is that samples of these performance measures are neither independent nor identically distributed. The samples are correlated because of the dynamic nature of the system. If one packet experienced an abnormally high delay then the next packet would likely also experience a high delay since the network will probably remain congested. The samples are not identically distributed if the network is started from a state that is not its steady state of operation. Thus the samples will not be identically distributed until the network reaches steady state.

One solution to the problem of getting the network into steady state is to run the simulation for a period of time sufficient for the network to reach steady state before any data sampling is done. A problem with this approach is that it is difficult to determine the amount of time required for the simulation to reach steady state.

The problem of correlated samples is too complicated to handle with classical statistics [6]. To handle this problem directly necessitates the use of methods of time series analysis instead of classical statistics [7].

There are alternative methods available to solve this problem and instead obtain independent identically distributed samples of the performance measure of interest. Three of these methods are independent replication runs [34], the method of batch means [34], and the regenerative method [7,9,11,34].

The method of independent replication runs uses the means of the performance measures as the independent samples needed to calculate a confidence interval. By making each sample consist of the mean from a simulation run that is driven by different random number streams we are assured that the means will independent. A disadvantage to this method is that a fairly large number of runs will be needed to apply the central limit theorem to calculate the confidence interval width. This can be inefficient since each simulation run must be run for a period of time for the simulated network to reach steady state.

The method of batch means can improve the efficiency of the simulation with respect to independent replications. In this method the simulation is started with a subrun to reach steady state, then a number of runs are made without resetting the network conditions, so each successive run is started using the end of the last run as the initial condition (see figure 2.1). This obviously introduces some correlation between the samples because for example if the network was congested at the end of one sample it would be congested at the start of the next sample. This eliminates the need the network to come to steady state before each run but, if it is just does introduce the possibility of calculating too small a confidence interval width because of correlation between the samples.

A better method of calculating the confidence intervals was developed by Crane and Iglehart [7,9] and Fishman [11]. If the simulation can be divided into a series of cycles that begin and end in a known state, then samples of random variables based on these cycles are independent and identically distributed. The simulation results can be divided into regenerative cycles if the system will always return to a given state. This condition will always be met in the simulation of a stable queue. Usually

3.0 Approaches to Efficient Network Simulation

This chapter will describe two important approaches used to develop computationally efficient techniques for network simulation. For each of these approaches some common computationally efficient techniques will be defined.

Two general approaches to reducing the computational demands of a computer network simulation are hybrid modeling and the application of variance reduction techniques. Hybrid modeling increases the efficiency of a simulation by combining an analytical model with a simulation model. A variance reduction technique increases the computational efficiency by using statistical methods to obtain more accurate performance measures from the simulation. Variance reduction techniques can either produce a more accurate simulation with the same expenditure of computational resources or can reduce the computational resource requirements for the simulation.

3.1 Hybrid Simulation Techniques

A hybrid technique works by only simulating part of the system and using analytical techniques where possible. A hybrid simulation may be able to provide some of the detailed modeling available with simulation at a much lower computational cost.

lower computational cost. In section 4 applications of hybrid modeling techniques to computer network simulation will be examined.

3.2 Variance Reduction Techniques

Unlike hybrid modeling techniques, variance reduction techniques exploit the statistical properties of the random variables used in the simulation. There are many types of variance reduction techniques including antithetic variates, common random numbers, control variates, and importance sampling. These techniques are discussed in detail in some texts on discrete simulation particularly Kleijnen [28] and Law [34].

An antithetic variate technique works by running the simulation twice, one with one random number stream and the other with a complementary random number stream that will have the opposite effect on the performance variables of interest. The two runs are then averaged. This will produce an estimate closer to the mean value than the brute force approach.

The application of the common random number technique reduces the random variations between simulation runs by using the same random number streams in each of several simulation runs.

Variance reduction by using control variates works by finding a correlation between the performance measure and a control variable which can be simulated and also has a known mean. Then if the simulated value of both the control variable and the performance measure are known, it is possible to make an improved estimate of the mean of the performance measure using mean of the control variable and the correlation between the performance measure and the control variable.

Importance sampling involves the use of a simulation that concentrates its use of computational resources in simulating the random inputs that make the largest contribution to the performance measure of interest. Importance sampling reduces the variance of the performance measure by providing more of the samples that have a major effect on the parameter of interest.

Often variance reduction techniques can provide substantial savings by using efficient sampling and observation techniques.

In sections 4 and 5 it will be shown that both hybrid techniques and variance reduction techniques can be applied to computer communications networks. The advantages and disadvantages of each technique will be discussed.

4.0 Application of Hybrid Techniques

In a hybrid technique a simulation model is combined with an analytical model. When using hybrid modeling to reduce the computational cost with respect to simulation, usually an analytical model is developed for the more mundane parts of the model and simulation is used to model the behavior of the parts of the system which are impossible to model analytically.

This paper will cover only applications of hybrid modeling to communications network simulation and not applications to computer performance simulation which is another common application of hybrid techniques [19,45,4,50].

4.1 Types of Hybrid Modeling Techniques

Two types of hybrid simulation methods are the decomposition of the system being modeled and the use of conditional expectation. Decomposition involves partitioning the system into subsystems of which some can be solved analytically and others require simulation. Conditional expectation is useful when an analytic model is available where some of the quantities needed are not known but can be simulated.

In a paper presenting a unifying framework for hybrid modeling Shanthikumar and Sargent [46] identified four classes of hybrid modeling according to the interaction of the simulation and analytical models. Their classes I and

II are types of decomposition models, but their class III corresponds to a conditional expectation type of model. Class IV was a simulation model that used an analytic model to calculate certain parameters for the simulation. None of the applications studied here used a class IV type of hybrid.

4.1.1 Decomposition Techniques

In the class I of hybrid decomposition technique the analytical and simulation models alternate over time. For example a network which could be described analytically except for transmission errors and link failures could be simulated efficiently by letting the analytical technique handle the normal operation of the network and by switching to simulation for errors and link failures. This can result in a very considerable savings by eliminating the need to simulate the system except during the relatively small amounts of time that are required for error handling. Figure 4.1 illustrates the structure of a Class I hybrid model.

Another class of hybrid decomposition models is one in which both the simulation and hybrid models operate together at the same time. An example of this type of modeling would be the modeling of a large network where a few stations were simulated in detail while analytical expressions for the traffic at background nodes such as

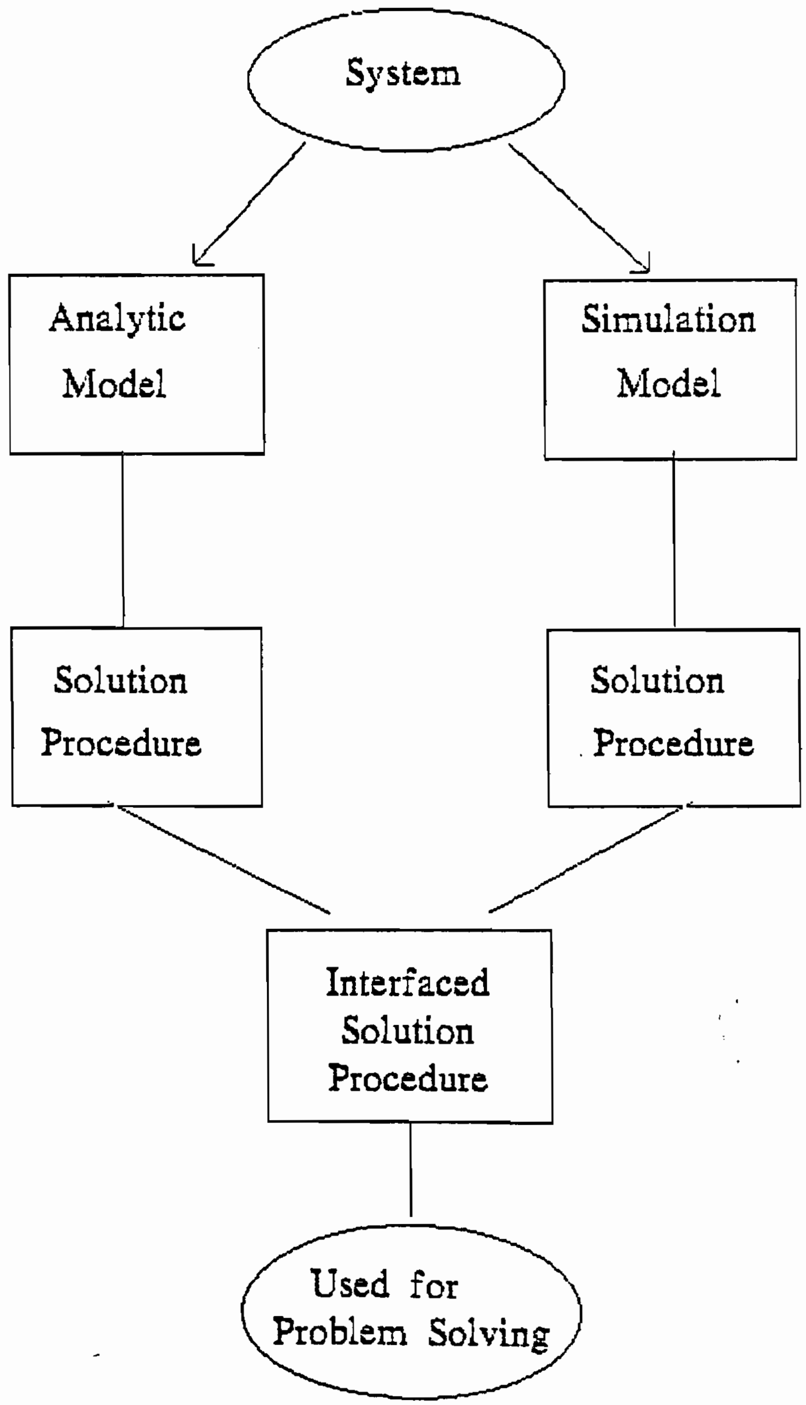

Figure 4.1 Class I hybrid simulation/analytic model [46].

was done for a CSMA/CD model by O'Reilly and Hammond [40]. This class of hybrid models reduces the cost of simulation by reducing the scope of the simulation. Figure 4.2 illustrates the structure of a Class II hybrid model.

Decomposition is most useful in situations when one or more of the submodels can be solved analytically, when one or more of the submodels has a fixed set of parameters in the situation to be studied, or when large differences in time scales exist between submodels [35]. If one or more of the submodels can be solved analytically then the computational burden will obviously be lessened.

When one of the submodels has a fixed set of parameters the simulation for that submodel can be run only once and those results used in all the simulations of the entire model. Without decomposition it would be necessary to simulate the fixed parameter submodel every for every simulation run.

If widely different time scales are present in the model decomposition into frequent and infrequent types of events will lead to a reduction in computational burden. The reduction will result because it is possible to accurately simulate the submodel with frequent events in a much shorter simulated time than the submodel with infrequent events. This results because the certainty of a statistical experiment depends on the sample size.

There are some important disadvantages to

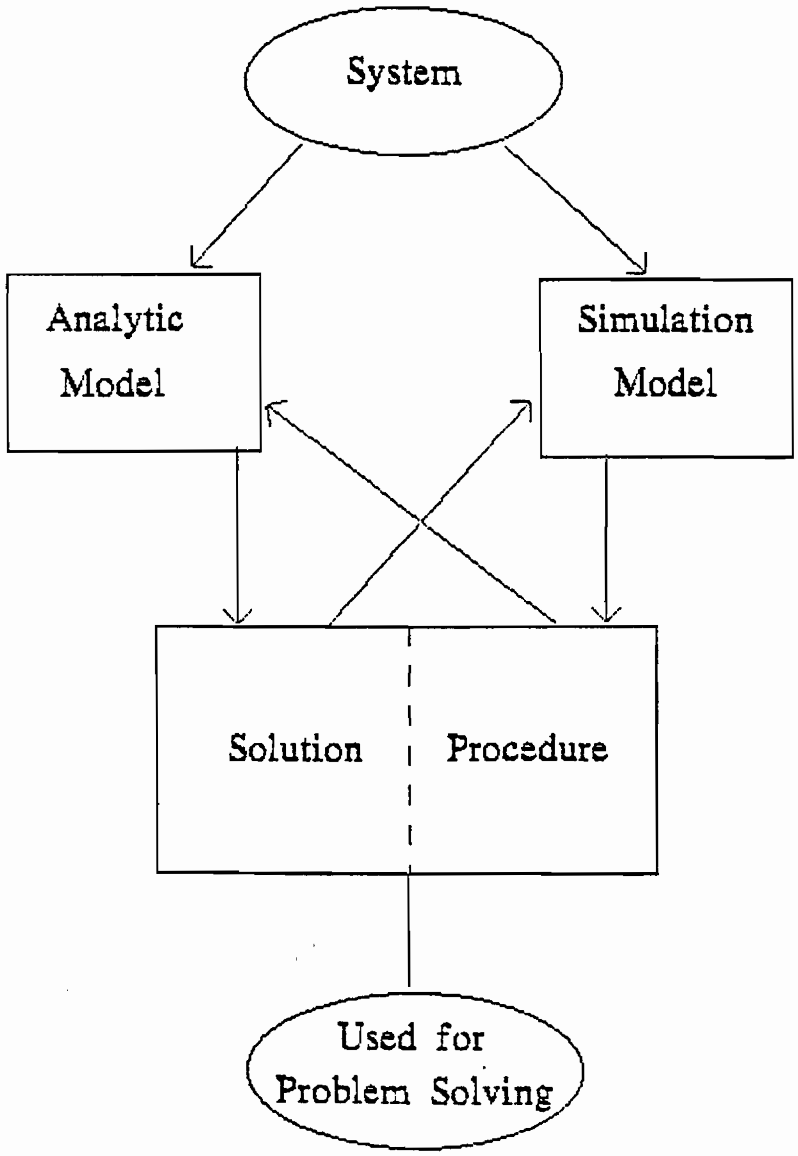

Figure 4.2 Class II hybrid/simulation/analytic model [46]

decomposition techniques. The application of these techniques is very specific to the model of the system under study. Also in decomposing a model into submodels, some of the advantages of a simulation study are lost.

