Admission Control and Routing in ATM Networks using Inferences from Measured Buffer Occupancy

C. Courcoubetis, G. Kesidis, *Member, IEEE*, A. Ridder, J. Walrand, *Fellow, IEEE*, and R. Weber

***Abstract*— We address the issue of call acceptance and routing in ATM networks. Our goal is to design an algorithm that guarantees bounds on the fraction of cells lost by a call. The method we propose for call acceptance and routing does not require models describing the traffic. Each switch estimates the additional fraction of cells that would be lost if new calls were routed through the switch. The routing algorithm uses these estimates. The estimates are obtained by monitoring the switch operations and extrapolating to the situation where more calls are routed through the switch. The extrapolation is justified by a scaling property. To reduce the variance of the estimates, the switches calculate the cell loss that would occur with virtual buffers. A way to choose the sizes of the virtual buffers in order to minimize the variance is discussed. Thus, the switches constantly estimate their spare capacity. Simulations were performed using Markov fluid sources to test the validity of our approach.**

I. Introduction

ASYNCHRONOUS transfer mode (ATM) is a form of packet switching that is proposed for broadband networks. In ATM, data is divided into 53 byte cells that are multiplexed on a time-slotted channel. When network traffic is bursty, ATM's use of statistical multiplexing results in an efficient use of bandwidth [8]. ATM uses virtual circuits (VCs). Every cell of a call will use the same route. When a cell arrives at a switch, the switch determines its output link by looking at the VC number in the header of the cell, and using a lookup table in the switch's memory. The VC number of every call and the lookup tables of every switch are determined by the routing algorithm [14].

Calls share buffers in switches. The method of call acceptance described in this paper can be used with various switch architectures (e.g. output-buffer, shared-buffer, Batcher-banyan). If, for example, output buffer switches are used, each output link of a switch has an associated buffer. When the traffic offered to a link exceeds the link's capacity, cells begin to accumulate in the buffer. When a cell arrives at a full buffer, it is lost. Since cell losses are rare and delays are small, the statistics of a call does not significantly change along its virtual circuit. Also, we assume that no feedback congestion control method, such as windowing, is employed (such methods may not be feasible in ATM because of the large bandwidth$\times$ delay products). Therefore we assume that calls of the same type (e.g., video, speech, etc.) produce input traffic streams with identical statistics at every buffer through the network. Traditional quality-of-service requirements for a call accepted through the network require that its cell loss rate be less than a predefined amount that depends on the type of service provided by the call.

We consider calls that use a non-negligible fraction of the link bandwidth, such as video calls or bundles of voice or data transfer calls. Thus, the number of calls is small compared to the buffer size. In our analysis, we therefore consider asymptotics as the buffer size becomes large and the number of calls is fixed.

Our goal is to design an algorithm for call acceptance and routing that guarantees bounds on the fraction of cells lost by the calls because of buffer overflows. The method we propose does not require models describing the statistics of the traffic. This contrasts with algorithms based on parametric models that attempt to estimate the parameters from the traffic. We choose the former approach because realistic models may be complex and slow to fit. We make an analogy with direct vs. indirect adaptive control. In indirect adaptive control, a parametric model is first fitted to the observed traffic. The optimal policy for the estimated parameters is then used. In the direct approach, the quantity to be optimized is measured. The control actions are selected to optimize future values of this quantity.

Thus, by monitoring its buffer occupancy, each switch constantly estimates its spare capacity to accept new calls. The algorithm then accepts and routes calls by using these estimates. To estimate its spare capacity, the switch has to evaluate the value that the cell loss probability would have if more calls were using that switch. Since the loss probability is very small, estimators based on fractions of lost cells have a very large variance and are therefore very slow. To reduce the variance, we estimate the value that the cell loss probability would have if the switch buffers could store fewer cells. That is, we keep track of a virtual buffer occupancy corresponding to a smaller buffer capacity. To relate the statistics of cell losses of this smaller buffer to those of the actual buffer, we use results on the shape of the loss probability as a function of the buffer size. These shape results are derived using the theory of large deviations.

The paper is organized as follows. Section 2 describes the call acceptance and routing algorithm in some detail.

Paper approved by Debasis Mitra, the Editor for Teletraffic Analysis and Data Networks of the IEEE Communications Society. Manuscript received November 18, 1991; revised: March 28, 1992; August 4, 1992.

C. Courcoubetis is with the Dept of Computer Science, University of Crete, and Institute of Computer Science, FORTH, Greece

G. Kesidis was with the EECS Dept, Univ. of California at Berkeley. He is now with the E&CE Dept, Univ. of Waterloo, Waterloo, ON, Canada, N2L 3G1

A. Ridder was with the EECS Dept, Univ. of California at Berkeley. He is now with the Dept of Econometrics, Free University Amsterdam, de Boelelaan 1105, 1081 HV Amsterdam, the Netherlands

J. Walrand is with the EECS Dept, Univ. of California, Berkeley, CA94720

R. Weber is with Queen's College, University of Cambridge, Cambridge, U.K.

IEEE Log Number 9411644.

