Report Number: WUCS-92-09

1992

Exact and Approximate Solution of a Multiplexing Problem

Mario Bonatti, Andreas Bovopoulos, Apostolos Dailianas, and Alexei Gaivoronski

In this paper the detailed solution of the multiplexing problem of independent but not necessarily homogenous traffic sources is presented. The "curse of dimensionality" of problems of realistic dimensions is comprehensively addressed by a methodology employing state aggregation. A number of detailed examples provide further insight into the multiplexing problems and their exact and approximate solutions.

Exact and Approximate Solution of a Multiplexing Problem

Mario Bonatti, Andreas Bovopoulos, Apostolos Dailianas and Alexei Gaivoronski

WUCS-92-09

December 1992

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis MO 63130-4899

This work was supported by the National Science Foundation under grant NCR-9110183 and an industrial consortium of Bellcore, BNR, DEC, Italtel SIT, NEC, NTT and SynOptics.

Exact and Approximate Solution of a Multiplexing Problem

Mario Bonatti† Andreas Bovopoulos‡* Apostolos Dailianas‡* Alexei Gaivoronski†

Abstract

In this paper the detailed solution of the multiplexing problem of independent but not necessarily homogeneous traffic sources is presented. The "curse of dimensionality" of problems of realistic dimensions is comprehensively addressed by a methodology employing state aggregation. A number of detailed examples provide further insight into the multiplexing problems and their exact and approximate solutions.

1. Classification of Multiplexing Problems

The fundamental problem of the ATM technology stems from the fact that the performance of an ATM statistical multiplexer depends on the time behavior of the incoming traffic. As a result, without sufficient traffic control provisions, an ATM network may fail to provide grade of service (GoS) guarantees to end users. Each connection in the ATM layer results in a cell sequence (CS) that can be both analyzed and controlled at one or more of the following time scales: call, packet, burst, and cell [11, 8, 17]. At each of the time scales for which a CS control is provided, a resource allocation scheme can be introduced and classified as corresponding to the call, packet, burst, or cell time scale.

The most crucial part of an ATM network is the access segment [3]. In order to understand the multiplexing problems arising in the access segment, assume that I multiservice terminals (MST) are connected to the access segment. Each MST supports one or more services, and at the cell level, each terminal issues calls that generate "one" or multiple cell sequences (CSs). For example a digital voice conversation results in one CS, whereas a data terminal generates a family of CSs. The simplest possible traffic multiplexing problem that a multiplexer at the access segment must handle is the case in which each of the terminals generates the same CS. This is a classical multiplexing problem that has been extensively studied and is referred to henceforth as multiplexing problem A. A more complicated

†Central Research Laboratories, Italtel, SIT, Centimmo, Milanese, Milano, Italy

‡Computer & Communications Research Center, Bryan Hall 405 – Campus Box 1115, Washington University, One Brookings Drive, St. Louis MO 63130-4899

*This work was supported by the National Science Foundation under grant NCR-9110183, and an industrial consortium of Bellcore, BNR, DEC, Italtel SIT, NEC, NTT, SynOptics.

traffic multiplexing problem arises when each terminal generates one CS, while different terminals generate different CSs. Although this problem, referred to henceforth as multiplexing problem B, has recently received attention [4, 9, 14], it has not been fully addressed.

A third class of problems results when each terminal is a multiservice terminal and thus generates a family of CSs. If all terminals generate the same family of CSs, then the resulting multiplexing problem is referred to as multiplexing problem C_1. If the different multiservice terminals generate different families of CSs, then the resulting multiplexing problem is referred to henceforth as Multiplexing Problem C_2.

In this paper a comprehensive formulation of the multiplexing problem of type B is presented. Issues related to both exact as well as approximate solutions are addressed, and a number of numerical results are presented.

2. The Traffic Source Generator Model

In this section the parameters of the multiplexer are described. The multiplexer has a cell buffer of size K. Time is divided into units called slots, where the time interval corresponding to a slot equals the service time of a cell. Incoming cells are served by the multiplexer in the order of their arrival. The cell at the front of the buffer is served at the beginning of a slot. Incoming cells are stored in the cell buffer as long as the buffer is not full; if the buffer is full, cells are discarded. A number of independent, but not necessarily identical, sources are multiplexed.

The time axis is partitioned in renewal intervals. Each renewal interval is D slots long. The slot boundaries which coincide with renewal boundaries are defined as *renewal epochs*. Each of the multiplexed sources can be described as follows [5]: At the renewal epochs, the behavior of the cell source i is described by an embedded S_i-state homogeneous, irreducible and aperiodic Markov chain, with states $\{0, 1, \cdots, S_i - 1\}$. Let σ and $\hat{\sigma}$ be any two states of the embedded Markov chain for source i. Let $p^i_{\sigma\hat{\sigma}}$ be the probability that source i is in state $\hat{\sigma}$ at a renewal epoch, given that its state at the previous renewal epoch was σ. In addition let p^i_σ be the equilibrium probability that the source is in state σ at a renewal epoch. Similarly let ${}^\sigma A^i(u)$, for every u, $u \geq 0$, be the total number of arrivals in the u units of time immediately following a renewal epoch, given that the state of the source i at the renewal epoch was σ. Further, let ${}^\sigma\alpha^i_k(u) \stackrel{\text{def}}{=} P[{}^\sigma A^i(u) = k]$, for every u, $u \geq 0$. For simplicity the notation ${}^\sigma\alpha^i_k$ will be used to refer to ${}^\sigma\alpha^i_k(D)$. Given that the state of the source at the renewal epoch is σ, let ${}^\sigma A^i_n(u)$, for $0 \leq u \leq 1$, be the total number of arrivals in the first u units of time of the n^{th} slot immediately following the renewal epoch, where $0 \leq n \leq D$. Notice that ${}^\sigma A^i_n(u) = {}^\sigma A^i(n+u) - {}^\sigma A^i(n)$. Let ${}^\sigma\alpha^{n,i}_k(u) \stackrel{\text{def}}{=} P[{}^\sigma A^i_n(u) = k]$. For simplicity the notation ${}^\sigma A^i_n$ will be used to refer to ${}^\sigma A^i_n(1)$, and the notation ${}^\sigma\alpha^{n,i}_k$ will be used to refer to ${}^\sigma\alpha^{n,i}_k(1)$.

The source i traffic load is the mean number of arriving cells from source i in a slot time interval and is given by

$$Load_i \stackrel{\text{def}}{=} \frac{1}{D}\sum_{\forall\sigma}\sum_{k=1}^{\infty} k \times {}^\sigma\alpha^i_k \times p^i_\sigma = \frac{1}{D}\sum_{\forall\sigma}\sum_{n=0}^{D-1}\sum_{k=1}^{\infty} k \times {}^\sigma\alpha^{n,i}_k \times p^i_\sigma \quad . \tag{1}$$

In Section 6, a number of well known source models are shown to be special cases of this model.

The class of Markov renewal CSs of the type described above is closed under superposition, *i.e.* the superposition of a number of CSs of the type described above results in another CS of the same type [7]. If the index i representing the specific source i is removed from the above equations, then the corresponding parameters refer to the collective behavior of all sources (*i.e.* the superposition of all the incoming CSs). The traffic source model described above is rich yet amenable to performance analysis [5].

3. Exact Analysis of the Behavior of a Multiplexer

The state of the multiplexer immediately before the beginning of the n^{th} slot since the last renewal, for all n, $0 \leq n \leq D-1$, is described by (j_c, σ), where j_c is the total number of cells in the multiplexer at the end of the previous slot, and σ is the state of the cell sources respectively at the beginning of the 0^{th} slot.