Hybrid decomposition models require more human effort to apply because they are very system dependent. Each new system model must be examined to find where it can be broken into submodels, and each of the submodels must be examined to see whether a simulation or analytical treatment is more appropriate. Not all systems can be decomposed in a reasonable manner, and the amount of computational savings will vary widely from system to system.

Some of the advantages of simulation can also be lost when a decomposition approach is taken. Usually some assumptions must be made in order to decompose the system into noninteracting submodels. These assumptions mean that the decomposition model is only an approximation to the system, thus the major advantage of being able to model a system in intricate detail with simulation is lost. Also, depending on the structure of the decomposition, only the means but not the variances of the performance measures may be obtained using decomposition.

4.1.2 Conditional Expectation Techniques

In a conditional expectation hybrid model the simulation model is used in a subordinate way to the analytic model. Thus the simulation model is used to provide some parameters for the analytic model. Figure 4.3 show the structure of a conditional expectation type of hybrid model.

In some situations conditional expectation can be guaranteed to improve the efficiency of a computer simulation. Suppose that we are trying to find the mean of the random variable of interest Y and that X is another random variable which can be simulated. If the value of $E[Y|X]$ is known we can simulate X and use its value to calculate $E[Y]$. The variance of the expected value of Y calculated from a simulated value of X would be [34]:

Eqn 4.1 $$Var[E(Y|X)] = Var(Y) - E[Var(Y|X)] \leq Var(Y)$$

Equation 4.1 assures that a calculation of $E[Y]$ based the simulation of X and the subsequent calculation of $E[Y]$ will have a variance equal or smaller than a direct simulation of Y where the same number of observations of X and Y are used. The equality will apply only if Y is completely determined by X so that $E[Var(Y|X)]$ is zero. If it does not take more computational resources to generate samples X than Y then it is assured that using conditional expectation will increase the efficiency of the simulation. This is an advantage of conditional expectation which is not shared by all other computationally efficient techniques.

Conditional expectation is best applied when an

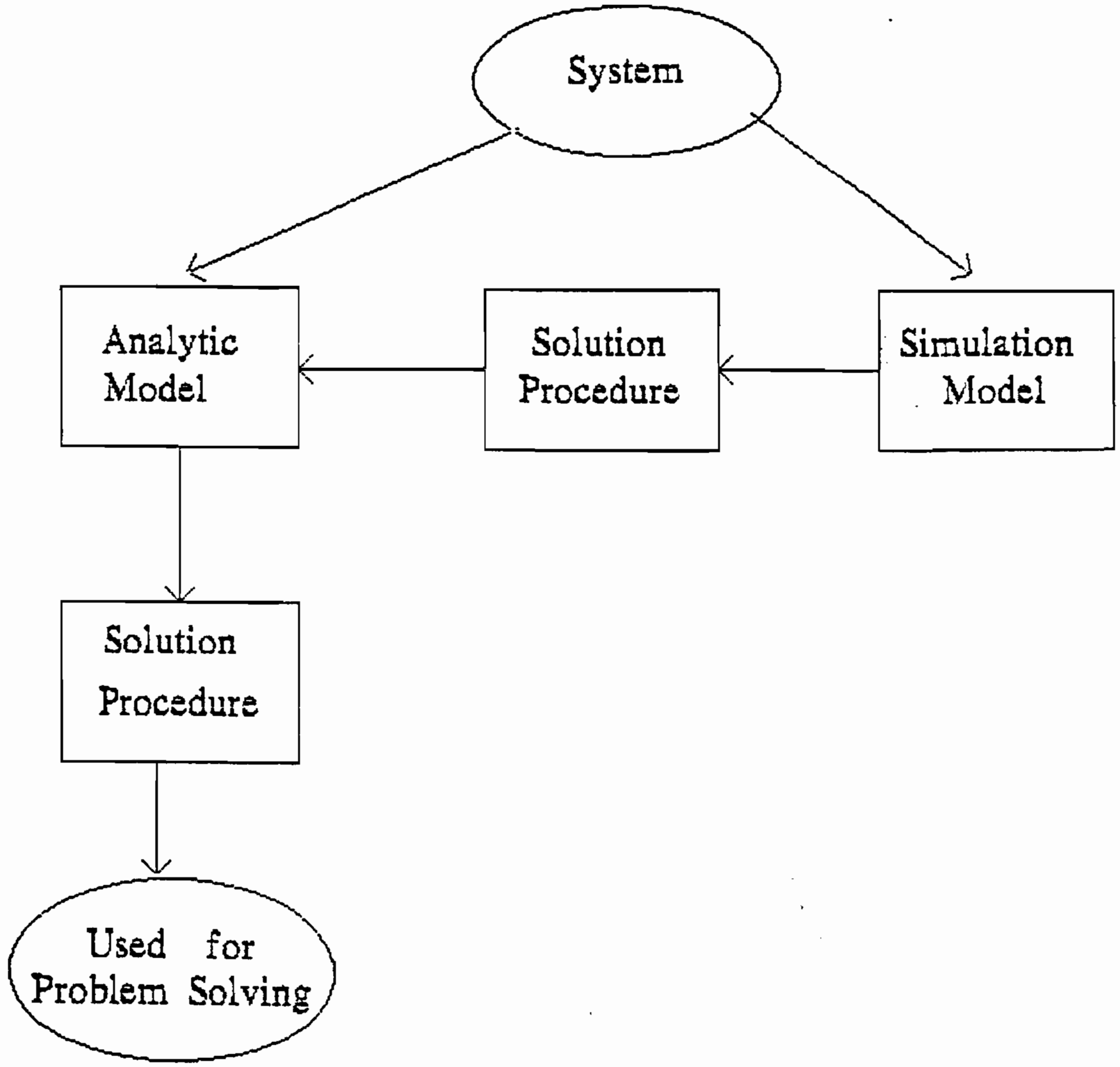

Figure 4.3 Class III hybrid simulation/analytic model [46].

obviously suitable conditioning quantity whose conditional expectation can be found easily. Otherwise the advantages of reducing the computational resource expenditures must be balanced against the the human labor necessary to find a suitable conditioning quantity.

Conditional expectation like decomposition is very model dependent. Also like decomposition some of the advantages of simulation may be lost. Unless the conditional expectation for the variance is known, it will usually be impossible to estimate the variance of the quantity of interest. Unlike decomposition, conditional expectation is not an approximate simulation of the model unless the condition expectation is approximate.

4.2 Applications of Hybrid Techniques to Computer Networks

Hybrid simulation has been successfully applied to the modeling of computer communications networks.

O'reilly and Hammond [40] used hybrid modeling for use in modeling a large multiple access network. Their method worked by partitioning the system into two classes of stations: 1) a small number of stations to be modeled with detailed simulation, and 2) a large number of background stations to be modeled analytically. This type of partitioning would make this model a class II hybrid model in the discussion above because of the interaction between the simulated stations and the analytically modeled stations. In the model it is assumed that each of the background nodes has an identical Poisson arrival process and a fixed packet length. It is also assumed that there is no queueing at the background nodes.

The background stations are modeled as generating a set of busy and idle periods on the network. The statistical properties of these busy and idle periods accurately models the composite behavior of background stations in the network. The composite behavior is determined by the algorithm from the number of stations in the background, their traffic characteristics, and the protocols used.

The background model and the simulation model interact as the simulation proceeds. The background traffic and the simulation compete for access to the media and interact through the contention handling of the CSMA/CD protocol.

They found that this technique is more efficient than brute force simulation when more than 100 to 1000 background stations are in the network being modeled. The advantage of the hybrid technique increased as the traffic intensity in the network increased.

This partitioning method is applicable to communications networks that have a large number of nodes competing for the same transmission resource. The disadvantage of this technique is that a network must have a large number of nodes that can be modeled as a homogeneous group for this technique to be useful.

Wong, Moura, and Field used a multilevel hierarchy hybrid technique to model the three lower levels of the ISO network hierarchy during file transfers [51]. The model was decomposed into a three level hybrid that would be considered to be class I since the levels did not interact with each other during the simulation. In this approach the media access layer, a token ring, was modeled first then those results were used as inputs to the model of the link access layer, and finally those results were used in the network layer model. Figure 4.4 shows the architecture used in this model.

In order to separate the network into three submodels which could be evaluated without interactions between the layers some simplifying assumptions need to be made. In order to separate levels one and two it was necessary to assume an error free channel and to limit the control packets to only receive ready. To model such features as error recovery and piggybacked acknowledgements require information from the link level to be made available to the media access level which is incompatible with the hierarchy of the hybrid model. Thus the model is only approximate since error recovery and piggybacked acknowledgements cannot be modeled.

Level one models the transmission of data packets

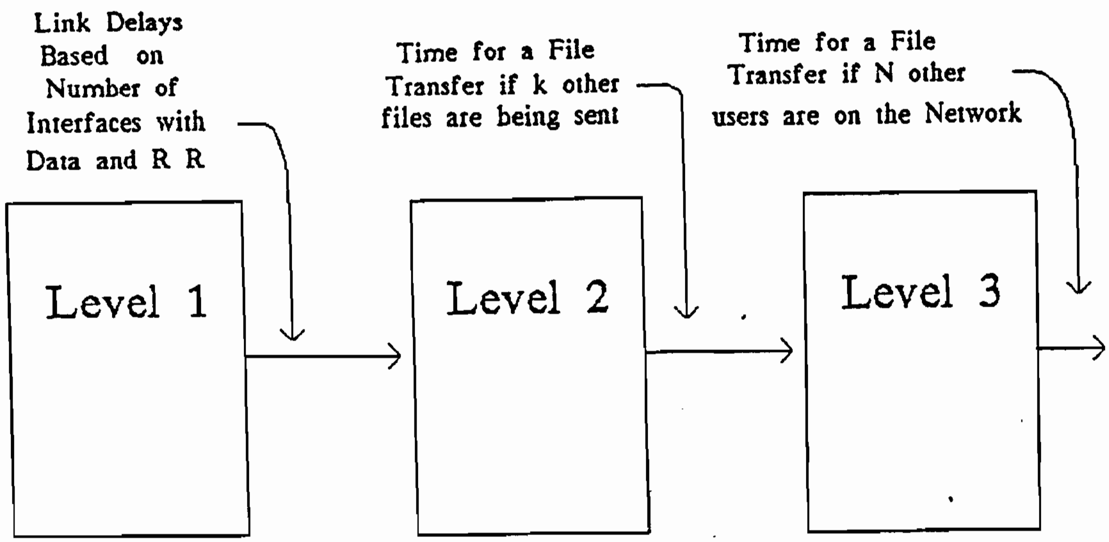

Figure 4.4 Conceptual representation of the hierarchical modeling technique.

and receive ready packets at a given interface. The mean delays from queueing, contention, and transmission of data and receive ready packets as a function of the number of interfaces servicing packets are passed to the second level of the model. Under the simplifying assumptions of the paper [51] level one can be modeled using analytical techniques.

The second level models effects such as window size, file size, and acknowledgement times. This layer uses the average packet delays from level one to calculate the level two performance measure t(k) which is the mean time for a file transfer when k other file transfers are in progress. The effects on level two are to complicated to analyze so simulation is used.

The third level of the hierarchy models the effects of file transfers on the subscribers to the network. It is assumed that each subscriber alternates between file transfers and independent processing. The performance estimate generated on this level is T(N) the mean time required for a file transfer given that N subscribers are present on the network. Under the assumptions made in the paper level three can be solved using Markov chain analysis.

This technique [51] produced impressive increases in computational efficiency without significant loss in modeling accuracy for system studied. Typically the computer resource requirements were reduced by a factor of 50 over a direct simulation. The inaccuracies introduced by not modeling the interactions of the first and second layers of the model were typically less than 5% for the system studied.

This type of technique would be useful to find an overall performance measure of a network multiple level protocol such as the ISO. The disadvantage is that a great deal of modeling versatility is lost by removing all of the interactions between the layers. Unfortunately this technique sacrifices the ability to model details like error recovery and piggybacked acknowledgements.

Lavenberg and Welch utilized conditional expectation to increase the efficiency of a discrete event network simulation[31]. They used conditional expectation to model a closed network of 4 to 6 queues and with 15 to 25 customers. They had an expected value formula for the expected waiting time at a node conditioned number of customers at that node and elapsed service time of the customer in service. Both the mean and variance of the waiting times were obtained from the conditional simulation.

The variance in the expected waiting times were reduced by factors ranging from 1.1 to 5. Since there was a negligible increase in computer resource utilization for the conditional simulation, the variance reduction translates into an equivalent gain in computational efficiency which is greatest for service centers not in heavy use. Although a closed queueing network is usually used in computer performance modeling, it is possible to generate similar results in some type of communications networks, such as store and forward packet networks.

Frost, LaRue, and Shanmugan [15] developed a conditional expectation type of hybrid technique for the efficient simulation of multiple access communications networks. This technique has produced impressive computational saving in a simulation of CSMA/CD local area network without compromising the versatility of the simulation.