0090-6778/95$4.00

In section 3 we assume the input sources are of the same type, and prove a predictive scaling property of the probability of buffer overflow in a busy cycle. We relate this quantity to the fraction of cells lost. In section 4, we describe two methods of variance reduction. Our simulations are described in section 5. Finally, in section 6, we discuss a way to handle multiple types of calls with this method and draw conclusions.

II. Monitor to Infer Network Overflow Statistics (MINOS)

We describe an algorithm which can be used by a switch to predict its spare capacity. This algorithm is a Monitor to Infer Network Overflow Statistics (MINOS). Another sample path estimation method for queuing networks was described in [7].

As explained above, we want to estimate the loss probability in the switch buffers. Consider a given buffer of size B cells shared by $N > 0$ virtual circuits, and served by a fiber with transmission rate c cells/s. Let $F(N, B, c)$ be the fraction of cells lost due to buffer overflows. We want to estimate the fraction $F(N(1+\epsilon), B, c)$ of cells lost when a fraction ϵ more calls are added. In section 3, we show that the probability of buffer overflow in a busy cycle, Φ, has the following property: for large B,

$$\Phi(N(1+\epsilon), B, c) \quad \approx \quad \Phi(N, B, \frac{c}{1+\epsilon}). \qquad (1)$$

By expressing F in terms of Φ we conclude that F has this property as well. Thus, we can estimate F when a fraction ϵ more calls are added by estimating F with the current number of calls, N, and the service rate reduced by the same fraction. To estimate $F(N(1+\epsilon), B, c)$, a device is added to the switch that keeps track of the buffer occupancy, $Y(t)$, when the service rate is $\frac{c}{1+\epsilon}$. Specifically, when a cell arrives at the buffer, $Y(t)$ is incremented by one. Also, $Y(t)$ is decremented by one every $\frac{1+\epsilon}{c}$ seconds when $Y(t) > 0$. This function could be realized by a chip implementing the above algorithm. The problem now is to estimate $F(N, B, \frac{c}{1+\epsilon})$ by monitoring the buffer. Because this loss probability is very small (typically about 10^{-9}), a direct estimator based on the fraction of lost cells has a very large variance.

To improve the estimator, the device will estimate the losses for smaller buffers (called virtual buffers) so as to increase the frequency of buffer overflows, and therefore speed-up the collection of "important" samples. There is a tradeoff in choosing the size of the virtual buffers. If these virtual buffers are still too large, our estimates will be too slow. However, if these virtual buffers are too small, the original system system is over-distorted and we have a large error when we extrapolate back to B. Let B/k be the size of a virtual buffer for some $k > 1$.

The virtual buffer estimate, $F(N, \frac{B}{k}, \frac{c}{1+\epsilon})$, is related to $F(N, B, \frac{c}{1+\epsilon})$, in the following way. We can show (next section) that F has the following form:

$$F(N, B, \frac{c}{1+\epsilon}) \quad = \quad \exp(-BI(N, \frac{c}{1+\epsilon}) + o(B)).$$

Assuming $e^{o(B)}$ has the form $AB^{-\xi}$, we obtain estimates of

$$F(N, \frac{B}{k}, \frac{c}{1+\epsilon}) \quad = \quad A\left(\frac{B}{k}\right)^{-\xi} \exp\left(-\frac{B}{k} I(N, \frac{c}{1+\epsilon})\right).$$

Because we have three unknowns (A, ξ, I), we will carry out this estimate for three values of k: $k_0 > k_1 > k_2 > 1$. These three equations can be solved for A, ξ and $I(N, \frac{c}{1+\epsilon})$. We can then plug in A, ξ and $I(N, \frac{c}{1+\epsilon})$ into the expression for $F(N, B, \frac{c}{1+\epsilon})$, and thus compute the desired quantity $F(N(1+\epsilon), B, c) \approx F(N, B, \frac{c}{1+\epsilon})$.

To summarize the above, the estimation algorithm in the device keeps track of three "virtual buffer" occupancy processes with buffers of size B/k_i, $i = 1, 2, 3$, and service rate $\frac{c}{1+\epsilon}$. Note that these computations can be done in parallel with the normal operation of the switch so that the estimates of $F(N, \frac{B}{k_i}, \frac{c}{1+\epsilon})$ are constantly available to the routing algorithm. In section 4 we describe another way to reduce the variance of an estimate of F.

The difference between MINOS and change-of-measure methods ([5],[12]) is that it is impossible to alter the traffic parameters in the problem addressed here. That is why we resort to an estimation method that monitors the actual traffic.

