Performability of an Algorithm for Connection Admission Control

John F. Meyer, *Fellow*, *IEEE*

Abstract—Connection admission control (CAC) in broadband, ATM-based telecommunication networks is a problem of recognized importance. Its solution calls for real-time algorithms that can accurately decide whether a connection request may be granted without compromising specified quality of service (QoS) requirements. We consider an algorithm for this purpose that accommodates variable bit rate (VBR) connections and is based on effective bandwidth computations. Arriving VBR traffic streams are assumed to be leaky-bucket regulated and are represented by worst-case, periodic, on-off, fluid sources that are randomly phased. Moreover, certain traffic classes may benefit from statistical multiplexing (S-VBR), while others may not (NS-VBR). We then evaluate the ability of such a CAC algorithm to perform in the presence of fluctuating channel capacity, where the performability variable Y_T is the fraction of some specified busy period T during which the quality of service requirement is violated. A general base model that supports Y_T is then formulated and instances of the resulting performability model are constructed and solved using UltraSAN. In particular, the evaluation experiments reveal some interesting differences in how QoS degrades with respect to the control of S-VBR vs. NS-VBR traffic.

Index Terms—Broadband telecommunication networks, connection admission control (CAC), effective bandwidth, performability, variable bit rate (VBR) traffic.

1 Introduction

Regarding methods of connection admission control (CAC) in ATM-based telecommunication networks, techniques that employ effective bandwidth computations have dealt primarily with congestion at the burst level. Moreover, two basic approaches have been generally recognized for this purpose (see [1], for example). The first is suited to *burst scale delay* systems which employ large buffers to accommodate burst congestion. The principal quality of service (QoS) measure in this case is the probability of buffer overflow and, with regard to burst admission, a multiplexed load is acceptable if this probability is less than or equal to a very small number (typically 10^{-9}). Alternatively, one can consider *burst scale loss* systems which employ smaller buffers, provided they are large enough to guarantee loss QoS at the ATM-cell level. In this case, the principal QoS measure is the probability of saturation, i.e., the probability P_{sat} that the aggregate bit arrival rate exceeds capacity, thus causing bursts to be lost. Here, a multiplexed load is regarded as being acceptable if P_{sat} is no greater than some very small value (again typically 10^{-9}).

The investigation that follows assumes the second scenario where, historically, an important set of initial results are those obtained by Hui [2]. (Hui's paper also considers congestion analysis at the cell and call levels.) The model used by Hui to evaluate the burst scale loss probability P_{sat} presumes unbuffered resources; hence, it initially appeared to have limited applicability in an ATM setting. Since then, however, a number of studies have shown how Hui's approach can be extended to buffered resources. Effort in this direction was led by Kelly [3], who derived several alternative formulations of buffer-related effective bandwidths (EBs) according to the nature of the QoS requirement. More recently, Elwalid et al. [4] have exploited Hui's results via an innovative transformation of a two-resource (buffer-channel) problem into one involving a single resource (buffer or channel). This is done by imposing a reasonable constraint on how both resources are allocated relative to a specified traffic class. Moreover, by focusing the analysis on the buffer output (channel input), the "bufferless results" of Hui can then be applied.

Extending the analysis of [4], which uses the the basic Chernov bound to estimate P_{sat}, we have been able to formulate improved EBs using the modified Chernov bound [5]. This patent application also describes a *Table-Based CAC Algorithm* that circumvents known difficulties associated with controlling the admission of mixed statistically multiplexable (S-VBR) and nonstatistically multiplexable (NS-VBR) sources. Relevant material from both [4] and [5] is reviewed in Section 2.

In particular, the Table-Based (TB) Algorithm just cited relies on an ordering of traffic classes according to their "relative multiplexabilities" and a table of precomputed EBs for incremental values of residual capacity. The latter suggests that the evaluation of this algorithm can be extended quite naturally to channels with fluctuating capacity. For example, such capacity variations could result from link faults and subsequent recovery actions, a setting that is well-suited to performability modeling (as surveyed in [6], [7], for example). A general performability model for this purpose is introduced in Section 3, followed by the formulation of a performability variable Y_T that quantifies the extent to which the QoS requirement is violated during

• *The author is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109. E-mail: jfm@eecs.umich.edu.*

Manuscript received 23 Aug. 2000; accepted 3 Jan. 2001.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 112765.

0018-9340/01/$10.00 © 2001 IEEE

a specified busy period T. Section 4 describes the model's construction and solution (with respect the prescribed measure) using UltraSAN.[1] Results of an initial evaluation study based on this model are then presented in Section 5.

Although the model presumes the admission criteria and EB computations associated with the TB Algorithm [5], alternative assumptions concerning how connections are actually admitted can be accommodated by altering the representation of accepted connections. In this sense, the model is not specific to the TB Algorithm. Moreover, the model has many other features that permit extensions in several directions. These are summarized in Section 6, along with a number of suggestions for further research.

This paper revises and extends a conference version [9] presented at the International Conference on Dependable Systems and Networks (DSN 2000). Extensions include an interesting conjecture based on the results reported in Section 5. For the most part, the revisions can be credited to helpful suggestions from reviewers whose assistance in this regard is truly appreciated.

2 Background

2.1 Traffic Assumptions

As in [4], traffic sources are represented by worst-case, regulated rate processes. Specifically, given a leaky bucket characterized by parameters r (the token rate, which regulates the mean input rate), B_T (the token buffer capacity, which bounds the burst size), and P (the peak rate permitted, where $r \leq P$), a source, when so regulated, behaves as follows: If $r < P$, then the output of the regulator is a deterministic, periodic, on-off rate process. Moreover, before regulation, the source is worst-case with respect to limits imposed by the leaky bucket, i.e., 1) when on, its rate is the allowed peak rate P, 2) the amount of information (burst size) permitted is maximum, and 3) the mean rate in during each period is r. Consequently, the regulated process is on ($\text{rate} = P$) for a duration $T_{\text{on}} = B_T/(P - r)$ and then off ($\text{rate} = 0$) for a duration $T_{\text{off}} = B_T/r$. Individual sources are therefore identical except for their phases, where the latter are assumed to be statistically independent random variables uniformly distributed over the period $T_{\text{on}} + T_{\text{off}}$. If $r = P$, then the off period disappears, i.e., the output of the regulator becomes a constant bit rate (CBR) process with rate r. Heterogeneity is introduced by considering a set $\mathcal{J} = \{1, 2, \ldots, J\}$ of J traffic classes that are distinguished by leaky buckets with differing parameter values, i.e., for each $j \in \mathcal{J}$, source class j is characterized by the parameter triple $(r_j, B_{T,j}, P_j)$.