If σ is the state of the sources and j_c is the number of cells in the multiplexer buffer immediately before the beginning of the n^{th} slot, let ${}^{\sigma}b_{j_c}^{j_c^*}(n)$ be the probability that the multiplexer will have j_c^* cells at the end of the n^{th} slot for all n, $0 \leq n \leq D-1$. Using the notation $\lceil x \rceil^M \stackrel{\text{def}}{=} \min\{x, M\}$, and $\lfloor x \rfloor_M \stackrel{\text{def}}{=} \max\{x, M\}$, the transition probabilities are defined for $0 \leq j_c, j_c^* \leq K$, and $0 \leq n \leq D-1$ as follows:

$$
{}^{\sigma}b_{j_c}^{j_c^*}(n) = \begin{cases} {}^{\sigma}\alpha^{n}_{j_c^*-j_c+1(j_c>0)} & \text{if } j_c^* < K \\ 1 - \sum_{j=\lfloor j_c-1 \rfloor_0}^{K} {}^{\sigma}\alpha^{0}_{j-j_c+1(j_c>0)} & \text{if } j_c^* = K \\ 0 & \text{otherwise} \end{cases} \tag{2}
$$

Let the square matrix ${}^{\sigma}\mathbf{P}_n$ be defined such that ${}^{\sigma}\mathbf{P}_n(j_c, j_c^*) \stackrel{\text{def}}{=} {}^{\sigma}b_{j_c}^{j_c^*}(n)$ for $0 \leq j_c, j_c^* \leq K$ and $0 \leq n \leq D-1$. In short,

$$
{}^{\sigma}\mathbf{P}_n \stackrel{\text{def}}{=} \left[{}^{\sigma}b_{j_c}^{j_c^*}(n) \right] .
$$

With ${}^{\sigma}\pi_{j_c}^{(n)}$ defined as the equilibrium probability that the system is in state (j_c, σ) immediately before the beginning of the n^{th} slot since the last renewal epoch, let ${}^{\sigma_i}\pi^{(n)} \stackrel{\text{def}}{=} [{}^{\sigma_i}\pi_0^{(n)}, {}^{\sigma_i}\pi_1^{(n)}, \cdots, {}^{\sigma_i}\pi_K^{(n)}]$ for every l, $0 \leq l \leq T-1$, for every n, $0 \leq n \leq D-1$ and for every i, $0 \leq i \leq S-1$. Notice that $\sum_{i=0}^{S-1} {}^{\sigma_i}\pi^{(n)}\mathrm{e} = 1/D$, for all n, $0 \leq n \leq D-1$. The algorithm used to compute the equilibrium probabilities ${}^{\sigma_i}\pi^{(n)}$ for all n, $0 \leq n \leq D-1$, for all l, $0 \leq l \leq T-1$ and for every i, $0 \leq i \leq S-1$, is the Gauss-Seidel overrelaxation algorithm presented in [16].

Let L_s be the random variable describing the number of cells lost in a slot time interval. Further, let ${}^{\sigma}Y_{j_c}^{(k;n)}$ be the probability that k cells are lost during the n^{th} slot since the last renewal epoch, where $0 \leq n \leq D-1$, given that the system is in state (j_c, σ) immediately before the beginning of the n^{th} slot. Then for $0 \leq n \leq D-1$,

$$
{}^{\sigma}Y_{j_c}^{(k;n)} \stackrel{\text{def}}{=} \begin{cases} \sum_{j_c^*=\lfloor j_c-1 \rfloor_0}^{K} {}^{\sigma}\alpha^{0}_{j_c^*-j_c+1(j_c>0)} & \text{if } k = 0 \\ {}^{\sigma}\alpha^{n}_{K+k-j_c+1(j_c>0)} & \text{if } k \geq 1 \end{cases} . \tag{3}
$$

If ${}^{\sigma}\mathbf{Y}_k^{(n)} \stackrel{\text{def}}{=} [{}^{\sigma}Y_0^{(k;n)}, {}^{\sigma}Y_1^{(k;n)}, \cdots, {}^{\sigma}Y_K^{(k;n)}]^T$ for $0 \le n \le D-1$, then for $k \ge 0$, the probability that k cells are lost in a slot time interval is given by

$$P[L_s = k] \;=\; \sum_{\forall \sigma} \sum_{n=0}^{D-1} {}^{\sigma}\pi^{(n)\sigma}\mathbf{Y}_k^{(n)} \quad . \tag{4}$$

3.1. The Waiting Time Distribution of a Traffic Source at the Multiplexer

A cell arriving u time units after the beginning of the n^{th} slot following the last renewal is tagged, where $0 \le u < 1$. Let the multiplexer state be (j_c, σ) immediately before the beginning of the n^{th} slot since the last renewal, for all n, $0 \le n \le D-1$. If $j_c + {}^{\sigma}A_n(u) \le K$, then the cell is accepted; otherwise it is rejected. Accepted cells are served in the order of their arrival. In what follows a methodology is introduced for the computation of the waiting time distribution of an accepted cell.

All slots are enumerated according to their proximity to a renewal epoch. Each time slot is assigned a number between 0 and $D-1$. According to the enumeration rule, the first slot following the renewal epoch is numbered 0, the next slot is numbered 1, and so on.

Let $W_k^n(u, j_c, \sigma)$ be the random variable describing the waiting time of the k^{th} accepted cell arriving u units of time after the beginning of the n^{th} slot since the last renewal, for $0 \le n \le D-1$ and $0 \le u < 1$, given that the multiplexer state immediately before the beginning of the n^{th} slot is (j_c, σ). If $K_{max}^n(j_c, \sigma) \stackrel{\text{def}}{=} K - j_c + 1(j_c > 0)$,

$$W_k^n(u, j_c, \sigma) \;=\; 1 - u + \lfloor j_c - 1 \rfloor_0 + k \quad , \tag{5}$$

for $1 \le k \le K_{max}^n$.

Given that the state of the source at the last renewal epoch was σ, let $F_k^{(\sigma,n)}(u)$ be the distribution of the arrival time of the k^{th} cell during the n^{th} slot following the last renewal epoch, for $0 \le u < 1$ and $0 \le n \le D-1$. Then,

$$P[W^n(j_c, \sigma) > x] \;=\; \frac{\sum_{k=1}^{K_{max}^n(j_c,\sigma)} \int_0^1 1(W_k^n(u, j_c, \sigma) > x) dF_k^{(\sigma,n)}(u)}{\sum_{k=1}^{K_{max}^n(j_c,\sigma)} \int_0^1 dF_k^{(\sigma,n)}(u)} \quad . \tag{6}$$

Notice that $W_k^n(u, j_c, \sigma)$ is a *decreasing* function of u for all k, $k \ge 0$. Let $U_k^n(x, j_c, \sigma)$ be the maximum value of u in the interval $[0, 1)$ for which $W_k^n(u, j_c, \sigma) > x$. Then,

$$P[W^n(j_c, \sigma) > x] \;=\; \begin{cases} \frac{\sum_{k=1}^{K_{max}^n(j_c,\sigma)} P[{}^{\sigma}A_n(U_k^n(x, j_c, \sigma)) \ge k]}{\sum_{k=1}^{K_{max}^n(j_c,\sigma)} P[{}^{\sigma}A_n \ge k]}, & \text{if } \sum_{k=1}^{K_{max}^n(j_c,\sigma)} P[{}^{\sigma}A_n \ge k] > 0; \\ undefined, & \text{otherwise,} \end{cases} \tag{7}$$

for all n, $0 \le n \le D-1$.

Let

$$\bar{P}[W^n(j_c, \sigma) > x] \;=\; \begin{cases} P[W^n(j_c, \sigma) > x], & \text{if } \sum_{k=1}^{K_{max}^n(j_c,\sigma)} P[{}^{\sigma}A_n \ge k] > 0; \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

for all n, $0 \leq n \leq D-1$. Let the column vector ${}^{\sigma}\bar{\mathbf{P}}[W^n > x]$ be defined such that its $(j_c)^{th}$ element is $\bar{P}[W^n(j_c, \sigma) > x]$, for all possible values of j_c. Then, if $P[W \leq x]$ is the probability that any incoming cell will be delayed at most x units of time, then for x, $x \geq 0$,

$$P[W > x] \;=\; \frac{\sum_{\forall\sigma}\sum_{n=0}^{D-1} {}^{\sigma}\pi^{(n)\sigma}\bar{\mathbf{P}}[W^n > x]}{\sum_{\forall\sigma}\sum_{n=0}^{D-1} {}^{\sigma}\pi^{(n)\sigma}\bar{\mathbf{P}}[W^n > 0]} \quad . \tag{9}$$

4. Computation of Performance Bounds

The algorithms presented in the previous sections give exact description of the steady state behavior of the multiplexer. From this it is easy to define various performance measures which are needed for the multiplexer dimensioning. Unfortunately, the requirements in terms of CPU time may be excessive for realistic systems. In order to reduce the computational burden, observe that in many cases it is not necessary to know all steady state probabilities. Often the values of one or several performance measures f, such as the average cell loss or the average waiting time, suffice. Such measures are linear functions* of the steady state probabilities, and more efficient algorithms for the computation of the upper and lower bounds of the quantities of interest can be derived [10, 13, 15, 16, 2].

