Robust Dynamic Admission Control for Unified Cell and Call QoS in Statistical Multiplexers*

Debasis Mitra
Bell Laboratories
Lucent Technologies
Murray Hill, NJ

Martin I. Reiman
Bell Laboratories
Lucent Technologies
Murray Hill, NJ

Jie Wang
AT&T Labs
Holmdel, NJ

November 18, 1997
Revised: March 12, 1998

Abstract

The design of connection admission control (CAC) for a simple Markovian model of a multiservice statistical multiplexer is considered. The paper begins by laying the foundation through several fundamental analytic concepts, such as a Semi-Markov Decision Process formulation of the design problem and time scale decomposition, before progressively leading up to real world requirements, like robustness and simplicity of design. Several numerical illustrations are given. The salient contributions of the paper are : (1) A unified treatment of multiclass cell and call QoS. (2) A CAC design which is robust, i.e., fair and efficient. (3) Simplicity in the CAC design, together with an evaluation of the tradeoff with performance. (4) An analytic technique for computing the feasibility region in the space of call arrival rates where some control exists to satisfy QoS. (5) The discovery of near linearity of the boundary of the feasible region, which is then used to decompose the design problem. (6) A unified treatment of aggressive and conservative forms of CAC, the latter being conventional and the former yielding better call level performance. (7) An effective bandwidth definition based on the aggressive form of CAC, which influences the CAC design. (8) Demonstration of the beneficial impact on performance of cell level control. (9) An asymptotic theory of the joint behavior of cell loss and call blocking. (10) A rigorous development of time scale decomposition. (11) A numerical evaluation of the accuracy of the notion of Nearly Completely Decomposable Markov chains.

1 Introduction

This paper considers the design of connection admission control (CAC) for a simple Markovian model of a multiservice statistical multiplexer. The dominant characteristic of the CAC considered here is that it addresses both cell and call Quality of Service (QoS) issues, which is different from the conventional focus on the cell level only. The paper begins by dealing rigorously with several fundamental analytic concepts, such as time scale decomposition between the burst and call level and a Semi-Markov Decision Process formulation

*Partial results have been reported in the 34th IEEE Conference on Decision and Control [20] and the 15th International Teletraffic Congress [17].

of the call admission control problem. This is accompanied by several numerical examples. The paper progressively leads up to real world requirements, such as robustness and simplicity of the CAC design. These are meta-problems for which there are no simple problem statements, yet good designs are usually easy to identify. For the analytically intractable design issues, in the latter part of the paper, we rely on a combination of sound fundamental bases, empiricism and validation.

First, why combine cell and call issues in CAC design? Because ultimately, the service provider will dimension facilities to meet both cell and call level QoS requirements (where the call level QoS requirement is that the blocking probability of each class be below some level), and ignoring call level QoS in CAC design will lead to overengineered systems, as we now argue. Recall that the principal goal of connection admission control in a broadband system is to simultaneously maintain the QoS for several traffic streams with different characteristics. There is a substantial literature in this area (see [13], [21], [11], [23] and references therein). With very few exceptions, such as [4] and [11], the published work has focused on cell level QoS, such as cell loss ratio, cell delay and cell delay variation. Most CAC schemes are based on the well known concept of effective bandwidth, which has been studied extensively (see [3], [6], [10], [14]). The formulation of effective bandwidth in the literature is, in turn, mainly based on cell level QoS. The basic call admission principle is tacitly *complete sharing* (CS), i.e., a new call is admitted if by doing so the previously agreed cell level QoS of this and other calls already in progress will not be violated. However, in order to take advantage of statistical multiplexing, it is necessary to carry traffic with different statistical characteristics on the same network and CS can cause a significant difference in call blocking probabilities between sources with different bandwidth requirements. Thus, in order to meet the call level QoS requirement for higher bandwidth calls under CS, the facility size will be such that low bandwidth calls will see call blocking probabilities substantially lower than required. Hence, in order to obtain efficient resource sharing, it is necessary to consider not only the cell level QoS but also the call level QoS when designing the CAC. This leads naturally to call admission control schemes that are not CS. In this paper we propose two ways to take call level QoS into consideration when studying call admission control. First, we introduce a new measure of bandwidth requirement for connections that takes into account both cell and call level QoS requirements. Secondly, based on the new bandwidth requirement, which we call (yet another) *effective bandwidth*, we propose a new method for call admission control, which guarantees not only cell level QoS, but also regulates traffic at the call level to meet the call level QoS requirements in a robust manner.

There is a particular notion of robustness employed here. For each service or traffic class we have a corresponding engineered load, with the complete set of such loads forming part of the input to the CAC design problem. In reality, of course, the offered load for a class may be more or less than its engineered load, and we use the terms "overload" and "underload", respectively, to distinguish the loading status of the class. Our notion of robustness and fairness requires the underloaded classes to be protected, i.e., their received QoS is only marginally affected when the total load exceeds the engineered level. This protective or isolating feature pulls the design in the direction of *complete partitioning*, which is at the other end of the spectrum from complete sharing. The CAC design we propose here is based on the resource-sharing technique of *virtual partitioning* (VP) [18] [5]. This technique aims to achieve controlled sharing, which strikes a balance between unrestricted sharing and isolation. Instead of each class of traffic having a fixed priority, as in traditional trunk reservation ([1], [15]), in virtual partitioning the priorities depend on the state of the system.

In the first part of the paper we examine several topics that are important for providing the bases of our CAC design. We begin in Section 2 by introducing our model, which consists of a single bufferless link with multiple call classes. The call classes may have quite different peak rates and burstiness properties, and each call behaves as an on-off fluid source while in the system. We then consider the following optimization problem: Given some maximum tolerable cell loss, devise a call admission control procedure that provides acceptable cell loss and maximizes the revenue due to carried traffic. We consider the optimization problem both with and without constraints on the call blocking probabilities. The optimization problem is formulated as a Semi-Markov Decision Process (SMDP) with constraints.

For any system of realistic size, the optimization problem of Section 2 is so computationally intensive that, in effect, it is numerically impractical. This leads us to seek some method that would enable us to reduce the computational burden. There is a natural time scale decomposition in our system that arises due to the large disparity between the lifetime of a burst and the lifetime of a call. This time scale decomposition allows us to apply the notion of Nearly Completely Decomposable (NCD) Markov chains ([2],[11],[9]) to our system. The NCD approximation reduces the dimension of the state, and the reduced state optimization is again formulated as an SMDP in Section 3. The reduced state optimization problem is numerically feasible for systems of realistic size. Two related versions of the problem are considered: conservative and aggressive. In the former approach we require that the cell loss constraints are satisfied for every state, i.e., the number of calls of each class in progress. In the aggressive approach, we only require that the cell loss constraints be satisfied on a long run average basis. Next, also in Section 3, we describe how cell level control, i.e., selective cell discarding based on the cell class, can be used in a two-class problem to reduce two loss constraints to one constraint. Accompanying numerical results show that the gain in capacity from cell level control is significant, especially if the required cell loss ratios are quite different. (Although the cell level control does not play an explicit role in the design of our simple CAC, in formulating the constraints on cell level behavior we implicitly assume that an optimal cell level control is being used.) The final topic considered in Section 3 is the accuracy of the NCD approximation. Specifically, we give results and insights from a numerical investigation on the solution of the SMDP problem, with and without the NCD approximation. The results indicate that the NCD approximation works well when the average number of on-off cycles during a call's holding time is about 100 or larger.

Section 4 is a short, but important, bridge between the first half of the paper, which is dominated by the SMDP and NCD concepts, and the second half, where the focus is on simple and robust CAC designs. In this section we ask if, given the statistical parameters of the burst and call level behaviors of each of two classes, and the cell and call QoS parameters, there exists a CAC such that the QoS constraints are satisfied ? This is a feasibility question and, not surprisingly, it is determined by the feasibility of an LP. The answer to the question is exhibited as a region, called the "feasibility region", in λ- space, i.e., where the coordinates are the arrival rates, λ_i, of calls. An important discovery is that the boundary of the feasible region exhibits near - linearity. Exploitation of this feature allows multiple class problems to be treated in the next section as multiple single class problems. Another feature of the numerical results of this section is that the feasible region for the aggressive approach is significantly larger than for the conservative approach. With this as motivation, we consider only the aggressive approach in the remainder of the paper.

Starting with Section 5 we consider the realization of simple and robust CAC for feasible λ. Section 5 considers a single class system and proposes the new effective bandwidth.

Importantly, the effective bandwidth as calculated here is less conservative than traditionally calculated values which ignore call dynamics. This is because in the latter the worst case call configuration dominates, while here the call distribution is taken into account. An asymptotic analysis of the system is given in Section 6 for the scaling in which the call arrival rates and the link bandwidth are simultaneously made large. Although these asymptotics are not used for our CAC design, they provide a valuable insight into the parameter scalings that arise as a consequence of the qualitatively different scalings in the cell and call level QoS constraints. In Section 7 we discuss the standard policies of complete sharing and trunk reservation. Virtual partitioning is introduced in Section 8. In Section 9 we present our numerical results on the performances of various call admission control policies, with emphasis on robustness. Our results clearly indicate that our proposed CAC is able to protect the underloaded class in the presence of traffic that deviates from the engineered load, and that neither complete sharing nor trunk reservation provides this protection. New directions are discussed in Section 10.

We end this section by summarizing the salient contributions of this paper. (1) A unified treatment of multiclass cell and call QoS in CAC design. (2) The concept of robust CAC design based on a balance of fairness and efficiency. (3) Simplicity of design. The trade-off between simplicity and performance is highlighted in the numerical work, where it is shown that a small portion of the feasibility region is traded for simplicity. (4) An analytic technique for computing the feasibility region. The technique handles small cell loss probabilities, such as 10^{-9}, which simulations cannot. (5) The discovery of near-linearity of the boundary of the feasibility region. This is an example of the empirical content of the paper. (6) A unified treatment of conservative and aggressive forms of CAC. The conservative form has been conventionally accepted, yet the paper provides solid grounds based on performance for favoring the latter. (7) An effective bandwidth definition based on the aggressive form of CAC. (8) The concept of cell level control and a quantification of the gain in performance. (9) The asymptotic theory of the joint behavior of cell loss and call blocking, where the scaling is for exponentially small cell loss and critical loading at the call-level. (10) A rigorous development of time scale decomposition. (11) A numerical evaluation of the accuracy of the NCD technique.

2 Call Admission Control: A Semi-Markov Decision Process Formulation

2.1 The Model

We consider the call admission control problem for a statistical multiplexer system that consists of a single link with transmission rate R and no buffer. There are J classes of calls arriving according to independent Poisson processes, and class j calls arrive at rate λ_j, $j = 1, \ldots, J$. After being admitted into the system, a class j ($1 \leq j \leq J$) call behaves according to a two state Markov process depicted in Figure 1. The two states represent "on" (bursting) and "off". The initial state is "on" with probability r_j, and "off" with probability $1 - r_j$. The mean holding times in the "on" and "off" states are β_j^{-1} and α_j^{-1} respectively. A call leaving the "on" ("off") state has a probability of q_j^1 (q_j^0) of departing from the system. When a class j call is in the "on" state, it generates cells as fluid at rate ν_j. When a class j call is admitted, the system collects a fixed amount of revenue w_j. Note that, in contrast with [8], where a decision theoretic framework is used to deal with unknown parameters, we are assuming that the network has information on the statistical characteristics of the traffic at the burst and call levels.