The technique is based on a formulation for the waiting time of a GI/G/1 queue [36]. A GI/G/1 queue has one server but the interarrival and service time distributions can be general. If W is the waiting time for a stable queue in steady state operation then the moments of W are given as:

Eqn 4.2

$$E[W] = \alpha + r = E[U^2]/(-2E[U]) - E[I^2]/(2E[I])$$

$$\text{where: } \alpha = E[U^2]/(-2E[U]) \text{ and } r = - E[I^22]/(2E[I])$$

$$\mathrm{Var}[W] = E[U^3]/(-3E[U]) + \{E[U^2]/(-2E[U])\}^2 + E[I^3]/3E[I] - \{E[I^2]/2E[I]\}^2$$

Where U is $t_n - s_n$ and t_n and s_n are the interarrival and service times of the nth customer to arrive. I is a random variable representing length of the idle portion of the busy cycles. Since this is a stable single server queue, the I's are independent and identically distributed. The moments of I can be regarded the conditioning quantity.

The first term of the formulation for the expected waiting time, α, is commonly used as a heavy traffic approximation for the expected waiting time [36]. As ρ the traffic intensity approaches 1 the expected waiting time will approach α. The second term, r, is a correction for the heavy traffic approximation. It is possible to solve analytically for α if the moments of the interarrival and service distributions can be found. If the interarrival distribution is exponential then r can be found analytically because the memoryless property of exponential distributions guarantees that I will have the same distributions as the interarrivals. With other interarrival distributions it is impossible to analytically find r.

In [36] it was suggested that equation 4.2 can be used to formulate a conditional expectation type of hybrid simulation by simulating r and using it in equation 4.2. If $\alpha \gg r$ as, is the case with heavy traffic, then only a small portion of E[W] will be subject to random variations. Under such conditions the hybrid estimate of E[W] will have a much narrower confidence interval than brute force estimate of E[W].

A pilot study was run using M/M/1 queue to find the conditions under which a computational saving can result from using this conditional expectation technique. The results of this pilot simulation are given in figures 4.5 and 4.6. Figure 4.5 show the ratio between the required cpu times for the simulation of an M/M/1 queue using the brute force and the conditional expectation technique. Figure 4.6 shows the relative confidence interval widths as a function of traffic intensity for both the brute force simulation and the conditional expectation technique. As can be seen from the graphs, the confidence interval width for a brute force simulation increases rapidly with increasing traffic. In contrast the conditional expectation technique's confidence interval width decreases with increasing traffic. As can be seen from figures 4.5 and 4.6 the conditional expectation technique is more efficient for traffic intensities above about 0.4. At a traffic intensity of .9 the computer time required to brute for simulate with a confidence interval equal to the conditional expectation simulation is about 5000 times as great.

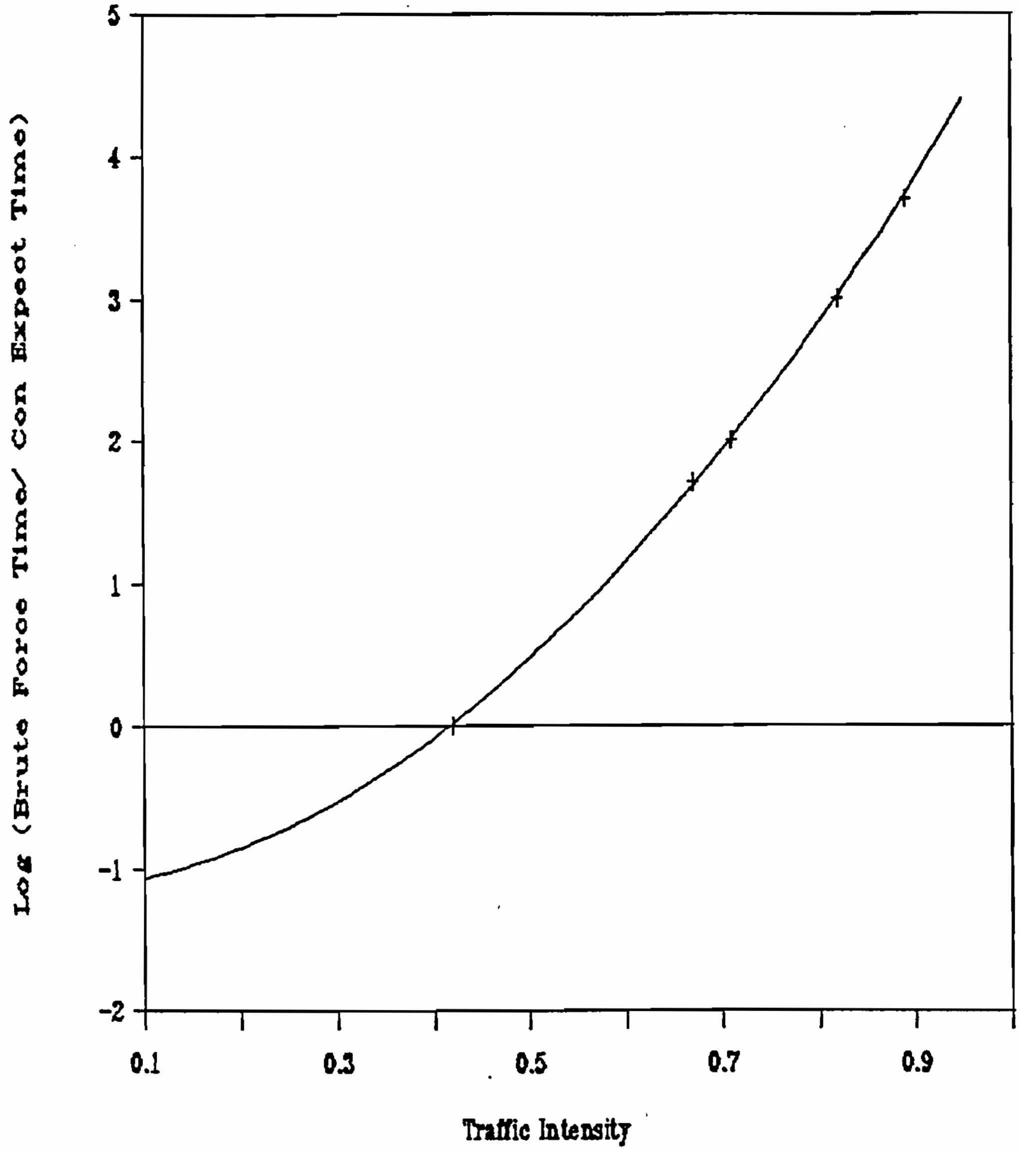

Figure 4.5 Ratio of cpu times required to achieve the same confidence interval width for an M/M/1 queue.
(Brute force simulation time / conditional expectation simulation time)

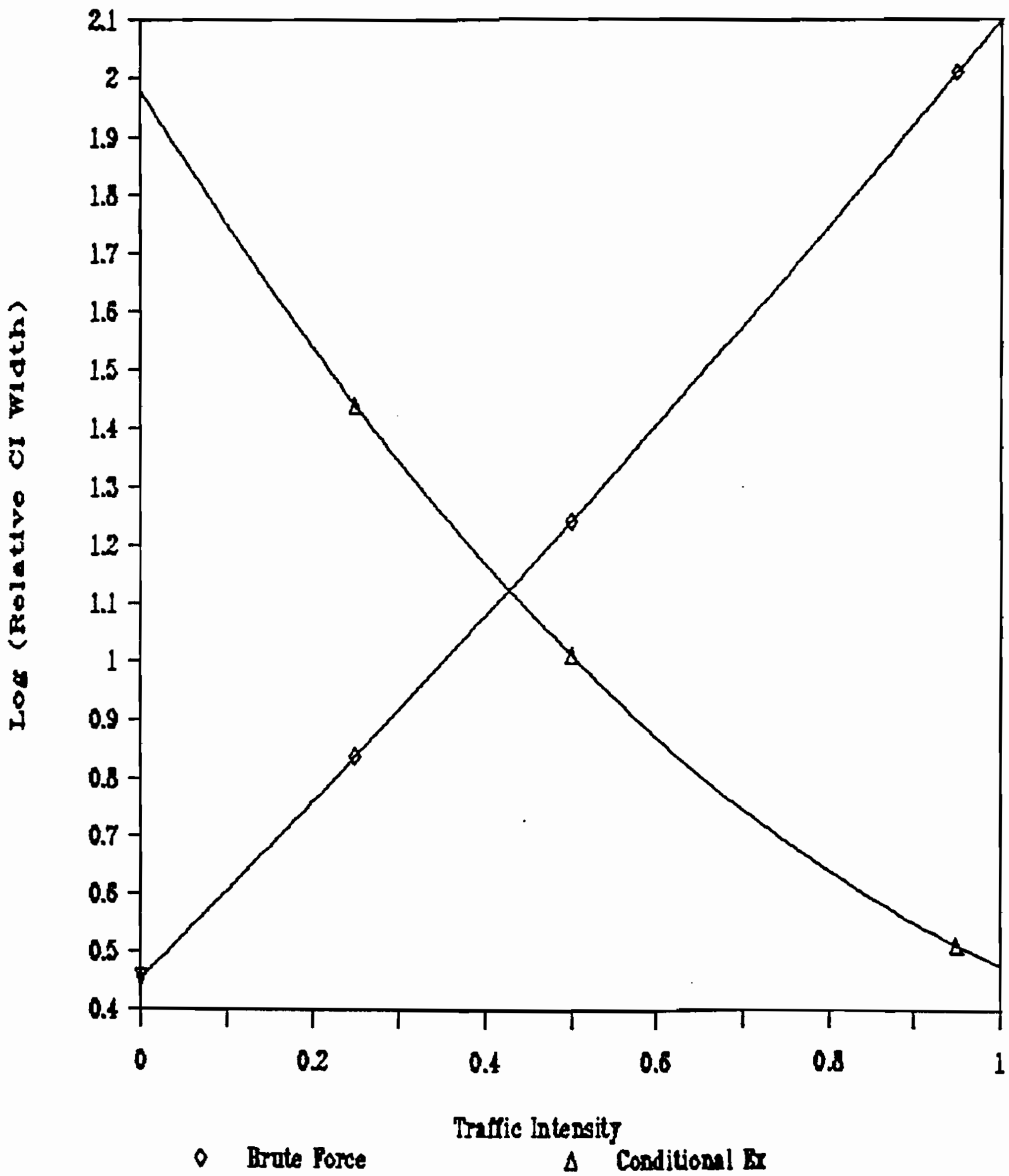

Figure 4.6 Relative confidence interval widths as a function of traffic intensity.

network with exponential interarrival times of mean 0.050 seconds at each node was simulated. In the network tested the messages were assumed to have exponential message length. The mean message length was varied to test the conditional expectation simulation at traffic intensities between 0.5 and 0.9. The slot time used was 512 us, the interframe gap was 96 bits, and the jam time was 32 us. The attempt limit was set to 16 and the backoff limit to 10. These parameters were chosen model a network based on the IEEE 802.3 standard [14].

Figures 4.7, 4.8 and 4.9 were produced from the results of ten independent simulation replications of the simulation model for each of 6 different traffic intensities. Each run represents a simulation time of 10 seconds of real time both for the brute force and conditional expectation technique.

As can be seen from figure 4.7 and 4.8 the agreement in the mean waiting times between the conditional expectation and the brute force simulation is extremely good. There is a worst case deviation of less than one half the width of a 95 % confidence interval.

Figure 4.9 shows the width of the confidence intervals relative to the means for the expected waiting time from a brute force and a conditional expectation type of simulation as a function of traffic intensity. As can be seen from the figure 4.9 the conditional expectation

$\mu = 10000.0 * \rho$

$\lambda = 0.00002$ for each of five queues

ρ	Brute	Force		Idle Cycle			Improvement
	μ	$\pm$	$\pm/\mu$	μ	$\pm$	$\pm/\mu$	
0.50	2674	233	0.087	2681	403	0.150	0.58
0.65	6464	730	0.113	6204	412	0.066	1.70
0.75	11986	1610	0.134	11691	571	0.0488	2.75
0.80	16547	2355	0.142	16411	625	0.0381	3.73
0.85	21625	3203	0.148	22708	795	0.0350	4.23
0.90	33276	5285	0.159	35421	1358	0.0383	4.15

Figure 4.7 Summary of the results of a CSMA/CD simulation comparing a brute force and a conditional expectation technique.