The problem in question can be formulated as follows: $\mathcal{S} \stackrel{def}{=} \{0, 1, \cdots, S-1\}$. Find

$$f \;=\; c\pi^T \tag{10}$$

subject to constraints

$$\pi \;=\; \pi P, \quad \pi \geq 0, \quad \sum_{i\in\mathcal{S}} \pi_i = 1 \quad , \tag{11}$$

where π and c are row vectors of dimension S and define respectively the equilibrium probabilities and the state contribution to the performance measure. P is the state transition matrix of the discrete Markov chain. Under well known assumptions Equation 11 has a unique solution π which defines the steady state distribution of the Markov chain. However, formally the problem in Equations 10-11 can be considered a linear programming problem:

$$\max_{\pi} c\pi^T \tag{12}$$

subject to constraints

$$\pi(P-I) \;=\; 0, \quad \pi \geq 0, \quad \sum_{i\in\mathcal{S}} \pi_i = 1 \quad . \tag{13}$$

The dual of the above optimization problem is [12]

$$\min_{v,z} z \tag{14}$$

subject to

$$(P-I)v^T + \bar{z}^T \geq c^T \quad , \tag{15}$$

*The expected waiting time is a ratio of two linear functions. It can be expressed as the unique solution of a linear program through a non-linear transformation [6].

where $\overline{z}$ is a vector of dimension S, with all components equal to z, and for arbitrary values of z and v.

If instead the problem in Equations 10-11 were viewed as a maximization problem, then the dual would have been the following minimization problem

$$\max_{v,z} z \tag{16}$$

subject to

$$(P - I)v^T + \overline{z}^T \leq c^T \tag{17}$$

where $\overline{z}$ is a vector of dimension S, with all components equal to z, and for arbitrary values of z and v.

Due to the ergodic property the solution of Equations 12-13 exists, and its optimal value f coincides with the optimal value of its dual problem [12]. But from the weak duality theorem of linear programming [12], any feasible solution of the minimization problem is an upper bound of any feasible solution of the maximization problem. Since an arbitrary value of v combined with a value of z given by the equation $z = \max_{k\in\mathcal{S}}(c_k + v_k - \sum_{i\in\mathcal{S}} v_i p_{ki})$ constitute a feasible solution of the Equations 14-15, the upper bound of the optimal solution of the maximization problem is given by

$$f \leq \max_{k\in\mathcal{S}}(c_k + v_k - \sum_{i\in\mathcal{S}} v_i p_{ki}) \quad . \tag{18}$$

Similarly, an arbitrary value of v combined with a value of z given by the equation $z = \min_{k\in\mathcal{S}}(c_k + v_k - \sum_{i\in\mathcal{S}} v_i p_{ki})$ constitute a feasible solution of the Equations 16-17, and a lower bound of the optimal solution of the maximization problem is given by

$$\min_{k\in\mathcal{S}}(c_k + v_k - \sum_{i\in\mathcal{S}} v_i p_{ki}) \leq f \quad . \tag{19}$$

Now consider the case in which all inequalities in Equations 15 and 17 are satisfied as equalities at the optimal solution, *i.e.*

$$v = vP^T + \overline{z} - c \quad . \tag{20}$$

This is the most interesting case because from the complementary slackness theorem of linear programming [12], it is known that if Equations 15 and 17 are strict inequalities, the problem is redundant in the sense that the corresponding rows of the matrix P do not influence the value of the performance measure. That is, if Equation 20 is not satisfied, then the transition probabilities from some state can be arbitrarily changed without changing the performance measure.

Equations 11 and 20 can now be compared. A solution to Equation 11 is the following fixed point algorithm

$$\pi^{s+1} = \pi^s P \quad , \tag{21}$$

which converges to the steady state distribution of the Markov chain as $s \to \infty$ under well known assumptions. The similar fixed point algorithm for the system of Equations 20 has the following form:

$$v^{s+1} = v^s P^T + \overline{z} - c \tag{22}$$

or equivalently,

$$v_k^{s+1} = \sum_{i \in \mathcal{S}} v_i^s p_{ki} + z - c_k \tag{23}$$

for all $k \in \mathcal{S}$. Observe that due to Equations 18 and 19, the following estimate on each iteration holds:

$$\min_{k \in \mathcal{S}} (c_k + v_k^s - \sum_{i \in \mathcal{S}} v_i^s p_{ki}) \leq f \leq \max_{k \in \mathcal{S}} (c_k + v_k^s - \sum_{i \in \mathcal{S}} v_i^s p_{ki}) \quad , \tag{24}$$

or equivalently,

$$z + \min_{k \in \mathcal{S}} (v_k^s - v_k^{s+1}) \leq f \leq z + \max_{k \in \mathcal{S}} (v_k^s - v_k^{s+1}) \tag{25}$$

where the difference between the left and right sides of the inequality tends to zero under the same assumptions which guarantee the convergence of Equation 21. Note that Equations 23 and 25 depend on the unknown quantity $\overline{z}$. The substitution $u^s \stackrel{def}{=} v^s + s\overline{z}$ results in $z + v^s - v^{s+1} = u^s - u^{s+1}$,

$$u^{s+1} = u^s P^T - c \tag{26}$$

and

$$\min_{k \in \mathcal{S}} (u_k^s - u_k^{s+1}) \leq f \leq \max_{k \in \mathcal{S}} (u_k^s - u_k^{s+1}) \quad . \tag{27}$$

The previous equations lead to the following algorithm:

$$u^{s+1} = u^s P^T - c \tag{28}$$

$$\overline{f} = \max_{k \in \mathcal{S}} (u_k^s - u_k^{s+1}) \tag{29}$$

$$\underline{f} = \min_{k \in \mathcal{S}} (u_k^s - u_k^{s+1}) \tag{30}$$

where $\overline{f} \geq f \geq \underline{f}$, $\overline{f} - \underline{f} \to 0$ as $s \to \infty$.

The above equations are the basis of following value-iteration algorithm [16].

Value-Iteration Algorithm for the Computation of the Performance Bounds

Step 0. Choose $u_k^0 = -c_k$ for all $k \in \mathcal{S}$. Let $s := 1$.
Step 1. Compute the value function $u_k^s, k \in \mathcal{S}$, from

$$u_k^s = \sum_{j \in \mathcal{S}} p_{kj} u_j^{s-1} - c_k \quad . \tag{31}$$

Step 2. Compute the bounds

$$\underline{f} = \min_{k \in \mathcal{S}} \{u_k^s - u_k^{s+1}\}$$

$$\overline{f} = \max_{j \in \mathcal{S}} \{u_k^s - u_k^{s+1}\}$$

If $0 \leq \overline{f} - \underline{f} \leq \epsilon \underline{f}$, where ϵ is a tolerance number, then stop. Otherwise, go to Step 3.
Step 3. $s := s + 1$ and go to Step 1.

The convergence rate of this value-iteration algorithm can be further accelerated by the introduction of a relaxation factor. The reader is referred to [16] for details. Unfortunately, for problems with a large state space, this algorithm may prove to be prohibitively slow. In this case approximations must be used. One approach to approximation is presented in Section 5, where the value iteration algorithm is combined with aggregation.

5. Guaranteed Bounds for Multiplexer Performance Through State Aggregation

As mentioned in [10] "the curse of dimentionality" forces the development of useful techniques for problems of practical significance. One of the techniques that has been considered in the literature is state aggregation [10, 1, 13, 15, 2]

In [10] fixed-weight aggregation is presented. In this context let $\mathcal{K} \stackrel{def}{=} \{0, 1, \cdots, K-1\}$ be the aggregated state space. The state k of the aggregated state space includes all states in the subset $\mathcal{A}_k$, for $k \in \mathcal{K}$ of the state space $\mathcal{S}$. Further, $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for all $i, j \in \mathcal{K}$ and $i \neq j$. $\cup_{k\in\mathcal{K}} \mathcal{A}_k = \mathcal{S}$. Let λ_s^k be the weight of state s in the aggregated state k. Then let q_{kl} be the transition probability in the aggregated Markov chain from the aggregated state k to the aggregated state l, and let r_k be the aggregated cost of the aggregated state k. Then,

$$q_{kl} = \sum_{s\in\mathcal{A}_k} \lambda_s^k \sum_{j\in\mathcal{A}_l} p_{sj} \tag{32}$$

$$r_k = \sum_{s\in\mathcal{A}_k} \lambda_s^k c_s \tag{33}$$

$$\sum_{s\in\mathcal{A}_k} \lambda_s^k = 1 \tag{34}$$

for all $k, l \in \mathcal{K}$.