2.2 Resource Allocation

With these traffic assumptions, the analysis in [4] concerns the probabilistic behavior of channel utilization at the output of a buffer with capacity B. At this point, such behavior can be viewed as the input to a bufferless resource, thus permitting application of some important results described in [2]. Quantifying loss at the burst level, the *saturation probability* P_{sat} is the probability that the total instantaneous demand S (aggregate burst rate) exceeds the channel capacity C, i.e.,

$$P_{\text{sat}} = \Pr[S > C]. \tag{1}$$

The QoS requirement places an upper bound on P_{sat}, i.e., for some specified positive real number γ,

$$P_{\text{sat}} \leq e^{-\gamma}. \tag{2}$$

This allowed saturation probability is assumed to be very small, e.g., 10^{-9}, in which case $\gamma \approx 20.72$.

To reduce the allocation problem to a single resource (either the buffer or the channel), it is assumed that buffer/bandwidth allocations for a class-j connection are such that both resources are exhausted simultaneously. This permits the formulation of the *no-saturation effective bandwidth* (NEB) $e_{0,j}$ of a class-j source in terms its characterizing parameters $(r_j, B_{T,j}, P_j)$ and the ratio B/C of buffer capacity B to channel capacity C. (See equation (15) on p. 1118 of [4], where this is referred to as the "effective bandwidth for lossless performance.") The term NEB for $e_{0,j}$ is due to the fact that a connection vector $n = (n_1, n_2, \ldots, n_J)$, where n_j is the number of sources in class j, is admissible with no saturation ($P_{\text{sat}} = 0$) if and only if

$$\sum_{j=1}^{J} n_j e_{0,j} \leq C. \tag{3}$$

Further, letting w_j denote the fraction of time that the channel is utilized by a class-j source, the product $w_j e_{0,j}$ is equal to the mean rate r_j permitted by the regulator, whence

$$w_j = \frac{r_j}{e_{0,j}}. \tag{4}$$

Consequently, for an assumed value of the ratio B/C, the two quantities $e_{0,j}$ and w_j suffice to characterize traffic class j.

2.3 Chernov Bounds

As in [4], but conforming more closely with the notation of [3], let

- X_{ji}: Load due to the ith class-j source.
- $W_j(x)$: Probability distribution function (PDF) of X_{ji} (sources in a given class are the same except for random phasing).
- $M_j(s) = \log E[e^{sX_{ji}}] = \log \int_0^\infty e^{sx} dW_j(x)$: Logarithmic moment generating function (LMGF) of X_{ji}.
- $S = \sum_{j=1}^{J} \sum_{i=1}^{n_j} X_{ji}$: Aggregate load for connection vector n.
- $M_S(s) = \sum_{j=1}^{J} n_j M_j(s)$: LMGF of S.

Then, given that saturation is possible, Chernov's bound applies to the random variable $S - C$, yielding

$$P_{\text{sat}} \leq e^{-F_n(s^*)}, \tag{5}$$

where $F_n(s) = sC - M_S(s)$ and s^* is the unique value of s that maximizes $F_n(s)$, i.e.,

1. UltraSAN is a software package for model-based performability evaluation of systems represented by stochastic activity networks (SANs); see [8], for example.

$$F_n(s^*) = \sup_{s \geq 0} F_n(s).$$

The multiplexability distinctions and subsequent analysis of [4] are based on the basic Chernov bound (BCB) given by (5). An improved bound, first considered by [2] in this context, is the modified Chernov bound (MCB)

$$P_{\text{sat}} \leq \frac{e^{-F_n(s^*)}}{s^* \sqrt{2\pi\sigma_n^2(s^*)}}, \psi \tag{6}$$

where

$$\sigma_n^2(s) = \frac{\partial^2}{\partial s^2} M_S(s) = \sum_{j=1}^{J} n_j M_j''(s).$$

Although the use of (6) results in analytic complications (due to the nature of the denominator) that do not exist for the basic bound (5), we have shown [5] that they can be dealt with effectively.

Letting $K_n(s^*) = s^* \sqrt{2\pi\sigma_n^2(s^*)}$ and assuming that saturation is possible, by taking logarithms on both sides of the MCB, the QoS requirement will be satisfied by connection vector n if $-F_n(s^*) - \log K_n(s^*) \leq -\gamma$. Equivalently, letting

$$G_n(s^*) = F_n(s^*) + \log K_n(s^*), \psi \tag{7}$$

the admission criterion is simply

$$G_n(s^*) \geq \gamma. \psi \tag{8}$$

When compared with the BCB-based criterion $F_n(s^*) \geq \gamma$, this is more liberal since, except for extreme values of n, $\log K_n(s^*) > 0$.

2.4 Multiplexability

If bandwidth is allocated to class-*j* VBR sources in isolation, then class j is *statistically multiplexable* (S-VBR) if the maximum number of admissible sources exceeds the maximum $n_{0,j} = C/e_{0,j}$ permitted with no saturation; otherwise, it is *nonstatistically multiplexable* (NS-VBR). With a slight abuse of notation, let n_j denote the connection vector consisting of n_j class-*j* sources only, i.e., $n_j = (0, 0, \ldots, 0, n_j, 0, \ldots, 0)$. Then, ignoring the QoS requirement, saturation occurs in the interval

$$N_j = \{n_j \mid n_{0,j} < n_j \leq n_{0,j}/w_j\}, \psi \tag{9}$$

where the upper bound insures stability. Note that if $w_j = 1$, then the interval N_j is empty and nothing more needs to be done. This is consistent with the definition (4) of w_j, where its value is 1 if class j is bandwidth limited, including the CBR extreme where $e_{0,j} = P_j = r_j$.

If we now impose the QoS constraint then, by the MCB-based admission criterion (8), source class j is S-VBR if, for some $n_j \in N_j, G_{n_j}(s^*) \geq \gamma$; otherwise, it is NS-VBR. Formulating the summands of (7) in terms of n_j, $n_{0,j} = C/e_{0,j}$ and the class-*j* utilization fraction w_j, it can be shown [5] that

$$F_{n_j}(s^*) = n_{0,j}\left[\log\left(\frac{n_{0,j}}{n_j w_j}\right) + (n_j/n_{0,j} - 1)\log\left(\frac{n_j - n_{0,j}}{n_j(1 - w_j)}\right)\right] \tag{10}$$

and

$$\log K_{n_j}(s^*) = \frac{1}{2}\log\left(\frac{2\,\pi\, n_{0,j}\,(n_j - n_{0,j})}{n_j}\right) + \log\left(\log\left(\frac{n_{0,j}\,(1 - w_j)}{(n_j - n_{0,j})\,w_j}\right)\right), \psi \tag{11}$$

thus providing a closed-form solution of $G_{n_j}(s^*)$ for isolated class j. In the limit as n_j approaches either its lower bound $n_{0,j}$ or upper bound $n_{0,j}/w_j$, $G_{n_j}(s^*) \to -\infty$. However, with the exception of values very near these extremes (which will not arise in practice), $G_{n_j}(s^*)$ is a nicely behaved function of n_j over the interval N_j. Specifically, it has a maximum value which occurs when n_j is quite close to $n_{0,j}$; beyond this point, G_{n_j} decreases montonically with increasing n_j and remains larger than $F_{n_j}(s^*)$ until n_j is very close to $n_{0,j}/w_j$.