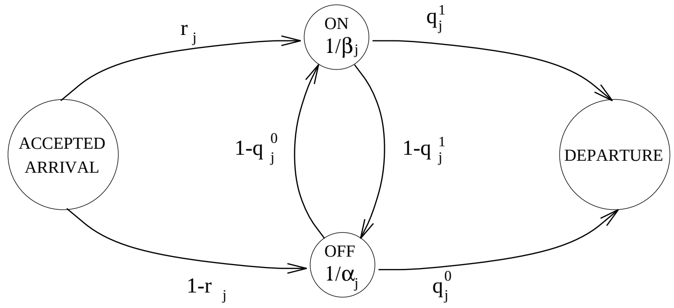

Figure 1: A Markov Chain Describing Call and Burst Dynamic

Based on the above model, we can calculate the average call holding time for class j, $1/\mu_j$, as a time to exit a two state transient semi Markov process. The states are 0(off), and 1(on), and the transition matrix of the embedded Markov Chain is

$$P^{(j)} = \begin{pmatrix} 0 & 1-q_j^0 \\ 1-q_j^1 & 0 \end{pmatrix}.$$

The mean residence time in state 0 is $1/\alpha_j$, and the mean residence time in state 1 is $1/\beta_j$. Let $N^{(j)} = (I - P^{(j)})^{-1}$. Then

$$\frac{1}{\mu_j} = (1-r_j)[N_{00}^{(j)}/\alpha_j + N_{01}^{(j)}/\beta_j] + r_j[N_{10}^{(j)}/\alpha_j + N_{11}^{(j)}/\beta_j].$$

A straightforward calculation shows that

$$N^{(j)} = \frac{1}{1-(1-q_j^0)(1-q_j^1)} \begin{pmatrix} 1 & 1-q_j^0 \\ 1-q_j^1 & 1 \end{pmatrix},$$

so that

$$\frac{1}{\mu_j} = \frac{\beta_j(1-r_jq_j^1)+\alpha_j(1-(1-r_j)q_j^0)}{\alpha_j\beta_j[1-(1-q_j^0)(1-q_j^1)]}. \tag{1}$$

The fraction of time a class j call spends in the "on" state is

$$p_j = \frac{\alpha_j(1-(1-r_j)q_j^0)}{\alpha_j\beta_j[1-(1-q_j^0)(1-q_j^1)]}. \tag{2}$$

When cells are generated at a rate exceeding the link transmission capacity, those cells that cannot be transmitted are lost.

Two types of Quality of Service (QoS) are of interest: the cell level QoS and the call level QoS. At the cell level, we want the cell loss ratio to be smaller than p^{cell}. A commonly used bound is $p^{\text{cell}} = 10^{-9}$. The call level QoS is reflected in call blocking probability bounded by p^{call}.

2.2 The Semi-Markov Decision Process (SMDP) and Linear Programming

Let k_j be the number of class j calls in progress and n_j the number of class j calls in the "on" state. Then for a stationary admission control policy, the process is Markovian and $(\mathbf{k},\mathbf{n}) = (k_1,\ldots,k_J; n_1,\ldots,n_J)$ is a state descriptor. When there are n_j class j calls in the "on" state, $j = 1,\ldots,J$, the total cell loss rate is $[\sum_{j=1}^J n_j\nu_j - R]^+$, while the cell arrival rate is $\sum_{j=1}^J n_j\nu_j$.

The first call admission control we consider here is the solution to the following optimization problem: maximize the long run average revenue while satisfying the cell level QoS requirement. We formulate this problem as a SMDP and use Linear Programming to solve it. We also consider the same problem with the addition of call level QoS requirements.

2.2.1 State Space and Action Sets

Let I denote the state space, which is the union of two sets, the set of call arrival states and the set of states corresponding to the rest of the events. A call arrival state has the format $(\mathbf{k},\mathbf{n},j)$ which corresponds to the arrival of a new class j call that finds $\mathbf{k}$ calls in progress of which $\mathbf{n}$ calls are in the "on" state. The rest of the events correspond to either a call departure, a call being turned "on", or a call being turned "off". We use $(\mathbf{k},\mathbf{n})$ to denote a state corresponding to those events, which indicates that there are $\mathbf{k}$ calls in progress of which $\mathbf{n}$ calls are in the "on" state *after* the event. In order to obtain a solution to the SMDP from linear programming, the state space must be finite. Let M be a fixed large positive integer. We thus consider as our state space $I(M)$, where

$$I(M) = \left\{(\mathbf{k},\mathbf{n},j) : k_i, n_i \in Z^+,\ n_i \le k_i,\ 1 \le i \le J;\ 1 \le j \le J;\ \text{ and } \sum_{i=1}^J k_i \le M\right\}$$

$$\cup \left\{(\mathbf{k},\mathbf{n}) : k_i, n_i \in Z^+,\ n_i \le k_i,\ 1 \le i \le J;\ \text{ and } \sum_{i=1}^J k_i \le M\right\}.$$

For each state in the first set, $(\mathbf{k},\mathbf{n},j)$, the corresponding action set is $A(\mathbf{k},\mathbf{n},j) = \{0,1\}$, where 0 denotes rejection and 1 denotes acceptance. For states of the form $(\mathbf{k},\mathbf{n})$, there is no action to be taken and we set $A(\mathbf{k},\mathbf{n}) = \{0\}$.

2.2.2 Mean Sojourn Times

Let $\tau[(\mathbf{k},\mathbf{n},j);a]$ denote the average time spent in state $(\mathbf{k},\mathbf{n},j)$ until next decision epoch if action a is taken; and $\tau[(\mathbf{k},\mathbf{n});0]$ denote the average time spent in state $(\mathbf{k},\mathbf{n})$ until next decision epoch. In addition, let $S(\mathbf{k},\mathbf{n}) = \sum_{j=1}^J \lambda_j + \sum_{j=1}^J n_j\beta_j + \sum_{j=1}^J (k_j - n_j)\alpha_j$. Then

$$\tau((\mathbf{k},\mathbf{n},j);a) = \begin{cases} \frac{r_j}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j)} + \frac{1-r_j}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n})} & a = 1, \\ \\ \frac{1}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n})} & a = 0. \end{cases}$$

$$\tau[(\mathbf{k},\mathbf{n});0] = \frac{1}{S(\mathbf{k},\mathbf{n})}.$$

2.2.3 Transition Probabilities

Let $p[(\mathbf{k},\mathbf{n},j),\mathbf{i},a]$ denote the transition probability from state $(\mathbf{k},\mathbf{n},j)$ to $\mathbf{i}$ if action a is chosen, where $a = 0,\ 1$; and $p[(\mathbf{k},\mathbf{n}),\mathbf{i}]$ denote the transition probability from state $(\mathbf{k},\mathbf{n})$ to $\mathbf{i}$. Recall that, for state $(\mathbf{k},\mathbf{n},j)$, there are $\mathbf{k}$ calls in progress of which $\mathbf{n}$ calls are in the "on" state *before* the arrival event; and for state $(\mathbf{k},\mathbf{n})$ there are $\mathbf{k}$ calls in progress of which $\mathbf{n}$ calls are in the "on" state *after* the event.

Recall that r_j is the probability that an accepted class j call starts in "on" state first. Hence we have for any $k_j \geq n_j \geq 0$,

$$p[(\mathbf{k},\mathbf{n},j),\mathbf{i},1] = \begin{cases} \frac{r_j\lambda_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j)} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j,l)\ \ 1\leq l\leq J \\ \frac{(1-r_j)\lambda_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n})} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j,\mathbf{n},l)\ \ 1\leq l\leq J \\ \frac{r_j(1-q_l^1)(n_l+1_{\{j=l\}})\beta_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j)} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j-\mathbf{e}_l)\ \ 1\leq l\leq J \\ \frac{(1-r_j)(1-q_l^1)n_l\beta_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n})} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j,\mathbf{n}-\mathbf{e}_l)\ \ 1\leq l\leq J \\ \frac{r_jq_l^1(n_l+1_{\{j=l\}})\beta_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j)} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j-\mathbf{e}_l,\mathbf{n}+\mathbf{e}_j-\mathbf{e}_l)\ \ 1\leq l\leq J \\ \frac{(1-r_j)q_l^1n_l\beta_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n})} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j-\mathbf{e}_l,\mathbf{n}-\mathbf{e}_l)\ \ 1\leq l\leq J \\ \frac{r_j(1-q_l^0)(k_l-n_l-1_{\{j=l\}})\alpha_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j)} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j+\mathbf{e}_l) \\ \frac{(1-r_j)(1-q_l^0)(k_l-n_l)\alpha_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n})} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j-\mathbf{e}_l,\mathbf{n}+\mathbf{e}_j) \\ \frac{r_jq_l^0(k_l-n_l-1_{\{j=l\}})\alpha_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n}+\mathbf{e}_j)} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j-\mathbf{e}_l,\mathbf{n}+\mathbf{e}_j) \\ \frac{(1-r_j)q_l^0(k_l-n_l)\alpha_l}{S(\mathbf{k}+\mathbf{e}_j,\mathbf{n})} & \mathbf{i} = (\mathbf{k}+\mathbf{e}_j-\mathbf{e}_l,\mathbf{n}) \\ 0 & \text{otherwise.} \end{cases}$$

$$p[(\mathbf{k},\mathbf{n},j),\mathbf{i},0] = p[(\mathbf{k},\mathbf{n}),\mathbf{i}] = \begin{cases} \frac{\lambda_j}{S(\mathbf{k},\mathbf{n})} & \mathbf{i} = (\mathbf{k},\mathbf{n},j) \\ \\ \frac{q_j^1 n_j \beta_j}{S(\mathbf{k},\mathbf{n})} & \mathbf{i} = (\mathbf{k},\mathbf{n}-\mathbf{e}_j) \\ \\ \frac{(1-q_j^1) n_j \beta_j}{S(\mathbf{k},\mathbf{n})} & \mathbf{i} = (\mathbf{k}-\mathbf{e}_j,\mathbf{n}-\mathbf{e}_j) \\ \\ \frac{q_j^0 (k_j - n_j)\alpha_j}{S(\mathbf{k},\mathbf{n})} & \mathbf{i} = (\mathbf{k},\mathbf{n}+\mathbf{e}_j) \\ \\ \frac{(1-q_j^0)(k_j - n_j)\alpha_j}{S(\mathbf{k},\mathbf{n})} & \mathbf{i} = (\mathbf{k}-\mathbf{e}_j,\mathbf{n}) \\ \\ 0 & \text{otherwise.} \end{cases}$$

2.2.4 Rewards

The reward can only be earned in state (k,n,a) when action $a=1$ is taken.

$$r[(\mathbf{k},\mathbf{n},j);a] = \begin{cases} 0 & a=0, \\ \\ w_j & a=1. \end{cases}$$

$$r[(\mathbf{k},\mathbf{n});0] = 0.$$

2.2.5 Cell Loss and Arrival Rates

Recall the previously stated cell loss and arrival rates when n_j class j calls are in the "on" state. Let $l_m[(\mathbf{k},\mathbf{n},j);a]$ and $g_m[(\mathbf{k},\mathbf{n},j);a]$ denote the expected amount of cells of class m lost and arriving until next decision epoch when in state $(\mathbf{k},\mathbf{n},j)$ and action a is taken; $l_m[(\mathbf{k},\mathbf{n})]$ and $g_m[(\mathbf{k},\mathbf{n})]$ denote the expected amount of cells of class m calls lost and arriving in state $(\mathbf{k},\mathbf{n})$ until next decision epoch. Then, with no priority cell discarding,

$$\begin{aligned} \frac{l_m[(\mathbf{k},\mathbf{n},j);1)]}{\tau[(\mathbf{k},\mathbf{n},j);1]} &= r_j[\sum_{i=1}^{J} n_i\nu_i + \nu_j - R]^+ \frac{n_m\nu_m + 1_{\{m=j\}}\nu_m}{\sum_{i=1}^{J} n_i\nu_i + \nu_j} + (1-r_j)[\sum_{i=1}^{J} n_i\nu_i - R]^+ \frac{n_m\nu_m}{\sum_{i=1}^{J} n_i\nu_i}, \\ \frac{l_m[(\mathbf{k},\mathbf{n},j);0)]}{\tau[(\mathbf{k},\mathbf{n},j);0]} &= \frac{l_m[(\mathbf{k},\mathbf{n});0]}{\tau[(\mathbf{k},\mathbf{n});0]} = [\sum_{i=1}^{J} n_i\nu_i - R]^+ \frac{n_m\nu_m}{\sum_{i=1}^{J} n_i\nu_i}, \\ \frac{g_m[(\mathbf{k},\mathbf{n},j);1)]}{\tau[(\mathbf{k},\mathbf{n},j);1]} &= n_m\nu_m + 1_{\{m=j\}} r_j\nu_j, \\ \frac{g_m[(\mathbf{k},\mathbf{n},j);0)]}{\tau[(\mathbf{k},\mathbf{n},j);0]} &= \frac{g_m[(\mathbf{k},\mathbf{n});0]}{\tau[(\mathbf{k},\mathbf{n});0]} = n_m\nu_m. \end{aligned}$$

