Traffic Modeling For Telecommunications Networks

As new communications services evolve, professionals must create better models to predict system performance.

Victor S. Frost and Benjamin Melamed

The efficient flow of information is a key element in today's technology and business environments. This flow is supported by a complex computer and communications infrastructure that, if properly designed and operated, is invisible to end users. High-speed network transport mechanisms, such as ATM, serve as enabling technologies for new classes of communications services, such as multimedia and video on demand, that are typically grouped under the heading of B-ISDN.

Asynchronous Transfer Mode

As these new communications services evolve and the needs of users change, the enterprise must respond by modifying existing communications systems or by implementing entirely new ones. To this end, telecommunications professionals are being called upon to design and manage these systems in the face of fast-moving technology and a climate of increasing customer expectations. Design and management decisions require predictions of network performance; decisions based on poor predictions may adversely affect network customers' perception of the new technology. Analytical techniques, computer simulation, projections from existing experience, and experimentation are methods that are used to evaluate and compare network designs and protocols. Independent of the prediction methodology, however, design and management decisions often must be made with incomplete knowledge of impending user demands and how the system will evolve.

methods

The goal of this article is to provide an overview of computer simulation modeling for communications networks, as well as some important related modeling issues. This article is intended to be neither a detailed tutorial on computer simulation of communications networks, nor a monograph for experts in discrete-event simulation, nor a review of specific simulation tools (refer to [1] in this issue); several excellent texts provide comprehensive treatments of various aspects of simulation [2-10]. Rather, we give a brief overview of discrete-event simulation and single out two important modeling issues that are germane to extant and emerging networks: traffic modeling and rare-event estimation.

Monte Carlo computer simulation is a flexible performance prediction tool used widely in science and engineering. Its flexibility stems from the fact that it consists of a computer program that "behaves" like the system under study. Unlike analytical models, which often require many assumptions and are too restrictive for most real-world systems, simulation modeling places few restrictions on the classes of systems under study. For communications networks, developing a simulation program requires:

- Modeling random user demands for network resources.
- Characterizing network resources needed for processing those demands.
- Estimating system performance based on output data generated by the simulation.

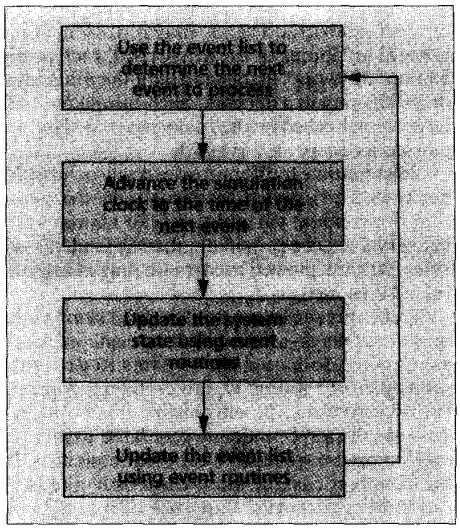
The operation of communications networks can be conveniently described by simulation programs. The stochastic nature of demands for network resources is modeled using pseudo-random number generators. The execution of a computer simulation model is comparable to conducting an in-vitro experiment on the target system. All in all, a Monte Carlo simulation program can serve as a flexible testbed for conducting system experimentation without disturbing production networks or constructing software/hardware prototypes.

Several factors contribute to the difficulty of successfully applying simulation technology to performance evaluation. To begin with, the nature of user demands for network resources may be incompletely known or poorly understood. Further, networks are in a perpetual state of flux, because user services and networking technologies constantly alter usage patterns. Thus, any model of user traffic would necessarily be approximate or even speculative, especially when new services are under consideration. Traffic models are reviewed subsequently in this article.

VICTOR S. FROST is director of the Telecommunications and Information Sciences Laboratory at the University of Kansas.

BENJAMIN MELAMED is head of the Performance Analysis Department, C&C Research Laboratories, NEC USA Inc.

0163-6804/94/$04.00 © 1994 IEEE



■ **Figure 1.** *DES flow design.*

Simulation programs can be used to closely model the processing of user demands for network services. Such models can contain tremendous detail, especially for large networks. However, the execution of detailed models may require prohibitive amounts of computational resources. It is not uncommon for network simulations to require days of processing time on a modern workstation. The analyst must be aware of the trade-off between model detail and simulation execution time. Frugal modeling is often called for, i.e., modeling that includes only those network functions that have an appreciable impact on the desired performance metrics. The simulation of a large and complex network is treated in a companion paper [11].

Because executing a simulation is analogous to conducting an experiment involving randomness, simulation outputs must be treated as random observations. In a similar vein, because a model is viewed as a faithful representation of the target system, instrumentation is required in order to collect statistics and formulate performance predictions. The success of a simulation study hinges on identifying appropriate performance metrics and then devising a strategy for exploring the ensuing performance response surface. A system response surface is rarely one-dimensional; more often, the desired response of the system is a function of a parameter vector. For example, the cell-loss rate in an ATM network is a function of a set of congestion-control parameters. Correlation in the measured observations must be taken into account when forming statistical performance estimates. For example, positive autocorrelation in a sequence of delay times would manifest itself as bursts of long (short) delays; i.e., if message k has experienced a long (short) delay, then it is likely that message $k+1$ will also experience a long (short) delay. The autocorrelated nature of samples obtained from simulation (or measurement records from operational systems) complicates the task of forming performance predictions. Methods such as replication, batch means, and regeneration are used to address sample dependence [2]. Furthermore, rare events, such as ATM cell losses, are key metrics that characterize the performance of emerging broadband networks. Using simulation to estimate the probability of rare events and their effect on performance is problematic, because vast computational resources may be required to generate a sufficient number of events from which statistical estimates may be formed with adequate statistical confidence. Simulation techniques to estimate rare-event probabilities in communications networks are addressed in a following section.

The rest of this article provides an overview of simulation, with a slant toward telecommunications networks, and discussions of two modeling issues of special significance to today's networks: traffic models appropriate for high-speed networks and importance sampling as a tool for estimating rare-event probabilities in broadband networks.

Discrete-Event Simulation

From a high-level perspective, telecommunications networks can be seen as users who generate demands for network resources, and protocols (distributed algorithms) that control the allocation of network resources to satisfy those demands. The generation of user demands and their satisfaction are encapsulated in simulation events, which are ordered by their time of occurrence. The action of the protocols depends on the state of the network at the time the demand was issued. A simple routing algorithm, for instance, may send packets to the output link with the shortest buffer. This event-based processing lends itself to a method known as discrete-event simulation (DES) [12]. Most simulation tools for telecommunications networks are based on DES [1]. An important characteristic of DES models is that they keep time via simulation clocks, which change by random increments. The basic executable unit in DES models is an event (a program that is executed at discrete simulation times).

In DES, the state of the simulated system is stored in a set of system state variables. Event routines cause state variables to be modified. An event list is used to control the execution sequence of these event routines; the list consists of events in increasing chronological order. Event routines can add or delete items from the event list, and pseudo-random number generators in the event routines provide the requisite randomness for modifying and scheduling of future events. From a high-level viewpoint, running a simulation is, in essence, the repeated execution of a loop, where at each iteration the most imminent event (the one with the earliest scheduled time) is executed in turn. A flow diagram for DES is shown in Fig. 1.

Typical events in a communications network simulation include the arrival of demands for network resources. A description of network resources, which is needed to satisfy the demand, is associated with each arrival. The time between successive demand arrivals as well as the nature of the required network resources are important elements of traffic modeling to be discussed later.

In communications network simulation, entities acted upon by event routines include calls, messages, packets, and cells. These entities are represented internally by data structures, which are often closely related to the message/packet format defined by the protocol. For example, source and destination addresses as well as control infor-

The execution of a computer simulation model is comparable to conducting an in-vitro experiment on the target system.

An understanding of the nature of traffic in the target system and selection of an appropriate random traffic model are critical to the success of the modeling enterprise.

mation and data might be organized in standard packet formats. A data structure representing a packet would contain these elements in addition to simulation-specific information. A packet-creation time stamp, for instance, could be used for statistics collection. The data field might contain a length indication or, in more detailed models, a pointer to another data structure that represents a network-layer packet. Such encapsulation of data structures is a common feature in communications networks and is also supported by object-oriented programming languages. Clearly, every DES must have an initialization mechanism to establish the initial system state, statistical collection routines to obtain measurements, a post-processor to transform the collected statistics into the desired performance estimates, and a coordinating program to control the event list, post-processor, and initiation and termination of the simulation.