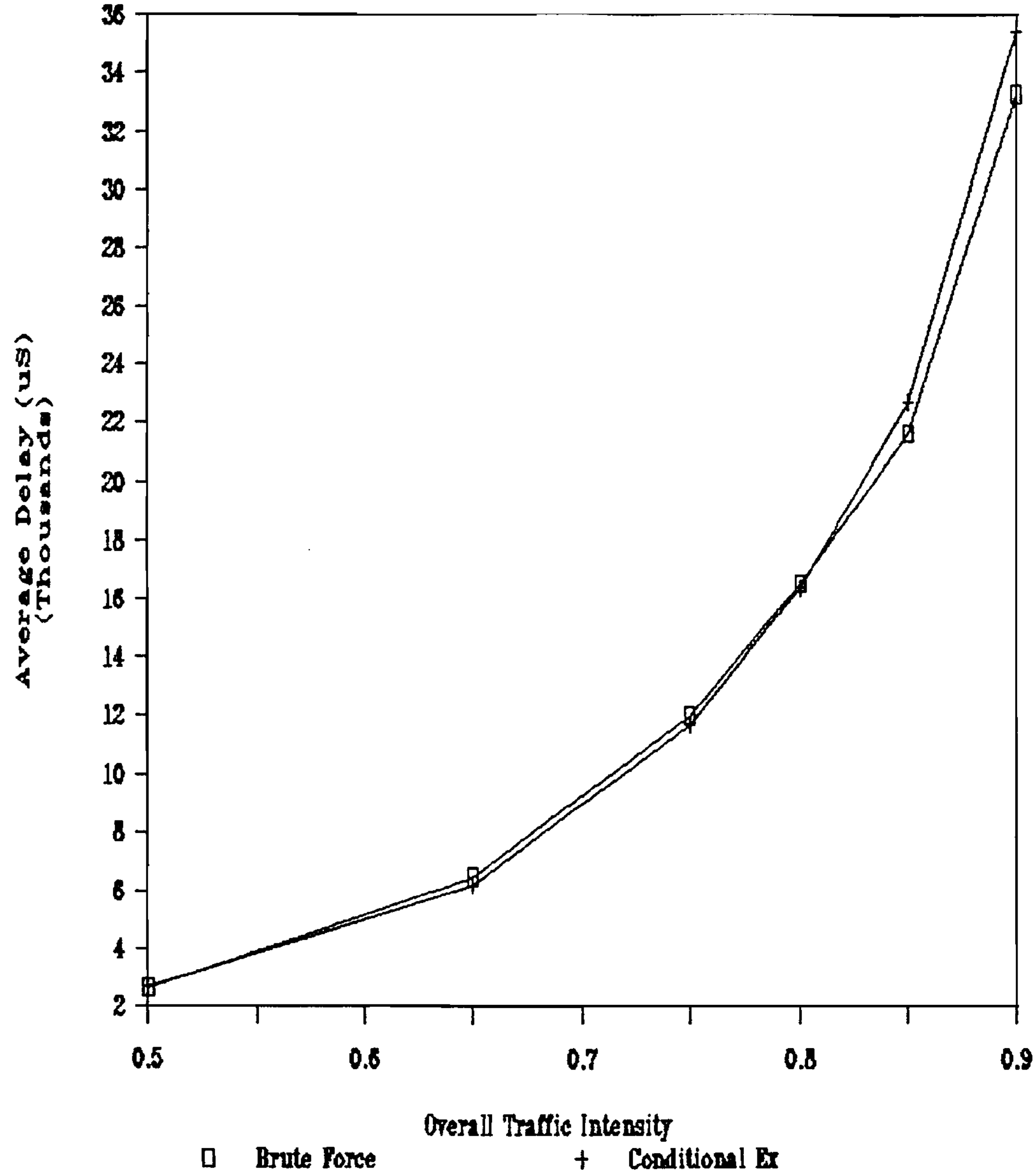

Figure 4.8 Comparison of the mean waiting times found by brute force simulation and by using the conditional expectation technique for a 5 node CSMA/CD network.

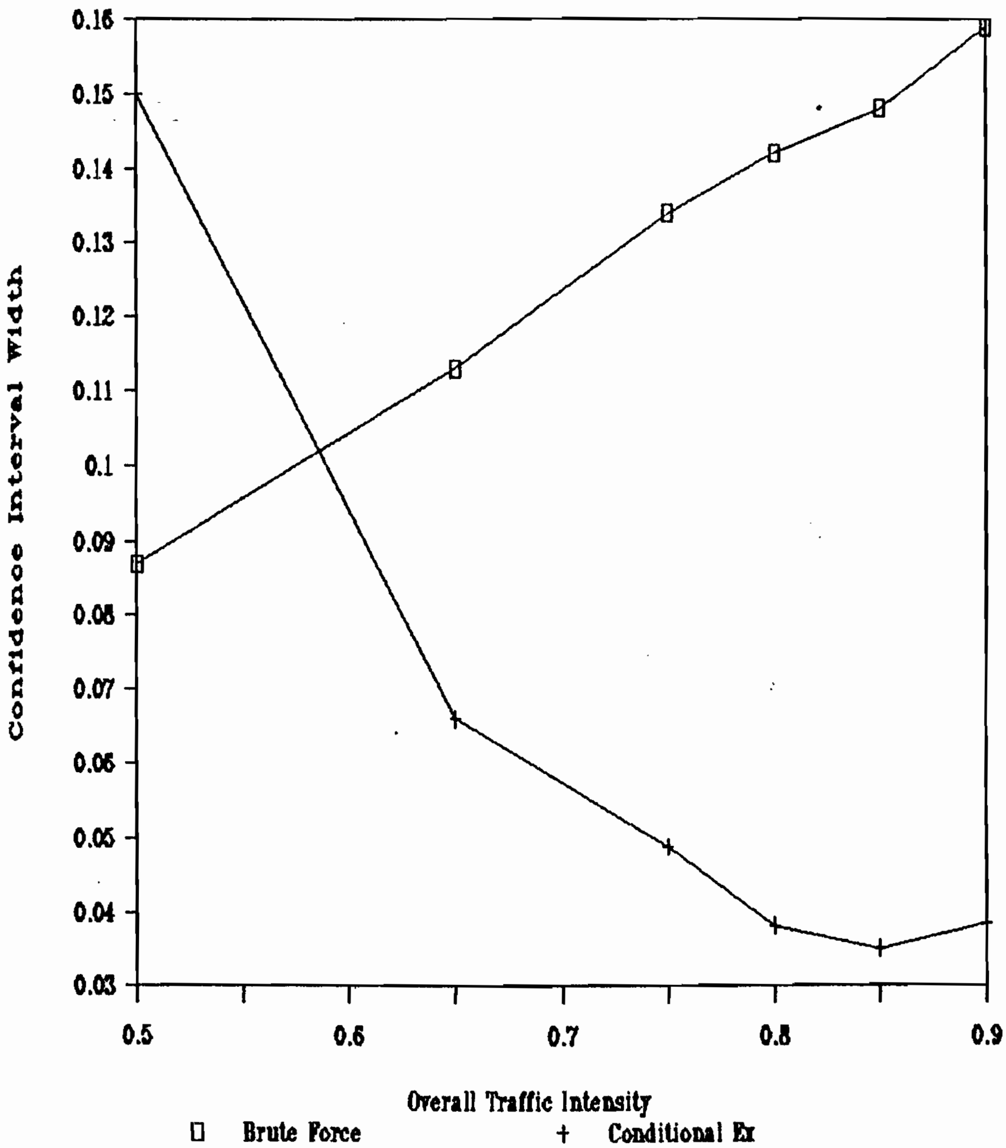

Figure 4.9 Confidence interval widths comparison of brute force and conditional expectation techniques for the simulation of a 5 node CSMA/CD network.

are competing for the same transmission resource. This technique is only applicable at high traffic intensities, but high traffic intensities present the greatest computational burden using brute force simulation.

4.3 Summary

At its best decomposition combines the advantages of the low computational cost of analytical modeling with the detailed modeling available with simulation. This is most easily realized when the most costly part of a brute force simulation can be accurately modeled analytically. The amount of human effort to apply decomposition to a given situation must be balanced against the resulting computation savings.

The use of conditional expectation techniques can guarantee a savings in computer utilization without affecting the accuracy of the simulation. Like decomposition the model dependency necessitates a large amount of human direction in applying this technique.

Decomposition and conditional expectation can be viewed as two different approaches to hybrid modeling which is the combination of analytical and simulation modeling of the same system. In decomposition an engineering approach is taken where the system is broken into functional blocks which can be analyzed separately. In conditional expectation a mathematical approach is taken but the quantities that are needed to calculate the results and can not be found analytically are simulated.

In short hybrid techniques have the advantages of substantial computational saving and proven applicability to network simulations. They are best applied to situations where the same network structure must be simulated a number of times so that the human effort invested in the technique can be recovered with large computational savings.

5.0 Variance Reduction Techniques

A different approach to the computationally efficient simulation of computer networks is the use of variance reduction techniques. Varicnce reduction techniques exploit the statistical properties of the simulation model to reduce the uncertainty in the output. This chapter will describe four types of variance reduction techniques. The theory, applications, and usefulness for communication network simulation of antithetic sampling, common random numbers, control variates, and importance sampling will be discussed.

5.1 Antithetic Sampling

Antithetic Variates is a variance reduction technique that works by running two simulations in parallel with random number streams that are complementary. To achieve a reduction in the variance of the performance measure it is necessary to have a method to produce complementary random number streams which will have an opposite effect upon the performance measure. If these complementary streams produce two outputs which are negatively correlated then average of the output random variables will be a more efficient estimator for the mean than either of the two outputs.

If the simulation uses a U(0,1) random number generator, which generates numbers uniformly over the interval from 0 to 1, to produce the random samples for the simulation then a system independent method exists for producing a common random number stream. Let s be a sample from a U(0,1) random number generator, and let s' = 1 - s. A common means of producing complementary streams is to use samples from S as one stream and the corresponding sample of S' as samples for the complementary stream.

It is crucial that the two simulations remain synchronized [34]. For example in a queueing simulation it is important that for a random sample that is used as a service time in one simulation is also used as a service time in the complementary simulation. If a random sample was used sample as a service time in the primary simulation and in the complementary simulation the complement as an interarrival time, the effect would be to correlate the two runs. For example, if the sample generated a small service time in the primary simulation, its complement would result in a large interarrival time in the complementary simulation. The effect would be to shorten the waiting time in both simulations. The increase in correlation of an unsynchronized antithetic sampling scheme could result in no variance reduction or a variance increase. The synchronization problem can be solved by using a separate stream of random number for each random variable that is to be sampled in the simulation.

Burt and Garman [2] successfully combined antithetic sampling with a conditional Monte Carlo technique to reduce the variance in a network simulation. In the simulation of a five queue network, with the topology of a Wheatstone bridge (see figure 5.1) they found that by using antithetic sampling they could achieve the same confidence interval widths with only about 20% as much simulated time compared to using conditional expectation alone. They found that antithetic sampling resulted in a variance reduction about 80% of the time. Sometimes there was no variance reduction due to statistical fluctuations.

Fishman [10] used a generalized form of antithetic sampling called rotation sampling to speed the simulation of Markov Chains. In rotation sampling instead of only two complementary runs being made in parallel, many runs are made in parallel with the random samples all generated from one stream of random numbers designed to reduce the overall simulation variance.

As a test of this method he simulated a single server queue with finite capacity. The variance reductions that resulted depended upon the capacity of the queue and number of runs being made in parallel. The variance reductions ranged from about 1.4 up to about 70. The method described in [10] is only applicable to finite capacity systems and the variance reduction becomes smaller as the capacity of the system increases.

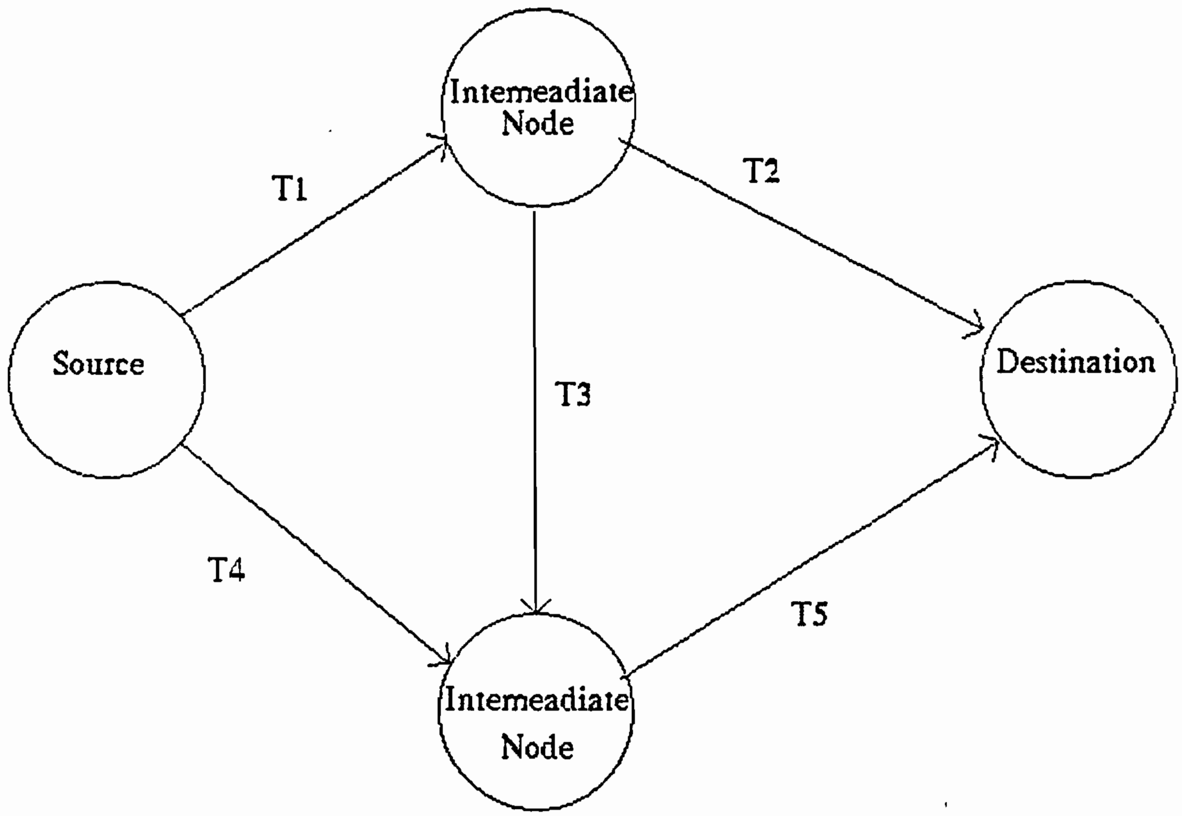

Figure 5.1 Wheatstone bridge network topology used by Burt and Garman [2].

Antithetic sampling has some important advantages. One is that it is very easily understood and implemented. Another advantage is that it is widely applicable to a variety of network simulation situations because it is not very implementation dependent. Also it is possible that combining antithetic variates with another variance reduction technique may result in a greater variance reduction than would otherwise be achieved.