2.2.6 Linear Programming Formulation

Linear programming is a classical technique for solving both Markov decision processes and semi-Markov decision processes, with and without constraints, c.f. [22]. With multiple constraints in an SMDP setting, Feinberg [7] shows that it suffices to consider randomized stationary policies for this problem, and the optimal solution subject to class l calls' long-run average cell loss ratio $\leq p_l^{\text{cell}}$ can be obtained from a linear program. The variables in the LP are $z_{\mathbf{i}a}$, $\mathbf{i} \in I(M)$, $a \in A(\mathbf{i})$, where $z_{\mathbf{i}a}\tau(\mathbf{i}, a)$ corresponds to the fraction of time spent in state $\mathbf{i}$ with action a chosen. The LP is

$$\text{Maximize} \sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} r(\mathbf{i}, a) z_{\mathbf{i}a}$$

subject to

$$\begin{aligned}
\sum_{a\in A(\mathbf{j})} z_{\mathbf{j}a} - \sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} p(\mathbf{i}, \mathbf{j}, a) z_{\mathbf{i}a} &= 0, \quad \mathbf{j} \in I(M) \\
\sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} [l_j(\mathbf{i}, a) - p_j^{\text{cell}} g_j(\mathbf{i}, a)] z_{\mathbf{i}a} &\leq 0, \quad 1 \leq j \leq J \\
\sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} \tau(\mathbf{i}, a) z_{\mathbf{i}a} &= 1, \\
z_{\mathbf{i}a} &\geq 0, \quad i \in I(M), \quad a \in A(\mathbf{i}).
\end{aligned}$$

The above LP is feasible, because $z_{\mathbf{0},0} = [\tau(\mathbf{0}, 0)]^{-1} = \sum_j \lambda_j$, and $z_{\mathbf{i}a} = 0$ otherwise is a feasible solution (corresponding to never allowing any calls to enter). Let $\phi(\mathbf{k}, \mathbf{n}, j)$ denote the probability that action 1 (accept) is chosen in state $(\mathbf{k}, \mathbf{n}, j)$. Given an optimal solution, z, of the LP, we obtain an optimal randomized stationary policy as

$$\phi(\mathbf{k}, \mathbf{n}, j) = \frac{z_{(\mathbf{k},\mathbf{n},j);1}}{z_{(\mathbf{k},\mathbf{n},j),0} + z_{(\mathbf{k},\mathbf{n},j);1}}$$

if $z_{(\mathbf{k},\mathbf{n},j),0} + z_{(\mathbf{k},\mathbf{n},j);1} > 0$, and $\phi(\mathbf{k}, \mathbf{n}, j) = 0$ otherwise.

2.2.7 Both Call and Cell Level QoS Requirements

We now consider the above optimization problem with the J additional constraints that the call blocking probability for class j should not exceed p_j^{call}, $1 \leq j \leq J$. The treatment in [7] easily enables us to add these constraints to the linear program. It is now possible, however, that the LP will be infeasible: There may be no admission control rule that can simultaneously satisfy all cell level and call level quality of service requirements. (With $J = 2$ and all parameters other than λ_1 and λ_2 fixed, in Section 4 we provide a related LP for the reduced state problem introduced in Section 3, and numerically determine the set of (λ_1, λ_2) for which the LP is feasible.)

We define, for fixed M, the set of arrival states corresponding to class j calls as

$$I_j^A(M) = \left\{(\mathbf{k}, \mathbf{n}, j) : k_i,\ n_i \in Z^+,\ n_i \leq k_i, 1 \leq i \leq J;\ \sum_{i=1}^{J} k_i \leq M\right\},\ 1 \leq j \leq J\,.$$

Recall that, in the LP formulation, $\tau(\mathbf{i}, a) z_{\mathbf{i}a}$ corresponds to the fraction of time spent in state i with action a chosen. Thus, the constraint on class j blocking probability is

$$\sum_{\mathbf{i}\in I_j^A(M)} z_{i0} - p_j^{\text{call}} \lambda_j \leq 0\,.$$

The full linear program with both cell and call QoS constraints is thus

$$\text{Maximize} \sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} r(\mathbf{i}, a) z_{\mathbf{i}a}$$

subject to

$$\sum_{a\in A(\mathbf{j})} z_{\mathbf{j}a} - \sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} p(\mathbf{i},\mathbf{j},a) z_{\mathbf{i}a} = 0,\ \mathbf{j}\in I(M)$$

$$\sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} \left[\ell_j(\mathbf{i},a) - p_j^{\text{cell}} g_j(\mathbf{i},a)\right] z_{\mathbf{i}a} \le 0,\ 1\le j\le J$$

$$\sum_{\mathbf{i}\in I_j^A(M)} z_{\mathbf{i}0} - p_j^{\text{call}}\lambda_j \le 0,\ 1\le j\le J$$

$$\sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} \tau(\mathbf{i},a) z_{\mathbf{i}a} = 1$$

$$z_{\mathbf{i}a}\ge 0,\quad i\in I(M),\ a\in A(\mathbf{i})\,.$$

3 Nearly Complete Decomposability: Implications and Accuracy

The SMDP introduced in Section 2 is $2J$ dimensional for a J class system. For $J > 1$, this leads to computational problems of excessive size. The notion of Nearly Completely Decomposable (NCD) Markov chains can be used to reduce this to a J-dimensional problem. (Although the discussion presented here is heuristic, the problem reduction is rigorously justified by Theorem 11 of [2].) Intuitively, this limiting regime is a good approximation when the process describing the number of calls in the "on" state reaches equilibrium between any change in the number of calls in progress due to call arrival/departure. Hence, when making a call admission decision, the number of calls of each class in progress is important, but the number of calls of each class in the on state is not, because these quantities oscillate too rapidly.

We handle the above idea mathematically as follows. Note that the mean time between call events (call arrival and departure) of class i is $1/(k_i\mu_i + \lambda_i)$. For this quantity to be large relative to the mean burst cycle time, as is required for the NCD limiting regime to hold, it is required that

$$\left(\frac{\alpha_i\beta_i}{\alpha_i+\beta_i} - \lambda_i\right)/\mu_i >> k_i \quad 1\le i\le J. \tag{3}$$

Now consider a family of system indexed by $\epsilon\to 0$, in which $\alpha_i(\epsilon) = O(1)$, $\beta_i(\epsilon) = O(1)$ and $r_i(\epsilon) = O(1)$. Assume that

$$q_i^1(\epsilon) = \epsilon q_i^1 \ \text{ and } q_i^0(\epsilon) = \epsilon q_i^0.$$

Then since the average call duration is (see Section 2.2)

$$\frac{1}{\mu_i(\epsilon)} = \frac{1 - r_i q_i^1(\epsilon)}{\alpha_i(\epsilon)[1-(1-q_i^1(\epsilon))(1-q_i^0(\epsilon))]} + \frac{1-(1-r_i)q_i^0(\epsilon)}{\beta_i(\epsilon)[1-(1-q_i^1(\epsilon))(1-q_i^0(\epsilon))]},$$

it follows that $\mu_i(\epsilon) = O(\epsilon)$ and

$$\frac{\alpha_i(\epsilon)\beta_i(\epsilon)}{\alpha_i(\epsilon)+\beta_i(\epsilon)}\frac{1}{\mu_i(\epsilon)} = O\left(\frac{1}{\epsilon}\right). \tag{4}$$

Now let us further assume that

$$\frac{\lambda_i(\epsilon)}{\mu_i(\epsilon)} = o(\frac{1}{\epsilon}). \tag{5}$$

So that, from (4) and (5),

$$(\frac{\alpha_i\beta_i}{\alpha_i+\beta_i} - \lambda_i)/\mu_i = O(\frac{1}{\epsilon}). \tag{6}$$

On comparing (3) and (6), we see that the NCD approximation may apply even with a large number of calls in progress. Note that QoS considerations typically prevent the number of calls in progress from being excessively large relative to capacity, as we shall see in our applications. However, whenever the NCD approximation is used, it is necessary to check that the condition in (3) is satisfied. This is the case in the examples treated in the paper.

In the NCD limit, the holding time distribution of a call will be exponential for any on and off time distribution, as long as the memoryless process contained in our model for terminating a call is used. Although it would be possible to incorporate non-exponential call holding times, this would not be entirely straightforward. A Semi-Markov Decision Process formulation would require that we keep track of the elapsed holding time for each call in progress. (The current phase of each call would be sufficient for a phase-type holding time distribution.) The optimal policy would typically depend in a non-monotone manner on this elapsed holding time information, making it difficult to implement. An alternative would be to implement a policy that only depends on the number of calls in progress, and ignores the elapsed holding time information. Such a policy might do well, but it might not.

As $\epsilon \to 0$, the $\mathbf{n}$ component of the state becomes noise on the time scale where call arrival and departure rates are $O(1)$, and can be ignored for admission control purposes. This part of the state *does* affect the loss rate, so it must be 'averaged' properly. For ϵ small, the $\mathbf{n}$ process reaches equilibrium between changes in the $\mathbf{k}$ process. The equilibrium corresponds to fixed $\mathbf{k}$ and is given by the binomial distribution:

$$\psi(\mathbf{k},\mathbf{n}) = \prod_{i=1}^{J} \binom{k_i}{n_i} p_i^{n_i}(1-p_i)^{k_i-n_i} \ .$$

When the total cell arrival rate is $\sum_i n_i\nu_i$, the cell loss rate is $[\sum_i n_i\nu_i - R]^+$. The average class i cell loss rate with $\mathbf{k}$ is thus

$$b_i(\mathbf{k}) \ = \ \sum_{n_1=0}^{k_1} \cdots \sum_{n_J=0}^{k_J} \psi(\mathbf{k},\mathbf{n})[\sum_j n_j\nu_j - R]^+ \frac{n_i\nu_i}{\sum_j n_j\nu_j} \ . \tag{7}$$

This again reflects the fact that there is no priority — the cell loss rate of a class is proportional to its input rate. It seems intuitively clear that the exponential distribution for on and off times plays no essential role in (7). Indeed, the results of [2] apply for on and off times having any phase type distributions.

3.1 Formulation of the Reduced Problem as a SMDP

3.1.1 States, Actions, and Transition Probabilities

We now formulate the limit control problem as an SMDP. As in the original problem admission decisions are made upon call arrival. The state space I is again the union of two sets, corresponding to states associated with call arrivals and states associated with call departures. To make this a finite state problem we may need to place an a priori bound on the number of calls that can be accepted in the system. This is taken care of below. Arrival states take the form $(\mathbf{k}, j)$, with $k_j \geq 0$, and $1 \leq j \leq J$. The state $(\mathbf{k}, j)$ corresponds to an arrival of class j call when there are k_i class i calls in progress, $1 \leq i \leq J$. These are the only states in which decisions need to be made. The set of actions available is $A(\mathbf{k}, j) = \{0, 1\}$, where 0 denotes rejection and 1 denotes acceptance. The state $\mathbf{k}$ is a departure state where the departing call leaves k_i class i calls behind, $1 \leq i \leq J$. For states of the form $\mathbf{k}$ where no decision needs to be made we set $A(\mathbf{k}) = \{0\}$.