Let us now describe how the routing algorithm can use the above estimates. For simplicity, we assume that all the virtual circuits carry calls of the same type. The case of different traffic types is discussed in section 6. Denote by $F_n = F(N_n, B_n, c_n)$ the current fraction of cells lost at buffer n, for all buffers n in the network. Assuming a first-come-first-served queuing discipline in each buffer, F_n is the fraction of cells lost at buffer n by *each* call that uses buffer n. If call i uses buffers $1, 2, ..., m$, the fraction of cells lost by that call is $1 - \prod_{n=1}^{m}(1 - F_n) \approx \sum_{n=1}^{m} F_n$.

Now say we are trying to route a new call. Using the above method, buffer n estimates $F'_n = F(N_n(1+\epsilon), B_n, c_n)$. We attempt to find a path for the new call that satisfies

$$G^{\text{new}} \quad \geq \quad \sum_{n \in \text{path}} F'_n$$

where G^{new} is the fraction of lost cells acceptable to the new call. Moreover, the router must ensure that, by choosing a particular path for the new call, the above constraint is not violated for any existing, previously routed call i (with guarantee G^i) which uses all or part of that path. If no path is found that satisfies these constraints, the new call is refused.

The routing policy just described is myopic. More sophisticated strategies should be investigated. For instance, given statistics about the generation of new calls, one can formulate a dynamic programming problem which is solved by the optimal policy. We are currently exploring such extensions.

III. A Predictive Scaling Property

We start by heuristically deriving an expression for $\Phi(N,B,c)$ for general source models. Consider a buffer of size B with service rate c shared by N i.i.d., stationary and ergodic sources. For all M greater than the average rate of cells produced by a source, assume that the probability that a source produces MT cells over a period of time of length T is approximately $\exp(-TH(M))$ where H is strictly convex and non-negative. This assumption is motivated by large deviations results for empirical distributions of Markov chains[3]. We will give the expression for H for Markov fluid models and thereby derive the asymptotic results in [1] in a completely different way. The expression for H for a large class of stationary and ergodic discrete-time sources is given in [10].

By independence, the probability that, for $j = 1, ..., N$, the j^{th} source produces $\mu_j T$ cells over time T is about

$$\exp\left(-T\sum_{j=1}^{N} H(\mu_j)\right).$$

Consequently, the probability that all sources produce a total of NMT cells over large time T is about

$$\sum_{\mu:\sum\mu_j=NM} \exp\left(-T\sum_{j=1}^{N} H(\mu_j)\right)$$

where $\mu = (\mu_1, ..., \mu_N)$. Indeed, each choice of μ such that $\sum \mu_j = NM$ is one particular way for NMT cells to get produced. This sum of exponentials can be approximated by the largest term (originally an argument of Laplace):

$$\begin{aligned}\sum_{\mu:\sum\mu_j=NM} &\exp\left(-T\sum_{j=1}^{N} H(\mu_j)\right)\\ &\approx \exp\left(-\inf_{\mu:\sum\mu_j=NM} T\sum_{j=1}^{N} H(\mu_j)\right)\\ &= \exp(-TNH(M))\end{aligned}$$

where the last equality is due to the convexity of H.

Thus, the probability that, starting from an empty buffer, the sources produce cells at rate NM until the buffer overflows is

$$\exp\left(-B\frac{NH(M)}{NM-c}\right).$$

Indeed $T = B/(NM-c)$ is the time the buffer occupancy takes to reach B when the aggregate cell arrival rate is NM. By the argument of Laplace,

$$\Phi(N,B,c) \approx \exp\left(-B\inf_{M>\frac{c}{N}}\frac{NH(M)}{NM-c}\right) \quad (2)$$

$$= \Phi(1,B,\frac{c}{N}) \quad (3)$$

We now show the relationship between cell loss probability and the probability of buffer overflow in a busy cycle. Assuming that the processes are stationary and ergodic, we can express F in terms of the average number D of cells lost in an overflowing cycle, the average number C of cells arriving in a cycle, and Φ:

$$F = \frac{\Phi D}{C}. \quad (4)$$

In finding an expression for C we ignore the buffer size B because the number of cells arriving in an overflowing cycle is of order B and the probability of an overflowing cycle is of order $\exp(-BI + o(B))$. This gives us a contribution to C of approximately $\exp(-BI+o(B))$ which is negligible for large B. In finding an expression for D we assume that the typical value of the time-derivative of the buffer occupancy $\dot{Y}$ when an overflow occurs is not a function of B; indeed, that typical value is

$$\hat{M} := \arg\inf_{M>\frac{c}{N}}\frac{NH(M)}{NM-c}.$$

Also, we assume a negligible number of cells will be lost after $\dot{Y}$ returns to zero [2]. Under these assumptions, D will have a negligible dependence on B as well (for large B). Thus $D/C = \exp(o(B))$ which implies $F = \Phi\exp(o(B))$.

A. Markov Fluid Example

A source of a buffer is called a Markov fluid if its time-derivative is a continuous-time Markov chain on a finite state space. If the arrival process to a buffer with deterministic service rate is a superposition of independent Markov fluids, then the buffer occupancy has piecewise-linear trajectories with random slopes.