2.5 Characterization of S-VBR Classes; EB Computations

In view of the above, whether a traffic class can profit from statistical multiplexing can be decided as follows: Let $m_j(C)$ denote the maximum value of $G_{n_j}(s^*)$, i.e.,

$$m_j(C) = \max_{n_j \in N_j} G_{n_j}(s^*), \psi \tag{12}$$

Then, in terms of (12) and the value of the QoS exponent γ, it can be shown [5] that

$$\text{class } j \text{ is S-VBR for capacity } C \text{ if and only if } m_j(C) \geq \gamma. \tag{13}$$

Although this characterization is similar to that of [4], it differs quantitatively, thereby altering the S-VBR/NS-VBR dichotomy. In particular, a class that is NS-VBR according to the BCB-based definition may be S-VBR according to (13).

As in the BCB-based analysis of [4], but in terms of our modified S-VBR criterion, the problem of computing effective bandwidths (EBs) differs considerably according to the following three cases.

Case 1: $\max_{j \in \mathcal{J}} m_j(C) < \gamma$. Here, by (13), no class is S-VBR (all are NS-VBR).

Case 2: $\min_{j \in \mathcal{J}} m_j(C) \geq \gamma$. Again, by (13), this says that all source classes are S-VBR.

Case 3: Neither of the above, i.e., there is a proper mix of S-VBR and NS-VBR classes.

In Case 1, nothing can be gained by statistical multiplexing since all classes are NS-VBR; hence, the EBs are just the NEBs $e_{0,j}$ for all $j \in \mathcal{J}$. In Case 2, the boundary of the set of admissible connection vectors can be characterized strictly in terms of the modified Chernov bound. Moreover, as described in [5], the EBs can be computed via an approximating hyperplane that accurately and conservatively bounds the admissible set.

By its definition, Case 3 involves a proper mix of S-VBR and NS-VBR classes, resulting in an admission boundary that gets distorted by the influence of the NS-VBR classes. As demonstrated in [4] and emphasized in their concluding remarks, this interaction can preclude an accurate linear approximation of the admissible set. Indeed, this was the challenge that motivated our development of the TB Algorithm [5]. In particular, this algorithm relies on precomputed EBs for various values of residual channel

capacity, suggesting an extension to channels whose capacities fluctuate randomly for certain reasons.

3 CAC Performability

3.1 Fluctuating Capacity

In what follows, we assume that capacity fluctuations are due to link faults and subsequent fault recovery actions. However, other interpretations can likewise be considered, either by adjusting certain parameter values in the capacity submodel or perhaps altering this submodel to fit the needs of a given interpretation. More formally, we consider a channel whose physical capacity ranges over a set $\mathcal{C}$ of $K+1$ equally spaced values

$$\mathcal{C} = \left\{ \frac{kC}{K} \mid k = 0, 1, \ldots, K \right\},$$

where, here, C denotes the fault-free capacity. In turn, we let

$$Z = \{Z_t \mid t \in [0, \infty)\} \tag{14}$$

be a stochastic process that represents how capacity varies as a function of continuous time, where the random variables Z_t take values in the state space $\mathcal{C}$. In the presence of such fluctuations, we need to address issues concerning both how connections are admitted (e.g., to what extent is the admission algorithm aware of the actual capacity Z_t) and how such connections are affected by capacity changes once they have been admitted.

Generally, a network node will admit connections based on the amount of channel capacity it believes to be available for allocation. However, if capacity fluctuates, this knowledge will likely differ from the capacity Z_t that is actually available at time t. In particular, if Z_t is less than the assumed capacity, then the admission algorithm may make decisions that violate the QoS requirement (2). The extent of such QoS degradation obviously depends on how quickly and accurately a node is informed of the bandwidth actually available for allocation (as represented by the process Z). Note that, even in the ideal case where a node has exact knowledge of Z, if a connection is admitted at time t and if $Z_{t'} < Z_t$ at some time $t' > t$ before the connection is released, this can also result in a saturation probability that exceeds the QoS requirement.

3.2 A Performability Model

The above suggests a variety of possible studies concerning the ability of a CAC algorithm to perform in the presence of fluctuating capacity. As noted at the outset, we assume that connection admissions are decided according to the criteria and EB computations developed for the TB Algorithm. However, to simplify things for this initial investigation, we suppose that the traffic is homogeneous, i.e., there is but one traffic class ($J = 1$). Accordingly, the possibility of mixed S-VBR/NS-VBR traffic (Case 3; see Section 2.5) is excluded. This implies that the EB e_1^{m} of a class-1 connection (the superscript m indicates computation relative to the modified Chernov bound) is either its NEB $e_{0,1}$ (Case 1) or is determined by Case 2 computations. Moreover, since there is only one traffic class, the hyperplane construction for Case 2 reduces to determining a single point, namely the maximum number $n_{\max,1}$ of class-1 sources that can be connected without exceeding the QoS bound $e^{-\gamma}$ on P_{sat}.

In either case, this EB depends on the total amount of bandwidth (channel capacity) that is assumed to be available for allocation. For the study that follows (and again in the interest of simplifying things initially), we suppose that the CAC algorithm has no knowledge of fault-caused fluctuations in the channel's capacity. In other words, we are assuming that EB computations for the purpose deciding admissibility are based on the fault-free capacity C. Therefore, in Case 1, the maximum number of connections that can be admitted is $n_{0,1} = C/e_{0,1}$. In Case 2 (recalling the definition (12) of $m_j(C)$ and applying it to class 1), from the development in [5], it follows that $n_{\max,1}$ is either the value of n_1 for which $G_{n_1}(s^*) = m_1(C)$ (if $m_1(C) = \gamma$) or the larger of the two values of n_1 that satisfy $G_{n_1}(s^*) = \gamma$ (if $m_1(C) > \gamma$).