In general the weight-factors should be a function of the equilibrium probabilities. Under the aggregated form the problem in question can be formulated as follows: Find

$$f^a = r\hat{\pi}^T \tag{35}$$

subject to constraints

$$\hat{\pi} = \hat{\pi}Q, \quad \hat{\pi} \geq 0, \quad \sum_{\forall i} \hat{\pi}_i = 1 \quad , \tag{36}$$

where $\hat{\pi}$ and r are row vectors of dimension K and define respectively the equilibrium aggregated probabilities and the aggregated state contribution to the performance measure. Using a methodology simular to as that used to solve the non-aggregated problem in the previous section, it can be proven that

$$u^{s+1} = u^s Q^T - r \tag{37}$$

and

$$\min_{k\in\mathcal{K}}(u_k^s - u_k^{s+1}) \leq f^a \leq \max_{k\in\mathcal{K}}(u_k^s - u_k^{s+1}) \quad . \tag{38}$$

The previous equations lead to the following algorithm:

$$u^{s+1} = u^s Q^T - r \tag{39}$$

$$\overline{f} = \max_{k\in\mathcal{K}}(u_k^s - u_k^{s+1}) \tag{40}$$

$$\underline{f} = \min_{k\in\mathcal{K}}(u_k^s - u_k^{s+1}) \tag{41}$$

where $\overline{f} \geq f \geq \underline{f}$, $\overline{f} - \overline{f} \to 0$ and $\underline{f} - \underline{f} \to 0$ as $s \to \infty$.

Equation 40 can be further analyzed:

$$\overline{f} = \max_{k\in\mathcal{K}}(u_k^s - u_k^{s+1}) =$$

$$= \max_{k\in\mathcal{K}}\Big(u_k^s - \sum_{l\in\mathcal{K}} u_l^s q_{kl} + r_k\Big)$$

$$= \max_{k\in\mathcal{K}}\Big(u_k^s - \sum_{l\in\mathcal{K}} u_l^s \sum_{s\in\mathcal{A}_k} \lambda_s^k \sum_{j\in\mathcal{A}_l} p_{sj} + \sum_{s\in\mathcal{A}_k} \lambda_s^k c_s\Big)$$

$$= \max_{k\in\mathcal{K}}\Big\{\sum_{s\in\mathcal{A}_k} \lambda_s^k \Big(c_s - \sum_{l\in\mathcal{K}} u_l^s \sum_{j\in\mathcal{A}_l} p_{sj}\Big) + u_k^s\Big\} \quad . \tag{42}$$

From Equation 42 a number of different upper bounds can be derived. For example if all states in a group are equally weighted, then

$$\overline{f} = \max_{k\in\mathcal{K}}\Big\{\sum_{s\in\mathcal{A}_k} \frac{1}{|\mathcal{A}_k|}\Big(c_s - \sum_{l\in\mathcal{K}} u_l^s \sum_{j\in\mathcal{A}_l} p_{sj}\Big) + u_k^s\Big\} \tag{43}$$

where $|\mathcal{A}_k|$ is the cardinality of the set $\mathcal{A}_k$, for all $k \in \mathcal{K}$.

It is worth noting that if $\lambda_s^k = \frac{\pi_s}{\sum_{i\in\mathcal{A}_k}\pi_i}$ for $s \in \mathcal{S}$, then the non-aggregated and aggregated versions of the problem have the same objective function value. This is desirable because any lower and upper bounds for the aggregated version are lower and upper bounds for the non-aggregated problem as well.

An upper bound of $\overline{f}$ which holds for all possible choices of the weight-parameters in Equation 43 and thus is also an upper bound of the non-aggregated problem is given by

$$\overline{f} = \max_{k\in\mathcal{K}}\Big\{\max_{s\in\mathcal{A}_k}\Big(c_s - \sum_{l\in\mathcal{K}} u_l^s \sum_{j\in\mathcal{A}_l} p_{sj}\Big) + u_k^s\Big\} \quad . \tag{44}$$

Similarly, a lower bound of $\underline{f}$ which holds for all possible choices of the weight-parameters and thus is also a lower bound of the non-aggregated problem is given by

$$\underline{f} = \min_{k\in\mathcal{K}}\Big\{\max_{s\in\mathcal{A}_k}\Big(c_s - \sum_{l\in\mathcal{K}} u_l^s \sum_{j\in\mathcal{A}_l} p_{sj}\Big) + u_k^s\Big\} \quad . \tag{45}$$

The previous equations lead to the following algorithm [2]:

Value-Iteration Algorithm for the Computation of the Performance Bounds

Step 0. Choose $u_k^0 = \max_{j \in \mathcal{A}_k} c_j$ for all k. Set $\overline{f}^0 = \max_{j \in \mathcal{S}} c_j$ and $\underline{f}^0 = \min_{j \in \mathcal{S}} c_j$. Let $s := 1$.
Step 1. Compute the value function u_k^s, from

$$u_k^{s+1} = \max_{s \in \mathcal{A}_k} (\sum_{l \in \mathcal{K}} u_l^s \sum_{j \in \mathcal{A}_l} p_{sj} - c_s) \tag{46}$$

Step 2. Compute the bounds

$$\overline{f}^{s+1} = \max_{k \in \mathcal{K}} (u_k^s - u_k^{s+1}) \tag{47}$$

$$\underline{f}^{s+1} = \min_{k \in \mathcal{K}} (u_k^s - u_k^{s+1}) \tag{48}$$

If $|\overline{f}^{s+1} - \overline{f}^s| \leq \epsilon$ and $|\underline{f}^{s+1} - \underline{f}^s| \leq \epsilon$, where ϵ is a perspective tolerance number, then stop. Otherwise, go to Step 3.
Step 3. $s := s + 1$ and go to Step 1.

The reader is referred to [2] for further details.

6. Study of Different Traffic Sources

In order for a traffic source generator to be useful in an ATM environment, it must be capable of modeling a rich variety of traffic behaviors. A performance analysis based on such a traffic source generator is useful because it provides a uniform methodology through which a control infrastructure can be tested under diverse traffic conditions. In this section, a number of different traffic source models appearing in the ATM literature are presented and shown to be special cases of the traffic source model described in Section 3 and utilized throughout this paper. By showing how a wide variety of continuous and discrete time traffic source generators can be modeled using the traffic source model described in Section 3, it is demonstrated that the traffic generator on which the performance analysis presented in this paper is based is indeed capable of modeling a rich variety of traffic behaviors [5, 7].

6.1. A Poisson Process Traffic Source

A traffic source following a Poisson distribution with rate c can be described as follows: $S = 1$, $p_{00} = 1$ and $p_0 = 1$.

$$\alpha_k(u) \;=\; e^{-cu} \frac{(cu)^k}{k!} \quad ,$$

for $u \geq 0$, $0 \leq n \leq D - 1$, and $k \geq 0$. Finally,

$$\alpha_k^n(u) \;=\; \alpha_k(u) \quad ,$$

for $0 \leq u \leq 1$, and for $0 \leq n \leq D - 1$.

6.2. Multiplexing of N Independent and Identically Distributed Two-state Bursty Sources

A bursty source is modelled as a two-state Markov chain. The Markov chain makes a transition every $1/c$ units of time. When the Markov chain enters state 1, a cell is generated with probability p_{gen}. Notice that $p_{00} + p_{01} = 1$ and $p_{10} + p_{11} = 1$. Let

$$\mathbf{Q} \stackrel{\text{def}}{=} \begin{pmatrix} q_{00} & q_{01} \\ q_{10} & q_{11} \end{pmatrix} .$$

Let ${}^{\sigma}q(0,n)$ (respectively ${}^{\sigma}q(1,n)$) be the probability that after n transitions, the two-state Markov chain will be in state 0 (respectively 1), given that the state of the Markov chain at the last renewal epoch was σ. Also let ${}^{\sigma}q(n) \stackrel{\text{def}}{=} [{}^{\sigma}q(0,n)\ {}^{\sigma}q(1,n)]$. If $\sigma = 0$, then ${}^{0}q(0) \stackrel{\text{def}}{=} [1\ 0]$, and if $\sigma = 1$, then ${}^{1}q(0) \stackrel{\text{def}}{=} [0\ 1]$. ${}^{\sigma}q(n) = {}^{\sigma}q(0)\mathbf{Q}^{n}$, for $n \geq 0$ and $\sigma \in \{0,1\}$. The probability that k cells are generated in m consecutive transitions of the Markov chain is given by:

$$\sum_{k_1=0}^{1} \cdots \sum_{k_m=0}^{1} \prod_{n=1}^{m} {}^{\sigma}q(k_n, n) p_{gen}^{k} 1(\sum_{j=1}^{m} k_j = k) .$$

For this type of source model, $S = 2$, $\sigma_0 = 0$, $\sigma_1 = 1$, $p_{00} = {}^{0}q(0,D)$, $p_{01} = {}^{i}q(1,D)$, $p_{10} = {}^{0}q(0,D)$, and $p_{11} = {}^{1}q(1,D)$.

$${}^{\sigma}\alpha_k(u) = \begin{cases} \sum_{k_1=0}^{1} \cdots \sum_{k_{\lfloor uc \rfloor}=0}^{1} \prod_{n=1}^{\lfloor uc \rfloor} {}^{\sigma}q(k_n, n) p_{gen}^{k} 1(\sum_{j=1}^{\lfloor uc \rfloor} k_j = k) & \text{if } 1 \leq \lfloor uc \rfloor \\ 1 & \text{if } k = 0, \text{ and } 1 > \lfloor uc \rfloor \\ 0 & \text{otherwise} \end{cases} ,$$

for $\sigma \in \{0,1\}$, and $0 \leq u \leq D$, and $k \geq 0$. Finally,

$${}^{\sigma}\alpha_k^{n}(u) = {}^{\sigma}q(0, \lfloor nc \rfloor)^{i}\alpha_k(\lfloor (n+u)c \rfloor - \lfloor nc \rfloor) + {}^{\sigma}q(1, \lfloor nc \rfloor)^{a}\alpha_k(\lfloor (n+u)c \rfloor - \lfloor nc \rfloor) ,$$

for $\sigma \in \{0,1\}$, $k \geq 0$, $0 \leq n \leq D-1$, and $0 \leq u \leq 1$.