There are also several problems with antithetic sampling. One it may be difficult to generate a complementary random sample that results in a negative correlation in the performance measure between the sample runs. In large networks the complex interactions between between the queues make it impossible to achieve a substantial correlation in the performance measure by using antithetic sampling [42]. A related problem is that it is difficult to estimate the variance reduction that can be achieved by using antithetic variates. In some cases there will be no variance reduction or even a variance increase when applying antithetic sampling [34,42].

Since the use of antithetic variates is a fairly general variance reduction technique, it might be useful to make some small test runs to determine whether antithetic variates will result in an improvement for a given model.

Antithetic sampling techniques will not usually be applicable to the simulation of computer communications networks because the necessary correlations cannot be maintained between the input and output random variables. The interactions in a typical computer communications network are too complicated for antithetic sampling to be effective.

5.2 Common Random Numbers

Like antithetic sampling, common random numbers is a variance reduction technique that uses and input sampling strategy to induce a correlation in the output performance measure. The common random number technique uses the same random inputs to a set of simulations of similar systems to produce performance measures that are highly correlated. Variance reduction by using common random numbers is useful to reduce the random variation between systems that are being compared. The method works by using the same random samples for in each of the simulation runs with different parameters to reduce the random variation between the runs. If there is a high correlation of the performance measure between the runs that use common random numbers this technique will be effective.

Synchronization between the simulations is just as critical as with antithetic sampling [34,27]. One way to easily achieve the synchronization is to use a separate random number stream for each random variable which is sampled. Then each of the simulations are run with the same random number seeds which will assure that each simulation is driven by the same random numbers for each of the same random variables.

There are examples of the successful application of the common random numbers technique in the literature. Kleijnen reported a 38% variance reduction in modeling alternative bus repair strategies [27]. Heikes, Montgomery, and Rardin reported a successful application of common random numbers to an inventory problem, but did not quantify the improvement [22].

The advantages to the common random number technique are primarily the ease of understanding and using the technique, and its general applicability. Common random numbers is also the most intuitive of all the variance reduction techniques.

Unfortunately there are some important disadvantages to using a common random numbers technique. One disadvantage to this technique is that it can only be applied to a comparison of alternative systems, and not to reduce the variance of the simulation of any one system. Another disadvantage of this technique is that in a complex computer network, it is doubtful that a significant correlation in the performance measures will be achieved by using the same input random number streams.

Again it is difficult to accurately predict the variance reduction that will be attained, if any, from the use of common random numbers.

The correlation between the performance measures depend on both the complexity of the system being simulated and on the similarity between the systems being compared. Very similar systems that have the same random inputs will obviously have highly correlated outputs. The best use for the common random number technique would seem to be for the comparison of several parameter sets for use in a model or parameter optimization. In these applications a high correlation between runs with the same sets of random numbers would be very likely.

Common random number techniques would be applicable to the comparison of very similar communications networks. This technique would be useful for determining the effect of changing some system parameter.

5.3 Control Variates

The method of Control Variates works by finding the correlation between the output random variable of interest X and a control random variable Y which has a known expected value. If the correlation can be found then an unbiased estimator for X the variable of interest can be formed using the simulated value of Y, the simulated value of X, the expected value of Y , and the correlation between X and Y.

Eqn 5.1 $\hat{X} = X - a(Y - E[Y])$

The variance of this estimator of X will be less than the variance of the simulated value of X if there is enough correlation between X and Y and the variance of Y is small enough [34].

Eqn 5.2 $Var(\hat{X}) = Var(X) + a^2 Var(Y) - 2 a Cov(X,Y)$

The constant a is chosen through a linear regression using the simulation data to minimize the variance of the estimator in 5.2.

The control variable can be generated as a part of the system simulation in which case it is an internal control. The control variable can also be generate with an auxiliary simulation in which case it is an external control variable. This technique is easily generalized to allow the use of more than one control variable [45,32,34].

In the simulation of a queueing system where the variable of interest is the expected waiting time, a possible internal control variable is the service time. The service time would be correlated with the expected waiting time and typically the mean of the service time would be user input for the simulation. Intuitively if the simulated mean service time for a simulation was above the expected value then the simulated mean waiting time would be expected to be above the expected value of the waiting time. The best estimate that could be made of the expected waiting time would be lower than the simulated mean waiting time.

An external control variable could be the random variable of interest in a modified version of the system that was simplified so that the expected value of the variable of interest can be calculated. With an external control variable it should be possible to have a much higher correlation between the control variable and the variable of interest which could result in a much large reduction in the variance. This advantage of using external control variables must be weighed against the cost of an auxiliary simulation for the control variable which is not needed when internal control variables are used.

Iglehart and Lewis report that a 50% reduction in variance is usually easily obtainable by using internal control variates in the simulation of GI/G/1 queueing system [24]. They used a function of the number of customers serviced in a regenerative period as a control variable. The effectiveness of this method proved to be independent of the traffic intensity.

Lavenberg, Moeller, and Welch applied control variables to queueing network simulation for computer performance modeling [32]. They found that with a closed network of 5 nodes it was possible to achieve variance reductions of the order of 25% to 55 %.

Heidelberger developed a method similar to the use of control variables that does not require the mean of the control variable to be known [20]. Instead of choosing the control variable to have a known mean, it is chosen to have a mean equal to the parameter of interest. The effect of this technique is to generate multiple estimates of the parameter of interest and to weight each estimate inversely proportionally to its variance. This method was able to produce a cpu usage reduction of 2 to 20 times when compared with a brute force technique. The variance reduction was partially offset by a requirement for 2.25 times more computer resources for each customer simulated. The amount of reduction depended upon the traffic intensity being simulated. The amount of variance reduction decreases as the traffic intensity increased, therefore this technique is least effective at the high traffic intensities where a variance reduction technique is needed the most.

The disadvantages of using control variates are the difficulty of finding a good control variable, the additional computer costs needed to calculate the control variable, and the effort required to calculate the correlation from the simulation data. It is often impossible to find an internal control variable with a known mean and a high enough correlation with the variable of interest. An external control variate technique may use so much computer time running the auxiliary simulation that the actual simulation cost cannot be reduced.

Variance reduction by the use of control variates is most practical when there is an internal or easily simulated external control variable that has a high correlation with the parameter of interest and a known mean. Since the effectiveness of this technique is dependent on creating a correlation between the performance measure and the control variable, this technique also becomes less effective in the simulation of large and complicated system.

It is difficult to apply control variates to the simulation of computer networks because it is difficult or impossible to calculate the mean for any suitable control variable due to the complexity of a realistic network simulation.

5.4 Importance Sampling

Importance sampling increases the efficiency of a simulation by concentrating the computational effort on the values of the random inputs which have the greatest effect on the performance measures being simulated.

Importance sampling is an attempt to sample the random inputs according to their effect on the performance measure of interest instead of according to their natural distribution.

Importance sampling is implemented by simulating a related random process with a biased input distribution which has the same expected value for the performance measure. The results from this simulation has a smaller variance because the sampling has been done primarily from the values of random inputs that cause greatest fluctuations in the performance measures. Although importance sampling has not often been applied to computer network simulation, it has proved very effective in reducing the amount of computation required for a bit error estimate in a communications link simulation [45].

Moy used importance sampling to efficiently simulate a realistic model of a truck loading dock system [38]. This model included a multiserver queue, a four level priority structure, and nonstationary arrival and service time distributions.

Simulation results using importance sampling on this model resulted in a variance reduction of between 61% and 74%, depending on the values of the simulation input parameters, in simulating the truck loading dock model. Since there was a small increase in the computer run time resulting from the addition of importance sampling, the resulting gain in computational efficiency was between 57% and 73%.

Moy concluded that importance sampling is a significantly more efficient alternative to a brute force simulation of a periodic queueing system.

5.4.1 A New Importance Sampling Technique

Frost, LaRue and Shanmugan developed a new variance technique by combining importance sampling with concepts from regenerative simulation theory [15].The application of this technique to M/M/1 and M/G/1 queues will be described in detail.

In this technique the importance sampling is accomplished by biasing (eg increasing) the arrival rate at the queue. Increasing the arrival rate will concentrate the simulation effort on regenerative cycles which have large numbers of entities. The regenerative cycles with large numbers of entities are also the cycles that have the greatest effect on the mean waiting time.

Clearly the expected waiting time in a regenerative queue can be found from equation 5.3 where S is the sum of the waiting times for all entities in the busy portion of a regenerative period, and N is the number of entities processed in that busy period.

Eqn 5.3 $$E[W] = \frac{E[S]}{E[N]}$$

S_B and N_B are the sum of the waiting times in a busy period and the number of entities processed in a busy period respectively for the system using a biased arrival process. Equations 5.4 and 5.5 would then describe the joint probability of the natural and biased processes respective.

Eqn 5.4 $P(S=s, N=n) = P(S=s | N=n)\ P(N=n)$

Eqn 5.5 $P(S_B=s, N_B=n) = P(S_B=s | N_B=n)\ P(N_B=n)$

Equation 5.6 is obtained by dividing equation 5.4 by equation 5.5.

Eqn 5.6 $$\frac{P(S=s,\ N=n)}{P(S_B=s,\ N_B=n)} = \frac{P(S=s | N=n)\ P(N=n)}{P(S_B=s | N_B=n)\ P(N_B=n)}$$

Since only the arrival process is being biased, and the service process is unchanged, The total waiting time distribution for a cycle of n entities will be the same for both the biased and unbiased case. Thus the statistical properties for busy cycles with the same number of entities will be the the same in both the biased and unbiased processes. Only the probability of busy cycles with n entities will be different in the biased and unbiased processes. This gives equation 5.7 and equation 5.8.

Eqn 5.7 $P(S=s | N=n) = P(S_B=s | N_B=n)$

Eqn 5.8 $$P(S=s,\ N=n) = P(S_B=s,\ N_B=n)\ \frac{P(N=n)}{P(N_B=n)}$$

The ratio of the probability of a busy cycle occurring naturally to the probability that the cycle occurring in the biased case is the weight that should be assigned to simulation information from the biased simulation. The weighting function is defined in equation 5.9.

$$\text{Eqn 5.9} \qquad w(n) = \frac{P(N=n)}{P(N_B=n)}$$

Let S^* and N^* be estimates of the unbiased process based on samples of the biased process and the weighting function. S^* and N^* are defined in equations 5.10 and 5.11.

$$\text{Eqn 5.10} \qquad S^* = S_B\, w(n)$$

$$\text{Eqn 5.11} \qquad N^* = N_B\, w(n)$$

It is desired to show that S^* is an unbiased estimator for S and that N^* is an unbiased estimator for N. To show that S^* is an unbiased estimator for S, it must be shown that $E[S^*] = E[S]$. To find $E[S^*]$ the definition for the expected value of S^* is used along with equations 5.9 - 5.11.

$$\text{Eqn 5.12} \qquad E[S^*] = E[S_B\, w(n)]$$

$$\text{Eqn 5.13} \qquad = \sum_{\forall s} \sum_{\forall n} s\, w(n)\, P(S_B=s, N_B=n)$$

Eqn 5.14 $$= \sum_{\forall s} \sum_{\forall n} s \, P(S_B = s, N_B = n)$$

Eqn 5.15 $$= \sum_{\forall s} \sum_{\forall n} s \, P(S = s, N = n)$$

But by the definition of expected value 5.15 is the expected value of S.

Eqn 5.16 $$E[S^*] = E[S]$$

A similar proof for that N* is an unbiased estimator of N is given in equations 5.17 to 5.21.

Eqn 5.17 $$E[N^*] = E[N_B \, w(n)]$$

Eqn 5.18 $$= \sum_{\forall s} \sum_{\forall n} n \, w(n) \, P(S_B = s, N_B = n)$$

Eqn 5.18 $$= \sum_{\forall s} \sum_{\forall n} n \, P(S_B = s, N_B = n)$$

Eqn 5.19 $$= \sum_{\forall s} \sum_{\forall n} n \, P(S = s, N = n)$$

Eqn 5.20 $$E[N^*] = E[N]$$

Since N* and S* are unbiased estimators for S and N it is possible to construct an estimator for W based on 5.22.

Eqn 5.22 $$E[W] = \frac{E[S]}{E[N]} \qquad \frac{E[S_B \, w(n)]}{E[N_B \, w(n)]}$$

By substituting the expected values of S* and N* based on M samples, an unbiased estimator for W can be found.

$$\text{Eqn 5.23} \qquad W = \frac{s_i\, w(n_i)}{n_i\, w(n_i)}$$

The weighting function must also be calculated from the time history of the simulation since it is necessary to know the length of the busy and idle periods to find $w(n_i)$. If the length of the busy period in the regenerative cycle is D = di and the idle period is I = pi then equation 5.24 is a formulation for $w(n_i)$.

$$\text{Eqn 5.24} \qquad w(n_i) = \frac{P(N=n_i \text{ in } D=di \cap N=0 \text{ in } I=pi)}{P(N_B=n_i \text{ in } D=di \cap N_B=0 \text{ in } I=pi)}$$

If the arrival process is Poisson then number of arrivals in D is independent of the number of arrivals in I because of the memoryless property of Poisson process.

$$\text{Eqn 5.25} \qquad w(n_i) = \frac{P(N=n_i \mid D=di)\, P(N=0 \mid I=pi)}{P(N_b=n_i \mid D=di)\, P(N_B \mid I=pi)}$$

5.4.2 Application to M/M/1 queueing system

If the arrival process is Poisson then equation 5.26 gives the probability of n arrivals in a busy period of length di for the unbiased process. Equation 5.27 gives the same probability for the biased process. Equations 5.28 and 5.29 give the probabilities for no arrivals during an idle period of length pi for the unbiased and biased arrival processes respectively.