To complete the specification of the SMDP we need to provide transition probabilities, mean sojourn times, rewards, and costs for each state-action pair. Let $\tau(\mathbf{i}, a)$ denote the average time spent in state $\mathbf{i}$ (until the next decision epoch) if action $a \in A(\mathbf{i})$ is chosen. Then

$$\tau(\mathbf{k}, 0) = (\sum_{i=1}^{J} \lambda_i + k_i\mu_i)^{-1}$$

and

$$\tau((\mathbf{k}, j), a) \quad = \quad \begin{cases} (\sum_{i=1}^{J} \lambda_i + k_i\mu_i)^{-1}, & a = 0 \\ (\sum_{i=1}^{J} \lambda_i + k_i\mu_i + \mu_j)^{-1}, & a = 1 \ . \end{cases}$$

Let $p(\mathbf{i}, \mathbf{i}', a)$ denote the transition probability from state $\mathbf{i}$ to $\mathbf{i}'$ if action a is chosen. Let

$$\overline{p}(\mathbf{k}, \mathbf{i}) \quad = \quad \begin{cases} k_l\mu_l/(\sum_{i=1}^{J} \lambda_i + k_i\mu_i) & \text{if } \mathbf{i} = (\mathbf{k} - \mathbf{e}_l) \\ \lambda_l/(\sum_{i=1}^{J} \lambda_i + k_i\mu_i) & \text{if } i = (\mathbf{k}, l), \end{cases}$$

and 0 otherwise. Then

$$p(\mathbf{k}, \mathbf{i}, 0) = \overline{p}(\mathbf{k}, \mathbf{i}) \ ,$$

and

$$p((\mathbf{k}, j), \mathbf{i}, a) = \begin{cases} \overline{p}(\mathbf{k}, \mathbf{i}), & a = 0, \\ \overline{p}(\mathbf{k} + \mathbf{e}_j, \mathbf{i}), & a = 1. \end{cases}$$

3.1.2 The Conservative Approach to the Cell Loss Constraint

We formulate a conservative approach to the cell loss constraint as follows. Let

$$C = \{\mathbf{k} : b_i(\mathbf{k}) \leq p_i^{\text{cell}} k_i p_i \nu_i, \ \ 1 \leq i \leq J\}. \tag{8}$$

Then C is the set of $\mathbf{k}$ such that when there are k_i class i calls in the system *forever*, $1 \leq i \leq J$, the long run average cell loss rate constraints are always satisfied. The set C uniquely determines a state space I: for any j, $(\mathbf{k}, j) \in I$ if and only if $\mathbf{k} \in C$; and $\mathbf{k} \in I$ if and only if there exists a j such that $\mathbf{k} + \mathbf{e}_j \in C$. The action sets for the states on the boundary of I require minor modification: given $\mathbf{k} \in C$, if $\mathbf{k} + \mathbf{e}_j \notin C$, then $A((\mathbf{k}, j)) = \{0\}$, $1 \leq j \leq J$. In this manner C uniquely determines a SMDP. Note that given p_j^{cell}, $1 \leq j \leq J$, and R, the set C is always finite. The optimal policy obtained by solving this SMDP is conservative in terms of cell loss constraints because it will never go into any state for *any period of time* where the cell loss constraints will be violated if we stay there *forever*. This SMDP can be solved using either value iteration or LP. We used LP. For the 2 class system whose parameters are given in Table 1 (with $R = 45$), using CPLEX it took 88 seconds on a Sparc Server 3000 to solve the LP.

Class i	μ_i	ν_i	p_i	p_i^{cell}	p_i^{call}
1	0.1	6.0	0.025	10^{-9}	10^{-2}
2	1.0	1.5	0.100	10^{-9}	10^{-2}

Table 1: Parameters

3.1.3 An Aggressive Approach to the Cell Loss Constraint

We now describe an 'aggressive' approach to the cell loss constraint. In order to be able to apply standard numerical solution procedures to find the optimal policy, we need to make the state space of the SMDP finite. We achieve this by placing an a priori bound on the number of calls that can be accepted into the system. Let M be a fixed large positive integer. We consider a state space $I(M)$ of the form

$$I(M) = \{(\mathbf{k}, j) : k_i \in \mathbf{Z}^+, \quad i = 1, \ldots, J;\ 1 \le j \le J;\ \text{and } \textstyle\sum_{i=1}^J k_i \le M\}$$
$$\cup \{\mathbf{k} : k_i \in \mathbf{Z}^+, \quad i = 1, \ldots, J \quad \text{and } \textstyle\sum_{i=1}^J k_i \le M\}\,.$$

For states $(\mathbf{k}, j)$ such that $\sum_{i=1}^J k_i = M$, we let $A((\mathbf{k}, j)) = \{0\}$.

Since M was chosen arbitrarily to make the problem finite, we need to solve a series of problems with increasing M's, until the associated optimal policy and reward stop changing. Then the optimal policy for $M = \infty$ will have been obtained.

Let $r(\mathbf{k}, j, a)$ denote the reward earned in state $(\mathbf{k}, j)$ when action a is chosen. (The reward earned in states of the form $\mathbf{k}$ is 0.) Then

$$r(\mathbf{k}, j, a) = \begin{cases} 0, & a = 0 \\ w_j, & a = 1\,. \end{cases}$$

We also define 'costs' which will play a role in the constraint on loss probability.

Let

$$\bar{l}_\ell(\mathbf{k}) = b_\ell(\mathbf{k})\tau((\mathbf{k}), 0), \quad \ell = 1, \ldots, J\,.$$

We define

$$l_\ell((\mathbf{k}), 0) \;=\; \bar{l}_\ell(\mathbf{k})\,,$$

and

$$l_\ell((\mathbf{k}, j), a) \;=\; \begin{cases} \bar{l}_\ell(\mathbf{k}), & a = 0 \\ \bar{l}_\ell(\mathbf{k} + \mathbf{e}_j), & a = 1,\ j = 1, \ldots, J. \end{cases}$$

As defined above, $c_\ell(\mathbf{i}, a)$ is the expected amount of class ℓ traffic lost until the next decision epoch. Let

$$\begin{aligned} \bar{g}_\ell(\mathbf{k}) &= p_\ell k_\ell \alpha_\ell \tau((\mathbf{k}), 0)\,, \\ g_\ell(\mathbf{k}, 0) &= \bar{g}_\ell(\mathbf{k})\,, \end{aligned}$$

and

$$g_\ell((\mathbf{k},j),a) \;=\; \begin{cases} \overline{g}_\ell(\mathbf{k}), & a=0 \\ \overline{g}_\ell(\mathbf{k}+\mathbf{e}_j), & a=1,\; j=1,\ldots,J. \end{cases}$$

Then $g_\ell(\mathbf{i},a)$ is the expected amount of class ℓ traffic to arrive until the next decision epoch.

We can again use the results of [7], which enable us to obtain the optimal control from the following linear program:

$$\text{maximize} \quad \sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} r(\mathbf{i},a) z_{\mathbf{i}a} \tag{9}$$

subject to

$$\sum_{a\in A(\mathbf{j})} z_{\mathbf{j}a} - \sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} p(\mathbf{i},\mathbf{j},a) z_{\mathbf{i}a} = 0, \quad \mathbf{j}\in I(M), \tag{10}$$

$$\sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} [l_\ell(\mathbf{i},a) - p_\ell^{\text{cell}} g_\ell(\mathbf{i},a)] z_{\mathbf{i}a} \le 0, \quad \ell = 1,\ldots,J, \tag{11}$$

$$\sum_{\mathbf{i}\in I(M)} \sum_{a\in A(\mathbf{i})} \tau(\mathbf{i},a) z_{\mathbf{i}a} = 1, \tag{12}$$

$$z_{\mathbf{i}a} \ge 0\ , \qquad \mathbf{i}\in I(M),\; a\in A(\mathbf{i}). \tag{13}$$

As was the case in Section 2.2.6, the above LP is clearly feasible, because $z_{\mathbf{0},0} = [\tau(\mathbf{0},0)]^{-1} = \sum_{i=1}^J \lambda_i$, and $z_{\mathbf{i}a} = 0$ otherwise is a feasible solution. Let $\phi(\mathbf{k},j)$ denote the probability that action 1 (accept) is chosen in state $(\mathbf{k},j)$. Given an optimal solution, z, of the LP, we obtain an optimal randomized stationary policy as

$$\phi(\mathbf{k},j) = \frac{z_{(\mathbf{k},j),1}}{z_{(\mathbf{k},j),0} + z_{(\mathbf{k},j),1}}$$

if $z_{(\mathbf{k},j),0} + z_{(\mathbf{k},j),1} > 0$, and $\phi(\mathbf{k},j) = 0$ otherwise. Again using the parameters in Table 1 (with $R = 45$) and using CPLEX on a Sparc Server 3000 it took 402 seconds to solve the LP.

3.1.4 Cell Level Control

Both the aggressive and conservative approaches involve J cell level constraints, one for each class. In both cases it is sometimes possible to transform the problem to one involving one constraint using cell level control, which consists of deciding how many cells of each class are lost. This transformation will result in improved performance of the associated optimal control. For $J = 2$, equation (7) presents the cell loss rate under a 'no priority' assumption. Not surprisingly, typically only one of the two cell level constraints is tight in the solution. Thus, by giving priority to the class whose constraint is tight it seems clear that we can improve the solution. Related results and discussion are contained in Bean [3].

Consider the conservative approach for $J = 2$, for which the 'acceptance region', C, is given by (8). Let

$$b(\mathbf{k}) = \sum_{n_1=0}^{k_1} \sum_{n_2=0}^{k_2} \psi(\mathbf{k},\mathbf{n})[n_1\nu_1 + n_2\nu_2 - R]^+\ . \tag{14}$$

There is a cell level control (it will depend on k_1, k_2) that can achieve

$$b_i(\mathbf{k}) \le p_i^{\text{cell}} k_i p_i \nu_i\ , \quad i = 1,2 \tag{15}$$

if

$$b(k_1, k_2) \leq p_1^{\text{cell}} k_1 p_1 \nu_1 + p_2^{\text{cell}} k_2 p_2 \nu_2 \ , \tag{16a}$$

$$b(k_1, 0) \leq p_1^{\text{cell}} k_1 p_1 \nu_1 \ , \tag{16b}$$

$$\text{and} \quad b(0, k_2) \leq p_2^{\text{cell}} k_2 p_2 \nu_2 \ . \tag{16c}$$

Furthermore, if $p_1^{\text{cell}} = p_2^{\text{cell}}$, then (16a) implies (16b) and (16c), and the two constraints of (15) are replaced by the one constraint (16a). In this case, the acceptance region becomes

$$C = \{(k_1, k_2) : b(\mathbf{k}) \leq p^{\text{cell}}[k_1 p_1 \nu_1 + k_2 p_2 \nu_2]\} \ , \tag{17}$$

where $p^{\text{cell}} = p_1^{\text{cell}} = p_2^{\text{cell}}$.

The aggressive case is a bit more complicated because the constraints involve the stationary distribution over all states. Consider the constraint

$$\sum_{i \in I} \sum_{a \in A(i)} \left[c_1(i,a) + c_2(i,a) - p_1^{\text{cell}} c_3(i,a) - p_2^{\text{cell}} c_4(i,a)\right] z_{ia} \ \leq \ 0 \ . \tag{18}$$

For the case $p_1^{\text{cell}} = p_2^{\text{cell}}$, if we solve the LP (9), (10), (12), (13), (18), we can find a cell level control such that the constraints (11) are satisfied. A related result holds with $p_1^{\text{cell}} \neq p_2^{\text{cell}}$.