When N of the above sources are multiplexed, the state of the multiplexed sources is the number of active sources.

$${}^{\sigma}\alpha_k^{n}(u) \stackrel{\text{def}}{=} \sum_{k_0 + \cdots k_{N-1} = k} {}^{\sigma_0}\alpha_{k_0}^{n,0}(u) \otimes \cdots \otimes {}^{\sigma_{N-1}}\alpha_{k_{N-1}}^{n,N-1}(u) 1(\sum_{j=0}^{N-1} \sigma_j = \sigma) ,$$

for all $k \geq 0, k_0 \geq 0, \cdots k_{N-1} \geq 0$, $0 \leq n \leq D-1$, and $0 \leq u \leq 1$. Furthermore the probability that the state of the multiplexed sources makes a transition from state σ_1 to σ_2 is given by

$$p_{\sigma_1 \sigma_2} = \sum_{k_1=0}^{\sigma_1} \sum_{k_2=0}^{N-1-\sigma_1} \binom{N}{\sigma_1 - k_1, k_1, N - \sigma_1 - k_2, k_2} p_{11}^{k_1} p_{10}^{\sigma_1 - k_1} p_{01}^{k_2} p_{00}^{N-\sigma_1-k_2} 1(k_1 + k_2 = \sigma_2) .$$

for all $\sigma_1, \sigma_2 \in \{0, 1, \cdots, N-1\}$.

6.3. Binomial Traffic Source Model

Cells are generated every $1/c$ units of time with probability p. The binomial traffic source model can be seen as a traffic source with $S = 1$, $p_{00} = 1$ and $p_0 = 1$.

$$\alpha_k(u) \;=\; \binom{\lfloor uc \rfloor}{k} p^k (1-p)^{\lfloor uc \rfloor - k} \quad ,$$

for $0 \leq u \leq D$, and $k \geq 0$. Finally,

$$\alpha_k^n(u) \;=\; \binom{\lfloor (n+u)c \rfloor - \lfloor nc \rfloor}{k} p^k (1-p)^{\lfloor (n+u)c \rfloor - \lfloor nc \rfloor - k} \quad ,$$

for $0 \leq n \leq D-1$, $0 < u \leq 1$, and $k \geq 0$.

6.4. The Constant Bit Rate Traffic (CBRT) Source

The CBRT source is another example of a single state traffic source. One cell is generated every $1/c$ units of time. Let D be an integer multiplier of $1/c$. For this type of traffic source, $S = 1$, $p_{00} = 1$ and $p_0 = 1$.

$$\alpha_k(u) \;=\; P[\frac{k}{c} \leq u < \frac{k+1}{c}] \;=\; 1(\lfloor uc \rfloor \;=\; k) \quad ,$$

for $0 \leq u \leq D$, and $k \geq 0$. Finally,

$$\alpha_k^n(u) \;=\; P[\lfloor (n+u)c \rfloor \;-\; \lfloor nc \rfloor \;=\; k] \;=\; 1(\lfloor (n+u)c \rfloor \;-\; \lfloor nc \rfloor \;=\; k) \quad ,$$

for $0 \leq n \leq D-1$, $0 < u \leq 1$, and $k \geq 0$.

Notice that the binomial source model can be seen as a special case of the two-state Markov chain model for $q_{01} = q_{10} = p$, and that the CBRT source can be seen as a special case of the binomial source model for $p = 1$.

6.5. Multiplexing of N Non-Homogeneous Sources

The arrival process of source i is completely described by the quantities $\alpha_k^{n,i}(u)$ for $0 \leq n \leq D-1$, $0 \leq u \leq 1$, and $k \geq 0$. The overall arrival process of N cell streams is described by the quantities

$$^{\sigma}\alpha_k^n(u) \;\stackrel{\text{def}}{=}\; \sum_{k_0 + \cdots k_{N-1} = k} {}^{\sigma_0}\alpha_{k_0}^{n,0}(u) \otimes \cdots \otimes {}^{\sigma_{N-1}}\alpha_{k_{N-1}}^{n,N-1}(u) 1(\sigma = f(\sigma_0, \cdots, \sigma_{N-1})) \quad ,$$

for all $k \geq 0, k_0 \geq 0, \cdots k_{N-1} \geq 0$, $0 \leq n \leq D-1$, and $0 \leq u \leq 1$.

Special attention has to be given to the appropriate definition of the state σ which in general is a function $f(\sigma_0, \cdots, \sigma_{N-1})$ of the states $\sigma_0, \cdots, \sigma_{N-1}$. The number of states of the multiplexed sources cannot be higher than $S_0 \times \cdots \times S_{N-1}$ and is usually much less [7]. If one of the sources, say the first, has one state, that is if $S_0 = 1$, then the total number of states of the superposition equals the number of states of the superposition of the remaining of the sources. Therefore, the presence of single state sources does not affect the state space of the multiplexed sources.

The expected number of lost cells from the i^{th} source is given by

$$EL_s^i = \sum_{n=0}^{D-1} \sum_{j_c=0}^{K} \sum_{\sigma_0 \in S_0} \cdots \sum_{\sigma_{N-1} \in S_{N-1}} \sum_{j_0 + \cdots + j_{N-1} = j_c} \frac{j_i}{\sum_{k=0}^{N-1} j_k} {}^{\sigma}\pi_{j_c} \prod_{l=0}^{N-1} {}^{\sigma_l}\alpha_{j_l}^{n,l} \lfloor \sum_{s=0}^{N-1} j_s - K + \lfloor jc - 1 \rfloor_0 \rfloor_0 \quad ,$$

for $i \in \{0, \cdots, N-1\}$. For further details the reader is referred to [7].

7. Examples and Discussion

In this section, a number of examples illustrating the results derived in this paper are presented. The first class of examples (illustrated in Figures 1 through 4) is based on the multiplexing of independent and identically distributed (i.i.d.) bursty sources. A bursty source is modeled by the two-state discrete Markov chain model introduced in Section 6.2. Let

$$Activity = Prob[\text{Source is active}] = \frac{p_{01}}{p_{01} + p_{10}} \quad . \tag{49}$$

Further, let

$$Burstiness = \frac{\text{Maximum Source Rate}}{\text{Mean Source Rate}} = \frac{\frac{1}{c}}{\frac{1}{c} \times P_{gen} \times Prob[\text{source is active}]} \quad . \tag{50}$$

Therefore,

$$Burstiness = \frac{1}{Activity \times P_{gen}} \quad . \tag{51}$$

The mean burst length is $\frac{1}{p_{10}}$ cells, and the mean silence length is $\frac{1}{p_{01}}$. Further, notice that

$$\frac{1}{c} = \frac{\text{Output Link Rate}}{\text{Maximum Source Rate}} \quad . \tag{52}$$

In Figure 1, the buffering required for the multiplexing of a number of i.i.d bursty sources under a specific cell loss probability GoS is computed. As expected, the increased number of multiplexed sources leads to a linear increase in the traffic activity of the multiplexer, which in turn results in a nonlinear increase in the buffering required if the same cell loss probability is to be maintained.

The analysis presented in Figure 1 is based on the exact description of the source behaviors and is computationally expensive. In general, if $S-1$ two-state discrete Markov chain sources are superimposed, the complexity of the problem is proportional to $S \times (K+1)$. In an ATM environment and for a 140 Mbps link supporting low bursty (up to 2 Mbps) sources, $S-1$ would likely be greater than 100. The complexity of the corresponding problem would therefore be on the order of $100K$.

In order to reduce complexity, an approximation must be introduced. In the example illustrated in Figure 2, a three-state approximation of the behavior of the original S-state Markov chain is employed. This aggregation reduces the number of states to $3(K+1)D$.