Traffic modeling is a key element in simulating communications networks. A clear understanding of the nature of traffic in the target system and subsequent selection of an appropriate random traffic model are critical to the success of the modeling enterprise.

An Overview of Traffic Modeling

In this section, we survey commonly used traffic models. Such models are employed in two fundamental ways: either as part of an analytical model, or to drive a discrete-event simulation. The most common modeling context is queueing, where traffic is offered to a queue or a network of queues and various performance measures are calculated.

Simple traffic consists of single arrivals of discrete entities (packets, cells, etc). It can be mathematically described as a point process [13], consisting of a sequence of arrival instants $T_1, T_2 \ldots, T_n \ldots$ measured from the origin 0; by convention, $T_0 = 0$. There are two additional equivalent descriptions of point processes: counting processes and interarrival time processes. A counting process $\{N(t)\}_{t=0}^{\infty}$ is a continuous-time, non-negative integer-valued stochastic process, where $N(t) = \max\{n: T_n \leq t\}$ is the number of (traffic) arrivals in the interval $(0,t]$. An interarrival time process is a non-negative random sequence $\{A_n\}_{n=1}^{\infty}$, where $A_n = T_n - T_{n-1}$ is the length of the time interval separating the nth arrival from the previous one. The equivalence of these descriptions follows from the equality of events:

$$\{N(t) = n\} = \{T_n \leq t < T_{n+1}\}$$
$$= \left\{\sum_{k=1}^{n} A_k \leq t < \sum_{k=1}^{n+1} A_k\right\}$$

since $T_n = \sum_{k=1}^{n} A_k$. Unless otherwise stated, we assume throughout that $\{A_n\}$ is a stationary sequence and that the common variance of the A_n is finite.

Compound traffic consists of batch arrivals; that is, arrivals may consist of more than one unit at an arrival instant T_n. To fully describe compound traffic, one also needs to specify a non-negative random sequence $\{B_n\}_{n=1}^{\infty}$, where B_n is the (random) number of units in the batch. At a higher level of abstraction, B_n may represent some general attributes of the nth arrival, such as the amount of "work" associated with the nth arrival or its itinerary in a network. Such compound traffic processes, called marked point processes [14], are outside the scope of this article.

Discrete-time traffic processes correspond to the case when time is slotted. Mathematically, this means that the random variables A_n can assume only integer values, or equivalently, that the random variables $N(t)$ are allowed to increase only at integer-valued time instants T_n.

Traffic processes are used to drive simulations in several ways, all of which use one or more pseudo-random number streams to generate sequences of random variables via appropriate transformations. To emphasize this point, we shall use the term randomly generated to refer to such computer-generated random sequences. In the simplest case, a simulation only needs to randomly generate a sequence of interarrival times $\{A_n\}$.

The traffic-generation mechanism that would be contained in the event algorithms is straightforward. Initially, the simulation clock is set to $T_0 = 0$. Next, A_1 is randomly generated and an arrival event is scheduled for time $T_1 = A_1$. This arrival event is placed on the chronologically-ordered event list or calendar. Eventually, that arrival event will become the most imminent one, the simulation clock will be set to A_1, and that arrival event will be processed. Arrival generation proceeds inductively. At simulation time T_n, the nth arrival event is processed, the next interarrival time A_{n+1} is randomly generated, and an arrival event is scheduled for simulation time $T_{n+1} = T_n + A_{n+1}$ and so on. For compound traffic, the simulation randomly generates a batch size B_n (in addition to the interarrival time A_n), and implements the arrival of B_n units at simulation time T_n. Most models call for the sequences $\{A_n\}$ and $\{B_n\}$ to be stochastically independent.

In addition to arrival times and batch sizes, it is often useful (and sometimes essential) to incorporate the notion of workload into the traffic description. The workload is a general concept describing the amount of work $\{W_n\}$ brought to a system by the nth arriving unit; it is usually assumed independent of interarrival times and batch sizes. A typical example is the sequence of service time requirements of arrivals at a queueing system, although in queueing, one usually refers to the arrival process alone as traffic. On the other hand, traffic reduces to workload description when interarrival times are deterministic. A case in point is compressed video, also known as VBR (variable bit rate) video, where coded frames (arrivals) have variable and random size (bit rate), and these must be delivered deterministically every 1/30 of a second or so, for high-quality video. The workload consists of coded frame sizes (say, in bits), because frame size is roughly proportional to its transmission time (service requirement).

In this article, we describe generic models that can be used to randomly generate any component of traffic description, be it $\{A_n\}$, $\{B_n\}$, or $\{W_n\}$, but we emphasize simple traffic, described by $\{A_n\}$. We also point out that different traffic streams, corresponding to different telecommunications services (voice, video, file transfer, etc.)

can be superposed (multiplexed) to form a realistic heterogeneous mixture of traffic.

Traffic Burstiness

A recurrent theme relating to traffic in broadband networks is the traffic "burstiness" exhibited by key services such as compressed video, file transfer, etc. Burstiness is present in a traffic process if the arrival points $\{T_n\}$ appear to form visual clusters; that is, $\{A_n\}$ tends to give rise to runs of several relatively short interarrival times followed by a relatively long one. The mathematical underpinning of burstiness is more complex. Two main sources of burstiness are due to the shapes of the marginal distribution and autocorrelation function of $\{A_n\}$. For example, burstiness would be facilitated by a bimodal marginal distribution of $\{A_n\}$, or by short-term autocorrelations in $\{A_n\}$. Strong positive autocorrelations are a particularly major cause of burstiness. Since there seems to be no single widely-accepted notion of burstiness, we shall briefly describe some of the commonly-used mathematical measures that attempt to capture it.

The two simplest measures of burstiness take account only of first-order properties of traffic (they are each a function of the marginal distribution only of interarrival times). The first one is the ratio of peak rate to mean rate — a very crude measure, which also has the shortcoming of dependence on the interval length utilized for rate measurement. A more elaborate measure of burstiness is the coefficient of variation, defined as the ratio of standard deviation to mean $c_A = \sigma[A_n]/E[A_n]$ of interarrival intervals.

In contrast, the peakedness measure [15] and the index-of-dispersion measure [16] do take account of temporal dependence in traffic (second-order properties). For a given time interval of length τ, the index of dispersion for counts (IDC) is the function $I_c(\tau) = Var[N(\tau)]/E[N(\tau)]$; i.e., the variance-to-mean ratio of the number of arrivals in the interval $[0,\tau]$. Since the number of arrivals is related to the sum of interarrival intervals via Eq. (1), the numerator of the IDC includes the autocorrelations of $\{A_n\}$. The peakedness concept is related, but more involved. Assume that the traffic stream $\{A_n\}$ is offered to an infinite server group consisting of independent servers with common service time distribution F. Let S be the equilibrium number of busy servers. The peakedness is the functional $z_A[F] = Var[S]/E[S]$, which maps a service time distribution to a real number. A commonly used peakedness is $z_{exp}(0)$, obtained as a limiting case for an exponential service distribution with service rate approaching 0.

Finally, the Hurst parameter [40] can be used as a measure of burstiness via the concept of self-similarity. This notion is discussed in a following section.

Renewal Traffic Models

This section introduces renewal traffic processes and the important special cases of Poisson and Bernoulli processes. Renewal models have a long history, because of their relative mathematical simplicity. In a renewal traffic process, the A_n are independent, identically distributed (IID), but their distribution is allowed to be general. Unfortunately, with few exceptions, the superposition of independent renewal processes does not yield a renewal process. The ones that do, however, occupy a special position in traffic theory and practice. Queueing models historically have routinely assumed a renewal-offered traffic.

Renewal processes, while simple analytically, have a severe modeling drawback — the autocorrelation function of $\{A_n\}$ vanishes identically for all nonzero lags. The importance of capturing autocorrelations stems from the role of the autocorrelation function as a statistical proxy for temporal dependence in time series. Moreover, recall that positive autocorrelations in $\{A_n\}$ can explain, to a large extent, the phenomenon of traffic burstiness. Bursty traffic is expected to dominate broadband networks, and when offered to a queueing system, it gives rise to much worse performance (such as mean waiting times) as compared to renewal traffic (which lacks temporal dependence); see [17] for a detailed discussion. Consequently, models that capture the autocorrelated nature of traffic are essential for predicting the performance of emerging broadband networks.

