SIMULATING THE NONSTATIONARY BEHAVIOR OF COMMUNICATION NETWORKS

D. Tipper, W. Lovegrove, J. Hammond, K. Pawlikowski
Electrical & Computer Engineering Department
Clemson University
Clemson, SC 29634-0915

ABSTRACT

Discrete event simulation has emerged as the basic tool for conducting detailed design and performance evaluation studies of communication networks. However, current simulation techniques focus almost exclusively on evaluating the steady-state behavior of the system under study. Recently, it has been increasingly noted that communication networks must not only have acceptable steady-state performance but must also deliver optimal performance under transient or nonstationary conditions. In this paper, the problem of simulating communication networks under nonstationary conditions is considered. The critical issues in simulating the nonstationary behavior, such as determining appropriate performance measures, calculating confidence intervals, selecting initial conditions, etc., are discussed and solution approaches to these issues proposed.

1.0 INTRODUCTION

In the recent times, it has increasingly been noted that communication networks must not only have acceptable steady-state performance but must also deliver optimal performance under transient or nonstationary conditions [1-4]. For example, congestion in communication networks is in general a transient phenomenon, but congestion controls are designed and their performance evaluated by a steady state analysis. Obviously the performance of congestion control schemes designed in such a fashion may be far from optimal when nonstationary conditions prevail in the network.

Nonstationary conditions occur in communication networks when the packet/message/call arrival processes to the various network queues or the service rates of the queues vary with time. A simple evaluation of the time constant (relaxation time) of the widely used $M/M/1$ queueing model of a network transmission link [5] indicates that the time taken by the queue at the link to reach steady-state after an event that generates transient conditions will be quite long, particularly when the link is heavily loaded, and hence periods of nonstationary or transient behavior prevail during much of the time.

Typical events that give rise to transient conditions are: load sharing, changes in routing and flow control parameters (i.e., adaptive routing and flow control), failures of links, nodes, or other network resources, topological changes, network start-up and shut-down, and most importantly nonstationary input loads. It is well known that in many packet switched communication networks, the user demand for data communication varies so rapidly that the load is essentially nonstationary for large time periods. In fact, it is in recognition of the nonstationary conditions that exist in most packet switched wide area networks (WAN's) that there has been such a considerable effort at developing adaptive routing and flow control methods. Furthermore, it has very recently been noted [1] that a class of data communication networks called rapidly reconfigurable networks (RRN's) exists. These networks are subject to frequent, if not continuous, changes in the combination of network geometry, user demand for data communication, and transmission link capacity, such that nonstationary conditions always exist and the transient behavior is the only meaningful measure of performance. Also, in the telecommunication environment it has been repeatedly noted [3,4] that for Integrated Digital Services Networks (ISDN) the data communication sessions will be in a transient state due to the priority given to voice sessions which results in a time-varying transmission bandwidth for the data sessions. Thus there exists a need for a new performance evaluation techniques aimed at analyzing the time-varying behavior of communication networks.

In this paper, the problem of studying communication networks under nonstationary conditions is considered with
the aim of identifying and developing simulation techniques that can be used in the performance evaluation and the design of control procedures.

2.0 MODELING TECHNIQUES

Models of communication networks focus on the contention for shared resources and the resultant queueing for these resources. Quite naturally, the most widely used models are based on queueing theory and represent the system as a network of interconnected queues. While there has been a considerable effort in analyzing queueing networks with particular emphasis on their application to data communication networks [6], the results are almost exclusively limited to determining steady state behavior. Precise analytical solutions for the nonstationary behavior of queueing systems are largely unavailable and difficult to obtain. For example, the exact expression for the time varying average queue length in the simple M/M/1 queue involves an infinite sum of Bessel functions. This is difficult to work with and exact solutions for queueing networks would seen intractable. The principal useful analytical results are approximate expressions for the settling time of queueing systems [5,7]. The settling time is defined as the time taken by the system to return to steady state after a perturbation. The available results indicate that the time it takes to reach steady state is very long when the system is heavily loaded and the system is almost always in a transient state.

Recently [7] an alternative approach to trying to find closed form analytical expressions has been proposed. The general idea is to formulate a differential or partial differential equation model of the queueing system (i.e. Chapman-Kolomogorov equations, diffusion models, etc ...), which can be solved using numerical methods such as Runge-Kutta techniques. The principal drawbacks of this approach are the difficulty in formulating performance measures and the computational complexity involved in solving large numbers of differential equations.