Let u (for underload), n (for normal load), and o (for overload) refer to the three aggregated states. Let $\mathcal{A}_u$ be the subset of the $S-1$ state space composed of the underload states. Let $\mathcal{A}_n$ be the subset of the state space composed of the normal load states. Finally let $\mathcal{A}_o$ be the subset composed of the remaining states. Then,

$$P_{\sigma\hat{\sigma}} = \frac{\sum_{i\in\mathcal{A}_\sigma}\sum_{j\in\mathcal{A}_{\hat{\sigma}}} p_i p_{ij}}{\sum_{i\in\mathcal{A}_\sigma} p_i} \quad ,$$

for $\sigma, \hat{\sigma} \in \{u, n, o\}$. While in state σ, the aggregate source generates i cells with probability $\frac{p_i}{\sum_{i\in\mathcal{A}_\sigma} p_i}$, for $i \in \mathcal{A}_\sigma$, and $\sigma \in \{u, n, o\}$.

Figure 2 demonstrates results based on the three-state approximation of the source behavior. In particular 20 i.i.d. bursty sources are multiplexed. The overall behavior of the multiplexed sources is described by a 21-state discrete Markov chain, which is approximated by a 3-state Markov chain. The partition of the original 21 states into the aggregate three states is as follows: $\mathcal{A}_u = \{0, \cdots, Threshold1\}$, $\mathcal{A}_n = \{Threshold1{+}1, \cdots, Threshold2\}$, $\mathcal{A}_o = \{Threshold2{+}1, \cdots, 21\}$. Whereas the results presented in Figure 2 are quite good, this type of source aggregation does not always give good results. This is due to the fact that an efficient state aggregation cannot be identified without consideration of the parameters of the multiplexer, such as buffer size and service rate [7].

In Figures 3 and 4, it is demonstrated that the output rate of the multiplexer divided by the peak input traffic rate is a fundamental quantity in the characterization of the cell loss probability. Observe that for buffer sizes less than this ratio, each source has at most one cell in the buffer, whereas for buffer sizes greater than this ratio, each source may have multiple cells in the buffer. The shape of the cell loss probability versus the buffer size raises the question of whether buffering can be considered an effective method for the solution of the congestion problem. It appears that buffering should be used only for a temporary surge of traffic.

In Figure 5, three classes of sources are statistically multiplexed. The first class contains five two-state Markov chain sources with $p_{01} = 0.00625$, $p_{10} = 0.025$, $c = 1$ and $p_{gen} = .2$. The second class contains five two-state Markov chain sources with $p_{01} = 0.003125$, $p_{10} = 0.0125$, $c = 1$ and $p_{gen} = .2$. The third class of traffic is a Poisson source with rate 0.2. Notice that the first and second classes include sources with identical activity and burstiness, whereas the second class sources have twice the average size of the idle and active periods compared to the first class sources. Figure 5 shows the loss probability of the overall multiplexed sources. Figure 6 shows the loss probability of a source belonging to each of the three classes. From Figure 6, it is observed that as anticipated for a given buffer size the least loss is experienced by the Poisson traffic source. Bursty CSs experience higher loss. Among the bursty sources, those with shorter active/idle periods experience less loss. From this result one concludes that the activity and burstiness factors do not uniquely characterize a CS. The expected length of an idle plus an active cycle is an additional parameter which plays a very important role in the overall characterization of a CS[7].

Figures 7, 8 and 9 present results of the approximations given in Section 5. In particular it is assumed that the initial buffer capacity equals the output rate divided by the peak input rate. The buffer sizes reported on the figures refer to the additional buffer capacity. In Figure 7 300 binomial sources with maximum arrival rate of 2 Mb/sec are multiplexed. The service rate of the multiplexer is 140Mb/sec. The original buffer is approximated by a buffer with 10 positions. Figure 8 depicts the maximal allowed number of binomial sources which have maximum arrival rate 2 Mb/sec and various values of the parameter p (which is the activity of a binomial source) under the condition that the overall cell loss probability does not exceed 10^{-8}. In this example $c = 1$. The actual size of the additional buffer capacity is 50 cells. It is clear that the minimum number of sources that can be multiplexed is 70, which is the case if each source receives the peak bandwidth allocation of 2 Mb/sec. In this case the cell loss probability will be zero for all possible activity values. The other curves are derived based on the upper bound value derived with the value iteration algorithm presented in Section 5. Notice that the solutions derived under the assumption of zero extra buffer are good enough. The curve that corresponds to the extra buffer size of 10(dotted curve) is almost indistinguishable from the exact solution. Experimentation [2] demonstrates that the quality of the approximation is maintained even if the sources are two-state Markov chains. In Figure 9 the states of source are not aggregated, but the buffer is approximated. The general behavior of the bounds is similar to the binomial case.

References

[1] Bertsekas, D. P., *Dynamic Programming,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[2] Bonatti, M., Gaivoronski, A., "Guaranteed approximations in ATM networks," ITALTEL Technical Report, 1992.

[3] M. Bonatti, M., Gallassi, G., Soto O. G. , "Role of ergodic approximations and of ergodic samples in IBC strategical planning: Lessons from POTS traffic engineering," *Proceedings of the Integrated Broadband Communications: Views from RACE,* Bonatti, Casali and Popple (Editors), Elsevier Science Publishers, 1991.

[4] Bonomi, F., Montagna, S., Paglino, R., "A further look at statistical multiplexing in ATM networks," preprint, March of 1992.

[5] Bovopoulos, A. D., "Performance Evaluation of a Traffic Control Module for ATM Networks," *Proceedings of the IEEE INFOCOM'92 Conference,* May 6-8, 1992, Florence, Italy, pp. 469-478.

[6] Bovopoulos, A. D., "Optimal Burst Level Admission Control in a Broadband Network" *STOCHASTIC MODELS,* Vol. 8, No 2, 1992, pp. 289- 305.

[7] Bovopoulos, A. D. and Dailianas, A., "Statistical Multiplexing of Non-Homogeneous Bursty Sources" paper in preparation, Department of Computer Science, Washington University, 1992.

[8] Boyer, P., "A Congestion Control for the ATM," Proceedings of the 7th ITC Seminar, New Jersey, October 1990, paper 4.3.

[9] Decina, M., Toniatti, T., Vaccari, P. and Verri, L., "Bandwidth assignment and virtual call blocking in ATM networks," *Proceedings of INFOCOM'90,* San Francisco, June 1990.

[10] Heyman, D. P., *Stochastic Models in Operations Research, Volume II,* McGraw-Hill Book Company, 1984.

[11] Hui, J, "Resource Allocation for Broadband Networks," *IEEE Journal on Selected Areas in Communications,* Vol. SAC-6, No. 9, December 1988.

[12] Luenberger, D. G., *Linear and Nonlinear Programming,* Addison Wesley, Second Edition, 1984.

[13] Mendelssohn, R., "An iterative aggregation procedure for Markov decision processes," *Operations Research,* Vol. 30, 1982, pp. 62-73.

[14] Montero, J., Gerla, M. and Fratta, L., "Statistical multiplexing within a bursty environment," *Proceedings of International Conference in Data Communications,* Barcelona, 1990.

[15] Schweitzer, P. J., Puterman, M. L., and Kindle, K. W., "Iterative aggregation-disaggregation procedures for solving discounted semi-Markovian reward processes," *Operations Research,* Vol. 33, 1985, pp. 589-605.

[16] Tijms, H. C., *Stochastic Modelling and Analysis,* John Wiley & Sons, 1986.

[17] Turner, J., "A Proposed bandwidth management and congestion control scheme for multicast ATM networks," Technical Report WUCCRC-91-1, Computer and Communications Research Center, Washington University in Saint Louis, 1991.

[18] Van Der Wal, J. and Schweitzer, P. J., "Iterative bounds on the equilibrium distribution of a Markov chain," *Probability in the Engineering and Informational Sciences,* Vol. 1, 1987, pp. 117-131.