Each Markov fluid source has state space $\Lambda = (\Lambda_1, ..., \Lambda_m)$ and has Markov time-derivative with transition rate matrix Q. We assume $\Lambda_j < \Lambda_{j+1} < \infty$ for all j. Let π be the invariant of Q ($\pi Q = 0$) and let $\overline{\Lambda} = \sum \pi_i \Lambda_i$ be the average arrival rate.

To define H, let J_Q be the large deviations action functional for the empirical distribution of a continuous-time Markov chain with transition rate matrix Q. Take

$$H(M) := \inf_{[\mu,\Lambda]=M} J_Q(\mu) \quad (5)$$

where the infimum is taken over the space Σ_m of distributions μ on Λ. Note that the convexity of J_Q on Σ_m implies that H is convex on (Λ_1, Λ_m). For completeness, we give the following expression for J_Q For $\mu \in \Sigma_m$,

$$J_Q(\mu) = \inf_{P\,:\,\mu P=0} G(P;Q)$$

where the infimum is taken over the space of transition rate matrices on Λ, G is the relative entropy rate between continuous-time Markov chains [9],

$$G(P;Q) := \sum_{i=1}^{m}\mu_i \sum_{j=1,j\neq i}^{m}\left(P_{i,j}\log\frac{P_{i,j}}{Q_{i,j}} + Q_{i,j} - P_{i,j}\right),$$

and μ is the invariant of P ($\mu P = 0$). This definition of J_Q is different but consistent–in the sense of the contraction mapping principle [11]–with that in [6] (see "level 2.5" large deviations in [9]).

If the Markov fluids are all of the on-off ($m = 2$) type, then (see equation (5))

$$H(M) = \frac{1}{\Lambda_2 - \Lambda_1}\left(\sqrt{q_1(\Lambda_2 - M)} - \sqrt{q_2(M - \Lambda_1)}\right)^2$$

where $q_1 = Q_{1,2}$ and $q_2 = Q_{2,1}$. We now use equations (2) and (4) to get an expression for $F(N, B, c) = \exp(-BI(N, c) + o(B))$. By direct calculation,

$$\begin{aligned} I(N,c) &:= \inf_{M > \frac{c}{N}} \frac{NH(M)}{NM - c} \\ &= \frac{(q_1 + q_2)(\frac{c}{N} - \overline{\Lambda})}{(\frac{c}{N} - \Lambda_1)(\Lambda_2 - \frac{c}{N})} \end{aligned}$$

and the minimizing $M > c/N$ is

$$\hat{M} = \frac{q_1\Lambda_1(\Lambda_2 - \frac{c}{N})^2 + q_2\Lambda_2(\frac{c}{N} - \Lambda_1)^2}{q_1(\Lambda_2 - \frac{c}{N})^2 + q_2(\frac{c}{N} - \Lambda_1)^2}.$$

Anick et al determined that $\exp(-BI(N, c) + o(B))$ is the stationary probability that the buffer occupancy exceeds B (for an infinite buffer) in [1]. In [4], it was shown that $F(1, B, c) = \exp(-BI(1, c) + o(B))$ using a first-step argument for on-off Markov fluids.

IV. Variance Reduction

Because we assume the buffer sources are stationary and ergodic, given enough time we can use the two variance reduction methods described below to obtain a good estimate of $\Phi(N, B, c/(1+\epsilon))$ with high probability. We attempt to analyze the performance of the first method by considering the sample standard deviation of our estimates.

Note that if a large deviation in the behaviour of the arriving traffic occurs, a poor estimate of $\Phi(N, B, c/(1+\epsilon))$ would result, and this could lead to an erroneous admission decision which, in turn, could result in excess loss or delay. This is a common problem of estimation methods of random quantities. The problem of how much time is required to obtain an accurate estimate $\Phi(N, B, c/(1+\epsilon))$ with high probability is addressed in the "small time" simulations of the next section.

A. Virtual Buffers

Recall that we proposed using virtual buffers to increase the frequency of "important" samples (buffer overflows) in order to reduce the variance of the estimate of F. The virtual buffers have sizes B/k_i, $i = 0, 1, 2$, with $k_0 > k_1 > k_2 > 1$. We will now explain a way to express the quantity we want to estimate

$$F \equiv F(N(1+\epsilon), B, c) \approx F(N, B, \frac{c}{1+\epsilon}) \approx AB^{-\xi}e^{-BI}$$

in terms of the virtual buffer estimates

$$F_i \equiv F(N, \frac{B}{k_i}, \frac{c}{1+\epsilon}) \approx A\left(\frac{B}{k_i}\right)^{-\xi}\exp(-\frac{B}{k_i}I)$$

$i = 0, 1, 2$. Let $a_{i,j} = \frac{1}{k_i} - \frac{1}{k_j}$, $e_0 = \frac{-k_0(k_2-1)}{k_0-k_2}$, $e_2 = \frac{k_2(k_0-1)}{k_0-k_2}$, and $\gamma = \log(k_0^{e_0}k_2^{e_2})/\log(k_0^{a_{1,2}}k_1^{a_{2,0}}k_2^{a_{0,1}})$. Solving for the three unknowns, A, ξ, and I, in terms of the F_i, and substituting into the expression for F we get,

$$\log F = l_0 \log F_0 + l_1 \log F_1 + l_2 \log F_2$$

where $l_0 = e_0 + \gamma a_{2,1}$, $l_1 = \gamma a_{0,2}$, and $l_2 = e_2 + \gamma a_{1,0}$.