An indication of the advantage of cell level control can be seen by comparing the acceptance regions for the conservative scheme. Figure 2 displays the acceptance regions associated with one and two constraints, as given by equations (17) and (18), for the parameter values of the two classes given in Table 1 (with $R = 45$).

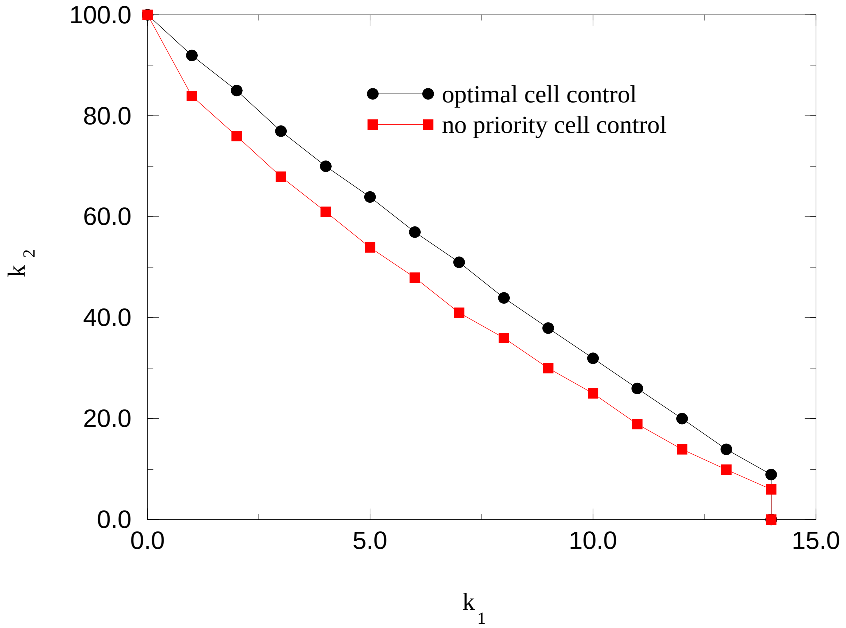

Figure 2: Acceptance Regions ($p_1^{\text{cell}} = p_2^{\text{cell}} = 10^{-9}$)

In Figure 3, where we use $p_1^{\text{cell}} = 10^{-9}$ and $p_2^{\text{cell}} = 10^{-7}$, the advantage of cell level control is more dramatic.

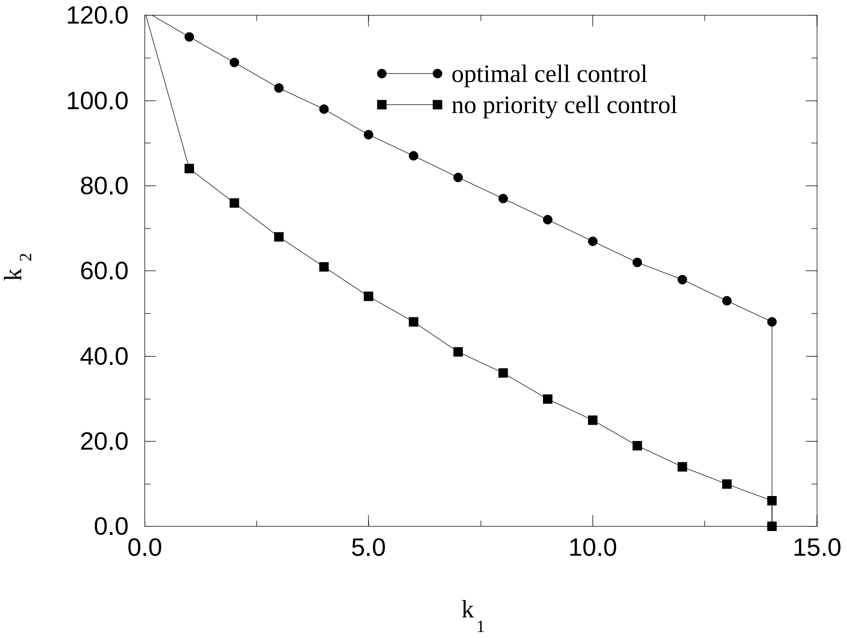

Figure 3: Acceptance Region ($p_1^{\text{cell}} = 10^{-9}, p_2^{\text{cell}} = 10^{-7}$)

3.2 Numerical Comparison of Call Admission Control from SMDP with and without NCD

Our goal is to compare the results from the exact SMDP with those obtained using NCD. For simplicity, we consider a single class system and omit the class index.

Consider a family of single class systems indexed by $0 < \epsilon \to 0$ with call arrival rate $\lambda(\epsilon)$, departing probability $q(\epsilon)$, average rate of a call leaving "on" state $\alpha(\epsilon)$ and average rate of a call leaving "off" state $\beta(\epsilon)$. We assume $\lambda(\epsilon) = O(1)$, $q(\epsilon) = O(\epsilon)$, $\alpha(\epsilon) = O(\frac{1}{\epsilon})$, $\beta(\epsilon) = O(\frac{1}{\epsilon})$. Hence $\mu(\epsilon) = O(1)$, $p(\epsilon) = O(1)$.

To make the comparison more meaningful, we use the normalized index $\epsilon = \frac{1}{\hat{x}}$, where $\hat{x}$ is the number of "on"-"off" cycles per unit call holding time. We apply SMDP to a system with parameters set as follows: we keep the parameters describing call level characteristics fixed when comparing the results of NCD with various ϵ values. More specifically, we fix λ, μ, p, $r = 1$. Set $q^1 = q$ and $q^0 = 0$. Then given a $\hat{x}$, we have from equations (1) and (2) that

$$q = \frac{\mu}{\hat{x} + \mu}, \quad \beta = \frac{\mu}{pq}, \quad \alpha = \frac{\mu(1-q)}{1-p}.$$

We observe the impact of $\hat{x}$ on the admission policies and traffic behavior. In particular, we are interested in the call blocking probability.

3.2.1 Example

We present one example to illustrate the impact of using NCD on call blocking. The capacity of the system is $R = 45$ and the value for cell loss ratio requirement is fixed at $p^{\text{cell}} = 10^{-9}$. We obtain the optimal admission policy for three different $\hat{x}$ values, 10.0, 100.0 and 500.0. The traffic considered is the first class in Table 1.

Our numerical results, summarized in Table 2, indicate that the NCD approximation performs well when the average number of on/off cycles during a holding time is about

100 or larger. In addition, it appears that the NCD approximation performs better with burstier sources. (With $p\nu$ fixed, smaller p is burstier.)

$\hat{x}$	q	β	α	call blocking
10.0	0.009901	404.0	10.2564	1.941×10^{-2}
100.0	0.000999	4004.0	102.564	2.119×10^{-2}
500.0	0.0002	20004.0	512.8205	2.128×10^{-2}
∞	0	∞	∞	2.136×10^{-2}

Table 2: Numerical Results for Class 1

The actual admission policy when $\hat{x} = 500.0$ is shown in Figure 4.

4 Feasibility of Call Admission Controls

In this section we discuss the feasibility problem which focuses on not only cell level QoS but also call level QoS. Given parameters $R, \lambda_1, \lambda_2, \mu_1, \mu_2, p_1, p_2, \nu_1, \nu_2$ and QoS constraints $p_1^{\text{cell}}, p_2^{\text{cell}}, p_1^{\text{call}}, p_2^{\text{call}}$, does there exist a CAC under which the QoS constraints are satisfied? The exact version of this problem was shown, in Section 2.2.7, to reduce to feasibility of an LP. Using the NCD results of Section 3 we can also obtain an LP for the reduced problem. For the aggressive approach the development is very similar to that in Section 2.2.7, so we can be brief.

We define the set of arrival states corresponding to class j calls as

$$I_j^A(M) = \left\{ (\mathbf{k}, j) : k_i \in Z^+,\ 1 \le i \le J;\ \sum_{i=1}^{J} k_i \le M \right\},\ 1 \le j \le J\,.$$

The resulting LP is

$$\text{maximize} \quad \sum_{\mathbf{i} \in I(M)} \sum_{a \in A(\mathbf{i})} r(\mathbf{i}, a) z_{\mathbf{i}a}$$

subject to

$$\begin{aligned}
&\sum_{a \in A(\mathbf{j})} z_{\mathbf{j}a} - \sum_{\mathbf{i} \in I(M)} \sum_{a \in A(\mathbf{i})} p(\mathbf{i}, \mathbf{j}, a) z_{\mathbf{i}a} = 0, \quad \mathbf{j} \in I(M), \\
&\sum_{\mathbf{i} \in I(M)} \sum_{a \in A(\mathbf{i})} [l_\ell(\mathbf{i}, a) - p_\ell^{\text{cell}} g_\ell(\mathbf{i}, a)] z_{\mathbf{i}a} \le 0, \quad \ell = 1, \ldots, J, \\
&\sum_{\mathbf{i} \in I_j^A(M)} z_{\mathbf{i}0} - p_j^{\text{call}} \lambda_j \le 0, \quad 1 \le j \le J \\
&\sum_{\mathbf{i} \in I(M)} \sum_{a \in A(\mathbf{i})} \tau(\mathbf{i}, a) z_{\mathbf{i}a} = 1, \\
&z_{\mathbf{i}a} \ge 0\,, \qquad \mathbf{i} \in I(M),\ a \in A(\mathbf{i}).
\end{aligned}$$

The above LP will be feasible if and only if there is a CAC under which both the cell and call level QoS constraints are met. For $J = 2$, holding all parameters other than λ_1 and λ_2 fixed we solved for the set of (λ_1, λ_2) for which the LP is feasible. We call this set the feasible region. An example of the feasible region is illustrated in Figure 5 for the parameter

```
Number of calls
are ``on'': n

25|                                                                            R
24|     A: Accept                                                           R  R
23|     R: Reject                                                        R  R  R
22|     O: Random Decision                                            R  R  R  R
21|                                                                R  R  R  R  R
20|                                                             R  R  R  R  R  R
19|                                                          R  R  R  R  R  R  R
18|                                                       R  R  R  R  R  R  R  R
17|                                                    R  R  R  R  R  R  R  R  R
16|                                                 R  R  R  R  R  R  R  R  R  R
15|                                              R  R  R  R  R  R  R  R  R  R  R
14|                                           R  R  R  R  R  R  R  R  R  R  R  R
13|                                        R  R  R  R  R  R  R  R  R  R  R  R  R
12|                                     R  R  R  R  R  R  R  R  R  R  R  R  R  R
11|                                  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R
10|                               R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R
09|                            R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R
08|                         R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R
07|                      R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R
06|                   R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R  R
05|                R  A  A  A  A  A  A  A  A  R  R  R  R  R  R  R  R  R  R  R  R
04|             A  A  A  A  A  A  A  A  A  A  A  A  A  A  R  R  R  R  R  R  R  R
03|          A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  R  R  R  R  R  R  R  R
02|       A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  R  R  R  R  R  R  R  R
01|    A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  R  R  R  R  R  R  R  R
00| A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  A  O  O  R  R  R  R  R  R
   ------------------------------------------------------------------------------
    00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

               the number of calls in progress: k
```

Figure 4: Policy for Case 1: $\hat{x} = 500.0$

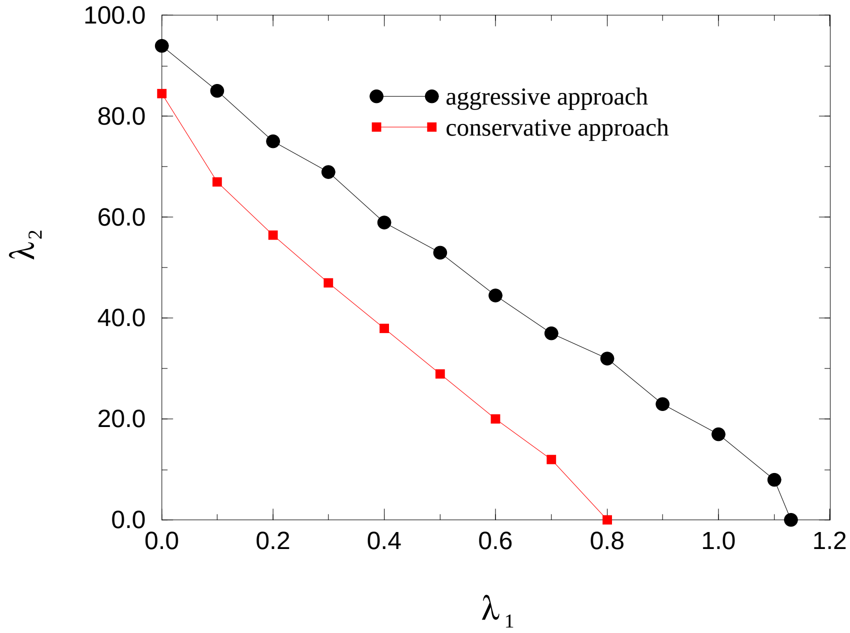

Figure 5: Feasible Regions: $p_i^{\text{cell}} = 10^{-9}$, $p_i^{\text{call}} = 0.01$, $i = 1, 2$

values in Table 1. (Other examples are contained in [20].) Remarkably, the boundary of this region exhibits near linearity. As a consequence of this (almost) linearity, the feasible region can be (approximately) determined from its corner points. These corner points can be obtained by solving two one-dimensional (single class) problems.