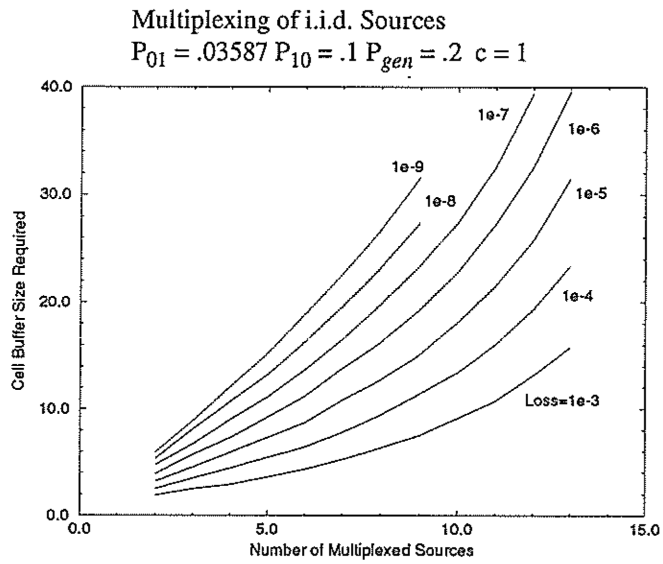

Figure 1: Buffering requirement for the multiplexing of a number of i.i.d. sources at a specific cell loss probability

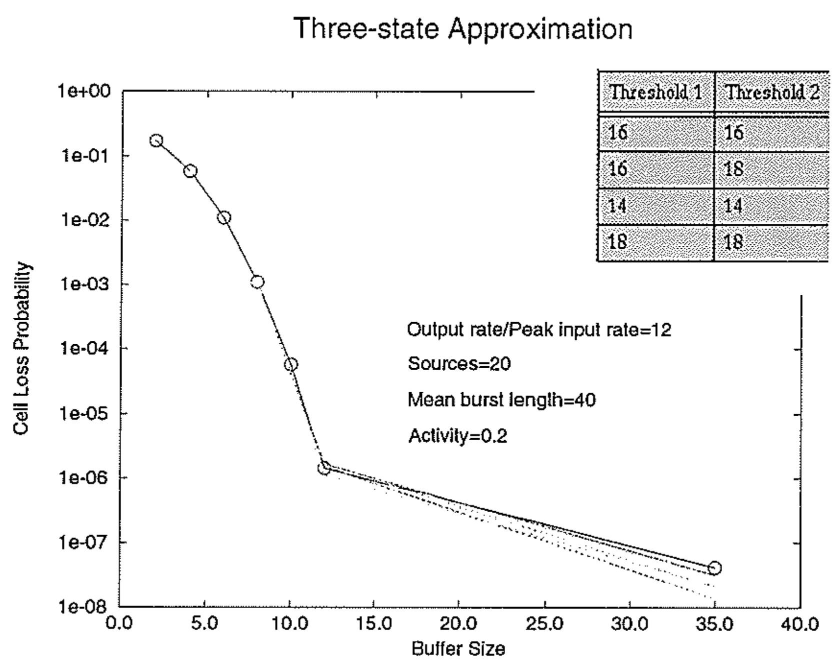

Figure 2: Evaluation of the effectiveness of approximating a multistate bursty source through a three state approximation

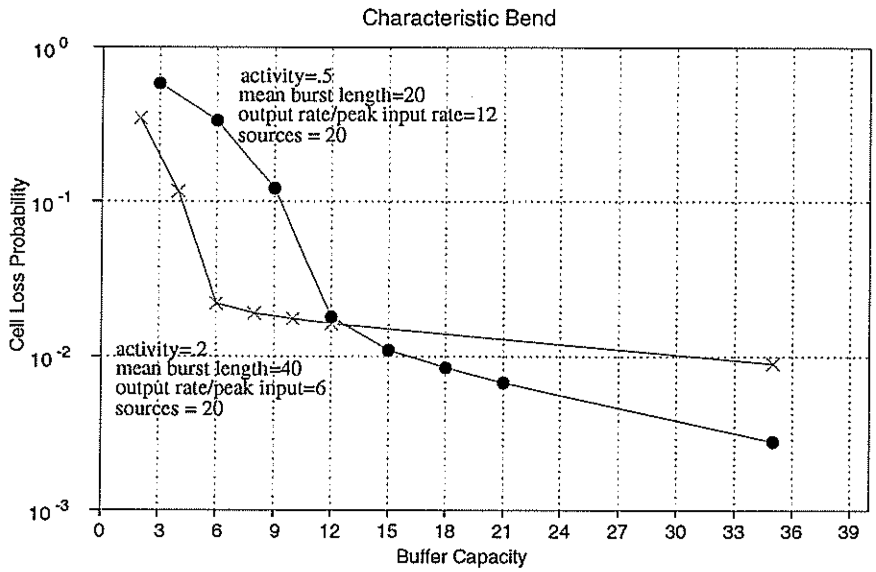

Figure 3: An example of the importance of the $\frac{\text{output service rate}}{\text{the maximum input rate}}$ in the performance of a multiplexer

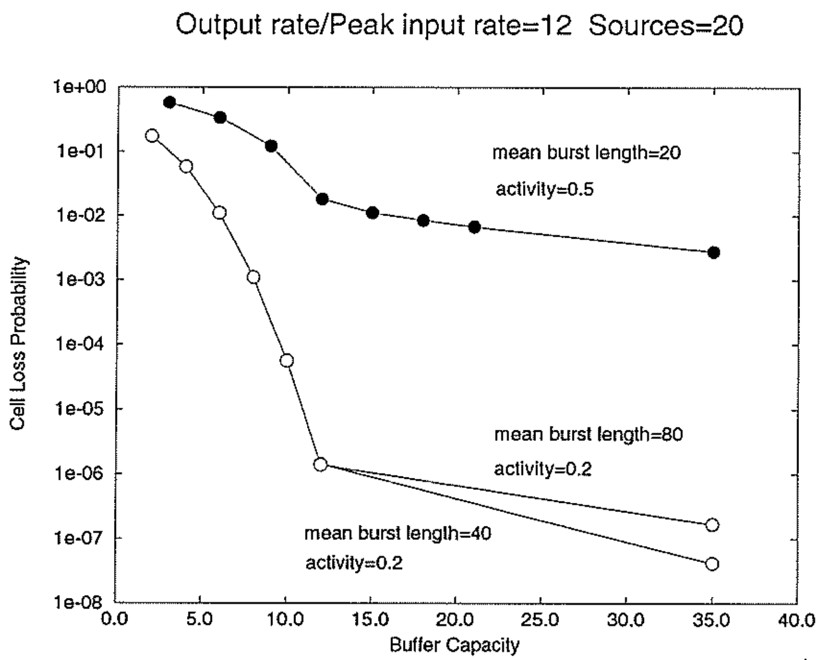

Figure 4: A further example of the importance of the $\frac{\text{output service rate}}{\text{the maximum input rate}}$ in the performance of a multiplexer

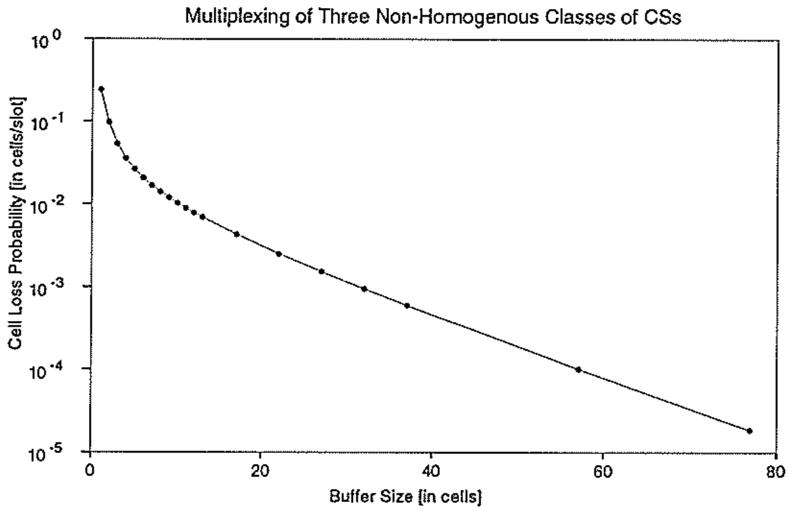

Figure 5: Cell loss probability for the three non-homogeneous CS class case

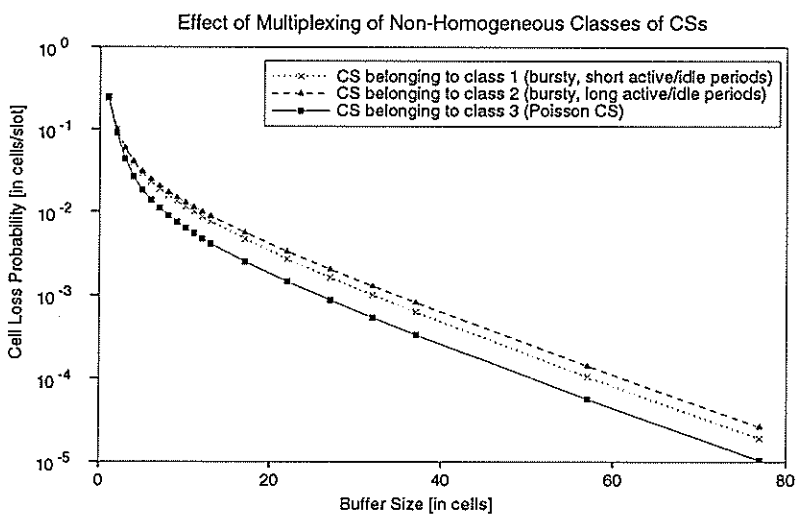

Figure 6: Effect of the multiplexing on each of the non-homogeneous CS classes

Figure 7 ~~Table 1~~. Dependence of the cell loss probability on the buffer length