Eqn 5.26 $\quad P(N=n_i \mid D=di) = \dfrac{\exp(-\lambda\ di)\ (\lambda\ di)^{ni}}{n_i\ !}$

Eqn 5.27 $\quad P(N=0 \mid I=pi) = \exp(-\lambda\ pi)$

Eqn 5.28 $\quad P(N_B=n_i \mid D=di) = \dfrac{\exp(-\lambda_B\ di)\ (l_B\ di)^{ni}}{n_i\ !}$

Eqn 5.29 $\quad P(N_B=0 \mid I = pi) = \exp(-\lambda_B\ pi)$

Equations 5.26 to 5.29 can be substituted into 5.25 to get the weighting function for the simulation of an M/M/1 queue.

Eqn 5.30 $\quad \hat{w}(n_i) = \exp((\lambda_B - \lambda)\ (di - pi))\ (\lambda/\lambda)^{ni}$

5.4.3 Simulation Results

This importance sampling technique was tested against brute force simulation and theoretical results for the M/M/1 queue. The system tested had a service time that was fixed at 1 and modeled for arrival rates between .1 and .8. The theoretical results were calculated using equation 5.31 [1]. Figure 5.2 shows the waiting times for this system. The brute force results were generated from the simulation of 25,000 packets. The biased simulation results were generated form a single biasing run of 25,000 packets at a traffic intensity of .85. Clearly both the brute force and the biased technique accurately estimate mean waiting time especially at lower arrival rates. Figure 5.3 contains the results from the simulations in tabular form.

$$\text{Eqn 5.31} \quad E[W] = \frac{\rho \mu}{1 - \rho}$$

The confidence intervals were calculated for both the brute force and biased simulations by using the method presented in [6]. The confidence interval estimation requires the series of samples of S and N, which are obtained directly in the brute force simulation and obtained from $s_B * w(n)$ and $n_B * w(n)$ for the biased simulation.

Figure 5.4 shows a comparison between the confidence interval widths generated from a brute force simulation of 25,000 packets and the results found from a biasing run of 25,000 packets made at a traffic intensity of .85. It is clear that at arrival rates of below .6 the brute force method has a considerably smaller confidence intervals. At arrival rates of .6 and greater the brute force and biased confidence intervals widths are similar.

The explanation for the poor performance of the biased method at low arrival rates is that because the biasing run was made at a .85 arrival rate much of the simulation effort is wasted simulating busy cycles that have so many arrivals that their probability of occurrence and hence weight is almost zero at arrival rates less than .60. It is important to note that the results from one

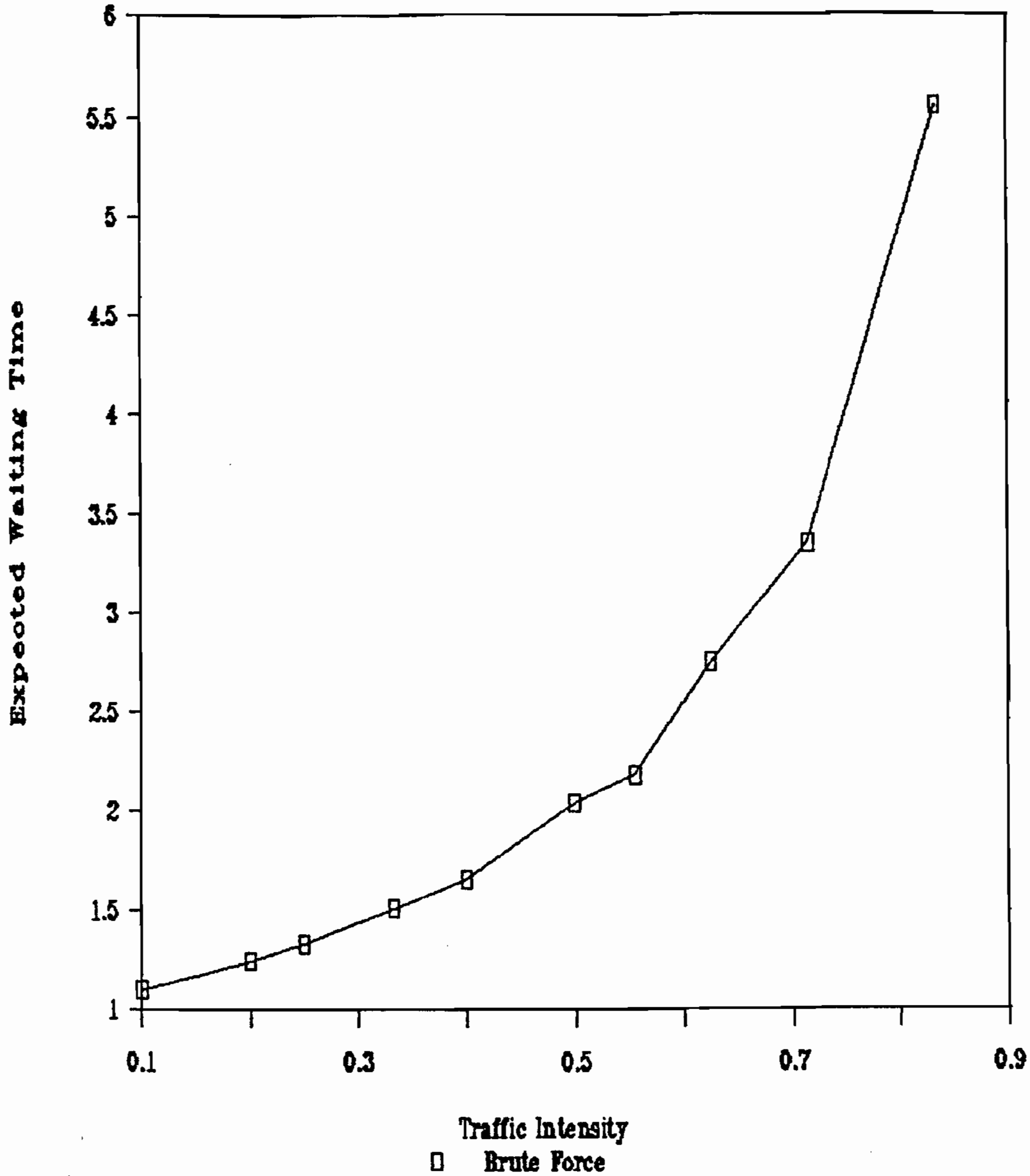

Figure 5.2 Expected waiting times for an M/M/1 queue found with brute force simulation.

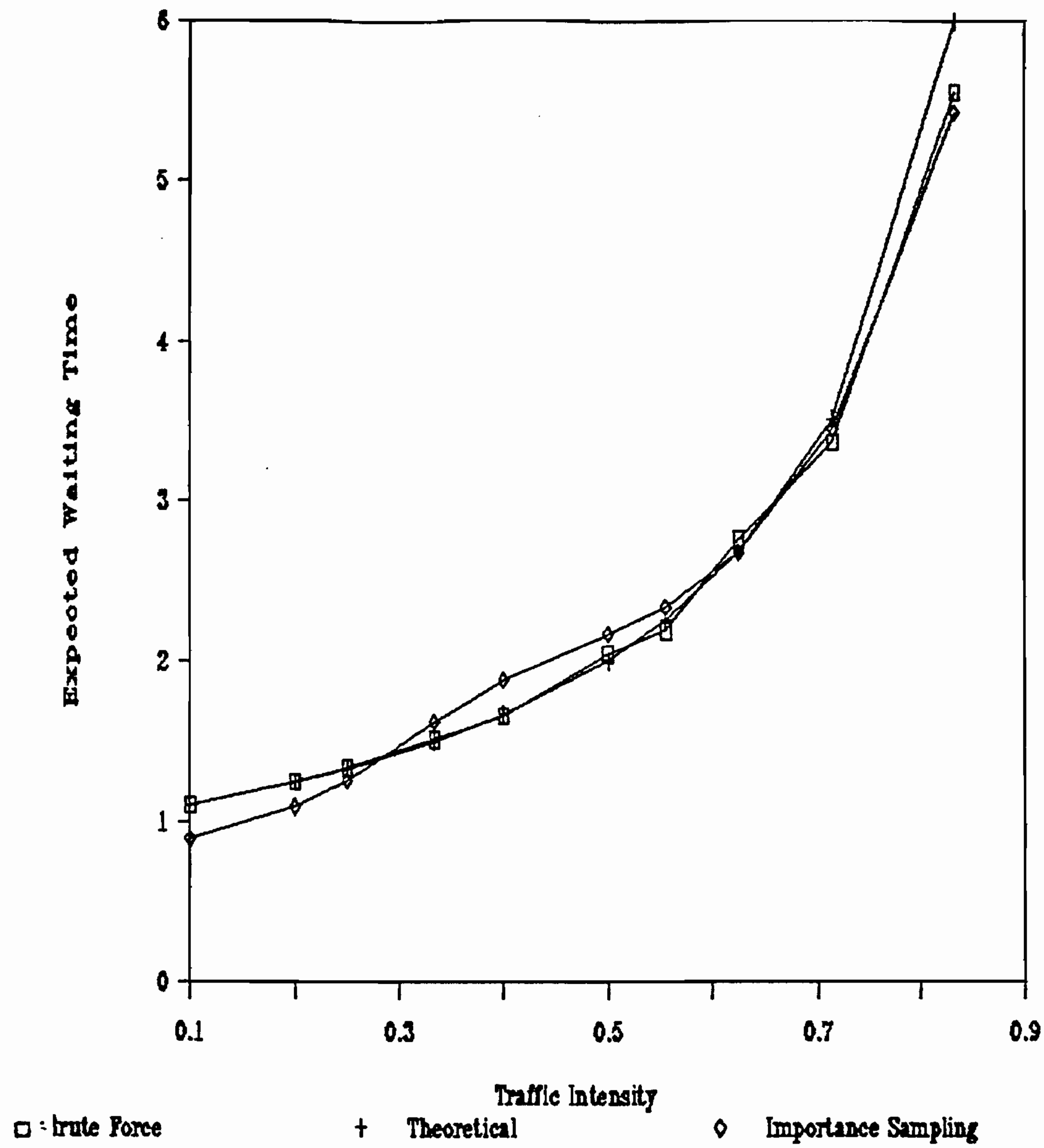

Figure 5.3 Comparison of biased method and brute force for the simulation of an M/M/1 queue.

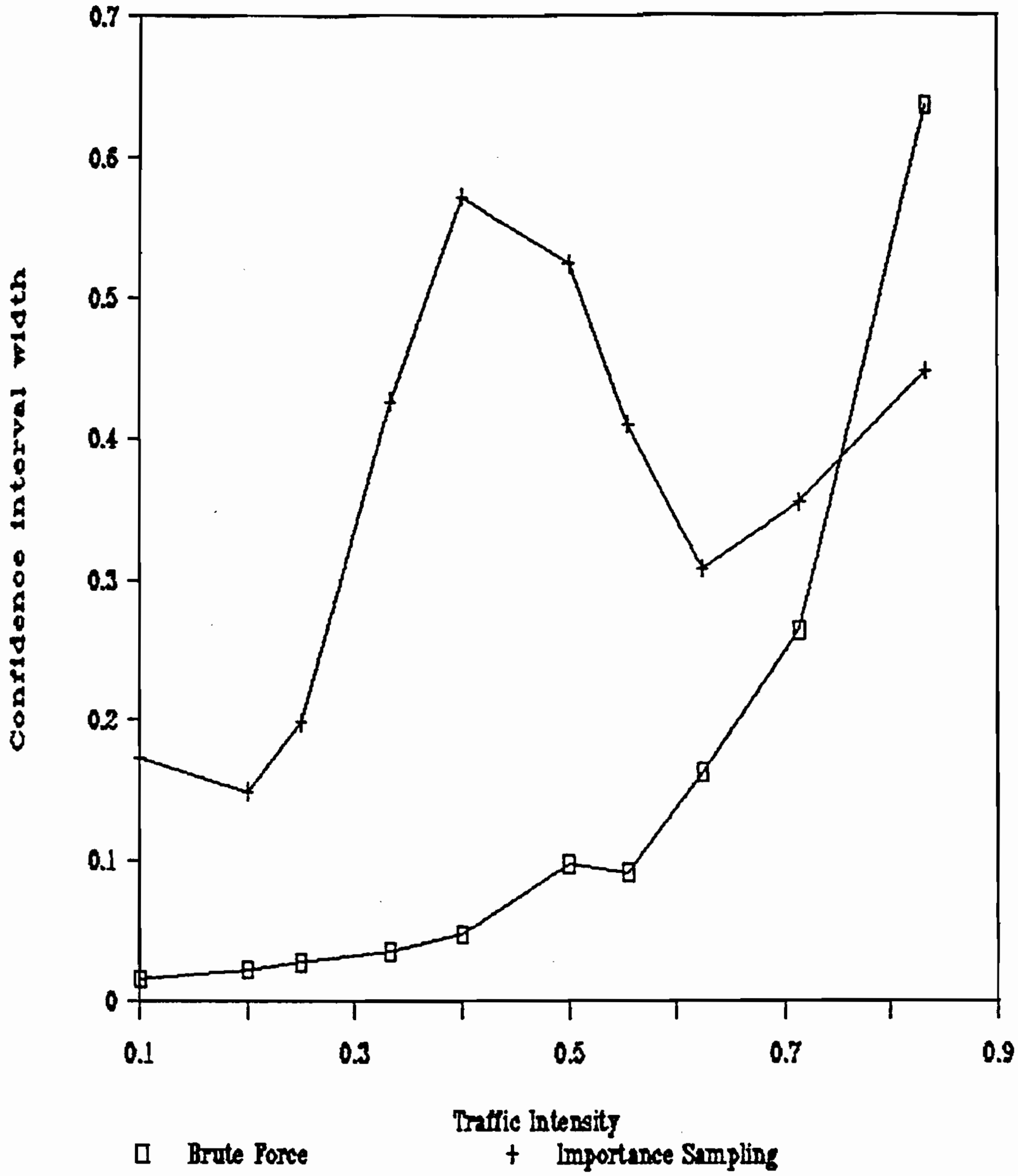

Figure 5.4 Confidence intervals for the waiting time of an M/M/1 queue. A Comparison between brute force and importance sampling.

biased run can be used to produce an estimated waiting time for all values of the arrival rate.