Poisson Processes — Poisson models are the oldest traffic models, dating back to the advent of telephony and the renowned pioneering telephone engineer A. K. Erlang. A Poisson process can be characterized as a renewal process whose interarrival times $\{A_n\}$ are exponentially distributed with rate parameter λ: $P\{A_n \le t\} = 1 - \exp(-\lambda t)$ [13]. Equivalently, it is a counting process, satisfying $P\{N(t)=n\} = \exp(-\lambda t)(\lambda t)^n/n!$, and the number of arrivals in disjoint intervals is statistically independent (a property known as independent increments).

Poisson processes enjoy some elegant analytical properties. First, the superposition of independent Poisson processes results in a new Poisson process whose rate is the sum of the component rates. Second, the independent increment property renders Poisson a memoryless process. This, in turn, greatly simplifies queueing problems involving Poisson arrivals. Third, Poisson processes are fairly common in traffic applications that physically comprise a large number of independent traffic streams, each of which may be quite general. The theoretical basis for this phenomenon is known as Palm's Theorem [18]. It roughly states that under suitable but mild regularity conditions, such multiplexed streams approach a Poisson process as the number of streams grows, but the individual rates decrease so as to keep the aggregate rate constant. Thus, traffic streams on main communications arteries are commonly believed to follow a Poisson process, as opposed to traffic on upstream tributaries, which are less likely to be Poisson. However, traffic aggregation (multiplexing) need not always result in a Poisson stream. A counter-example is provided in the section on self-similar traffic models that follows.

Time-dependent Poisson processes are defined by letting the rate parameter λ depend on time. Compound Poisson processes are defined in the obvious way, by specifying the distribution of the batch size, B_n, independent of the A_n.

Bernoulli Processes — Bernoulli processes are the discrete-time analog of Poisson processes (time-dependent and compound Bernoulli processes are defined in the natural way). Here the probability

A recurrent theme relating to traffic in broadband networks is the traffic "burstiness" exhibited by key services such as compressed video, file transfer, and so forth.

Markov-modulated models constitute an extremely important class of traffic models.

of an arrival in any time slot is p, independent of any other one. It follows that for slot k, the corresponding number of arrivals is binomial, $P\{N_k = n\} = \binom{k}{n} p^n (1-p)^{k-n}$, n between 0 and k. The time between arrivals is geometric with parameter p: $P\{A_n = j\} = p(1-p)^j$, j being a non-negative integer.

Phase-type Renewal Processes — An important special case of renewal models occurs when the inter-arrival times are of the so-called phase type. Phase-type interarrival times can be modeled as the time to absorption in a continuous-time Markov process $C = \{C(t)\}_{t=0}^{\infty}$ with state space $\{0,1,\ldots,m\}$; here, state 0 is absorbing, all other states are transient, and absorption is guaranteed in a finite time. To determine A_n, start the process C with some initial distribution π. When absorption occurs (i.e., when the process enters state 0), stop the process. The elapsed time is A_n, which implies that it is a probabilistic mixture of sums of exponentials. Then, restart with the same initial distribution π and repeat the procedure independently to get A_{n+1}.

Phase-type renewal processes give rise to relatively tractable traffic models. They also enjoy the property that any inter-arrival distribution can be approximated arbitrarily closely by phase-type distributions.

Markov and Markov-Renewal Traffic Models

Unlike renewal traffic models, Markov and Markov-renewal traffic models introduce dependence into the random sequence $\{A_n\}$ [13]. Consequently, they can potentially capture traffic burstiness, because of nonzero autocorrelations in $\{A_n\}$.

Consider a continuous-time Markov process $M = \{M(t)\}_{t=0}^{\infty}$ with a discrete state space. In this case, M behaves as follows: it stays in a state i for an exponentially distributed holding time with parameter λ_i, which depends on i alone; it then jumps to state j with probability p_{ij}, such that the matrix $P = [p_{ij}]$ is a probability matrix [13]. In a simple Markov traffic model, each jump of the Markov process is interpreted as signaling an arrival, so interarrival times are exponential, and their rate parameters depend on the state from with the jump occurred. This results in dependence among interarrival times as a consequence of the Markov property.

Markov models in slotted time can be defined for the process $\{A_n\}$ in terms of a Markov transition matrix $P = [p_{ij}]$ [13]. Here, state i corresponds to i idle slots separating successive arrivals, and p_{ij} is the probability of a j-slot separation, given that the previous one was an i-slot separation. Arrivals may be single units, a batch of units, or a continuous quantity. Batches may themselves be described by a Markov chain, whereas continuous-state, discrete-time Markov processes can model the (random) workload arriving synchronously at the system. In all cases, the Markov property introduces dependence into interarrival separation, batch sizes and successive workloads, respectively.

Markov-renewal models are more general than discrete-state Markov processes, yet retain a measure of simplicity and analytical tractability. A Markov renewal process $R = \{(M_n, \tau_n)\}_{n=0}^{\infty}$ is defined by a Markov chain $\{M_n\}$ and its associated jump times $\{\tau_n\}$, subject to the following constraint: the pair (M_{n+1}, τ_{n+1}) of next state and inter-jump time depends only on the current state M_n, but not on previous states nor on previous inter-jump times. Again, if we interpret jumps (transitions) of $\{M_n\}$ as signaling arrivals, we would have dependence in the arrival process. Also, unlike the Markov process case, the interarrival times can be arbitrarily distributed, and these distributions depend on both states straddling each interarrival interval [13].

The Markovian Arrival Process (MAP) is a broad and versatile subclass of Markov renewal traffic processes, enjoying analytical tractability [19]. Here, the interarrival times are phase-type but with a wrinkle: traffic arrivals still occur at absorption instants of the auxiliary Markov process M, but the latter is not restarted with the same initial distribution; rather, the restart state depends on the previous transient state from which absorption had just occurred. While MAP is analytically simple, it enjoys considerable versatility. Its formulation includes Poisson processes, phase-type renewal processes, and others as special cases [19]. It also has the appealing property that the superposition of independent MAP traffic streams results in a MAP traffic stream governed by a Markov process whose state space is the cross product of the component state spaces.

Markov-Modulated Traffic Models

Markov-modulated models constitute an extremely important class of traffic models. The idea is to introduce an explicit notion of state into the description of a traffic stream — an auxiliary Markov process is evolving in time and its current state controls (modulates) the probability law of the traffic mechanism.

Let $M = \{M(t)\}_{t=0}^{\infty}$ be a continuous-time Markov process, with state space $\{1,2,\ldots m\}$ (more complicated state spaces are possible). Now assume that while M is in state k, the probability law of traffic arrivals is completely determined by k, and this holds for every $1 \le k \le m$. Note that when M undergoes a transition to, say, state j, then a new probability law for arrivals takes effect for the duration of state j, and so on. Thus, the probability law for arrivals is modulated by the state of M (such systems are also called doubly stochastic, but the term "Markov modulation" makes it clearer that the traffic is stochastically subordinated to M).

The modulating process certainly can be more complicated than a continuous-time, discrete-state Markov process (so the holding times need not be restricted to exponential random variables), but such models are far less analytically tractable. For example, Markov Renewal-modulated processes constitute a natural generalization of Markov-modulated processes with generally-distributed interarrival times, but those will not be reviewed here.

Markov-Modulated Poisson Processes — The most commonly used Markov-modulated model is the Markov-Modulated Poisson Process (MMPP) model, which combines the simplicity of the modulating (Markov) process with that of the modulated (Poisson) process. In this case, the modulation mechanism simply stipulates that in state k of M, arrivals occur according to a Poisson process at rate λ_k. As the state changes, so does the rate.

MMPP models can be used in a number of ways.

Consider first a single traffic source with a variable rate. A simple traffic model would quantize the rate into a finite number of rates, and each rate would give rise to a state in some Markov modulating process. It remains to verify that exponential holding times of rates are an appropriate description, but the Markov transition matrix $Q=[Q_{kj}]$ of the putative M can be easily estimated from empirical data: simply quantize the empirical data, and then estimate Q_{kj} by calculating the fraction of times that M switched from state k to state j.

As a simple example, consider a two-state MMPP model, where one state is an "on" state with an associated positive Poisson rate, and the other is an "off" state with associated rate zero (such models are also known as interrupted Poisson for obvious reasons). These models have been widely used to model voice traffic sources [20]; the "on" state corresponds to a talk spurt (when the speaker emits sound), and the "off" state corresponds to a silence (when the speaker pauses for a break). This basic MMPP model can be extended to aggregations of independent traffic sources, each of which is an MMPP, modulated by an individual Markov process M_i, as described previously. Let $J(t)=(J_1(t),J_2(t),\ldots,J_r(t))$, where $J_i(t)$ is the number of active sources of traffic type i, and let $M(t)=(M_1(t),M_2(t),\ldots M_r(t))$ be the corresponding vector-valued Markov process taking values on all r-dimensional vectors with non-negative integer components. The arrival rate of class i traffic in state $(j_1,j_2,\ldots j_r)$ of $M(t)$ is $j_i\lambda_i$.