B. Analysis of Variance Reduction using Virtual Buffers

In this section, we estimate the variance reduction achieved by using the virtual buffers. For simplicity we take $\xi = 0$ and consider the variance reduction achieved by using two virtual buffers (instead of three) to estimate Φ (instead of F). Thus, we estimate

$$\Phi \equiv \Phi(N(1+\epsilon), B, c) \approx \Phi(N, B, \tfrac{c}{1+\epsilon}) \approx Ae^{-BI}$$

from the virtual buffer estimates

$$\Phi_i \equiv \Phi(N, \frac{B}{k_i}, \frac{c}{1+\epsilon}) \approx A\exp(-\frac{B}{k_i}I)$$

$i = 0, 2$. Substituting for A and I we get $\Phi = \Phi_0^{e_0}\Phi_2^{e_2}$.

Assume the time n (measured in busy cycles) to estimate the Φ_i's is fixed and is the same for both virtual buffers. Let σ_i be the standard deviation of the estimate of Φ_i so that $\sigma_i = \sqrt{\Phi_i(1-\Phi_i)/n} \approx \sqrt{\Phi_i/n}$, $i = 0, 2$. Thus, the relative error of the estimate of Φ, σ/Φ, satisfies

$$\frac{\sigma}{\Phi} \le -\frac{\sigma_0}{\Phi_0}e_0 + \frac{\sigma_2}{\Phi_2}e_2 =: f(k_0, k_2)$$

for σ_i sufficiently small. Note that $e_0 < 0$ and f is an upper bound for σ/Φ because we have ignored the fact that the Φ_i are positively correlated.

Minimizing f over (k_0, k_2) we get that the optimal k_0 is very large and the optimal k_2 minimizes $g(k) \equiv (k-1)\sqrt{1-A} + k\sqrt{\exp(BI/k) - A}$. Let n_k and n_{TA} be the number of cycles required to achieve $\epsilon \times 100\%$ relative error with 95% confidence [13] using two virtual buffers and direct time averaging respectively. A simple computation yields: $n_k/n_{TA} = g^2(k_2)\exp(-BI)$. In our simulations, we found $A << 1$ (which implies the optimal $k_2 \approx 0.4BI$), and $BI \approx 8$, so that $n_k/n_{TA} \approx 1/17$. The speed up factor is actually larger than 17 because $\sigma/\Phi < f$; using sample standard deviations, we found a speed up was about 100. Unfortunately, fixing $\xi = 0$ results in estimates of Φ that are consistently one order of magnitude too small. These calculations give us a rule of thumb for choosing the k_i, $i = 0, 1, 2$: choose k_0 large and k_2 small.

C. Variance Reduction using the Kullback-Leibler Distance

We now describe a faster method for estimating the probability of buffer overflow in a cycle. This method is useful when the estimation has to be performed very quickly, on the basis of few observations. The main point of this section is that estimators that improve upon those based on virtual buffers are possible. Instead of using three virtual buffers etc., we monitor the *peak buffer occupancy* in every cycle (call it Z_i for the i^{th} cycle). Let n be a given number of cycles, and $B^* = B/k_0$. For integers $b \geq B^* - 1$, define the empirical tail distribution of Z_i:

$$p(b) = \frac{1}{n} \times \begin{cases} \sum_{i=1}^{n} 1\{Z_i < B^*\} & \text{if } b = B^* - 1 \\ \sum_{i=1}^{n} 1\{Z_i = b\} & \text{if } b \geq B^*. \end{cases}$$

Also define

$$\phi(A, I, b) = \begin{cases} 1 - A\exp(-B^* I) & \text{if } b = B^* - 1 \\ A\exp(-bI) - A\exp(-(b+1)I) & \text{if } b \geq B^* \end{cases}$$

where we have taken $\xi = 0$. The Kullback-Leibler distance [3] between ϕ and p is

$$K(A, I) = \sum_{b=B^*-1}^{\infty} p(b) \log\left(\frac{p(b)}{\phi(A, I, b)}\right).$$

The values of A and I that minimize K are given by

$$I = \log\left(1 + \frac{1 - p(B^* - 1)}{\sum_{b=B^*}^{\infty} bp(b) - B^*(1 - p(B^* - 1))}\right) \text{ and}$$
$$A = (1 - p(B^* - 1))\exp(B^* I).$$

These expressions for A and I can be easily updated at the end of every cycle. Taking $\xi \neq 0$ so that K is a function of three parameters, we found no simple closed form solution to $\partial K/\partial\xi = \partial K/\partial I = \partial K/\partial A = 0$.