Accordingly, let

$$A = \{A_t \mid t \in [0, \infty)\} \tag{15}$$

be a stochastic process representing the dynamics of connection admission control, i.e., A_t is the number of (class-1) sources connected to the channel at time t. Then, under the above assumptions,

$$A_t \leq \begin{cases} n_{\max,1} & \text{if class 1 is S-VBR} \\ & \text{(as judged by the CAC algorithm)} \\ n_{0,1} & \text{else.} \end{cases}$$

At first glance, consideration of an NS-VBR class (the "else" case) appears uninteresting since its NEB is allocated to each admitted connection and, hence, saturation is not possible with respect to the fault-free capacity C. Moreover, since the class is NS-VBR for capacity C, it must be NS-VBR for any capacity $Z_t < C$. However, such a reduction in capacity may cause saturation and, as soon as this occurs, the QoS requirement is violated. Indeed, if an NS-VBR class is bandwidth limited ($w_1 = 1$, which is true in particular for any CBR class), then the conditions that characterize saturation, a QoS violation, and instability (the negation of condition (17) given below) are all equivalent.

Generally, saturation will be possible at time t if the number A_t of connections at time t exceeds the no-saturation maximum with respect to the actual capacity Z_t, i.e.,

$$A_t > Z_t/e_{0,1}. \tag{16}$$

If (16) holds and, moreover, the system is stable in the sense that the mean aggregate load does not exceed the capacity (as required in order to formulate the MCB; see (9)), i.e.,

$$A_t w_1 \leq Z_t/e_{0,1}, \tag{17}$$

then, using the MCB to approximate P_{sat} at time t,

$$P_{\mathrm{sat}}(t) \approx e^{-G_{A_t}(s^*)}; \text{ else } P_{\mathrm{sat}}(t) = 0. \tag{18}$$

Further, specializing (10) and (11) to class 1 and capacity Z_t, the variables n_1 and $n_{0,1}$ can then be expressed in terms of the random variables A_t and Z_t, namely, $n_1 = A_t$ and $n_{0,1} = Z_t/e_{0,1}$. Making these substitutions in the equations

just referred to, the random variable $G_{A_t}(s^*)$, which depends on Z_t as well as A_t, is given by the sum

$$G_{A_t}(s^*) = F_{A_t}(s^*) + \log K_{A_t}(s^*), \tag{19}$$

where

$$F_{A_t}(s^*) = \frac{Z_t}{e_{0,1}}\left[\log\left(\frac{Z_t e_{0,1}}{A_t w_1}\right) + (A_t e_{0,1}/Z_t - 1)\log\left(\frac{A_t - Z_t/e_{0,1}}{A_t(1-w_1)}\right)\right] \tag{20}$$

and

$$\log K_{A_t}(s^*) = \frac{1}{2}\log\left(\frac{2\pi Z_t\big(A_t - Z_t/e_{0,1}\big)}{A_t e_{0,1}}\right) + \log\left(\log\left(\frac{Z_t(1-w_1)}{(A_t e_{0,1} - Z_t)w_1}\right)\right). \tag{21}$$

These formulas imply that the base model

$$(A, Z) = \{(A_t, Z_t) \mid t \in [0, \infty)\} \tag{22}$$

suffices to describe the behavior of $G_{A_t}(s^*)$ and, hence, by (18), how the saturation probability $P_{\text{sat}}(t)$ varies randomly as a function of time. In turn, this permits a number of QoS-related performability measures to be supported by the base model (22), where an interesting example is the following.

Let $T = [u, v]$ denote a specified busy period ($u < v$) and let Y_T denote the fraction of period T during which the QoS requirement is violated (not satisfied). By definition (2), such a violation occurs at time t if

$$P_{\text{sat}}(t) > e^{-\gamma}. \tag{23}$$

Moreover, (23) will occur if and only if either

1. the system is stable (17), saturation is possible (16), and (assuming (18) to be exact) $G_{A_t}(s^*) < \gamma$, or
2. the system is unstable due to a mean aggregate load at time t that exceeds Z_t.

Then to formulate Y_T in terms of the base model, it suffices to consider a binary-valued reward variable whose defining predicate captures the above conditions, i.e., the variable

$$V_t = \begin{cases} 1 & \text{if } (A_t e_{0,1} > Z_t \geq A_t w_1 e_{0,1} \text{ and } G_{A_t}(s^*) < \gamma) \\ & \text{or } A_t w_1 e_{0,1} > Z_t \\ 0 & \text{else.} \end{cases} \tag{24}$$

The first half of the predicate (to the left of "or") states condition 1; the second half is just the negation of (17), thereby formalizing condition 2. Hence, $V_t = 1$ if and only if the QoS requirement is violated according to (23). In turn, the performability variable in question can be expressed as

$$Y_T = \frac{\int_u^v V_t dt}{v - u}, \tag{25}$$

permitting the evaluation of various measures of Y_T such as its mean and, if feasible, selected points of its probability distribution function.

4 Model Construction and Solution

Using stochastic activity networks (SANs) and UltraSAN (see footnote 1), instances of the performability model just described can be constructed and solved as follows. Throughout, we assume a fixed QoS requirement for the admitted connections, namely $P_{\text{sat}} \leq 10^{-9}$, implying a value of $\gamma = 20.72$ for the QoS exponent. We also assume that the fault-free capacity of the channel is fixed at $C = 150\text{Mbps}$. If desired, however, the model can be easily modified so as to incorporate both γ and C as parameters. The quantization factor K is likewise fixed, where we let $K = 15$. (A larger value of K can be considered by expanding the capacity submodel accordingly.) Hence, fluctuations in capacity range over $K + 1 = 16$ equally spaced levels, namely the (Mbps) values in the set

$$\mathcal{C} = \{0, 10, 20, \ldots, 150\},$$

All other numerical choices are parameterized, permitting a great deal of flexibility in experimenting with various instances of the resulting base model.

4.1 Admissions Submodel

By our general definition (15) of the process A that represents the admitted connections, the maximum value of A_t depends on whether the traffic class is S-VBR or NS-VBR with respect to capacity C. To decide this, the (single) traffic class in question is described by the two parameters whose values fully characterize the class with respect to its EB computation. Specifically, since there is but one traffic class, we simplify notation by letting $e_0 = e_{0,1}$ and $w = w_1$. Given the values of these parameters, we decide whether the class is S-VBR or NS-VBR and, in turn, determine the maximum number of connections that can be admitted according to its EB (see Section 3.2). For convenience, we introduce this as an additional parameter, denoted $n_{\max}$, where

$$n_{\max} = \begin{cases} \lfloor n_{\max,1} \rfloor & \text{if the traffic class is S-VBR} \\ \lfloor C/e_0 \rfloor & \text{else} \end{cases}$$

($\lfloor x \rfloor$ denotes the *floor of* x), permitting the quantity $n_{\max,1} = C/e_1^{\text{m}}$ to be computed offline in case the class is S-VBR.