Transition-Modulated Processes — Transition-modulated processes are a variation on the state modulation idea. Essentially, the modulating agent is a state transition rather than a state per se. A state transition, however, can be described simply by a pair of states, whose components are the one before transition and the one after it.

The generalization of a transition-modulated traffic model to continuous time is straightforward (the model in discrete time is described in [21]). Let $M=\{M(t)\}_{t=0}^{\infty}$ be a discrete-time Markov process on the positive integers. State transitions occur on slot boundaries, and are governed by an $m \times m$ Markov transition matrix $Q=[Q_{ij}]$. Let B_n denote the number of arrivals in slot n, and assume that the probabilities $P\{B_n=k \mid M_n=i,M_{n+1}=j\}=t_{ij}(k)$, are independent of any past state information (the parameters $t_{ij}(k)$ are assumed given). Notice that these probabilities are conditioned on transitions, (M_n,M_{n+1}), of M from state M_n to state M_{n+1} during slot n. Furthermore, the number of traffic arrivals during slot n is completely determined by the transition of the modulating chain (through the parameters $t_{ij}(k)$).

Markov-modulated traffic models are a special case of Markovian transition-modulated ones: simply take the special case when the conditioning event is $\{M_n=i\}$. That is, $t_{ij}(k)=t_i(k)$ depends only on the state i of the modulating chain in slot n, but is independent of its state j in the next slot $n+1$. Conversely, Markovian transition-modulated processes can be thought of as Markov-modulated ones, but on a larger state space. Indeed, if $\{M_n\}$ is Markov, so is the process $\{(M_n,M_{n+1})\}$ of its transitions.

As before, multiple transition-modulated traffic models can be defined, one for each traffic class of interest. The complete traffic model is obtained as the superposition of the individual traffic models. For queueing studies in discrete time, another wrinkle is the assignment of priorities to different classes, so as to order their arrivals in a buffer [21].

Fluid Traffic Models

The fluid traffic paradigm dispenses with individual traffic units. Instead, it views traffic as a stream of fluid, characterized by a flow rate (such as bits per second), so that a traffic count is replaced by a traffic volume.

Fluid models are appropriate to cases where individual units are numerous relative to a chosen time scale. In other words, an individual unit is by itself of little significance, just as one molecule more or less in a water pipeline has but an infinitesimal effect on the flow. In the B-ISDN context of ATM, all packets are fixed-size cells of relatively short length (53 bytes); in addition, the high transmission speeds (say, on the order of a gigabit per second) render the transmission impact of an individual cell negligible. The analogy of a cell to a fluid molecule is a plausible one. To further highlight this analogy, contrast an ATM cell with a much bigger transmission unit, such as a coded (compressed) high-quality video frame, which may consist of a thousand cells. A traffic arrival stream of coded frames should be modeled as a discrete stream of arrivals, because such frames are typically transmitted at the rate of 30 frames per second. A fluid model, however, is appropriate for the constituent cells.

Although an important advantage of fluid models is their conceptual simplicity, important benefits will also accrue to a simulation model of fluid traffic. As an example, consider again a broadband ATM scenario. If one is to distinguish among cells, then each of them would have to count as an event. The time granularity of event processing would be quite fine, and consequently, processing cell arrivals would consume vast CPU and possibly memory resources, even on simulated time scales of minutes. A statistically meaningful simulation may often be infeasible. In contrast, a fluid simulation would assume that the incoming fluid flow remains (roughly) constant over much longer time periods. Traffic fluctuations are modeled by events signaling a change of flow rate. Because these changes can be assumed to happen far less frequently than individual cell arrivals, one can realize enormous savings in computing. In fact, infeasible simulations of cell arrival models can be replaced by feasible simulations of fluid models of comparable accuracy. In a queueing context, it is easy to manipulate fluid buffers. Furthermore, the waiting time concept simply becomes the time it takes to serve (clear) the current buffer, and loss probabilities (at a finite buffer) can be calculated in terms of overflow volumes. Because fluid models assume a deterministic service rate, these statistics can be readily computed. Typically, though, larger traffic units (such as coded frames) are of greater interest than individual cells. Modeling the larger units as discrete traffic and their transport as fluid flow would give us the best of both worlds: we can measure wait-

The fluid traffic paradigm dispenses with individual traffic units. Instead, it views traffic as a stream of fluid, characterized by a flow rate, so that a traffic count is replaced by a traffic volume.

TES models provide another modeling approach geared toward capturing both marginals and autocorrelations of empirical records simultaneously, traffic included.

ing times and loss probabilities and enjoy savings on simulation computing resources.

Typical fluid models assume that sources are bursty — of the "on-off" type [22, 23]. While in the "off" state, traffic is switched off, whereas in the "on" state traffic arrives deterministically at a constant rate λ. For analytical tractability, the duration of "on" and "off" periods are assumed to be exponentially distributed and mutually independent (that is, they form an alternating renewal process). A Markov model of a set of quantized (fluid) traffic rates is presented in [24]. Fluid traffic models of these types can be analyzed as Markov-modulated constant rate traffic. The host of generalizations, described above for MMPP, carries over to fluid models as well, including multiple sources and multiple classes of sources.

Autoregressive Traffic Models

Autoregressive models define the next random variable in the sequence as an explicit function of previous ones within a time window stretching from the present into the past. Such models are particularly suitable for modeling VBR-coded video-a projected major consumer of bandwidth in emerging high-speed communications networks. The nature of video frames is such that successive frames within a video scene vary visually very little (recall that there are 30 frames per second in a high-quality video). Only scene changes (and other visual discontinuities) can cause abrupt changes in frame bit rate. Thus, the sequence of bit rates (frame sizes) comprising a video scene may be modeled by an autoregressive scheme (later, we describe another modeling approach), while scene changes can be modeled by some modulating mechanism, such as a Markov chain.

Linear Autoregressive Models — The class of linear autoregressive models [25] has this form:

$$X_n = a_0 + \sum_{r=1}^{p} a_r X_{n-r} + \varepsilon_r , \quad n > 0 , \quad (2)$$

where $X_0, \ldots, X_{p-1}$ are prescribed random variables, the a_r are real constants, and the ε_n are zero-mean, IID random variables, called residuals, which are independent of the X_n.

Eq. 2 describes the simplest form of a linear autoregression scheme, called $AR(p)$, where p is the order of the autoregression. In a good model, the residuals ought to be of a smaller magnitude than the X_n, in order to "explain" the empirical data.

The recursive form in Eq. 2 makes it clear how to randomly generate the next random element in the sequence $\{X_n\}_{n=0}^{\infty}$ from a previous one: this simplicity makes AR schemes popular candidates for modeling autocorrelated traffic. A simple AR(2) model has been used to model variable bit rate (VBR) coded video [26]. More elaborate models can be constructed out of $AR(p)$ models combined with other schemes. For example, video bit rate traffic was modeled as a sum, $R_n = X_n + Y_n + K_n C_n$, where the first two terms comprise independent $AR(1)$ schemes and the third term is a product of a simple Markov chain and an independent normal variate from an IID normal sequence [27]. The purpose of having two autoregressive schemes was to achieve a better fit to the empirical autocorrelation function; the third term was designed to capture sample path spikes due to video scene changes.

More complicated models, such as MA, ARMA, and ARIMA are outside the scope of this article [25]. Autoregressive models are typically used to fit the empirical autocorrelation function, but they cannot generally fit the empirical marginal distribution.

TES Models — Transform-expand-sample (TES) models provide another modeling approach geared toward capturing both marginals and autocorrelations of empirical records simultaneously [28, 29], traffic included [30]. The empirical TES methodology assumes that some stationary empirical time series (such as traffic measurements over time) is available. It aims to construct a model satisfying the following three fidelity requirements, simultaneously:

- The model's marginal distribution should match its empirical counterpart (a histogram, in practice).
- The model's leading autocorrelations should approximate their empirical counterparts up to a reasonable lag.
- The sample path realizations (histories) generated by simulating the model should "resemble" the empirical records.