Multiplexer 140 Mb, 300 binomial sources 2 Mb activity 0.2, 10 approximating buffer states

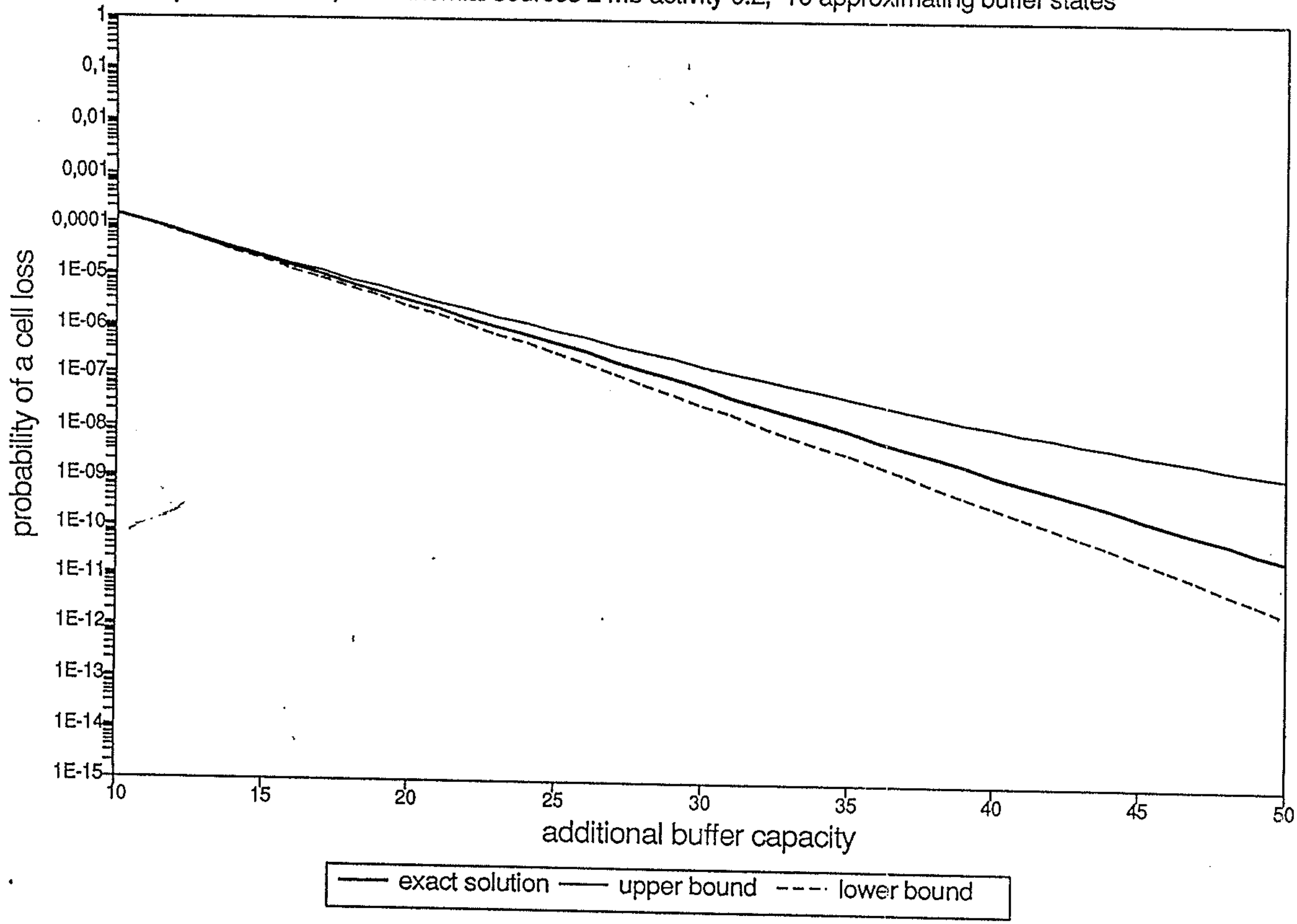

Figure 8

~~Table~~ 2. Multiplexer dimensioning for binomial sources

Multiplexer 140 Mb, sources 2 Mb, buffer length 50

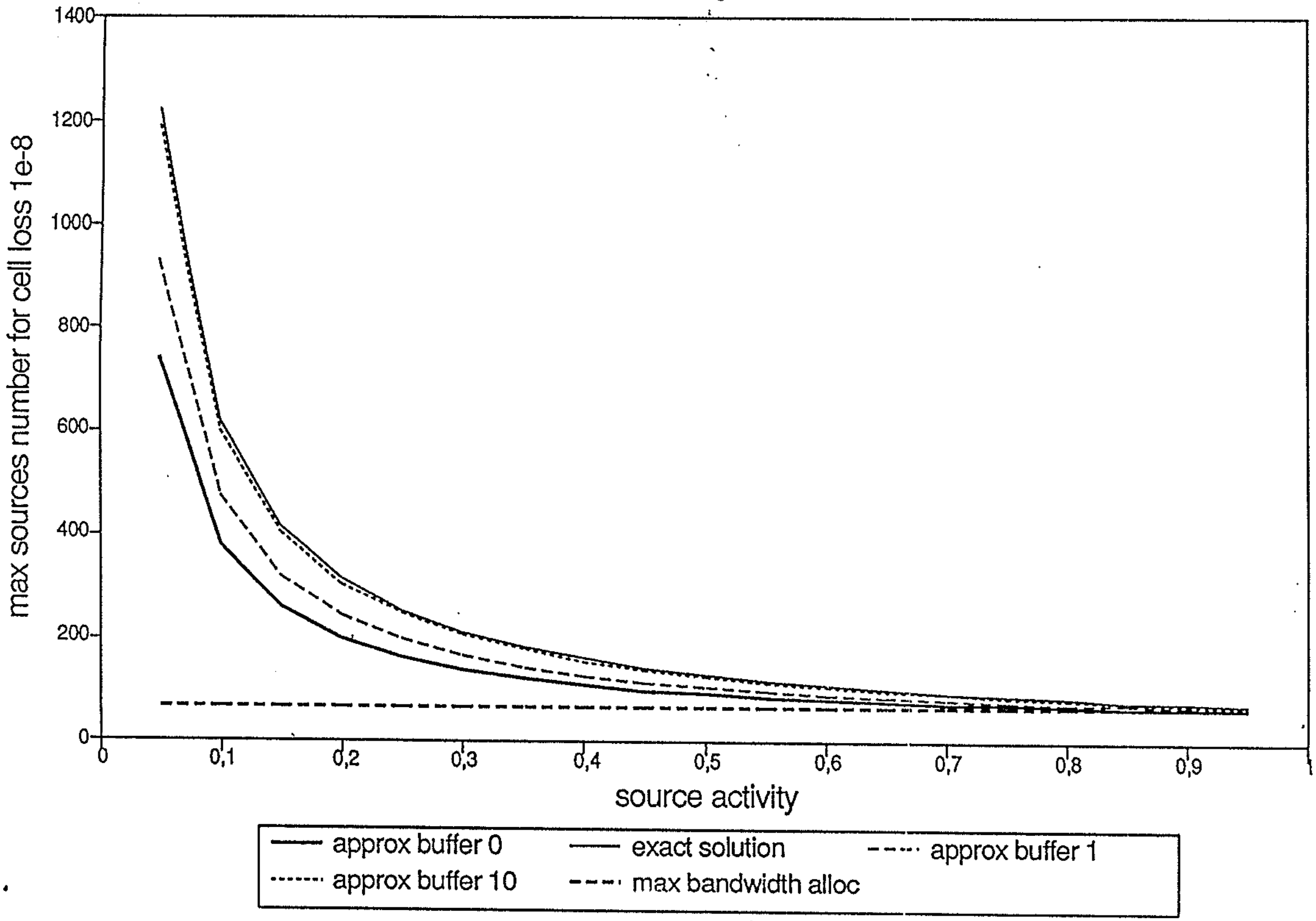

Figure 9

~~Table 7~~: Multiplexer dimensioning for on-off sources
multiplexer 140 Mb, source 2 Mb, activity 0.1