V. Simulations

The objective of the simulation experiments is to validate the methods used for estimating the spare capacity of switches. Recall that these methods are based on asymptotic results derived from the theory of large deviations. Specifically, we use two results: the exponentiality of the tail of the overflow probability distribution and the asymptotic scaling property which states that the loss probability is a function of the ratio of the number of calls over the transmission rate. Since these results are valid only in the limit, as the loss probability becomes small, it is important to verify that they are usable in the range of values of interest. The simulation experiments are used for verifying that validity.

The simulation experiments show that the asymptotic formulas introduce only a small error when used with realistic loss probabilities.

We conducted simulations using the on-off Markov fluid model described above for the sources to the buffer. The following values were chosen for the parameters: $B = 1800$ cells, $\Lambda_0 = 0$, $\Lambda_1 = 2500$ cells/s, $c = 15000$ cells/s (≈ 6 Mbps) and

$$Q_r = \begin{bmatrix} -10 & 10 \\ 20 & -20 \end{bmatrix}.$$

The number of sources, N, was varied. We took $\epsilon = 1/N$ and $N + 1 \in \{13, 14, .., 17\}$. The above parameters were chosen so that we would be simulating on-off sources with burst rates in the Mbps range (1 cell = 53 bytes). We observed that the fraction of overflowing cycles among non-"trivial" busy cycles was small: a trivial busy cycle being one in which a cell arrives to an empty buffer and leaves before the next cell arrives ($-c < \dot{Y} < 0$). The values of N were chosen in the above range to simulate heavy traffic conditions: $\rho(\frac{N}{c}) = \frac{N}{c}\pi_1\Lambda_1 = \frac{N}{18}$ where $\pi = (\pi_0, \pi_1) = (2/3, 1/3)$.

The first simulation checks the $N - c$ scaling property (equation (1)) for a finite B when a large amount of time is available to accurately estimate the F_i, $i = 1, 2, 3$. For $N + 1 \in 13, ..., 17$, we measured $F(N+1, B, c)$ using direct sample averaging (we stopped our simulation when the 95% confidence interval estimate of $F(N+1, B, c)$ was less than 0.3 [13]). Using three virtual buffers, we estimated $F(N+1, B, c)$ from estimates of $F_i = F(N, \frac{B}{k_i}, \frac{c}{1+N^{-1}})$ using the formulas above with $\epsilon = 1/N$ (we stopped our simulation when the 95% confidence interval estimate of $F(N, \frac{B}{k_i}, \frac{c}{1+N^{-1}})$ was less than 10%). We used two different sets of three k_i: (20,15,9) and (15,12,9). The simulation results are given in Table 1. That is, $\hat{F}$ is the measured value of $F(N+1, B, c)$, F^a is obtained by using three virtual buffers with $(N, c/(1+N^{-1}))$ and $k = (20, 15, 9)$, and F^b is obtained by using three virtual buffers with $(N, c/(1+N^{-1}))$ and $k = (15, 12, 9)$. Thus, F^a and F^b are both well within an order of magnitude of $\hat{F}$.

TABLE I
Large Time, Two-Rate Sources.

$N+1$	17	16	15	14	13
$-\log_{10}\hat{F}$	1.64	2.45	3.40	4.60	5.92
$-\log_{10}F^a$	1.84	2.52	3.42	4.48	5.74
$-\log_{10}F^b$	1.76	2.64	3.44	4.44	5.63

In the second simulation, we fixed $N + 1 = 13$ so that $\hat{F} := F(N+1, B, c) \approx 2 \times 10^{-6}$ and $\Phi(N+1, B, c) \approx 3 \times 10^{-9}$. We ran the simulation for $n = 10^7$ busy cycles of the "actual" buffer process (size B). Since the probability of even seeing one overflow in this amount of time is 30%, an estimate of F from direct sample averages would probably be zero. Table 2 shows the performance of the estimator of F using three virtual buffers for two sets of k_i: $k = (40, 20, 10)$ for F^a and $k = (20, 15, 10)$ for F^b. The sample standard deviation was less than the estimate in every trial. F^a and F^b are both well within an order of magnitude of $\hat{F}$.

TABLE II
Small Time, Two-Rate Sources, $\hat{F} = 2 \times 10^{-6}$.