This technique was also applied to the simulation of an M/G/1 queueing system with rare events. A rare event is an event that occurs infrequently, but has a major impact on the performance measure of interest. Some examples of rare events in a computer network model would be link failures and data file transfers. The presence of rare events greatly increases the cost of simulation for a system because they have a large effect on the performance measure, so for accurate results a large number of events must be simulated. The rarity of these events means that if a large number are to be simulated the computer resource utilization of the simulation will be very high.

5.4.4 Application to System with Rare Events

The importance sampling technique was applied to an M/G/1 system with rare events by biasing the rare event arrival rate. Consider a system with two Poisson message sources, one has a high arrival rate of $\lambda_n = 0.5$ and a short message length of $\mu_n = 1.0$, while the rare event has a small arrival rate λ_r and a large message length of $\mu_r = 1000.0$. A biasing run was made with $\lambda_r = .0004$, which gave an overall system traffic intensity of .9. The data from the biasing run was used with equation 5.32 to estimate the waiting time.

Eqn 5.32 $\quad w(n_i) = \exp((\lambda_{rB})(di - pi))\,(\lambda_r/\lambda_{rB})$

The biasing run was made with 1,000,000 packets processed and then the results were weighted to produce estimates of the expected delays for rare event arrival rates between .000025 and .0003. For comparison brute force simulations at the same arrival rates were made using runs of 1,000,000 entities each. The results of the brute force runs is given in figure 5.8, and the delays calculated from the biasing run is given in figure 5.9.

The waiting time for this type of M/G/1 queue can be found analytically to provide a test of the validity of the simulation techniques. The arrival process remains Poisson with a mean interarrival rate of λ_t where:

Eqn 5.33 $\lambda_t = \lambda_n + \lambda_r$

The service time distribution is no longer Poisson but is given in equation 5.34.

Eqn 5.34 $f_X(x) = p_1 \; f_X(x|M1) + p_2 \; f_X(x|M_2)$

where $p_1 = \frac{\lambda_n}{\lambda_t}$ and $p_2 = \frac{\lambda_r}{\lambda_t}$

If the message lengths for both normal and rare messages are exponential then Eqn 5.35 give the probability distribution for the message length.

Eqn 5.35 $f_X = \frac{\lambda_n}{\lambda_t \mu_n} \exp(-x/\mu_n) + \frac{\lambda_r}{\lambda_t \mu_n} \exp(-x/\mu_r)$

A analytical formula for the delay time of an

M/G/1 queue which requires only the first and second moments of the service time is given in [47] :

$$\text{Eqn 5.36} \quad E[T] = \frac{\lambda E[X^2]}{2(1-\rho)} + E[X]$$

$$\lambda = \lambda_t \text{ and } \rho = \mu_r\lambda_r + \mu_n\lambda_n$$

The first and second moments of the service time are given in equations 5.37 and 5.38.

$$\text{Eqn 5.37} \quad E[X] = \frac{\lambda_n + \mu_n + \lambda_r\mu_r}{\lambda_t}$$

$$\text{Eqn 5.38} \quad E[X^2] = \frac{2\lambda_n \mu_n^2 + 2\lambda_r\mu_r^2}{\lambda_t}$$

This leads to an analytical delay formulas given in 5.39 and 5.40.

$$\text{Eqn 5.39} \quad E[T] = \frac{\lambda_n \mu_n^2 + \lambda_r\mu_r^2}{1 - (\mu_n\lambda_n+\mu_r\lambda_r)} + \frac{\lambda_n \mu_n + \lambda_r\mu_r}{\lambda_r + \lambda_n}$$

$$\text{Eqn 5.40} \quad \text{if } \mu_r \gg \mu_n \text{ then } E[T] \approx \frac{\mu_r \rho_r}{1-\rho}$$

It can be seen from figure 5.5, a comparison between the waiting times that were determined analytically, with a brute force simulation and by using the biasing technique, that all three methods produce means that are very close.

Confidence intervals were calculated for the biased simulation by using S^* and N^* as independent samples as is guaranteed by the regenerative nature of

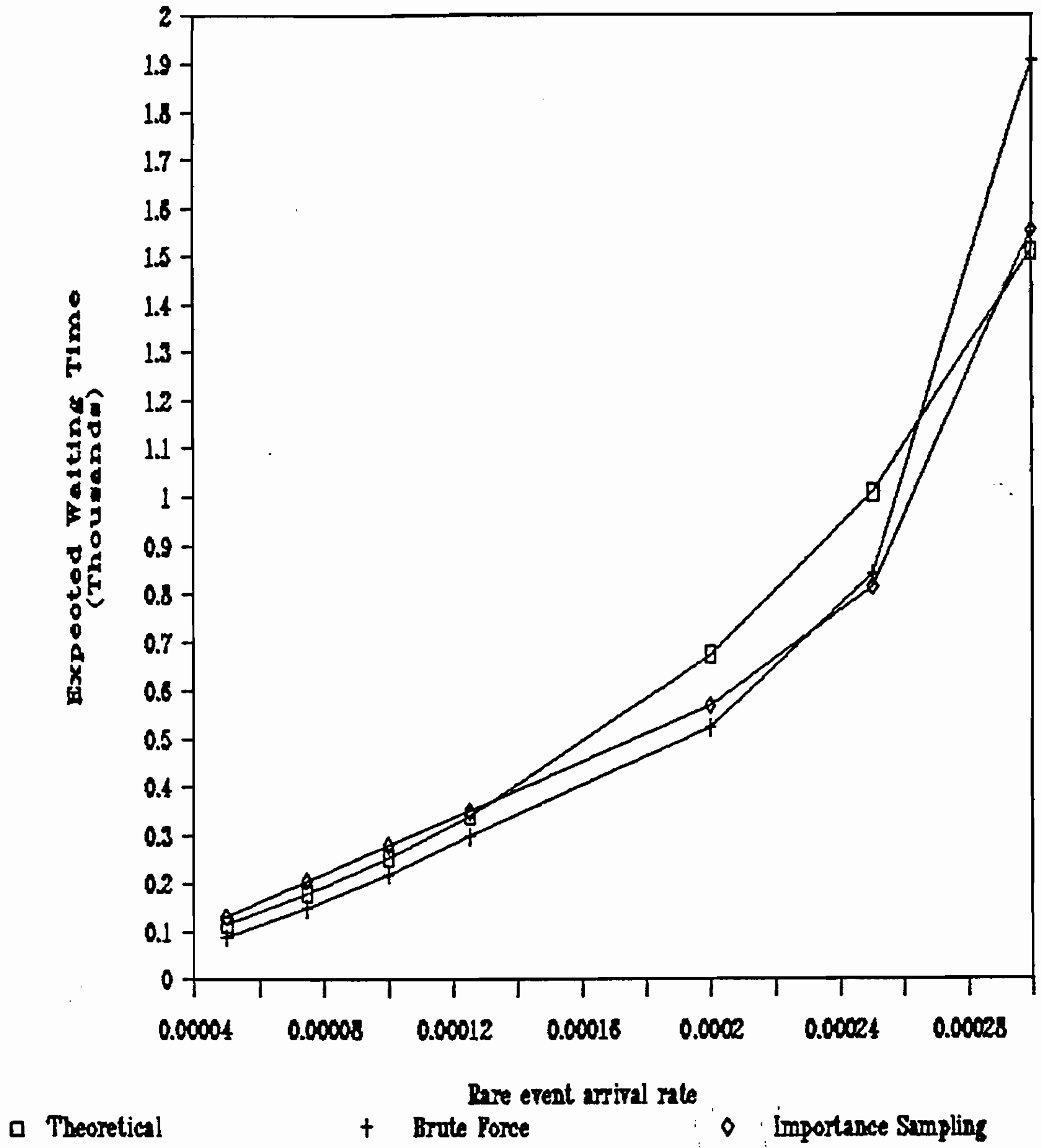

Figure 5.5 Expected waiting times for an M/G/1 queue with rare events. Comparison between analytical, importance sampling, and brute force.

the simulation. For the brute force case the confidence intervals were calculated by the using S and N as independent samples. Figure 5.6 is a plot of the confidence interval widths produced by using analytical, brute force, and biasing techniques. Figure 5.7 give the simulation results in tabular form.

It can be seen in figure 5.6 that arrival rates below 0.00015 cause the biased confidence interval widths to be much wider than the brute force confidence interval widths. This results from the fact that in the biasing run with a rare event arrival rate of .0004 most of the simulation effort is spent in simulating cycles with so many rare events that the probability of these cycles occurring at arrival rates of less than .00015 is almost zero. Thus these cycles are given a weight of near zero, so that they do not contribute to the simulation result.

For rare event arrival rates of .00015 and up the biased simulation produced confidence intervals of approximately the same width as a brute force simulation. While the simulation confidence intervals are not improved by the use of this method, it should be noted that an accurate prediction of the expected delay could be made over a wide range of arrival rates with a single biasing run.

Two obvious areas areas for future work with this new importance sampling technique are to reduce the confidence

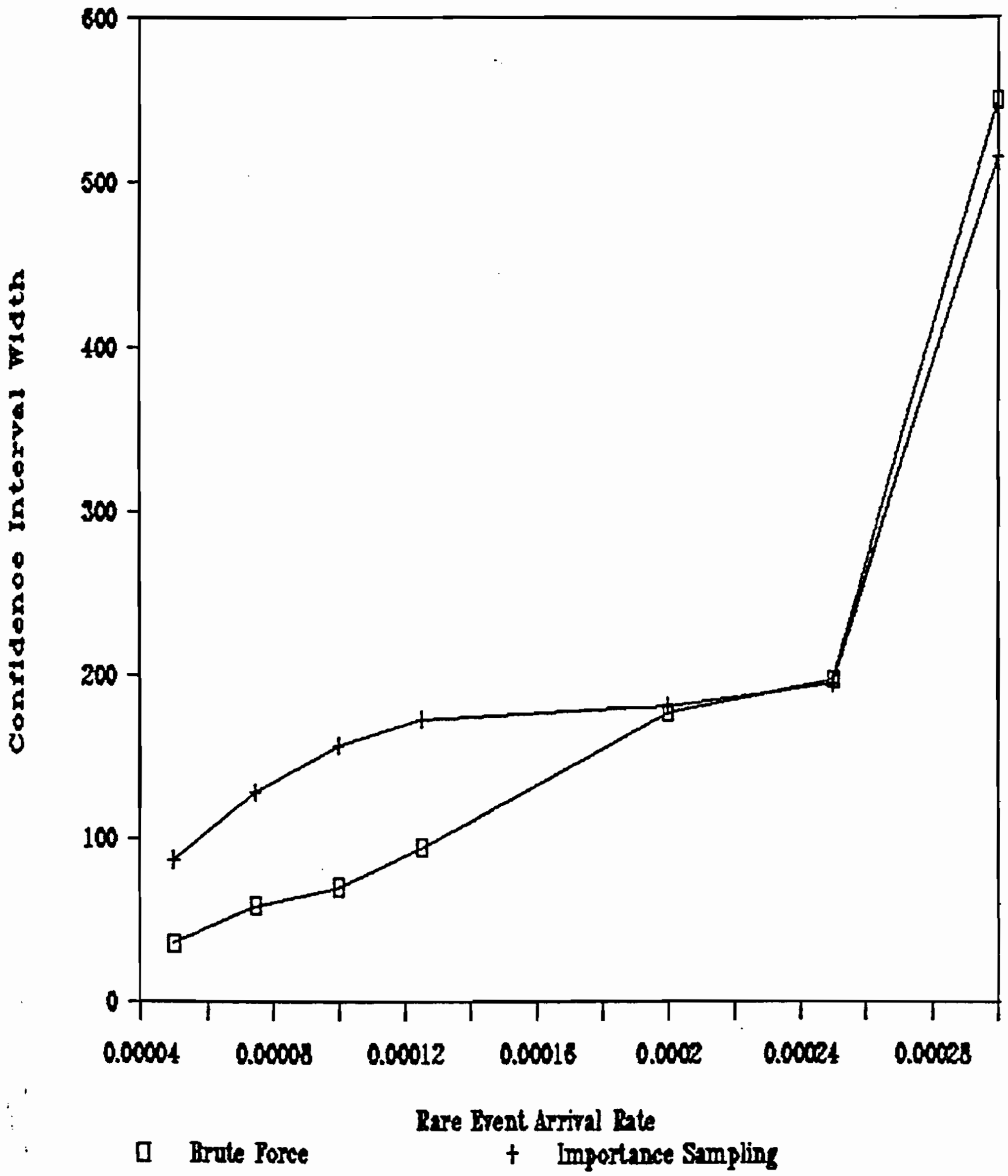

Figure 5.6 Confidence interval widths for waiting times of an M/G/1 queue. Comparison of brute force and importance sampling.