The first two are precise quantitative requirements, whereas the third is a heuristic qualitative one. Nevertheless, it is worth adopting this subjective requirement and keeping its interpretation at the intuitive level; after all, common sense tells us that if a model gives rise to time series which are entirely divorced in "appearance" from observed ones, then this would weaken our confidence in the model, and vice versa.

TES processes come in two flavors: TES^+ and TES^-. The superscript (plus or minus) is a mnemonic reminder of the fact that they give rise to *TES* processes with positive and negative lag-1 autocorrelations, respectively. *TES* models consist of two stochastic processes in lockstep, called background and foreground sequences, respectively. Background *TES* sequences have this form:

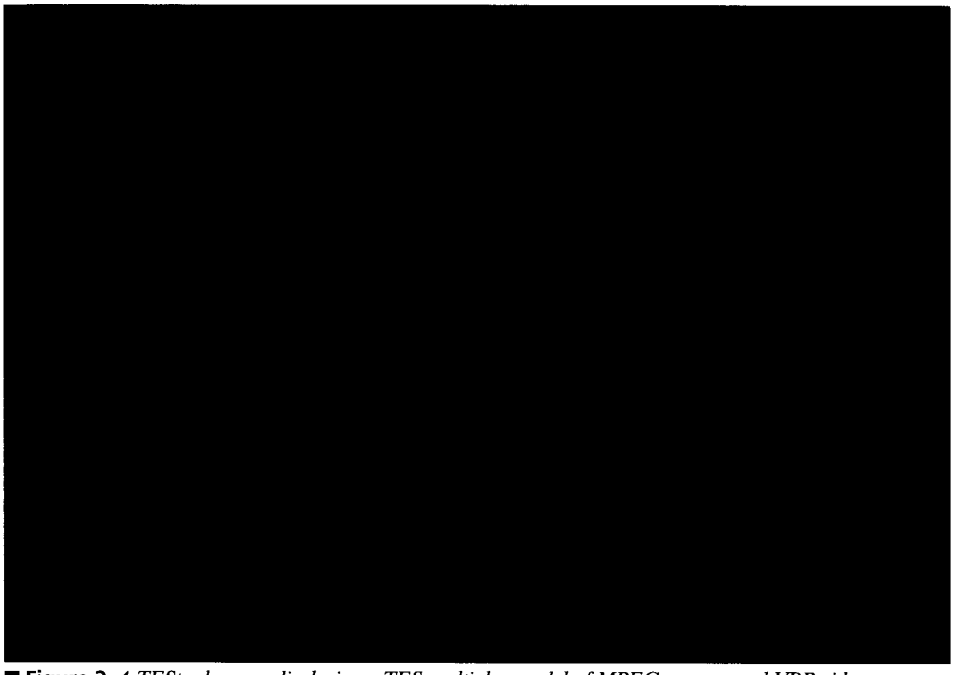
$$U_n^+ = \begin{cases} U_0 , & n=0 \\ \langle U_{n-1}^+ + V_n \rangle , & n>0 \end{cases} \quad U_n^- = \begin{cases} U_n^+ , & n \text{ even} \\ 1-U_n^+ , & n \text{ odd} \end{cases}$$

Here, U_0 is distributed uniformly on [0,1); $\{V_n\}_{n=1}^{\infty}$ is a sequence of IID random variables, independent of U_0, called the innovation sequence, and angular brackets denote the modulo-1 (fractional part) operator $\langle x \rangle = x - \max\{\text{integer } n: n \le x\}$. Background sequences play an auxiliary role. The real target are foreground sequences:

$$X_n^+ = D(U_n^+), \quad X_n^- = D(U_n^-) ,$$

where D is a transformation from [0,1) to the reals, called a distortion.

It can be shown that all background sequences are Markovian stationary, and their marginal distribution is uniform on [0,1), regardless of the probability law of the innovations $\{V_n\}$ [28]. However, the transition structure $\{U_n^+\}$ is time invariant, while that of $\{U_n^-\}$ is time dependent. The inversion method allows us to transform any background uniform variates to foreground ones with an arbitrary marginal distribution [31]. To illustrate this idea, consider an empirical time series $Y = \{Y_n\}_{n=0}^{N}$, from which one computes an empirical density $\hat{h}_Y$ and its associated distribution



■ **Figure 2.** *A TEStool screen displaying a TES multiplex model of MPEG-compressed VBR video.*

Stationary TES models can be combined to yield non-stationary composite ones. MPEG-coded video is a case in point.

function $\hat{H}_Y$. Then, the random variable $X=\hat{H}_Y^{-1}(U)$ has density $\hat{h}_Y$. Thus, TES foreground sequences can match any empirical distribution.

The empirical TES methodology actually employs a composite two-stage distortion:

$$D_{Y,\xi}(x) = \hat{H}_Y^{-1}(S_\xi(x)), \quad x \in [0,1] ,$$

where $\hat{H}_Y^{-1}$ is the inverse of the empirical histogram distribution based on Y, and S_ξ is a "smoothing" operation, called a stitching transformation, parameterized by $0 \le \xi \le 1$, and given by:

$$S_\xi(y) = \begin{cases} y/\xi, & 0 \le y < \xi \\ (1-y)/(1-\xi), & \xi \le y < 1 \end{cases}$$

For $0 < \xi < 1$, the effect of S_ξ is to render the sample paths of background TES sequences more "continuous-looking." Because stitching transformations preserve uniformity, the inversion method via $\hat{H}_Y^{-1}$ guarantees that the corresponding foreground sequence would have the prescribed marginal distribution $\hat{H}_Y$. The empirical TES modeling methodology takes advantage of this fact which effectively decouples the fitting requirements of the empirical distribution and the empirical autocorrelation function. Because the former is automatically guaranteed by TES, one can concentrate on fitting the latter. This is carried out by a heuristic search for a pair (ξ, f_V), where ξ is a stitching parameter and f_V is an innovation density; the search is declared a success on finding that the corresponding TES sequence gives rise to an autocorrelation function that adequately approximates its empirical counterpart, and whose simulated sample paths bear "adequate resemblance" to their empirical counterparts.

In practice, efficient searches of this kind must rely on software support. TEStool is a visual interactive software environment designed to support TES modeling [32]. TEStool allows the user to read in empirical time series and calculate their empirical statistics (histogram, autocorrelation function, and spectral density) in textual and graphical forms. It further provides services to generate and modify TES models and to superimpose the corresponding TES statistics on their empirical counterparts. The search proceeds in an interactive style, guided by visual feedback: each model modification triggers a recalculation and redisplay of the results. TES-model autocorrelations and spectral densities are calculated numerically from fast and accurate formulas developed in [28, 29]. This activity is further simplified by restricting the innovation densities f_V to be step functions. Such simple densities can be readily modified graphically, because steps are visually represented by rectangles that can be created, deleted, stretched, and moved easily with a mouse. A TES-modeling algorithm has been recently developed (based on nonlinear optimization) that has been shown to perform better and faster as compared to human heuristic search.

Stationary TES models can be combined to yield non-stationary composite ones. MPEG-coded video is a case in point [33]. It consists of three kinds of frames (called I-frames, P-frames, and B-Frames), interleaved in a deterministically repeating sequence (the basic cycle starts with an I-frame and ends just short of the next I-frame). Consequently, MPEG-coded VBR video is nonstationary, even if the corresponding I, P, and B subsequences of frames are stationary. A composite TES model can be obtained by modeling the I, B, and P subsequences separately, and then multiplexing the three streams in the correct order [50]. The resulting multiplexed TES model obtained from the corresponding TES models of the subsequences, but with autocorrelation injected into frames within the same cycle, are shown in Fig. 2, which includes a TEStool screen sub-divided into four canvas areas displaying var-

In the case of packet traffic, self-similarity is manifested in the absence of a natural length of a burst.

ious types of empirical statistics (bullets) against their TES model counterparts (diamonds). The upper-left canvas contains the empirical and TES model sample paths, the latter being generated by simulation; the upper-right canvas contains the corresponding histograms; the lower-left and lower-right canvases contain, respectively, the empirical autocorrelation function and spectral density plotted against their simulation-based TES model counterparts. Although the autocorrelation functions and spectral densities were formally computed from a single sample path as if the MPEG sequences were stationary, and therefore represent averaged estimates of different correlation coefficients, they nevertheless give an indication of how well the composite TES model captured temporal dependence in the empirical data, because they were all computed from sample paths in the same way. The general good agreement between the TES model statistics and their empirical counterparts is in accord with the three fidelity requirements stipulated at the beginning of this section. These TES source models can be used to generate synthetic streams of realistic traffic to drive simulations of communications networks.