TRIAL	1	2	3	4	5	6
$F^a \times 10^6$	3.45	1.75	3.44	2.16	2.01	2.98
$F^b \times 10^6$	2.10	2.13	3.11	2.02	2.31	2.54

These two simulations were repeated using four-rate Markov fluids with the following parameters chosen to have the traffic characteristics described above: $B = 1800$ cells, $\Lambda_i \in \{0, 2000, 3000, 4000\}$ cells/s, $c = 15000$ cells/s,

$$Q_r = \begin{bmatrix} -10 & 10 & 0 & 0 \\ 20 & -30 & 10 & 0 \\ 0 & 30 & -40 & 10 \\ 0 & 0 & 40 & -40 \end{bmatrix},$$

$N+1 \in \{9, 10, 11, 12\}$, and two different sets of three k_i: (30,15,5) for F^a and (20,15,5) for F^b for both small and large-time simulations. For the small-time simulation, we fixed $N+1 = 9$ and the simulation time $n = 5 \times 10^7$ cycles. Since the measured value of $\Phi(N+1, B, c) = 6 \times 10^{-9}$, we get that the probability of seeing an overflow in one n cycle trial is 30% as above. The results are shown in Tables 3 and 4 (as in Tables 1 and 2 respectively). Note that, F^a and F^b are both well within an order of magnitude of $\hat{F} = F(N+1, B, c)$ in both tables.

TABLE III
LARGE TIME, FOUR-RATE SOURCES.

$N+1$	12	11	10	9
$-\log_{10} \hat{F}$	3.01	3.88	5.11	6.74
$-\log_{10} F^a$	3.02	3.87	5.14	6.76
$-\log_{10} F^b$	3.62	4.28	5.00	6.55

TABLE IV
SMALL TIME, FOUR-RATE SOURCES, $\hat{F} = 2 \times 10^{-7}$.

TRIAL	1	2	3	4	5	6
$F^a \times 10^7$	2.77	2.98	4.66	1.50	2.90	2.13
$F^b \times 10^7$	2.63	2.78	4.22	1.56	2.32	2.01

We repeated the simulation with on-off Markov fluid sources to estimate Φ using the method based on minimizing the Kullback-Leibler distance. We fixed $N+1 = 15$ so that $\Phi(N+1, B, c) \approx 10^{-6}$ and ran the simulation for $n = 10^5$ busy cycles. Φ^a was obtained by using $B^* = B/15$ and $(N, c/(1+N^{-1}))$. In the six trials of this simulation (results shown Table 5), one overflow was observed in the actual buffer. The estimates are optimistic but are within an order of magnitude of the measured value $\Phi(N+1, B, c) = 1.0 \times 10^{-6}$. Also, the sample standard deviation was less than the estimate in every trial. For the same amount of time, the three virtual buffers estimate was very noisy.

When estimating Φ, trivial busy cycles were counted. When these cycles were not counted, the simulations yielded good results anyway; the values of Φ were, of course, much higher in this case. To show that the loss probability is very sensitive to the source and server burstiness, we consider an M/M/1 queue with the same traffic intensity $\rho = 15\pi_1\Lambda_1/c = 15/18$. Approximating Φ by the stationary probability that this M/M/1 queue exceeds B, we get $\Phi(N+1, B, c) \approx \rho^B \approx 10^{-140}$–an extremely poor estimate.

TABLE V
SMALL TIME, TWO-RATE SOURCES, $\hat{\Phi} = 1.0 \times 10^{-6}$.

TRIAL	1	2	3	4	5	6
$\Phi^a \times 10^7$	3.14	6.32	3.32	3.36	1.09	6.34

VI. Discussion and Conclusions

The above method can be used to handle multiple types of calls sharing a buffer. Say there are six voice calls (same type) and two video calls currently using the buffer, and we wish to estimate the effect of adding a video call. Define a new type of call that is the sum of three voice calls and one video call. Therefore there are two calls of the new type currently using the buffer. Instead of estimating the fraction of cells lost when another video call is added, we estimate F when another call of the new type is added. This, of course, may be a very conservative estimate of the affect of another video call on the buffer.

In order to estimate the number of Mips required by one virtual buffer to estimate F_i, we let the peak arrival rate into the buffer be $p \times c$ cells/s. The worst case occurs during cell loss when we have to handle the buffer occupancy and perform a comparison every $(pc+c)^{-1}$ seconds, and update the cells lost and cells arrived counters every $(pc)^{-1}$ seconds. Thus we require $2(pc+c) + 2pc = (4p+2)c$ Mips. For $c = 3.5 \times 10^5$ cells/s (150 Mbps) and $p = 5$ we get 7.7 Mips required by one virtual buffer.

In summary, we have described an algorithm for estimating the spare capacity of switches. This method monitors the traffic in a switch buffer and makes quick and direct estimates of the effect of routing more calls through that buffer on the fraction of cells lost in that buffer. The method can be used by a call acceptance and routing algorithm. The method is robust: it has been shown, in principle, to work under weak assumptions. Finally, simulations were conducted which demonstrated the predictive property of the algorithm as well as the significant variance reduction with finite buffer size.

Since many idealizations were made above, experiments on actual networks are clearly required. Moreover, a more exhaustive simulation study using more realistic ATM traffic sources should be conducted.

Acknowledgements

We are grateful to the reviewers for their constructive comments.

References

[1] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell Sys. Tech. J.*, Vol. 61, No.8:pp. 1871–1894, 1982.

[2] A. Bhargava and M.G. Hluchyj. Frame losses due to buffer overflows in fast packet networks. *Proc. IEEE INFOCOM*, Vol. 1:pp. 132–139, 1990.