$\lambda_n = 0.25$
$\mu_n = 2.0$
$\mu_r = 1000.0$

The biasing run was at $\lambda_r = 0.0004$

λ_r	Theoretical	Brute Force		Biased	
	μ	μ	±	μ	±
0.0003	1508	1905	550	1552	516
0.00025	1007	839	197	814	195
0.0002	673	522	177	567	181
0.000125	339	299	94	350	173
0.0001	255	220	70	280	157
0.000075	181	149	59	207	129
0.00005	116	89	36	132	87

Figure 5.7 Comparison between brute force and biased techniques for the simulation of M/G/1 queues with rare events.

interval width of the biased result and to extend the technique to networks of queues. The confidence interval width of the biased result can be reduce by choosing an optimum arrival rate for the biasing run and by eliminating the simulation of busy cycles with a very low weight. An extension to networks is promising since this technique is not dependent on maintaining a correlation between the input and output random variables. This correlation is the basis for many of the other types of variance reduction techniques, but it can not be maintained in a complex communications network.

5.4.5 An Alternative Technique for Rare Event Systems

An alternative biasing scheme for modeling a rare event queueing system was also investigated. In this technique, the simulation effort is concentrated on only the regenerative cycles in which one or more rare events occurred.

The expected value of the waiting time can be expressed as a sum of conditional probabilities.

Eqn 5.41 $$E[W] = E[W|k=0]\ P(k=0) + E[W|k=1]\ P(k=1) + \ldots.$$

Where k is the number of rare events in a busy cycle. In a brute force simulation most of the simulation effort is spent on simulating busy cycles with no rare events, even though the preponderance of the uncertainty in the waiting time comes from busy cycles in which rare events occur.

An efficient technique was developed where additional rare event cycles were simulated by forcing a rare event at the beginning of a busy cycle. In a pilot test for this method a reduction in variance by a factor of 5 was achieved for $\lambda_n = .6$, $\lambda_r = .0001$, $\mu_n = 1.0$, and $\mu_r = 1000$.

If the expected waiting times for cycles without rare events is known, then this method can be simplified to a decomposition type of hybrid technique by simulating only the cycles with rare events. A similar technique was applied by Carter and Ignall to the simulation of a policy for dispatching fire trucks in a system with severe fires being a rare event [3]. In this simulation study rare events were forced at various times during the simulation. This technique resulted in a computational savings of 2 to 20 times compared with a brute force simulation.

The technique of only sampling regenerative cycles with rare events is related to the simulation of systems with rare events done with the new biasing technique. Both techniques work by increasing the proportion of the simulation effort spent on cycles with rare events. The biasing technique does this by increasing the arrival rate of rare events while the rare event only technique only simulates cycles with rare events.

Importance sampling techniques are applicable to improving the efficiency of computer communications network simulations because they do not depend on either maintaining the correlation between the input and output random variables or on finding the mean value of a related quantity analytically. The problem in applying importance sampling techniques is the difficulty of finding suitable biasing techniques.

5.5 Summary

Importance sampling is promising for the application to communication network simulation. Although it is not as easy to apply as antithetic sampling or common random numbers, it does have the advantage that it does not depend on the correlation of the input and output random processes. This correlation is strong for a simple single server queue, but as the complexity of the system increases the correlation between input and output random processes will decrease leaving techniques such as antithetic sampling and common random numbers impotent.

Variance reduction techniques have the advantages that they are generally applicable and that they can provide substantial computational savings compared to brute force techniques. Antithetic variates and common random numbers are the most easily applied, but depend on a correlation between the input and output random variables which is difficult to achieve in simulations of large communications networks. Control variates and importance sampling have been proven to provide a significant benefit in some applications, but their application is much more model dependent. Significantly importance sampling does not depend of a correlation between input and output random variables, so it could be applicable to larger communications networks.

6.0 Conclusions

Ideally a computationally efficient network simulation technique should be accurate, generally applicable, flexible, require a minimum of effort to use, and provide a substantial computational savings over a brute force simulation. Unfortunately, all of the techniques discussed fall short in some respects.

At their best, hybrid techniques can provide substantial computational saving by using analytical techniques where possible and still provide the modeling flexibility that is available with simulation. Hybrid techniques have been proven effective in the simulation of realistic computer communications networks. The disadvantages to hybrid techniques are that they may not apply to all systems, they require a significant effort to apply, and they may limit the flexibility of the simulation by making it impossible to calculate the variance of the performance measures.

Variance reduction techniques can lessen the computational demands of a simulation by using statistical theory to get the most information from the data generated. Variance reduction techniques have the advantages that they are generally applicable, and sometimes be applied with little programming effort. Unfortunately the techniques that can be easily applied such as antithetic sampling and common random numbers depend upon a correlation between the input and output random variables. This correlation is difficult to achieve in the simulation of realistic computer network because of the complex interactions in the network. Importance sampling, however, does not depend on this correlation, but does require more effort to apply.

Obvious areas for future work include the extension of the conditional expectation simulation of CSMA/CD local area networks to other types of multiple access communications networks, and to provide local as well as global performance estimates. Another area for work is the application of variance reduction techniques such as importance sampling to computer network models.

Computationally efficient simulation techniques are needed to reduce the high computational cost of computer network simulations. The need to model more complex networks and multiple layer protocols is increasing the need for these techniques. Unfortunately they have not yet been widely applied because of the added difficulty of using these techniques, the lack of general purpose simulations programs which apply these techniques, and the uncertainty whether any of these techniques will be beneficial in a given case. The effort required to utilize a system dependent technique for each simulation is often to great to be justified by the potential savings.

References

[1] William G. Bulgren, Discrete Systems Simulation, Prentice Hall, Englewood Cliffs, New Jersey, 1982

[2] John M. Burt, Mark B. Garman, "Conditional Monte Carlo: A Simulation Technique for Stochastic Network Analysis," Management Science, Vol 18 No. 3 p 207-217, Nov 1971

[3] Grace Carter, Edward J. Ignall, "Virtual Measures: a Variance Reduction Technique for Simulation," Management Science Vol 21 No. 6 p 607-616, February 1975

[4] Willy W. Chiu, We-Min Chow, "A Performance Model of MVS," IBM System Journal, Vol 17 No 4 p 444-462, 1978

[5] Charles E. Clark, "Importance Sampling in Monte Carlo Analysis," Operations Research, Sep - Oct 1961, p 603 -620

[6] M. A. Crane, A. J. Lemoine, An Introduction to the Regenerative Method for Simulation Analysis, Springer-Verlag New-York 1977

[7] M. A., Crane, Donald L. Iglehart, "Simulating Stable Stochastic Systems, I: General Multiserver Queues", Journal of the Association for Computing Machinery, Vol 21 No. 1 p 104 - 113, Jan 1974

[8] Michael A. Crane, Donald L. Iglehart, "Simulating Stable Stochastic Systems, II: Markov Chains", Journal of the Association for Computing Machinery, Vol 21 No. 1 p 114-123, Jan 1974

[9] Michael A. Crane, Donald L. Iglehart, "Simulating Stable Stochastic Systems: III Regenerative Processes and Discrete Event Simulations," Operations Research, Vol 23, No 1, p 33 - 45, Jan - Feb 1975

[10] George S. Fishman, "Accelerated Accuracy in the Simulation of Markov Chains," Operations Research, Vol. 31 No 3, p 466 - 486, May - June 1983

[11] George S. Fishman "Discrete Time Methods for Simulating Continuous Time Markov Chains", Management Science, Vol 20, No. 3, p 772 -778, Nov 1972

[12] George S. Fishman, "Simulating a Markov Chain with a Superefficient Sampling Method," Technical Report No. UNC/ORSA/TR-82/3, April 1982

[13] George S. Fishman, "Correlated Simulation Experiments", Simulation, Vol 23, p 177-180, Dec 1974

[14] Victor S. Frost, Bradley J. Wood "A Performance Evaluation of Local Area Networks", Telecommunication and Information Sciences Laboratory University of Kansas, Feb 1986

[15] V. S. Frost, K.S. Shanmugan, "Hybrid Approaches to Network Simulation," IEEE, ICC '86 p 228-234, 1986

[16] M. B. Garman, "More on Conditioned Sampling in the Simulation of Stochastic Networks," Management Sci., Vol 19 p 90-95, 1972

[18] L. J. George, "Variance Reduction for a Replacement Process," Simulation, Vol. 29, No. 3, p 65-74, 1977

[19] Phillip Heidelberger, Stephen S. Lavenberg "Computer Performance Evaluation Methodology," IEEE transactions on Computers, Vol. c-33, No. 12, p 1195-1220, Dec 1984

[20] Philip Heidelberger, "Variance Reduction Techniques for the Simulation Markov Processes I: Multiple Estimates, IBM J. Res. Develop, Vol 24 No 5 p 570-581, Sep 1980

[21] Phillip Heidelberger, D. L. Iglehart "Comparing Stochastic Systems using Regenerative Sampling with Common Random Numbers" Adv Appl Prob, Vol. 11, p 804-819, 1979

[22] R.G. Heikes, D. C. Montgomery , R. L . Rardin, "Using Common Random Numbers in Simulation Experiments -- an Approach to Statistical Analysis," Simulation Vol 27 No 3 p 81 - 85, 1976

[23] Arie Hordijk, Donald L. Iglehart, Rolf Schassberger, "Discrete Time Methods for Simulating Continuous Time Markov Chains," Adv. Appl. Prob., Vol 8 p 772-778, Mar 1976

[24] Donald L. Iglehart, Peter A. W. Lewis, "Regenerative Simulation with Internal Controls," Journal of the Association for Computing Machinery, Vol 26, No. 2 p 271-282, April 1979

[25] Donald L. Iglehart, Gerald S. Shedler, "Simulation of Response Times in Finite-Capacity Open Networks of Queues," Operations Research, Vol 26 No 5 p 89 - 914, Sep-Oct 1978

[26] Michel C. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communications Systems," IEEE Journal on Selected Areas in Communications, Vol SAC-2 No 1, 153 -170, Jan 1984

[27] J.P.C. Kleijnen, Statistical Techniques in Simulation Vol I, Marcel Dekker, 1974

[28] J.P.C. Kleijnen, "Antithetic Variates, Common Random Numbers, and Optimal Computer Time Allocation in Simulation," Management Science, Vol 21, No 10 p 1186-1195, 1975

[29] Paul J. Kuehn, "Modelling and Analysis of Computer Networks - Decomposition Techniques, Transient Analysis and Protocol Implications," Int. Conference on Communications, Amsterdam, May 14-17, 1984

[30] S. S. Lavenberg, P. D. Welch 1981, "A Perspective on the Use of the Control Variables to Increase the Efficiency of Monte Carlo Simulation," Mgmt Sci. Vol 27, No. 3 p 322-336

[31] S.S. Lavenberg, P. D. Welch, " Using Conditional Expectation to Reduce Variance in Discrete Event Simulation," Procedings of the 1979 Winter Simulation Conference, p 291 - 294

[32] Stephen S. Lavenberg, Thomas L. Moeller, Peter D. Welch, "Statistical Results on Control Variables with Application to Queueing Network Simulation," Operations Research, Vol. 30 No. 1 182-202, Jan-Feb 1982

[33] Averill M. Law, "Efficient Estimators for Simulated Queueing Systems," Management Science, Vol 22 No 1 p 30 - 41, Sep 1975

[34] Averill M. Law, W. David Kelton, <u>Simulation Modeling and Analysis</u>, McGraw-Hill, 1982

[35] Edward A. Macnair, Charles H. Sauer, <u>Elements of Practical Performance Modelling</u>, Prentice-Hall, 1985

[36] Do Le Mihn, Ronald M. Sorli, "Simulating the GI/G/1 Queue in Heavy Traffic," Operations Research, Vol 31 No 5 p 967 - 971, Sep - Oct 1983

[37] Bill Mitchell, "Variance Reduction by Antithetic Variates in GI/G/1 Queueing Simulations", Operations Research, Vol 21 p 988-997, 1973

[38] William A. Moy, "Variance Reduction," T. H. Naylor, Computer Simulation Experiments with Models of Economic Systems, John Wiley & Sons, 1971

[39] Barry Nelson, "A Decomposition Approach to Variance Reduction," Proceedings of the 1985 Winter Simulation Conference, p 23-32

[40] Peter J. P. O'Reilly, Joseph L. Hammond Jr, :An efficient simulation technique for Performance Studies of CSMA/CD Local Networks," IEEE Journal on Selected Areas in Communications, Vol Sac-2, No. 1, 1984

[41] A. Pritsker, Introduction to Simulation and SLAM II, Systems Publishing Company, West Lafayette, Indiana, 1984

[42] T. E. Ramsay, R. D. Wright, " Variance Reduction Techniques Some Limitations and Efficiency Limits," Modeling and Simulation, Volume 10 p 49-55, 1979

[43] Martin Reiser, "Performance Evaluation of Data Communication Systems," Proceedings of the IEEE, Vol. 70, No. 2, p 171-196, Feb 1982

[44] Reuven Y. Rubinstein, Ruth Marcus, "Efficiency of Multivariate Control Variates in Monte Carlo Simulation," Operations Research, Vol. 33, No. 3, p 661-667, May - Jun 1985

[45] K. Sam Shanmugan, P. Balaban, "A modified Monte Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communications Systems," IEEE Transactions on Communications, vol COM-28, Nov 1980

[46] J. G. Shanthikumar, R. G. Sargent, "A Unifying View of Hybrid Simulation/Analytic Models and Modeling," Operations Research, Vol. 31 No. 6 p 1030 - 1052, Nov-Dec 1983

[47] B. W. Stuck, E. Arthurs, A Computer & Communications Network Performance Analysis Primer, Prentice-Hall, Englewood Cliffs, New Jersey, 1985

[48] Andrew S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, New Jersey, 1981