An LP for the conservative approach with call level QoS constraints can also be formulated, and solved. The feasible region for the conservative approach is also indicated in Figure 5. Note that the feasible region for the aggressive approach is substantially larger than for the conservative approach. We thus focus on the aggressive approach in the rest of the paper.

5 The Single Class System with Cell and Call QoS

Both cell and call level QoS are considered in the following study of a single class system. The insights gained from this study provide the basis for the techniques for multiclass systems.

5.1 The Model

The system and traffic model considered here is that obtained using NCD, with $J = 1$. Recall that for the system, the key parameters for the single class are denoted by λ, μ, p, ν; the QoS parameters are p^{cell} and p^{call}, and the link rate is R.

For the one dimensional reduced Markov chain, given a stationary call admission control policy, various steady-state performance measures of the system can be easily calculated. In this and the following sections, we do not specify any rewards or costs to be optimized. Our goal here is feasibility and robustness, so we do not provide any cost function. If we received a reward for each accepted call we would try to admit as many calls as possible subject to the two QoS requirements. Although the *optimal* admission control policy in this context normally has a randomized threshold due to the cell and call level QoS constraints, nonrandomized threshold policies, which are simpler, are sufficiently accurate for our purpose here.

5.2 Some Basic Relations

Let $\phi(\mathcal{K})$ be an admission control policy based on a threshold $\mathcal{K}$, i.e., a new call is admitted if there are fewer than $\mathcal{K}$ calls in progress, otherwise it is rejected. Based on the above model, we can calculate the following performance measures under policy $\phi(\mathcal{K})$. The mean cell arrival rate with k calls permanently in progress is $s(k) = k\nu p$. The mean cell loss rate with k calls permanently in progress is $b(k) = \sum_{n=0}^{k} \binom{k}{n} p^n (1-p)^{k-n} [n\nu - R]^+$. Let $g(\mathcal{K}) = \sum_{k=0}^{\mathcal{K}} (\lambda/\mu)^k / k!$. The stationary distribution for the number of calls in progress is $P(k \text{ calls}) = [(\lambda/\mu)^k/k!]/g(\mathcal{K}), \quad k = 0, 1, \ldots, \mathcal{K}$. The average cell arrival rate is $s_{\mathcal{K}} = [\frac{\sum_{k=1}^{\mathcal{K}} (\lambda/\mu)^k}{k!} s(k)]/g(\mathcal{K})$. The average cell loss rate is $b_{\mathcal{K}} = [\sum_{k=1}^{\mathcal{K}} \frac{(\lambda/\mu)^k}{k!} b(k)]/g(\mathcal{K})$. The average cell loss ratio is

$$L(\lambda, \mathcal{K}, R) = \frac{b_{\mathcal{K}}}{s_{\mathcal{K}}} = \frac{\sum_{k=1}^{\mathcal{K}} \frac{(\lambda/\mu)^k}{k!} b(k)}{\sum_{k=1}^{\mathcal{K}} \frac{(\lambda/\mu)^k}{k!} s(k)}. \tag{19}$$

The call blocking probability is given by the Erlang loss formula $B(\lambda, \mathcal{K}) = [(\lambda/\mu)^{\mathcal{K}}/\mathcal{K}!]/g(\mathcal{K})$.

5.3 Feasibility of Call Arrival Rate

Similarly to the treatment in Section 4, we can define the feasibility of the call arrival rate for a single class system as follows: a call arrival rate λ is said to be feasible if there exists an admission control policy from the class of threshold policies that meets both QoS requirements. That is, λ is feasible if there exists $\mathcal{K}$ such that $\phi(\mathcal{K})$ ensures both cell and call QoS. Recall that our cell and call QoS requirements are, respectively: $L(\lambda, \mathcal{K}, R) \leq p^{\text{cell}}$, and $B(\lambda, \mathcal{K}) \leq p^{\text{call}}$. For a given λ, let

$$\mathcal{K}_1(\lambda) = \max\{\mathcal{K} : L(\lambda, \mathcal{K}) \leq p^{\text{cell}}\} \quad \text{and} \quad \mathcal{K}_2(\lambda) = \min\{\mathcal{K} : B(\lambda, \mathcal{K}) \leq p^{\text{call}}\}.$$

If $\mathcal{K}_2(\lambda) \leq \mathcal{K}_1(\lambda)$, then there exists a threshold type policy $\phi(\mathcal{K})$ with $\mathcal{K}_2(\lambda) \leq \mathcal{K} \leq \mathcal{K}_1(\lambda)$, such that under $\phi(\mathcal{K})$ both cell and call QoS requirements are met, and we say λ is feasible.

Given p^{cell}, p^{call} and the traffic parameters described in Section 5.1, there is a maximum feasible call arrival rate, λ_{max}, such that for any $\lambda > \lambda_{max}$, $\mathcal{K}_2(\lambda) > \mathcal{K}_1(\lambda)$. It is not surprising that for most cases of practical interest, $\mathcal{K}_2(\lambda_{max}) = \mathcal{K}_1(\lambda_{max})$, which is assumed throughout this paper.

5.4 An Effective Bandwidth

Let $e = \frac{R}{\mathcal{K}_1(\lambda_{max})}$. Then e is a measure of the resources a call requires to satisfy *both* cell level and call level QoS requirements. Note that e is independent of the call arrival rate λ.

Note that a traditional effective bandwidth definition, see [6] for instance, assumes that calls last in perpetuity and addresses only cell level QoS:

$$e_{static} = \frac{R}{\mathcal{K}_{static}}, \quad \text{where} \quad \mathcal{K}_{static} = \max\left\{k : \frac{b(k)}{s(k)} \leq p^{\text{cell}}\right\}.$$

Note that $e \leq e_{static}$. Since e_{static} ignores call level dynamics it is more conservative.

Example 1 Consider a system with $R = 45.0$, and homogeneous sources with parameters corresponding to class 1 sources as listed in Table 1, i.e., $\mu = 0.1$, $\nu = 6.0$, $p = 0.025$, $p^{\text{cell}} = 10^{-9}$ and $p^{\text{call}} = 0.01$. In Figure 6 we illustrate the relation between $\mathcal{K}_1(\lambda)$, $\mathcal{K}_2(\lambda)$ and λ_{max}. $\mathcal{K}_1(\lambda)$ and $\mathcal{K}_2(\lambda)$ intersect at 20 when $\lambda = 1.125$. Hence, $\lambda_{max} = 1.125$, and $e = \frac{45}{20} = 2.25$. Note that $\mathcal{K}_{static} = 14$, hence $e_{static} = 3.21$.

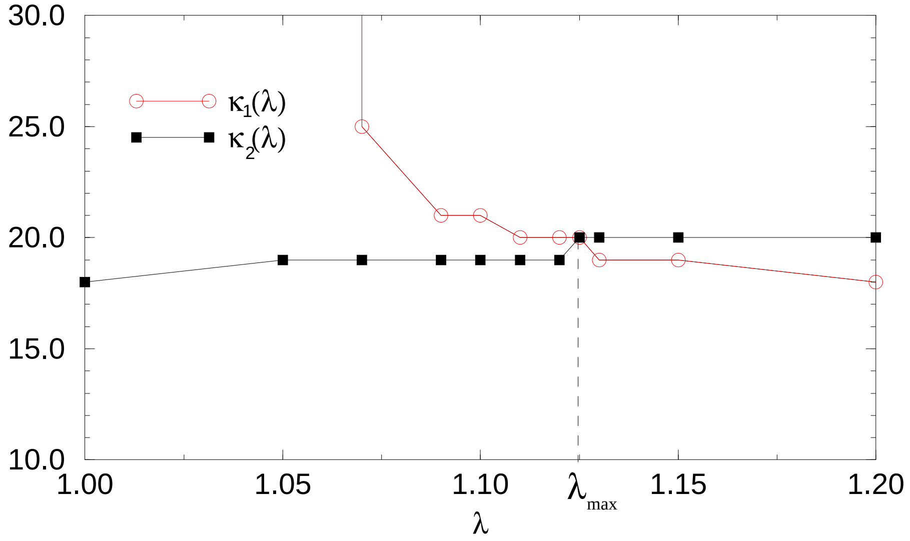

Figure 6: Calculating Effective Bandwidth for Class 1 Sources

Example 2 Now consider a system with the same R, p^{cell}, p^{call} as in Example 1, and sources corresponding to class 2, see Table 1, i.e., $\mu = 1.0$, $\nu = 1.5$ and $p = 0.1$. The relation between $\mathcal{K}_1(\lambda)$, $\mathcal{K}_2(\lambda)$ and λ_{max} are illustrated in Figure 7. $\mathcal{K}_1(\lambda)$ and $\mathcal{K}_2(\lambda)$ intersect at 111 when $\lambda = 94.0$. Hence, $\lambda_{max} = 94.0$, and $e = \frac{45}{111} = 0.4054$. Now $\mathcal{K}_{static} = 100$, and $e_{static} = 0.45$.