Self-Similar Traffic Models

Recent studies of high-quality, high-resolution traffic measurements have revealed a new phenomenon with potentially important ramifications to the modeling, design, and control of broadband networks. These include an analysis of hundreds of millions of observed packets over an Ethernet LAN in a R & D environment [34], and an analysis of a few millions of observed frame data generated by VBR video services [35]. In these studies, packet traffic appears to be statistically self-similar [36]. A self-similar (or fractal) phenomenon exhibits structural similarities across all (or at least a wide range) of time scales. In the case of packet traffic, self-similarity is manifested in the absence of a natural length of a burst: at every time scale ranging from a few milliseconds to minutes and hours, similar-looking traffic bursts are evident. Self-similar stochastic models include fractional Gaussian noise [37] and fractional ARIMA processes [38].

Self-similarity manifests itself in a variety of different ways: a spectral density that diverges at the origin ($1/f^{\alpha}$–noise, $0<\alpha<1$), a non-summable autocorrelation function (indicating long-range dependence), and a variance of the sample mean that decreases (as a function of the sample size n) more slowly than $1/n$ [39]. The key parameter characterizing these phenomena is the so-called Hurst parameter, H, which is designed to capture the degree of self-similarity in a given empirical record [40] as follows. Let $\{Y_k\}_{k=1}^{n}$ be an empirical time series with sample mean $\overline{Y}(n)$ and sample variance $S^2(n)$. The rescaled adjusted range, or R/S statistic, is given by $R(n)/S(n)$ with:

$$R(N) = \max\left\{\sum_{i=1}^{k}\left(Y_i - \overline{Y}(n)\right): 1 \le k \le n\right\} - \min\left\{\sum_{i=1}^{k}\left(Y_i - \overline{Y}(n)\right): 1 \le k \le n\right\} .$$

It has been found empirically that many naturally occurring time series appear to obey the relation:

$E[R(n)/S(n)] \cong n^H$, n large ,

with a H typically about 0.73. On the other hand, for renewal and Markovian sequences, it can be shown that the previous equation holds with $H=0.5$, for large n [37]. This discrepancy, generally referred to as the Hurst phenomenon, is a measure of the degree of self-similarity in time series, and can be estimated from empirical data [35].

From a mathematical point of view, self-similar traffic differs from other traffic models in the following way [39]. Let s be a time unit representing a time scale, such as $s = 10^m$ seconds ($m=0, \pm 1, \pm 2, \ldots$). For every time scale s, let $X^{(s)}=\{X_n^{(s)}\}$ denote the time series computed as the number of units (packets, bytes, cells, etc.) per time unit s in the traffic stream. Traditional traffic models have the property that, as s increases, the "aggregated" processes, $X^{(s)}$, tend to a sequence of IID random variables (covariance stationary white noise). On the other hand, the corresponding aggregation procedure of empirical traffic data yields time series $X^{(s)}$, which reveal two related types of behavior, when plotted against time. They either appear visually indistinguishable from one another ("exactly self-similar") but distinctively different from pure noise, or they converge to a time series with a nondegenerate autocorrelation structure ("asymptotically self-similar"). In contrast, simulations of traditional traffic models, rapidly converge to white noise after increasing the time scale by about 2 or 3 orders of magnitude. Similarly, when trying to fit traditional traffic models to self-similar traffic data, the number of parameters required typically grows, as the sample size increases. In contrast, self-similar traffic models are able to capture the observed fractal nature of packet traffic in a parsimonious manner (with about 1 to 4 parameters). Parameter estimation techniques are available for many self-similar models, as well as Monte Carlo methods for generating long traces of synthetic self-similar traffic [35].

Potential implications of self-similar traffic on issues related to design, control, and performance of high-speed, cell-based networks are currently under study. For example, it can be shown that many of the commonly used measures for burstiness do not characterize self-similar traffic [34]. Contrary to commonly held beliefs that multiplexing traffic streams tends to produce smoothed out aggregate traffic with reduced burstiness, aggregating self-similar traffic streams can actually intensify burstiness rather than diminish it [34]. From a practical vantage point, there are also indications that traffic congestion in self-similar networks may have broadly differing characteristics from those produced by traditional traffic models. Comparisons of queueing performance under self-similar versus traditional traffic scenarios are currently being studied.

Rare-Event Simulation

With the improved reliability of telecommunications networks, rare events have become an increasingly important ingredient of quality-of-service (QoS) metrics. The most common example is cell loss in ATM systems. The desired cell loss probability in such networks is in the range of 10^{-6} to 10^{-12}, depending on the application. This implies that 10^7 to 10^{13} statistically independent cells must be simulated to obtain estimates with reasonable

statistical confidence. Much longer simulations may be needed if the cell losses are autocorrelated as is the usual case with bursty traffic. Consequently, it is computationally costly to use conventional simulation techniques. This computational limitation is evident in most simulation studies of ATM systems where the lowest reported loss probability is usually around 10^{-6} to 10^{-5}. Two techniques have been proposed to deal with rare events in a telecommunications networking context: importance sampling (IS)[41] and the generalized extreme value theory (GEVT)[42], both of which have been successfully applied to elements of B-ISDN systems. Obtaining simulation-based performance predictions for large complex networks is still problematic, however; distributed processing and time-warping are other approaches that should be considered for large systems. (See [11] in this issue for a discussion of these techniques.)

The IS technique [43, 44] is a well-known statistical adjunct to simulation with a record of successful use in communications link and radar system design [45, 46]. Compared with conventional Monte Carlo simulation, IS may reduce the computational time by orders of magnitude.

The basic principle of IS is straightforward. Let L(x) be a function from real vectors to real numbers, and X a real random vector, so that L(X) is a random variable. Let $p^{(o)}(x)$ and $p(x)$ be two distinct joint probability density functions (pdf), where $p(x)$ is obtained by modifying or biasing $p^{(o)}(x)$. The essence of IS is the ability under suitable conditions to force the random variable L(X) to have the same mean with respect to $p^{(o)}(x)$ as L(X)•W(X) has with respect to $p(x)$, where $W(x)$ is a weighting function. The equivalent representations have this form:

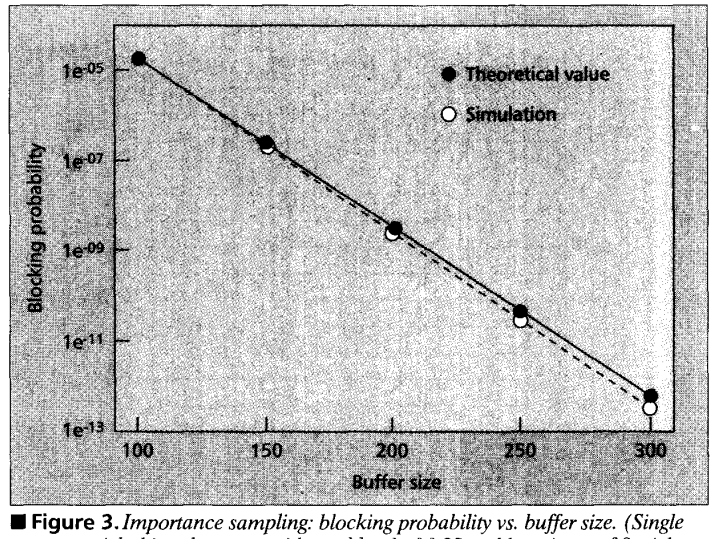
$$E[L(X)] = \int L(x)p^{(o)}(x)dx = \int L(x)W(x)p(x)dx ,$$

where fdx is shorthand for multiple integration.

The pdf $p^{(o)}(x)$ is referred to as the reference density, $p(x)$ as the biased density, and the change from $p^{(o)}(x)$ to $p(x)$ is commonly called a changed measure. In order to maintain equality in the above equation, the weighting function $W(x)$ must satisfy:

$$W(x) = p^{(o)}(x) / p(x) ,$$

provided $p^{(o)}(x)$ is zero whenever $p(x)$ is zero. Note that $p^{(o)}(x)$ is generally determined by a specific application but $p(x)$ can be any pdf that satisfies the above condition. The point of performing a change of measure is that even though L(X) under $p^{(o)}(x)$, and L(X)•W(X) under $p(x)$, have the same mean, their variances may be different. A significant variance reduction is possible if $p(x)$ is chosen properly; on the other hand, the variance can be inadvertently increased if an inappropriate biased pdf is selected. The fundamental idea behind the IS technique is to modify the probabilities that govern the outcomes of the simulation in such a way that the original low-probability events, governed by the reference pdf, occur more frequently under the biased pdf. That is, instead of simulating low-probability events under the reference pdf (requiring long simulation runs), we simulate relatively high-probability events under an appropriate biased pdf (requiring shorter simulation runs), and the results are weighted to compensate for the bias.