In light of the difficulty involved in conducting precise analytical studies of the behavior of data communication networks, simulation is an attractive alternative. Unlike analytical techniques which require a great number of simplifying assumptions to get a tractable model, simulation allows the network to be modeled to an arbitrary degree of accuracy and detail. In fact, discrete-event simulation has emerged as the basic tool for conducting detailed studies of communication networks under realistic conditions and several simulation languages and packages are presently available for performance evaluation and design purposes [8-12].

Due to the random nature of the events in discrete event simulation, the simulation represents a statistical experiment and a central issue is how to obtain meaningful performance values from the simulation output data. For example, if a simulation is run for a shorter or longer period of time or if different numbers were used to initialize the simulation, then different values of the system response and performance measures can be obtained. Confidence interval techniques are often used to quantify the quality of a performance estimate made from simulation data. Basically, a confidence interval represents how accurate a statistical estimate made from the simulation data is of the true behavior of the stochastic simulation model of the system. Specifically, a 1% confidence interval of width (a, b) on a particular performance measure y means that if the simulation were independently repeated a large number of times, the statistically estimated value of y obtained from each simulation run would fall in the interval (a, b) on approximately 1% of the runs. Several methods exist for generating confidence intervals [13-17], such as the method of batch means and the regenerative method.

While there exist many well established methods for constructing simulation models of computer networks and for performing statistical analysis of the simulation output data [9-17], these techniques have been focused almost exclusively on evaluating the steady-state performance of the system under study. There are important differences when the nonstationary behavior is to be simulated. As a simple example, note that in applications where nonstationary behavior is important, performance criteria based on time averages such as throughput, average time delay, and average number stored in a queue are not appropriate, since such quantities are not sensitive to what happens at specific times or over specific short time intervals.

3.0 SIMULATION METHODS

To best understand the differences in studying the stationary and nonstationary behavior of system by simulation we view the simulation within the framework of stochastic processes. Note that the output data of one simulation run can be viewed as a single sample path of a stochastic process $X(w; t)$, where $w$ is a random variable...
over the set or ensemble $\Omega$ of possible sample paths (simulation run outputs) and $t$ is time. In a specific realization of the stochastic process, called a sample path or sample function [13], $\omega$ is fixed (i.e. $\omega = \omega_i$) and the resulting $X(\omega_i;t)$ is a time function. On the other hand, if $t$ is fixed (i.e. $t = t_0$) the resulting $X(\omega;t_0)$ is a random variable. Of course, if both $\omega$ and $t$ are fixed, the resulting $X(\omega_i;t_0)$ is a single fixed number (observation). In this context, the observation of a communication network's response over time (queue lengths, network delay, throughput, etc ... ) in a simulation run corresponds to a single sample path $X(\omega_i;t)$ and an independent repetition of the simulation run will result in a different sample path $X(\omega_i;t)$. To illustrate the sample path behavior of a queueing simulation, a single $M/M/1/20$ queue which is initially empty and has arrival rate $\lambda = .8$ and service rate $\mu = 1.0$ was simulated. Figure 1a shows the typical behavior of the queue length in this system for two independent simulation runs, clearly demonstrating the sample path behavior.

In order to measure the performance of the simulated system, we note that averages for stochastic processes can be defined in two distinct ways: over a single sample path as an average on $t$ or over the ensemble of sample paths as an average on $\omega$. If the ensemble averages are independent of $t$, the process is said to be stationary (in a limited sense). Most of the work on discrete-event simulation applies to determining the steady-state behavior of the system for which it is assumed, sometimes tacitly, that time and ensemble averages are equal and that processes are stationary and ergodic. This assumption makes time average performance measures applicable and in the simulation of computer networks time average measures such as the average network delay, network throughput, and buffer utilization are commonly used. The assumption of ergodic processes often enables the steady-state behavior to be determined from a single long simulation run. When estimating the steady-state behavior, the initial transient response must be neglected in determining the performance measures in order to avoid obtaining biased values. This is normally accomplished by discarding the transient portion of the response. However, determining the length of the transient response is difficult and current techniques rely on heuristic rules, such as determining when the variance of a estimator of a mean quantity becomes small or looking at the output of pilot simulation runs [17].