6 Asymptotic Analysis

An asymptotic analysis of a single class system provides fundamental insights into the joint behavior of the cell loss ratio and call blocking probability for purposes of sizing and operations. The investigation is in the asymptotic framework of large systems, i.e., as $(\lambda, \mathcal{K}, R) \to \infty$ in a manner consistent with practical QoS requirements. Specifically, p^{cell} is expected to be in the range $10^{-6} - 10^{-9}$, while p^{call} is expected to be in the neighborhood of 10^{-2}. These numbers suggest the following important dichotomy: cell loss ratios decay exponentially in the large parameter, say $\mathcal{K}$, while call blocking probabilities decay polynomially, more specifically as $1/\sqrt{\mathcal{K}}$. While both elements are separately recognized in the literature, see for instance [24], we do not know of any prior analysis in which both elements are simultaneously present. The loadings at the cell and call levels are required to be such that the ultimate cell and call performances are, respectively, exponential and polynomial

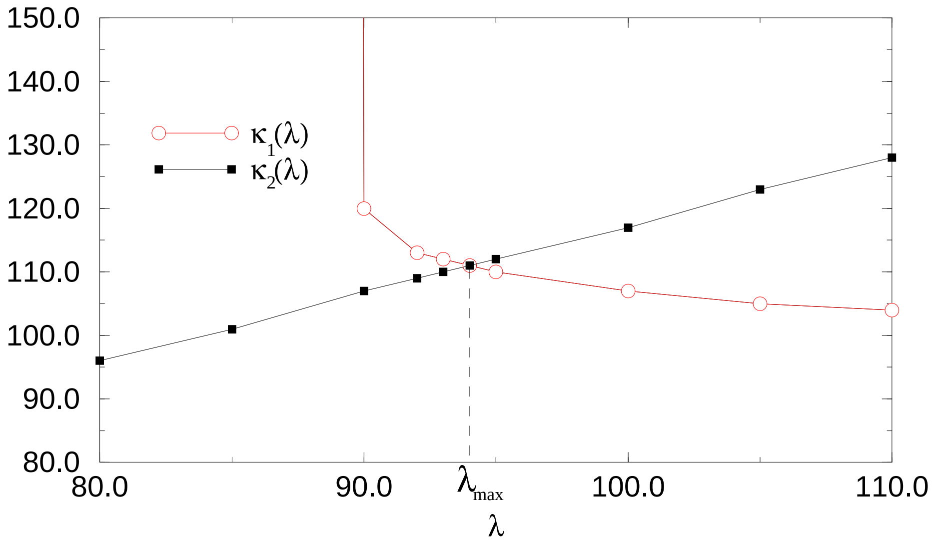

Figure 7: Calculating Effective Bandwidth for Class 2 Sources

in $\mathcal{K}$. We obtain such loading guidelines. We also obtain the coefficients associated with the exponential and polynomial behaviors. Thus the contributions here are both qualitative and quantitative.

First, we write the expression for the cell loss ratio in (19) as $L(\lambda, \mathcal{K}, C)$, where $C = R/\nu$ by definition. Also, we assume for convenience that C is an integer, and select the unit of time to be such that $\mu = 1$. Then,

$$L(\lambda, \mathcal{K}, C) = \mathrm{Num}/\mathrm{Den}, \tag{20}$$

where
$$\mathrm{Num} = 0 \qquad \text{if } \mathcal{K} \leq C \tag{21}$$
$$= \sum_{k=C+1}^{\mathcal{K}} \frac{\lambda^k}{k!} \sum_{n=C+1}^{k} (n-C) \binom{k}{n} p^n (1-p)^{k-n} \qquad \text{if } \mathcal{K} > C$$

and
$$\mathrm{Den} = p \sum_{k=0}^{\mathcal{K}} k\lambda^k/k!\ . \tag{22}$$

Note that we may assume that $\mathcal{K} > C$, as otherwise $L = 0$.

As mentioned above, we let $(\lambda, \mathcal{K}, C) \to \infty$, while p and ν are held fixed. For loading at the cell level we assume that the system is underloaded, i.e.,

$$\rho \triangleq \frac{\lambda p}{C} < 1\,, \tag{23}$$

and that critical loading holds at the call level, i.e.,

$$\gamma \triangleq \left(1 - \frac{\mathcal{K}}{\lambda}\right)\sqrt{\lambda} = O(1)\,, \tag{24}$$

i.e., γ is bounded. At the loss of some small generality we will make the convenient assumption that γ is a fixed constant, which can be either positive or negative. Hence $\mathcal{K} = \lambda - \gamma\sqrt{\lambda}$, so that, to leading order, $\mathcal{K}/\lambda \sim 1$.

Here we give an overview of the analysis, which includes the salient elements, with the details provided in the Appendix. We show that

$$\text{Num} = \sum_{l=0}^{\mathcal{K}-C} \frac{l(\lambda p)^{C+l}}{(C+l)!(\mathcal{K}-C-l)!} \int_0^\infty e^{-u}[\lambda(1-p)+u]^{\mathcal{K}-C-l} du \ , \tag{25}$$

where the integral is obtained by making use of Euler's representation, $n! = \int_0^\infty e^{-u} u^n du$, and the binomial theorem. Now,

$$(C+l)! = C!(C+1)\cdots(C+l) \geq C!\, C^l \ , \tag{26}$$

and also the bound is asymptotically exact as $C \to \infty$ and $l = O(1)$. After substituting (26) in (25) we show that

$$\begin{aligned}
\text{Num} &\sim \frac{1}{C!} \sum_{l=0}^{\mathcal{K}-C} \frac{l(\lambda p)^{C+l}}{C^l(\mathcal{K}-C-l)!} \int_0^\infty e^{-u}[\lambda(1-p)+u]^{\mathcal{K}-C-l} du \\
&= \frac{(\lambda p)^C}{C!(\mathcal{K}-C)!} \left[(\mathcal{K}-C) \int_0^\infty \int_0^\infty e^{-(u+v)} \rho v [\lambda(1-p)+u+\rho v]^{\mathcal{K}-C-1} du\, dv \right] \\
&= \frac{(\lambda p)^C \rho}{C!(\mathcal{K}-C)!} \frac{\partial I}{\partial \rho} \ ,
\end{aligned} \tag{27}$$

where

$$I = \int_0^\infty \int_0^\infty e^{-(u+v)} [\lambda(1-p)+u+\rho v]^{\mathcal{K}-C} du\, dv \ . \tag{28}$$

This is an important integral representation.

We obtain an asymptotic expression for the integral I for the scaling in (23) and (24):

$$I \sim \frac{\{\lambda(1-p)\}^{\mathcal{K}-C}}{(1-\alpha)(1-\rho\alpha)} \ , \tag{29}$$

where

$$\alpha \triangleq \frac{1-p/\rho}{1-p} \ . \tag{30}$$

The constant α arises since $(\mathcal{K}-C)/\{\lambda(1-p)\} \sim \alpha$ from the scaling. Note that $0 < \alpha < 1$, where the positivity is due to $\mathcal{K} > C$.

From (29),

$$\frac{\partial I}{\partial \rho} \sim \frac{\alpha\{\lambda(1-p)\}^{\mathcal{K}-C}}{(1-\alpha)(1-\rho\alpha)^2} \ . \tag{31}$$

Substitution in (27) together with use of Stirling's formula for $C!$ and $(\mathcal{K}-C)!$ gives the asymptotic expression for Num.

Turning to Den in (22), we find that

$$\text{Den} = \frac{p\lambda^{\mathcal{K}}}{(\mathcal{K}-1)!} \frac{1}{B(\lambda, \mathcal{K}-1)} \ , \tag{32}$$

where $B(\lambda, \mathcal{K}-1)$ is the Erlang loss formula. Since the call level loading is critical, we know from prior results [12] that to leading order $B(\lambda, \mathcal{K}-1)/B(\lambda, \mathcal{K}) \sim 1$, and that

$$\frac{1}{B(\lambda, \mathcal{K})} \sim \sqrt{\lambda} W_0(\gamma) \ , \tag{33}$$

where $W_0(\gamma)$ is a $O(1)$ constant which is obtained from the normal distribution and depends only on γ. In fact,

$$e^{-\gamma^2/2} W_0(\gamma) = \left(\frac{\pi}{2}\right)^{1/2} \text{erfc}\left(\frac{\gamma}{\sqrt{2}}\right) \ ,$$

where erfc is the complementary error function.

Finally, substituting the asymptotic expressions for Num and Den, we obtain

$$L(\lambda, \mathcal{K}, C) \sim \frac{A e^{-\delta \mathcal{K}}}{\mathcal{K}^2} \ , \tag{34}$$

where

$$\begin{aligned} A &= A(\rho, p, \gamma) = \frac{1}{\sqrt{2\pi}} \frac{\rho\alpha}{p(1-\alpha)(1-\rho\alpha)^2} \frac{1}{\sqrt{(1-p/\rho)p/\rho}} \cdot \frac{1}{W_0(\gamma)} \ , \\ \text{and} \qquad \delta &= \delta(\rho, p) = \log\left[\left(\frac{1-p/\rho}{1-p}\right)^{1-p/\rho} \left(\frac{1}{\rho}\right)^{p/\rho}\right] \ . \end{aligned} \tag{35}$$

It is easy to verify that for all (ρ, p) such that $0 < \rho < 1$ and $0 < p < 1$, $\delta(\rho, p) > 0$. Clearly $A(\rho, p, \gamma) = O(1)$. These facts prove the important result that for the asymptotic scaling in (23) and (24), the cell loss ratio $L(\lambda, \mathcal{K}, C)$ is exponentially small in the large parameter $\mathcal{K}$, with δ the constant in the exponential. The asymptotic call blocking probability for our scaling has already been obtained in (33), in which the $1/\sqrt{\lambda}$ (and therefore $1/\sqrt{\mathcal{K}}$) behavior is exhibited, and $1/W_0(\gamma)$ is the relevant constant.

7 CAC Design

We now return to the two class problem. We seek a CAC that satisfies all QoS requirements. For feasible parameters such a CAC can be obtained from the LP. We are seeking, however, a simple and robust CAC.

We consider all parameters except for λ_1 and λ_2 to be fixed, and consider a point (λ_1, λ_2) inside the feasible region described in Section 4 and depicted in Figure 5. We consider situations where class 1 is 'burstier'.

The two class system that we study has $R = 45$ and other parameters specified in Table 1. We know that when (λ_1, λ_2) is on or very close to the boundary of the feasible region, the call admission policies that meet all the QoS requirements necessarily have complicated structures. For just this reason, the engineered operating points are designed not to be too close to the boundary. Under the assumption that (λ_1, λ_2) are not very close to the boundary, the question of interest in this paper is whether we can find a *simple* connection admission policy that meets *both cell and call QoS requirements*. We want our policy to be robust in the sense that if the operating point drifts away from the engineered loads due to the unexpected high arrival rate of one class, the admission policy should be able to protect the other class.

The most widely studied call admission policy type is CS. Although it is easy to implement, it always favors calls requiring less bandwidth capacity, thus it can drive the blocking probability of calls with a larger bandwidth requirement up when the arrival rates of the calls with smaller bandwidth requirement are high. A very undesirable situation is when the class 1 blocking probability exceeds the QoS constraint while the blocking probability for class 2 calls is much lower than its constraint. With CS, regardless which class exceeds its engineered load, class 1 will suffer in terms of call blocking probability.

Trunk Reservation (TR) policies have been known to be able to provide protection to wideband calls. With properly chosen trunk reservation parameters, class 1 calls will not have unacceptably high call blocking probabilities due to high arrival rates of class 2 calls. However, the traditional trunk reservation gives priority to a fixed class, hence cannot protect the other class when the favored class has high arrival rate. Of particular interest is the following trunk reservation policy which we study numerically in Section 9: for single link systems with two classes of calls, narrowband and wideband, if we admit a new call (regardless of its class) only when the spare capacity in the system is at least the bandwidth of wideband calls, then this special trunk reservation policy balances the call blocking probabilities of the two classes, which can be easily seen with PASTA [16]. Furthermore, this policy is optimal among all the trunk reservation policies that balance the call blocking probabilities in the sense that the call blocking probabilities are the smallest. However, because this policy balances the call blocking probabilities, if one class has high call arrival rate, both classes will have high call blocking probabilities. In other words, it is not robust, which is far from desirable in many situations.

It is intuitively apparent that if (λ_1, λ_2) are small enough (close to the origin), a complete sharing policy will be sufficient. The interesting case is when (λ_1, λ_2) is not very close to the boundary of the feasible region and also not small. In this case, we believe a simple policy based on virtual partitioning with properly chosen parameters gives satisfactory performance.