■ **Figure 3.** *Importance sampling: blocking probability vs. buffer size. (Single server; eight biased sources with total load of 0.35 and burstiness of 8; eight unbiased sources with total load of 0.15 and burstiness of 4; run length: 5000 regenerative cycles; exponential bias with bias choice based on normalized sample variance.)*

For networking applications, IS is usually applied in a regenerative setting, which assumes that target random variables defined over successive regenerative cycles are statistically independent and identically distributed [47]. IS was used in the simulation of ATM systems [41] where the IS bias was applied to the reference-conditional cell arrival probabilities, $P_{ij}^{(o)}$ defined as:

$P_{ij}^{(o)}$= P(j cells arrive in the current slot | i arrived in the previous slot).

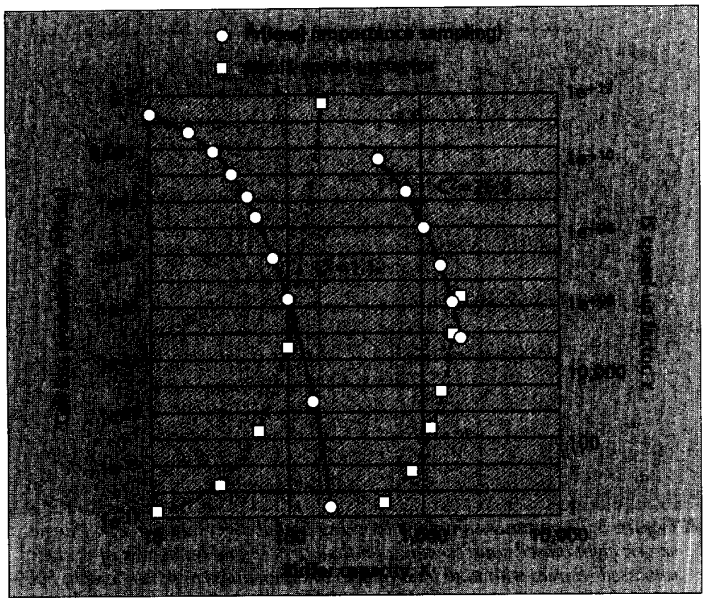
A geometric bias of this form:

$$P_{ij}^{(o)} = \frac{\theta^j P_{ij}^{(o)}}{K(i)} \qquad for\ i, j = 0,\ldots,N$$

where θ is the bias parameter and $K(i)$is a normalization factor, was proposed in [41]. A geometric bias was selected for several reasons: it is a one-parameter function with a particularly simple form, it has a fast rate of change, and for some simple systems, the optimal IS bias has an exponential form determined by large-deviation theory [48].

To keep the lengths of the regenerative cycles finite during the execution of this IS simulation procedure, the geometric bias was applied from the beginning of each regenerative cycle up until the occurrence of the first blocking event, at which point the transition probabilities were reset to their reference values. This procedure yielded computational improvement by a factor of 10^6 (Fig. 3) when estimating a cell loss probability of 10^{-13} for an ATM queueing model with heterogeneous traffic (a mixture of two groups of MMPP traffic sources).

Selection of the bias parameters presents the main problem with the IS methodology. Statistical optimization techniques of the bias parameters in IS are viable alternatives to analytical optimization methods [49]. These techniques use statistical estimates, either of simulation variance, or of



■ **Figure 4.** *Importance sampling with optimized bias parameters.*

gradients of the variance in order to find a near-optimal set of bias parameter values. Some cases of statistical optimization of the IS parameters have produced speedup factors ranging from 2 to 11 orders of magnitude for a number of communications systems at both the physical and network levels, including complex systems such as 16 x 16 Clos ATM switches [49]. A M-IBP+MMBBP/1/D/K queueing system (M-IBP is modified interrupted Bernoulli process and MMBBP is Markov modulated Bernoulli process with batches) was simulated with statistically optimized IS parameters in [49]; such models can be used to represent the first of several nodes in an ATM network, where the M-IBP traffic represents the stream under observation (e.g., a specific virtual circuit), and the MMBBP process represents the cross traffic (aggregation of all the other virtual circuits through the same node). A plot of blocking probabilities and speedup factors for this queueing model as a function of the buffer size, K, for two different values of the squared coefficient of variation, 1.1 and 26.9, representing different levels of traffic burstiness, is shown in Fig. 4 [49].

Conclusion

The advent of broadband networks is ushering in a Communications Age characterized by wide availability of new and varied services such as full-motion video-on-demand. The ensuing variety and ubiquity are spawning, in turn, ever more complex networks. Concomitant with these developments, telecommunications professionals are being called upon to design and manage an assortment of communications systems in the face of fast-moving technology and a climate of increasing customer expectations. Against this backdrop, analysts are frequently faced with incomplete knowledge of user demands, as well as uncertainty about the future evolution of these systems. If B-ISDN is to keep the promise of ubiquity, convenience, affordability, and reliability, accurate system models must be devised that are capable of yielding acceptably precise performance predictions in a reasonable amount of time.

Monte Carlo Simulation provides a powerful generic modeling methodology for predicting the performance of both extant systems under new scenarios, as well as of new systems currently being designed. Simulation models can be used (or abused) in designing and managing communication networks. Simulation per se, however, is not a panacea: its efficacy depends, of course, on the degree of fidelity incorporated into the simulated models. In particular, effective performance evaluation requires that special care be taken in modeling telecommunications traffic and the corresponding demands for network resources. The diverse traffic mixes planned for B-ISDN and the burstiness of some new services (such as compressed video) present formidable modeling challenges.

Needless to say, poor predictions invariably lead to inappropriate design and management decisions, which in addition to impacting users in tangible adverse ways, can also bring about a sense of disappointment and a perception that the new technology is being "oversold" to the public. As a glaring example we cite failure to account for autocorrelations in observed traffic; this typically results in over-optimistic performance predictions, which can easily deviate from observed performance by orders of magnitude.

We hope this review of generic modeling techniques, as well as specific models germane to emerging B-ISDN telecommunications networks, will serve to inform performance analysts of recent developments in traffic modeling and alert them to the potential pitfalls in modeling B-ISDN.

Acknowledgments

We are grateful to Walter Willinger for providing the material on self-similar traffic models and for a careful reading of the manuscript. Thanks are due to D. L. Jagerman and B. Sengupta for valuable discussions of traffic models and to K. Townsend for providing Fig. 4.

References

[1] A. M. Law and M. G. McComas, "Simulation Software for Communications Networks: The State of the Art," in this issue.
[2] A. M. Law and W. D. Kelton, Simulation Modeling and Analysis, Second Edition, (New York, NY: McGraw-Hill Inc., 1991).
[3] C. H. Sauer and E. A. MacNair, Simulation of Computer Communication Systems, (Englewood Cliffs, NJ: Prentice-Hall, 1983).
[4] M. Pidd, Computer Simulation in Management Science, (New York, NY: John Wiley and Sons, 1984).
[5] W. G. Bulgren, Discrete System Simulation, (Englewood Cliffs, NJ: Prentice-Hall, 1982).
[6] A. Pritsker, Introduction to Simulation and SLAM II, (New York, NY: John Wiley and Sons, 1984).
[7] R. E. Shannon, Systems Simulation, (Englewood Cliffs, NJ: Prentice-Hall, 1975).
[8] C. H. Sauer and E. A. MacNair, Simulation of Computer Communications Systems, (Englewood Cliffs, NJ: Prentice-Hall, 1983).
[9] A. M. Law and C. S. Larmey, "An Introduction to Simulation Using SIMSCRIPT II.5," C.A.C.I, Los Angeles, CA, 1984.
[10] R. Jain, The Art of Computer Systems Performance Analysis, (New York, NY: John Wiley and Sons, 1991).
[11] B. W. Unger, Douglas J. Goetz, and Stephen W. Maryka, "Simulation of SS7 Common Channel Signaling," in this issue.
[12] G. S. Fishman, Principles of Discrete Event Simulation, (New York, NY: John Wiley and Sons, 1978).
[13] E. Cinlar, Introduction to Stochastic Processes, (Englewood Cliffs, NJ: Prentice-Hall, 1975).
[14] P. Franken, D. Koenig, U. Arndt, and V. Schmidt, Queues and Point Processes, (Berlin: Akademie-Verlag, 1981).
[15] A. E. Eckberg, "Generalized Peakedness of Teletraffic Processes," Tenth International Teletraffic Conference, Montreal, Canada, 1983.
[16] K. Fendick and W. Whitt, "Measurements and Approximations to