To complete the submodel's formulation, we assume that connection admission requests arrive as a Poisson process with rate *ad_rate* (connections/hr) and that the holding time of a connection is exponentially distributed with mean $1/rel_rate$ (minutes). Moreover, as just discussed, A_t is bounded according to EB-based admission control, i.e., a connection request at time t is denied (blocked) if $A_t = n_{\max}$. Accordingly, the admissions submodel corresponds to an $\text{M/M}/n_{\max}/n_{\max}$ queue and is easily specified by a single-place SAN with two exponentially distributed timed activities. The marking of its place at time t is A_t, where a parameter, denoted n_{init}, permits specification of the initial marking (whence $\Pr[A_0 = n_{\text{init}}] = 1$). One of the two timed activities represents the admission of a connection (when the activity completes); the other represents the release of a connection.

Although the definition (24) of the performability variable Y_T does not presume steady-state conditions, it is nevertheless worth noting that the admissions submodel

has a very familiar steady-state behavior. Specifically, as $t \to \infty$, the probability distribution of A_t is a truncated (at $n_{\max}$) Poisson distribution with parameter $\rho = ad_rate/rel_rate$ (the offered load in Erlangs). In particular, the steady-state probability of rejecting a connection ($\Pr[A_t = n_{\max}]$) is given by Erlang's B formula.

It should also be noted that the above assumptions, while being reasonable for certain types of connections (e.g., ordinary telephone service), may not conform to what is experienced for other services. For example, holding times for data transfers between computers are typically non-exponential. Means of accommodating such differences via simulation are discussed in the concluding section (see Item 3) of Section 6).

4.2 Capacity Submodel

Given that the fault-free capacity is $C = 150$ Mbs, we assume that losses of channel capacity are caused by four different types of faults, denoted f_0, f_1, f_2, and f_3. Each fault type results in a different capacity reduction, where the loss caused by type f_{i+1} is twice that of f_i. More precisely, so as to conform with our earlier choice of $K + 1 = 16$ capacity levels, for $0 \le i \le 3$ we suppose that an occurrence of fault-type i results in a capacity loss of $10 \cdot 2^i$ Mbps. Thus, for example, if fault type f_0 occurs at time t, then the capacity Z_t is 10 Mbps less than it was just before the fault occurred; if type f_1 occurs, the reduction is 20 Mbps, etc. Once a fault of a given type occurs, a recovery action is initiated and, during the recovery period, we assume that no other faults of that type occur. However, faults of different types are presumed to occur independently. Hence, it is possible for all four types to be in effect simultaneously (prior to a recovery from any of them), in which case the total capacity reduction is $\sum_{i=0}^{3} 10 \cdot 2^i = 150$ Mbps. In other words, this state of Z corresponds to a complete loss of the channel.

Probabilistically, we make some choices which insure that Z is a time-homogeneous Markov process. Specifically, we assume that, for $0 \le i \le 3$, fault-type f_i arrives as a Poisson process with rate a_i (faults/hour) and has an exponentially distributed recovery period with mean duration $1/r_i$ (minutes). If all of these parameters have positive values, the result is a 16-state Markov process. Moreover, Z can be conveniently specified by a SAN with four places corresponding to the four fault types. Associated with each place are two exponentially distributed timed activities whose completions represent a fault arrival and a fault recovery, respectively. Assuming that the channel is initially fault-free ($\Pr[Z_0 = C] = 1$), the initial marking of each place is 1, signifying the absence all fault types at time $t = 0$.

4.3 Performability Variable Specification and Solution

When the two SANs just described for A and Z are composed (placed side by side), the resulting SAN specifies the product base model (A, Z). As so specified, the latter is a time-homogeneous Markov process with $(n_{\max} + 1) \cdot 16$ states. Although its nature is simple enough to permit manual specification of its generator matrix, as $n_{\max}$ gets larger (e.g., $n_{\max} = 90$ implies 1,456 states), this can be a time-consuming task. Using UltraSAN, this matrix is constructed automatically by invoking the tool's *reduced base model generator*.

However, a much more beneficial advantage of SAN specification is the variety of solution algorithms provided by UltraSAN, particularly the kind of reward model solvers that are suited to performability evaluation. Generally, using UltraSAN's *performability variable editor*, a reward variable is specified by

1. a predicate (stated in terms of place markings and model parameters) which, if true, results in the reward rate given by 2 (if false, the reward rate is 0), and
2. a function of place markings and parameters that specifies the reward rate, per se.

Specifically, for the reward variable V_t defined by (24), part 1 is obtained by expressing the random variables of the predicate

$$(A_t e_0 > Z_t \ge A_t w e_0 \text{ and } G_{A_t}(s^*) < 20.72) \text{ or } A_t w e_0 > Z_t \tag{26}$$

in terms of place markings and the model's parameters. In particular, A_t is just the marking of the single place of the admissions SAN and Z_t is the weighted sum

$$\sum_{i=0}^{3} 10 \cdot 2^i \cdot MARK(fi),$$

where place fi represents fault-type f_i in the capacity SAN. As already shown, the random variable $G_{A_t}(s^*)$ can be expressed in terms of A_t and Z_t as the sum of (20) and (21); hence, this part of the predicate is similarly accounted for. As for the reward-rate function (part 2), by the definition of V_t, this rate is always 1 (whenever the predicate holds); hence, this is specified simply as a constant having value 1.

Finally, regarding the performability variable Y_T, the definition given by (25) is generally referred to as *time-averaged accumulated reward* with respect to the reward variable V_t. Using UltraSAN, the expected value of Y_T (the performability measure) can be obtained via the tool's *accumulated reward solver*, where time interval $[u, v]$ is specified in the solver's control panel. In addition, UltraSAN's *pdf solver* can be used to evaluate selected points on the PDF of Y_T, i.e., probabilities of the form $\Pr[Y_T \le y]$ for selected values of y.

5 Evaluation Results and Their Implications

Using the model just described, the following summarizes the results of several evaluation experiments distinguished by different choices of parameter values for the base model (A, Z). Among other things, the results indicate that the model is indeed valid with respect to known theoretical properties that distinguish various traffic class types (S-VBR, NS-VBR, and bandwidth limited NS-VBR). Indeed, checks against such properties were very helpful in debugging the model's specification. Moreover, the results themselves are interesting since they illustrate how QoS degradation differs according to the multiplexability of the traffic class.

TABLE 1
Parameter Values for the Capacity Submodel

Fault type f_i	Arrival rate a_i (faults/hour)	MTTR $1/r_i$ (minutes)
f_0	1	2
f_1	0.5	5
f_2	0.25	10
f_3	0.1	60

All of the experiments assume the same choices of parameter values for the capacity submodel Z. These are given by Table 1, where MTTR denotes the *mean time to recover* from the corresponding fault type. There is no real justification for these choices; they simply provide what appears to be a reasonable example of random channel capacity fluctuations due to faults of the indicated types. Given these parameter values, the steady-state probability distribution of Z_t has a mean value of 137.34 Mbps and a standard deviation of approximately 26 Mbps. A complete loss of capacity occurs very rarely, i.e., in steady state, $\Pr[Z_t = 0] \approx 4 \cdot 10^{-5}$.