[3] J.A. Bucklew. *Large Deviation Techniques in Decision, Simulation and Estimation.* John Wiley and Sons, Inc., New York, NY, 1990.

[4] M. Butto, E. Cavallero, and A. Tonietti. Effectiveness of the leaky bucket policing mechanism in ATM networks. *IEEE JSAC*, Vol. 9, No. 3:pp. 335–342, April 1991.

[5] M. Cottrell, J-C. Fort, and G. Malgouyres. Large deviations and rare events in the study of stochastic algorithms. *IEEE Trans. Auto. Control*, Vol. 28, No. 9:pp. 907–920, 1983.

[6] J-D. Deuschel and D. W. Stroock. *Large Deviations.* Academic Press, New York, NY, 1989.

[7] Y.C. Ho and C. Cassandras. A new approach to the analysis of discrete event dynamic systems. *Automatica*, Vol. 19, No. 2:pp. 149–167, 1983.

[8] J.Y. Hui. Network, transport, and switching integration for broadband communications. *IEEE Network*, pp. 40–51, March 1988.

[9] G. Kesidis and J. Walrand. Relative entropy between Markov transition rate matrices. *IEEE Trans. Info. Th.*, Vol. 39, No. 3:pp. 1056,1057, May 1993.

[10] G. Kesidis, J. Walrand, and C.-S. Chang. Effective bandwidths for multiclass Markov fluids and other ATM sources. *IEEE/ACM Trans. Networking*, Vol. 1, No. 4:pp. 424–428, Aug. 1993.

[11] J. Lynch and J. Sethuraman. Large deviations for processes with independent increments. *Annals of Probability*, Vol. 15, No. 2:pp. 610–627, 1987.

[12] S. Parekh and J. Walrand. A quick simulation method for excessive backlogs in networks of queues. *IEEE Trans. Auto. Control*, Vol. 34, No. 1:pp. 54–66, 1989.

[13] S. M. Ross. *A Course in Simulation.* Macmillan, New York, NY, 1990.

[14] A. S. Tanenbaum. *Computer Networks, 2nd Ed.* Prentice Hall, Englewood Cliffs, NJ, 1988.

Costas Courcoubetis was born in Athens, Greece. He received his Diploma (1977) from the National Technical University of Athens, Greece, in Electrical and Mechanical Engineering, his M.S.(1980) and Ph.D (1982) from the University of California, Berkeley, in Electrical Engineering and Computer Science. He was a Research Assistant in ERL (Electronic Research Lab) in U.C. Berkeley, an AMTS (Associate Member of Technical Staff), Summer 1981, Bell Laboratories, an Adjunct Assistant Professor in the EE department, Columbia University, Visiting Lecturer at the EECS department, U. C. Berkeley, 1986, from 1982 until 1990 Member of the Technical Staff (MTS) in the Mathematical Sciences Research Center, Bell Laboratories, Murray Hill, NJ, and since 1992 Associate Professor at the Computer Science department, University of Crete, Heraklion, Greece, and researcher at the Institute of Computer Science, FORTH. His interests are performance analysis of computer systems and high-speed (ATM) networks, network management, formal methods for the specification and verification of concurrent and real-time systems.

George Kesidis (S'91-M'92) was born in Toronto, Canada, in 1964. He received a B.A.Sc. degree from the University of Waterloo in 1988 and the M.S. and Ph.D. degrees from the University of California at Berkeley in 1990 and 1992, respectively, all in electrical engineering. He is currently an assistant professor in the electrical and computer engineering department of the University of Waterloo, Waterloo, Ontario. His research interests include resource allocation, congestion control and performance evaluation of high-speed networks.

Ad Ridder received M.Sc. in Mathematics from the University of Amsterdam, in 1983, and the Ph.D. degree in Operations Research from the University of Leiden, in 1987. From 1987 to 1989 he worked for Devtech Inc., Heerlen, the Netherlands, in 1990 he taught at the University of California at Berkeley, and in 1991 and 1992 he was an assistant professor at the Erasmus University of Rotterdam. At present, he is an associate professor of Operations Research in the department of Econometrics of the Free University of Amsterdam. His research interests are in queueing theory, inventory models and applications of genetic algorithms.

Jean Walrand is a professor of Electrical Engineering and Computer Sciences at the University of California at Berkeley. He is the author of "An introduction to queueing networks" (Prentice-Hall, 1988) and of "Communication networks: A first course" (Irwin/Aksen, 1991). He is a Fellow of the IEEE and a recipient of the Lanchester Prize from the Operations Research Society of America.

Richard Weber is the Churchill Professor for the Mathematics of Operational Research in the Statistical Laboratory at University of Cambridge. His research is in stochastic processes and optimization in queueing, scheduling, bin packing, telecommunications, manufacturing and finance. The research reported here was conducted on the first of his many visits during the past five years to the Computer Science Institute and University of Crete. These visits have resulted in further progress on the analysis of ATM switches.