Describe the Offered Traffic and Predict the Average Workload in a Single-Server Queue," Proceedings of the IEEE, vol. 77, 171-194, 1989.
[17] M. Livny, B. Melamed, and A. K. Tsiolis, "The Impact of Autocorrelation on Queuing Systems," *Management Science*, vol. 39, no. 3, 1993, pp. 322-339.
[18] H. J. Larson and B. O. Shubert, Probabilistic Models in Engineering *Sciences*, (New York, NY: John Wiley and Sons, 1979).
[19] D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts, "A Single-Server Queue with Server Vacations and a Class of Non-Renewal Arrival Processes," *Adv. Appl. Prob.*, vol. 22, 1990, pp. 676-705.
[20] H. Heffes and D.M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data and Related Statistical Multiplexer Performance," *IEEE J. on Selected Areas in Commun.*, SAC-4, 1986, pp. 856-886.
[21] T. Takine, B. Sengupta, and T. Hasegawa, "An Analysis of a Discrete-Time Queue for Broadband ISDN with Priorities Among Traffic Classes," *IEEE Trans. on Commun.*, to be published.
[22] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic Theory of a Data-Handling System with Multiple Sources," *The Bell System Technical Journal*, vol. 61, no. 8, 1982, pp. 1871-1894.
[23] H. Kobayashi and Q. Ren, "A Mathematical Theory for Transient Analysis of Communications Networks," *IEICE Trans. on Commun.*, vol. E75-B, no. 12, 1992, pp. 1266-1276.
[24] P. Sen *et al.*, "Models for Packet Switching of Variable-Bit-Rate Video Sources," *J. on Selected Areas in Commun.*, vol. 7, no. 5, 1989, pp. 865-869.
[25] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control, (Englewood Cliffs, NJ: Prentice-Hall, 1976).
[26] D. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teletraffic in ATM Networks," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 2, 1992, pp. 49-59.
[27] G. Ramamurthy and B. Sengupta, "Modeling and Analysis of a Variable Bit Rate Video Multiplexor," *Proceedings of INFOCOM '92*, Florence, Italy, 1992, pp. 817-827.
[28] D. L. Jagerman and B. Melamed, "The Transition and Autocorrelation Structure of TES Processes Part I: General Theory," *Stochastic Models*, vol. 8, no. 2, 1992, pp. 193-219.
[29] D. L. Jagerman and B. Melamed, "The Transition and Autocorrelation Structure of TES Processes, Part II: Special Cases," *Stochastic Models*, vol. 8, no. 3, 1992, pp. 499-527.
[30] B. Melamed and B. Sengupta, "TES Modeling of Video Traffic," *IEICE Trans. on Commun.*, vol. E75-B, no. 12, 1992, pp.1292-1300.
[31] L. Devroye, *Non-Uniform Random Variate Generation*, Berlin, Heidelberg, (New York: Springer-Verlag, 1986).
[32] D. Geist and B. Melamed, "TEStool: An Environment for Visual Interactive Modeling of Autocorrelated Traffic," Proceedings of the 1992 International Conference on Communications, Chicago, Illinois. vol. 3, 1992, pp. 1285-89.
[33] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Commun. of the ACM*, vol. 34, 1991, pp. 46-58.
[34] W. E. Leland *et al.*, "On the Self-Similar Nature of Ethernet Traffic," in press, Bellcore, Morristown, New Jersey, 1993.
[35] J. Beran et al, "Variable-Bit-Rate Video Traffic and Long-Range Dependence," accepted for publication in *IEEE Trans. on Commun.*, 1992.
[36] B. B. Mandelbrot, The Fractal Geometry of Nature, (New York, NY: Freeman, 1983).
[37] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian Motions, Fractional Noises and Applications," *SIAM Review*, vol. 10, 1968, pp. 422-437.
[38] C. W.J. Grange and R. Joyeux, "An Introduction to Long-Memory Time Series Models and Fractional Differencing, " *Time Series Anal.*, vol. 1, 1980, pp. 15-29.
[39] D. R. Cox, "Long-Range Dependence: A Review," in Statistics: An Appraisal, H.A. David and H.T. David, Eds., (The Iowa State University Press, Ames, Iowa, 1984) pp. 55-74.
[40] H. E. Hurst, "Long-Term Storage Capacity of Reservoirs," *Trans. Amer. Soc. Civil Engineers*, vol. 116, 1951, pp. 770-799.
[41] Q. Wang and V. Frost, "Efficient Estimation of Cell Blocking Probability for ATM Systems," *IEEE Trans. on Networking*, April 1993.
[42] F. B. Bernabei *et al.*, "ATM System Buffer Design Under Very Low Cell Loss Probability Constraints," IEEE Conf. on Computer Commun., INFOCOM '91, Bal Harbor, FL, April 9-11, 1991.
[43] J. P. C. Kleijnen, *Statistical Techniques in Simulation*, Part I, New York, NY: Marcel Dekker, Inc., 1974.
[44] C. E. Clark, "Importance Sampling in Monte Carlo Analysis," *Operations Research*, Sept./Oct. 1961, pp. 603-620.
[45] K. S. Shanmugan and P. Balaban, "A Modified Monte Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," *IEEE Trans. on Commun.*, vol. 28, no. 11, Nov. 1980, pp. 1916-1924.
[46] V. G. Hansen, "Detection Performance of Some Nonparametric Rank Tests and an Application to Radar," *IEEE Trans. on Information Theory*, vol. 16, no. 3, May 1970, pp. 309-318.
[47] M. A. Crane, *An Introduction to the Regenerative Method for Simulation Analysis*, (Berlin, Heidelberg, New York: Springer-Verlag, 1977).
[48] J. A. Bucklew, *Large Deviation Techniques in Decision, Simulation, and Estimation*, (New York, NY: John Wiley & Sons, Inc., 1990).
[49] M. Devetsikiotis and K. Townsend, "Statistical Optimization of Dynamic Importance Sampling Parameters for Efficient Simulation of Communication Networks," to appear in *IEEE/ACM Trans. on Networking*.
[50] D. Reininger, B. Melamed, D. Raychandhuri, "Variable Bit Rate MPEG Video: Characteristics, Modeling, and Multiplexing," to appear in *ITC*, 1994.

Biographies

VICTOR S. FROST [SM '90] received B.S., M.S., and Ph.D. degrees from the University of Kansas, Lawrence in 1977, 1978, and 1982, respectively. He joined the faculty of the University of Kansas, Lawrence, Kansas, in 1982, where he is currently a professor of electrical engineering and computer science. He has been the director of the Telecommunications and Information Sciences Laboratory at the University of Kansas since 1987. His current research interests are in the areas of integrated communication networks, high-speed networks, communications system analysis, and simulation. He is currently involved in research on the MAGIC gigabit ATM testbed. He has received a Presidential Young Investigator Award from the National Science Foundation in 1984, an Air Force Summer Faculty Fellowship, a Ralph R. Teetor Educational Award from the Society of Automotive Engineers, and the Miller Professional Development Awards for Engineering Research and Service in 1986 and 1991 respectively. He is a member of Eta Kappa Nu, Tau Beta Pi, and he served as chair of the Kansas City Section of the IEEE Communications Society from June 1991 through December 1992.

BENJAMIN MELAMED [F '94] is head of the Performance Analysis Department at the C&C Research Laboratories, NEC USA, Inc., Princeton, New Jersey, where he has been employed since 1989. His research interests include system modeling and analysis, simulation, stochastic processes, and visual modeling environments. Melamed received a B.Sc. in mathematics and statistics from Tel Aviv University in 1972 and M.S. and Ph.D. degrees in computer science from the University of Michigan in 1973 and 1976, respectively. From 1977 to 1981 he taught at the department of Industrial Engineering and Management Science at Northwestern University. He joined the Performance Analysis Department at Bell Laboratories in 1981 and became an AT&T Bell Laboratories fellow in 1988.

Telecommunications professionals are being called upon to design and manage an assortment of communications systems in the face of fast-moving technology and a climate of increasing customer expectations.