In the simulation of nonstationary behavior, the assumption of stationary ergodic processes is not valid. Thus it is necessary to define different types of performance criteria which are time-dependent and can be statistically estimated from simulation output data. Ideally, we would like to be able to statistically estimate some performance criterion from a single specific sample path (simulation run) which describes the network's response in general. However, this is not possible since time averages are not applicable and to obtain generally applicable measures, it is necessary to consider statistical properties over the ensemble $\Omega$ at a particular time $t_0$. Such measures can take the form of time-varying probabilities like

$$P(X(\omega;t_0) < a) = b$$

or moments such as

$$AVE = E\{X(\omega;t_0)\}$$

and

$$VAR = E\{([X(\omega;t_0) - E\{X(\omega;t_0)\}]^2}\}.$$  \hspace{1cm} (3.3)

In the case of moments like (3.2) and (3.3), relations such as Tchebycheff's inequality, which give or bound the probability of a sample of a random variable falling outside a range about its mean are applicable. Determining meaningful performance measures appropriate for ensemble averages is a difficult problem, especially since many of the standard queueing relationships (such as, Little's formula, output rate of queue equal input rate) do not hold and the traditional measures such as average throughput and delay are inapplicable. Some examples of possible performance measures are: $L(t_0)$ the expected number of customers in the queue at time $t_0$ and $D(t_0)$ the average departure rate from the queue at time $t_0$. The time varying expected queue length $L(t_0)$ is determined by observing the queue length of each sample path at time $t_0$ and averaging the observations. Whereas the average departure rate $D(t_0)$ is determined by observing the busy/idle status of the server at time $t_0$ for each sample path, averaging the observations, and then multiplying the average by the service rate.

In order to calculate quantities such as (3.2), an ensemble of sample functions must be generated and this corresponds to performing a very large number of independent simulation runs and recording the network response. Due
to the large number of runs required to obtain an accurate ensemble, long sequences of random numbers are required and the quality of the pseudo-random number generator must be carefully considered with regard to the correlation effects between the simulation runs and repetition of the number generator. Also, data reduction methods may be necessary to store the outputs of all the simulation runs.

Confidence intervals can be constructed on the performance measures by noting that each observation \( X_i \) are realizations of independent and identically distributed random variables. Thus the central limit theorem can be used in the classical fashion [9] to construct confidence intervals. For example, the \( 100(1 - \alpha)\% \) confidence interval on the sample mean \( \bar{X} \) calculated from \( n \) observations is given by:

\[
\bar{X} \pm t_{n-1,1-\alpha/2} \frac{\bar{X}}{\sqrt{n}}
\]

where \( \bar{X} \) is the sample variance and \( t_{n-1,1-\alpha/2} \) is the upper \( (1 - \frac{\alpha}{2}) \) critical point obtained from the student - \( t \) distribution within \( n - 1 \) degrees of freedom.

Another difficulty in performing nonstationary simulation is the fact that there are very few techniques available for generating nonstationary random processes on the computer, with the exception of Poisson and Gaussian processes [9]. Furthermore, the techniques that do exist are based on the thinning approach and are not computationally efficient. Also, unlike in steady-state simulation, the initial state of the system is a crucial parameter in the transient behavior and techniques for determining and generating proper initial states are needed. Yet another problem is determining the time points \( t_0, t_1, \ldots, t_a \) to gather statistics. Obviously one wants to observe the system often enough to accurately determine the system response, but not so often as to add unnecessary computational burden. Thus, while simulation of communication networks is fairly straightforward when analyzing the steady-state behavior, several unsolved problems remain in determining the nonstationary response.

To illustrate some of the issues discussed above, consider the problem of determining the transient response of a finite \( M/M/1 \) queue. As noted previously the expected queue length as a function of time \( L(t) \) is an appropriate performance measure. The simplest case of when the arrival rate and service rate of the queue are constant is considered here with \( \lambda = .8, \mu = 1.0 \) and the queue in an initial empty state. Note that after an initial transient the system will attain a steady state. The theoretical behavior of the queue can be determined by numerically integrating the Chapman-Kolomogorov differential equation model as discussed in [7]. In Figure 1b, the theoretical average queue length is plotted along with the ensemble average of 50 and 5000 independent simulation runs. One can clearly see that a large number of independent runs must be generated to get an accurate portrayal of the system behavior, hence large amounts of computer run time are required. Evidently, such exercises become extremely computationally intensive when simulating large networks of queues, underscoring the need for new simulation methods which provide computationally efficient techniques for handling multiple simulation runs. Nevertheless, the possibility of using discrete-event simulation for the performance evaluation under nonstationary conditions, as outlined here, provides the network designer with a valuable tool.

REFERENCES