Regarding the admissions submodel A, the rate of connection admission requests is taken to be $ad_rate = 27\,\text{connections/hour}$ and the mean holding time is $1/rel_rate = 2\,\text{minutes}$, resulting in an offered load of 0.9 Erlangs. At time $t = 0$, we suppose that there are no admitted connections, i.e., we let $n_{\text{init}} = 0$, and, hence, $\Pr[A_0 = 0] = 1$. However, in view of the above parameter values, this process is in a steady-state condition after a relatively short period of time. The remaining parameters of this submodel characterize the traffic class, per se, where six classes are considered in the six experiments that follow (one class per experiment). These are distinguished by the parameter values indicated in Table 2. The NEB e_0 of a class is specified in Mbps. For the S-VBR classes, SMG refers to the *statistical multiplexing gain* of the class, i.e., the ratio of its NEB to its EB. For the NS-VBR classes, BL abbreviates "bandwidth limited" ($w = 1.0$).

The results obtained for each experiment are summarized in Tables 3 and 4 corresponding to two choices of the busy period $T = [u, v]$, both having a 1-hour duration. The first is $T = [0, 1]$, saying (since $Z_0 = \text{C}$ with probability 1) that the channel is fault-free just prior to its use. The busy period assumed for Table 4 is $T = [100, 101]$, at which time the capacity fluctuations have settled into a steady-state condition. (The behavior of the admissions submodel is essentially steady-state during either period.)

In addition to the performability variable Y_T in question (25), three other such variables are considered in order to better understand how admission-capacity differences affect the value of Y_T. These are likewise defined in terms of reward variables, namely

$$V_{1,t} = \begin{cases} 1 & \text{if } A_t e_0 > C \\ 0 & \text{else} \end{cases} \tag{27}$$

$$V_{2,t} = \begin{cases} 1 & \text{if } A_t e_0 > Z_t \\ 0 & \text{else} \end{cases} \tag{28}$$

TABLE 2
Traffic Classes for the Experiments

Exp.	e_0	w	$n_{\max}$	Comments
1	5	0.1	90	S-VBR; SMG ≈ 3.0
2	30	0.01	7	S-VBR; SMG ≈ 1.4
3	5	0.35	35	S-VBR; SMG ≈ 1.2
4	7.5	0.5	20	NS-VBR; not BL
5	10	0.3	15	NS-VBR; not BL
6	15	1.0	10	NS-VBR; BL

$$V_{3,t} = \begin{cases} 1 & \text{if } A_t w e_0 > Z_t \\ 0 & \text{else} \end{cases} \tag{29}$$

and are averaged over the same time interval $T = [u, v]$. Accordingly, the performability variable

$$Y_{1,T} = \frac{\int_u^v V_{1,t} dt}{v - u} \tag{30}$$

expresses the fraction of busy period T during which saturation would occur with respect to the CAC-assumed capacity C. Similarly, the variable

$$Y_{2,T} = \frac{\int_u^v V_{2,t} dt}{v - u} \tag{31}$$

is the fraction of T during which saturation occurs with respect to the actual capacity Z_t. In turn, the more narrowly defined variable

$$Y_{3,T} = \frac{\int_u^v V_{3,t} dt}{v - u} \tag{32}$$

expresses the fraction of T during which saturation is due to instability, i.e., the mean aggregate load exceeds the actual capacity. The measures tabulated for each experiment are the expected values of these variables.

5.1 Evidence of Model Correctness

We note first that these results satisfy certain conditions that must hold if the base model and performability variables have been correctly specified. For example, saturation at time t is necessary for a QoS violation at time t. More formally, in terms of the reward variables (24) and (28), $(\forall t \in T)[V_t = 1 \Rightarrow V_{2,t} = 1]$. Hence, by (25) and (31) and for any $y \geq 0$, $\Pr[Y_T \leq y] \geq \Pr[Y_{2,T} \leq y]$, implying (by a well-known characterization of expectation) that $E[Y_T] \leq E[Y_{2,T}]$. In the other direction, instability at time t is sufficient for a QoS violation at time t, which, in terms of (24) and (29), says that $(\forall t \in T)[V_{3,t} = 1 \Rightarrow V_t = 1]$. Reasoning as above, it follows that $E[Y_{3,T}] \leq E[Y_T]$. Hence, if the model is correct, the results obtained for these measures should satisfy both these inequalities, i.e.,

$$E[Y_{3,T}] \leq E[Y_T] \leq E[Y_{2,T}]. \tag{33}$$

An inspection of Tables 3 and 4 confirms that (33) indeed holds for each of the six experiments.

In the case of NS-VBR classes, some additional conditions can likewise be used to check the model's validity. For example, the CAC algorithm will not admit an NS-VBR

TABLE 3
Evaluation Results for Busy Period $T = [0, 1]$

Experiment	$E[Y_T]$	$E[Y_{1,T}]$	$E[Y_{2,T}]$	$E[Y_{3,T}]$
1	$1.787506e-05$	$1.682160e-36$	$1.787507e-05$	$7.649926e-06$
2	$3.832134e-03$	$3.124129e-04$	$4.346339e-03$	$7.649901e-06$
3	$1.787507e-05$	$1.682160e-36$	$1.787507e-05$	$7.703254e-06$
4	$5.779344e-05$	$0.000000e+00$	$5.779344e-05$	$1.747107e-05$
5	$8.060584e-05$	$0.000000e+00$	$8.060584e-05$	$9.747280e-06$
6	$3.727401e-04$	$0.000000e+00$	$3.727401e-04$	$3.727401e-04$

TABLE 4
Evaluation Results for Busy Period $T = [100, 101]$

Experiment	$E[Y_T]$	$E[Y_{1,T}]$	$E[Y_{2,T}]$	$E[Y_{3,T}]$
1	$5.718949e-05$	$1.940929e-36$	$5.718951e-05$	$2.455540e-05$
2	$9.268044e-03$	$3.386793e-04$	$9.800601e-03$	$2.455532e-05$
3	$5.718951e-05$	$1.940929e-36$	$5.718951e-05$	$2.472596e-05$
4	$1.831420e-04$	$0.000000e+00$	$1.831420e-04$	$5.594009e-05$
5	$2.527479e-04$	$0.000000e+00$	$2.527479e-04$	$3.125983e-05$
6	$1.100605e-03$	$0.000000e+00$	$1.100605e-03$	$1.100605e-03$

connection that causes saturation with respect to the assumed capacity C. This is borne out by the $E[Y_{1,T}] = 0$ outcomes for Experiments 4-6. Further, a QoS violation at time t coincides with actual channel saturation at time t; hence, equality must hold on the right side of (33). Comparing the values of measures $E[Y_T]$ and $E[Y_{2,T}]$ for each of Experiments 4-6, we see that they are indeed identical. Moreover, if the class is bandwidth limited ($w = 1$), then the conditions for saturation (16) and instability (17) are the same. This implies that equality holds on both sides of (33), i.e., all three of these measures should have identical values. The results of Experiment 6 demonstrate that the model is valid in this sense.

5.2 Dependence on Initial Conditions

Since the two busy periods considered ($T = [0, 1]$ and $T = [100, 101]$) have the same duration (both represent a busy hour), what distinguishes them is knowledge of the state Z_u of the channel at the time u when the period initiates. (There is also a distinction with respect to the initial state A_u of the admitted connections, but this turns out to be negligible for the parameter values considered.) As noted earlier, since the channel is assumed to be fault-free at time 0, the busy hour for Table 3 begins at a time when the channel is known to have its full capacity of 150 Mbps. On the other hand, the busy hour for Table 4 initiates when the capacity fluctuations are in a steady-state condition.

As one might expect, the mean amount of QoS loss, per the performability measure $E[Y_T]$, is worse for the steady-state busy hour. Specifically, examining the ratio

$$\frac{E[Y_{[100,101]}]}{E[Y_{[0,1]}]}$$

that compares a Table 4 entry with its corresponding entry in Table 3, this ratio ranges from 2.4 (Experiment 2) to 3.2 (Experiments 1 and 3). In other words, the extent of the QoS loss is approximately 2.5 to 3 times the amount experienced when the busy hour begins with a fault-free channel. This is an important observation since it says that even a small amount of added knowledge about the channel's status, namely that it is fully available when use begins, can improve a CAC algorithm's ability to perform in the presence of fluctuating capacity. In particular, this is encouraging for future work which, with various extensions of the admissions submodel, can examine more realistic scenarios regarding the extent of such knowledge. Such comparisons also demonstrate the importance of transient performability solutions since, without them, the evaluation data for Table 3 could not be obtained.

5.3 Dependence on the Traffic Class

5.3.1 S-VBR Classes

As for differences in the performability $E[Y_T]$ that exist for a given choice of T, among the S-VBR classes considered (Experiments 1-3), performability is the worst ($E[Y_T]$ is the largest) for Experiment 2. This is likely due to the fact that each connection in this case (see Table 2) requires a considerable allocation of bandwidth (30 Mbps for no saturation) and yet, due to extreme burstiness ($w = 0.01$), this class is S-VBR with respect to the fault-free capacity C. In particular, its SMG permits the CAC algorithm to admit $n_{\max} = 7$ connections, i.e., two more than the no-saturation maximum of 150/30 = 5 connections. Generally, it can be shown that, under steady-state conditions, the measure $E[Y_{1,T}]$ expresses the steady-state probability of exceeding the no-saturation maximum, i.e., from definitions (27) and (30), it follows that

$$E[Y_{1,T}] = \Pr[A_t > C/e_0]. \tag{34}$$

Hence, for Experiment 2, this probability (as given by Table 4) is relatively high, i.e., $3.3868 \cdot 10^{-4}$. By contrast, we see that such probabilities for Experiments 1 and 3 are on the order of 10^{-36}. As a consequence, the class assumed for

Experiment 2 is considerably more sensitive to QoS losses caused by capacity reductions.

For S-VBR classes in general, a tempting conjecture is that the value of the performability measure $E[Y_T]$ should vary inversely with SMG. This is based on the intuitive feeling that a traffic class which benefits from the statistical nature of its bit rate is less likely to be affected by random fluctuations in capacity. However, the SMG is determined by the maximum number of connections that can be admitted without compromising the QoS requirement. This ignores other factors, notably the probabilities of experiencing connection demands that are near or equal to this maximum number. Since these probabilities are bounded from above by the probability $\Pr[A_t > C/e_0]$, by (34), the measure $E[Y_{T,1}]$ is indicative of their values. In particular, comparing Experiments 2 and 3, although the class assumed in Experiment 2 has a higher SMG, as noted in the previous paragraph, the value of $E[Y_{1,T}]$ is much, much larger than that of Experiment 3. This appears to explain a resulting performability $E[Y_T]$ that, in comparison with Experiment 3, is worse by two orders of magnitude.

5.3.2 NS-VBR Classes

Regarding the three NS-VBR classes (Experiments 4-6), for either choice of T we see that the performability $E[Y_T]$ is worst for the BL class (Experiment 6). Indeed, comparing the values of $E[Y_T]$ for all six experiments in either Table 3 or Table 4, this class results in the greatest amount of QoS loss. However, this is not due to its BL nature. Generally, as we noted earlier (see the last paragraph of Section 5.1), NS-VBR connections experience a QoS violation whenever saturation occurs, i.e., the total allocated bandwidth exceeds the current capacity Z_t. Accordingly, the performability of the CAC algorithm is not affected by the fact that an NS-VBR class is BL. Indeed, given two NS-VBR classes having the same NEB, where one is BL ($w = 1$) and the other is not ($w < 1$), the only difference (all other things being equal) is the extent to which saturation is due to instability. In other words, the values of measure $E[Y_{3,T}]$ can differ, but the values of $E[Y_T] = E[Y_{2,T}]$ must be the same for both classes. Although this fact is not illustrated by the experiments of Tables 3 and 4, it has been observed by comparing the classes of Experiments 4 and 5 with their bandwidth limited counterparts.

Examining the NS-VBR results more closely, for either busy period we see that the performability $E[Y_T]$ gets worse (increases) as the NEB e_0 increases. This data, although not extensive, suggests the following conjecture.

Conjecture. Given two NS-VBR classes with differing NEBs e_0^1 and e_0^2, let $E[Y_T^1]$ and $E[Y_T^2]$ denote the respective performability outcomes. Then, for fixed values of the remaining parameters, $e_0^1 < e_0^2$ implies $E[Y_T^1] \leq E[Y_T^2]$.

To argue its plausibility, let A_t^i denote the accepted connections at time t for class i ($i = 1, 2$). (Note that this use of class names differs from that of Sections 2 and 3.) Then, due to control by the CAC algorithm, $A_t^i \leq n_{\max}^i = C/e_0^i$ and a QoS violation occurs (equivalently, saturation occurs) if and only if $A_t^i > Z_t/e_0^i$. Suppose now that $e_0^1 < e_0^2$ (the hypothesis of the conjecture). Then, $n_{\max}^1 > n_{\max}^2$, i.e., the CAC algorithm can admit more class-1 connections than class-2 connections. Although this appears to run counter to the conjecture, recall that the steady-state probability distribution of the admissions submodel A is a truncated Poisson distribution. Given that A reaches a steady-state condition rather quickly relative to the duration of T (as it does in this study), for a given number of connections n ($0 \leq n \leq \min(n_{\max}^1, n_{\max}^2) = n_{\max}^2$), it follows that $\Pr[A_t^1 = n] < \Pr[A_t^2 = n]$. Moreover, for a given value of Z_t, $Z_t/e_0^1 > Z_t/e_0^2$, saying that more connections are required for a QoS violation in the class-1 case. As a consequence of these two factors, the conjecture's conclusion appears to hold, i.e., $E[Y_T^1] \leq E[Y_T^2]$. In other words, CAC performability for class 1 (with the smaller NEB) is at least as good as that for class 2. Also, it appears likely that the inequality of the conclusion will typically be proper. However, to cover special cases, the equality option is necessary. For example, if there are no fluctuations in capacity (a degenerate capacity submodel where $Z_t = C$, for all t), then there are no QoS violations, i.e., $E[Y_T^1] = E[Y_T^2] = 0$, independent of how the NEBs of the two classes compare.

Additional experimentation is needed to obtain evidence that either further supports this conjecture or provides a counterexample. If the evidence is positive, a more careful theoretical analysis may suffice to prove that the conjectured relationship indeed holds.

6 Summary and Suggestions for Future Work

The research presented above is very encouraging in several respects. First of all, it demonstrates that CAC algorithm performability in the presence of fluctuating channel capacity can indeed be evaluated with respect to an appropriately specified base model. Further, the results of a handful of evaluation experiments have already provided valuable insight into how such performability, as quantified by $E[Y_T]$, varies according to the nature of the traffic class. And, perhaps most importantly, the base model described in Section 3 provides a framework for a number of worthwhile extensions. Specifically, some interesting possibilities in this regard are the following:

1. Generalize the admissions submodel A so as to represent the control of nonhomogeneous traffic ($J > 1$).
2. Extend the base model (A, Z) to permit the admissions submodel to have additional knowledge about the state of the capacity submodel Z. For example, this could represent information passed on by the network management system concerning its estimate of available capacity.
3. Replace the exponential distributions assumed in the current Markov model (those associated with connection arrivals, holding times, fault arrivals, and recovery times) with distributions that conform more closely to what is experienced in actual networks. Assuming that mean values of the old and new distributions coincide, estimations of $E[Y_T]$ obtained by simulation (using UltraSAN) can then

be compared with numerical values derived from the Markov model.

4. Alter the capacity submodel so as to represent less severe but more frequent changes in capacity due, say, to transient faults or phenomena such as interfering connection demands (other than those being controlled by the admission algorithm).
5. Replace the channel interpretation of the capacity submodel with that of another resource. For example, Z_t could represent the buffer space in a network node that is available for VBR traffic allocation at time t, where its random nature results from buffer-sharing with uncontrolled sources such as available bit rate (ABR) traffic.

Finally, the performability variable Y_T considered in this study is an instance of a general class of variables which can be defined and evaluated in the same manner. Accordingly, with respect to either the current model or any of the extensions suggested above, other aspects of how CAC is affected by a randomly varying resource can likewise be investigated.

ACKNOWLEDGMENTS

The CAC algorithm [5] evaluated herein resulted from joint work with Sergio Montagna, Roberto Paglino, and Alberto Puglisi during the author's 1996-1997 sabbatical leave at Italtel's Central Research Laboratory in Milan, Italy. Their valuable contributions, together with the day-by-day enjoyment that accompanied this collaboration, are gratefully acknowledged.

REFERENCES

[1] *Performance Evaluation and Design of Multiservice Networks, Final Report of the COST 224 Project,* J.W. Roberts, ed., Commission of the European Communities, Luxembourg, 1992.
[2] J.Y. Hui, "Resource Allocation for Broadband Networks," *IEEE J. Selected Areas in Comm.,* pp. 1598-1608, June 1988.
[3] F.P. Kelly, "Effective Bandwidth at Multi-Class Queues," *Queueing Systems,* pp. 5-16, Sept. 1991.
[4] A.I. Elwalid, D. Mitra, and R.H. Wentworth, "A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node," *IEEE J. Selected Areas in Comm.,* pp. 1115-1127, Aug. 1995.
[5] J.F. Meyer, S. Montagna, R. Paglino, and A. Puglisi, "Admission Control of Mixed VBR Sources in Broadband Networks," int'l patent application no. PCT/IT 98/00373, 1998.
[6] J.F. Meyer, "Performability: A Retrospective and Some Pointers to the Future," *Performance Evaluation,* vol. 14, nos. 3-4, pp. 139-156, Feb. 1992.
[7] J.F. Meyer, "Performability Evaluation: Where It Is and What Lies Ahead," *Proc. 1995 IEEE Int'l Computer Performance and Dependability Symp.,* pp. 334-343, Apr. 1995.
[8] W.H. Sanders et al., "The UltraSAN Modeling Environment," *Performance Evaluation,* vol. 24, no. 1, pp. 89-115, Oct.-Nov. 1995.
[9] J.F. Meyer, "Performability of Algorithms for Connection Admission Control," *Proc. Int'l Conf. Dependable Systems and Networks,* pp. 181-190, June 2000.

John F. Meyer (M'60-SM'71-F'85) received the BS degree from the University of Michigan in 1957, the MS degree from Stanford University in 1958, and the PhD degree from the University of Michigan in 1967. He is currently a professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. He has been active in computer and communication systems research for more than 40 years and has published widely in the areas of fault-tolerant computing and model-based evaluation of system performance, dependability, and performability. He joined the Michigan faculty in 1967 and, in addition to his university appointment, has held visiting research positions at laboratories in the US, France, Italy, Japan, and Sweden. Prior to 1967, he was a research engineer at the California Institute of Technology Jet Propulsion Laboratory, where his contributions included the first patent issued to the National Aeronautics and Space Administration.

Dr. Meyer is a fellow of the IEEE, a Golden Core Member of the IEEE Computer Society, and a Silver Core Member of IFIP for his service to Working Group 10.4 (Dependable Computing and Fault Tolerance). He is also a member of Eta Kappa Nu, Phi Kappa Phi, Sigma Xi, and Tau Beta Pi. At the IEEE Computer Society, he has served as chair of the Technical Committee on Fault-Tolerant Computing, a member of the Society's Governing Board, Publications Board, Technical Activities Board, and a guest editor of the *IEEE Transactions on Computers*.

▷ **For further information on this or any computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.