8 Virtual Partitioning

We now describe a call admission policy based on the notion of VP. Let e_1 and e_2 be the bandwidth requirements of class 1 and 2 calls, and K_1 and K_2 be the partitioning parameters which are two positive integers such that $K_1 e_1 + K_2 e_2 \geq R$. A call admission policy based on VP is summarized as follows: *When a call of class* 1 *arrives and finds* (k_1, k_2) *calls in progress, it is accepted if*

$$k_1 < K_1 \quad \text{and} \quad e_1 k_1 + e_2 k_2 \leq R - e_1,$$

$$\text{or,} \quad k_1 \geq K_1 \quad \text{and} \quad e_1 k_1 + e_2 k_2 \leq R - t_2 e_2 - e_1,$$

where $t_2 e_2$ *is the bandwidth reserved for (underloaded) class* 2 *calls. Similarly, a class* 2 *call is accepted if*

$$k_2 < K_2 \quad \text{and} \quad e_1 k_1 + e_2 k_2 \leq R - e_2 \quad \text{or}$$

$$k_2 \geq K_2 \quad \text{and} \quad e_1 k_1 + e_2 k_2 \leq R - t_1 e_1 - e_2,$$

where $t_1 e_1$ *is the bandwidth reserved for (underloaded) class* 1 *calls.*

Note that the call admission is performed on a set in (k_1, k_2) space defined by $k_1 e_1 + k_2 e_2 \leq R$. The motivation for selecting this set is derived from the linearity implicit in the notion of effective bandwidths. Admittedly there is no sound theoretical basis for the linearity at this time. In the absence of such a theory we take the precautionary step of verifying in our numerical investigations that cell level QoS requirements are satisfied by our admission control policies.

By choosing parameters K_1 and K_2, we partition the bandwidth between the two classes. The trunk reservation parameters t_1 and t_2 allow us to block calls from the overloaded class so as to reserve bandwidth for the underloaded class. The nature of the policy is to give the underloaded class higher priority, and the consequence is that the underloaded class is protected.

We expect t_1 and t_2 to be small nonnegative integers. When $t_1 = t_2 = 0$, the policy becomes a complete sharing policy over the whole admissible region (regardless of K_1 and K_2). If in this case the call blocking probabilities for *both* classes are still greater than the allowed limits (say, 1%), then the arrival rates are too high for there to exist any feasible policy. When t_1 and t_2 are not both zero, the policy is complete sharing on the set $\{(k_1, k_2) : k_1 \leq K_1,\ k_2 \leq K_2\}$, and dynamically prioritized outside the set. With complete sharing, the less bursty traffic enjoys lower call blocking, and consequently when the set in which complete sharing applies is bigger, fewer calls of the less bursty traffic are blocked.

9 Numerical Results for VP

In our numerical study, we attempt to show that we can use VP to design simple call admission policies such that when the system is subject to load at or below the engineered call arrival rates $(\lambda_1^e, \lambda_2^e)$ the admission policy will meet all the cell and call QoS requirements, and when the system is subject to load above the engineered loads due to a high arrival rate from one class, the other class is protected. More specifically, we observe the call blocking probabilities of the two classes, denoted by (B_1, B_2), under CS, TR and VP to illustrate the robustness of VP. The following load scenarios are studied: both classes are below the engineered load; both classes are at the engineered load; one of the two classes is at the engineered load while the other is at least 20% higher; both classes are 10% higher than the engineered loads.

The system and traffic sources are the ones specified in Section 7. From the analysis of the two single class problems in Section 5.4, we have $e_1 = 2.25$ and $e_2 = 0.4054$, which are used for all the cases in this section.

9.1 Case 1

The first set of numerical results are for the engineered load: $(\lambda_1^e, \lambda_2^e) = (0.6, 35.0)$. The bandwidth partitioning parameters used are $(K_1, K_2) = (9, 42)$ and the trunk reservation parameters used are $(t_1, t_2) = (2, 0)$. The call blocking probabilities for several different loads are listed in Table 3.

When the load is below or at the engineered load, all three policies give satisfactory call blocking probabilities. However, when one of the two classes has load higher than the engineered load, class 1 always suffers under CS and the blocking probabilities are driven above the allowed limit 1%. TR in general gives better performance than CS. However, when $(\lambda_1, \lambda_2) = (0.6, 45.0)$, TR gives balanced call blocking probabilities of 0.0106. Hence class 1 suffers because class 2 has an arrival rate higher than the engineered load. VP is able to protect the underloaded class. Because the engineered load is near the middle of the boundary of the feasible region, we need to protect each class when the other class exceeds its engineered load. Since on the complete sharing set class 2 is favored, by choosing the

Call arrival rates (λ_1, λ_2)	Call blocking Probabilities (B_1, B_2)		
	CS	TR	VP
(0.60, 35.0)	(0.0064, 0.0008)	(0.0031, 0.0031)	(0.0049, 0.0033)
(0.72, 35.0)	(0.0184, 0.0026)	(0.0096, 0.0096)	(0.0153, 0.0076)
(0.60, 45.0)	(0.0232, 0.0032)	(0.0106, 0.0106)	(0.0083, 0.0202)
(0.66, 38.5)	(0.0174, 0.0024)	(0.0085, 0.0085)	(0.0110, 0.0110)
(0.55, 33.0)	(0.0026, 0.0003)	(0.0013, 0.0013)	(0.0021, 0.0012)

Table 3: Numerical Results for Case 1

set large enough for VP we can protect class 2. Protection for class 1 in VP is achieved by choosing the complete sharing set not too large and selecting proper trunk reservation parameter t_1, which is set to 2 in this case.

9.2 Case 2

The second set of numerical results are for the engineered load $(\lambda_1^e, \lambda_2^e) = (0.2, 70.0)$. The bandwidth partitioning parameters used are $(K_1, K_2) = (7, 60)$ and the trunk reservation parameters used are $(t_1, t_2) = (1, 0)$. Table 4 summarizes the call blocking probabilities corresponding to the five scenarios.

Call arrival rates (λ_1, λ_2)	Call blocking Probabilities (B_1, B_2)		
	CS	TR	VP
(0.20, 70.0)	(0.0150, 0.0018)	(0.0048, 0.0048)	(0.0004, 0.0059)
(0.24, 70.0)	(0.0241, 0.0031)	(0.0079, 0.0079)	(0.0010, 0.0096)
(0.20, 84.0)	(0.1049, 0.0153)	(0.0327, 0.0327)	(0.0018, 0.0386)
(0.22, 77.0)	(0.0536, 0.0073)	(0.0171, 0.0171)	(0.0013, 0.0205)
(0.18, 63.0)	(0.0026, 0.0003)	(0.0009, 0.0009)	(0.0001, 0.0011)

Table 4: Numerical Results for Case 2

Now the engineered load is near the upper left corner of the feasible region. Again at $(\lambda_1, \lambda_2) = (0.20, 84.0)$, class 1 suffers under CS and TR when class 2 exceeds the engineered load, while VP is able to protect it. When the class 1 arrival rate exceeds the engineered load by 20%, the impact on class 2 is very small. On the other hand, increasing the arrival rate of class 2 has more dramatic impact on class 1 traffic, hence it seems now the focus should be on protecting class 1 against class 2.

9.3 Case 3

The third set of results are for the engineered load $(\lambda_1^e, \lambda_2^e) = (0.9, 10.0)$. The bandwidth partitioning parameters used are $(K_1, K_2) = (10, 25)$ and the trunk reservation parameters used are $(t_1, t_2) = (3, 0)$. Table 5 summarizes the call blocking probabilities for the five scenarios.

Call arrival rates	Call blocking Probabilities (B_1, B_2)		
(λ_1, λ_2)	CS	TR	VP
(0.9, 10.0)	(0.0038, 0.0005)	(0.0030, 0.0030)	(0.0038, 0.0005)
(1.08, 10.0)	(0.0156, 0.0022)	(0.0128, 0.0128)	(0.0156, 0.0022)
(0.90, 15.0)	(0.0073, 0.0010)	(0.0052, 0.0052)	(0.0072, 0.0017)
(0.99, 11.0)	(0.0093, 0.0013)	(0.0073, 0.0073)	(0.0093, 0.0014)
(0.81, 9.0)	(0.0013, 0.0002)	(0.0010, 0.0010)	(0.0013, 0.0002)

Table 5: Numerical Results for Case 3

This is a case where traffic is light so that VP is almost identical to CS. At $(\lambda_1, \lambda_2) = (1.08, 10.0)$, class 2 suffers under TR while both CS and VP are able protect class 2.

10 New Directions

Although our results have been presented for a two-class system, we believe that they would apply for more than two classes. The key to our approach is the near linearity of the feasible region depicted in Figure 5. If the feasible region is linear for the multiclass system, then the $J(\geq 2)$ class system can be analyzed using J single class systems, so that the CAC introduced here can be used for $J > 2$ as well. Further work on this topic is necessary.

In this paper we considered two versions of the cell loss constraint: conservative and aggressive. One might argue that the conservative approach is too conservative, and the aggressive approach is too aggressive, so that it would be nice to have something in between the two. An approach based on expected cell loss during the lifetime of a call in the system is considered in [19]. It is shown there that in the single class system the threshold based on this new approach indeed lies between the other two.

Acknowledgement

We are grateful to our colleagues John Morrison and K. G. Ramakrishnan for their invaluable help with the analysis in Section 6 and the solution of linear programs and related numerical problems, respectively.

References

[1] J.M. Akinpelu. The overload performance of engineered networks with nonhierarchical and hierarchical routing. *AT&T Bell Labs Technical Journal*, 63:1261–1281, 1984.

[2] E. Altman and V. A. Gaitsogory. Stability and Singular Perturbations in Constrained Markov Decision Problems. *IEEE Trans. Aut. Contr.*, 38:971–975, 1993.

[3] N.G. Bean. Effective bandwidths with different quality of service requirements. In *Integrated Broadband Communication Networks and Services*. V.B. Iverson (Ed.), IFIP, 1993.

[4] M. Beshai, R. Kositpaiboon, and J. Yan. Interaction of call blocking and cell loss in an atm network. *IEEE Journal on Selected Areas in Communications*, 12:1051–1058, 1994.

[5] S. Borst and D. Mitra. Virtual partitioning for robust resource sharing: Computational techniques for heterogeneous traffic. In this issue.

[6] A.I. Elwalid and D. Mitra. Effective bandwidth of general markovian traffic sources and admission control of high speed networks. *IEEE/ACM Transactions on Networking*, 1:329–343, 1993.

[7] E.A. Feinberg. Constrained Semi-Markov decision processes with average rewards. *ZOR-Mathematical Methods of Operations Research*, 39:257–288, 1994.

[8] R.J. Gibbens, F.P. Kelly, and P.B. Key. A decision-theoretic approach to call admission control in ATM networks. *IEEE J. Sel. Areas Commun.*, 13:1101–1114, 1995.

[9] F. Hubner and P. Tran-Gia. Quasi-Stationary Analysis of a Finite Capacity Asynchronous Multiplexer with Modulated Deterministic Input. *ITC-13, Copenhagen*, 1991.

[10] J.Y. Hui. Resource allocation for broadband networks. *IEEE J. Selected Areas in Communications*, 6, 1988.

[11] J.M. Hyman, A.A. Lazar, and G. Pacifici. A separation principle between scheduling and admission control for broadband switching. *IEEE Journal on Selected Areas in Communications*, 11:605–616, 1993.

[12] D.L. Jagerman. Some properties of the Erlang loss function. *Bell Systems Technical Journal*, 53:525–551, 1974.

[13] *Special Issue on* Advances in the Fundamentals of Networking - Part I, *IEEE Journal of Selected Area in Communications*, 13, 1995.

[14] F.P. Kelly. Effective bandwidths at multi-class queues. *Queueing Systems*, 9:5–16, 1991.

[15] P. Key. Optimal control and trunk reservation in loss networks. *Probability in the Engineering and Informational Sciences*, 4:203–242, 1990.

[16] K. Lindberger. Dimensioning and designing methods for integrated ATM networks. In *Proceedings of ITC-14*, pages 897–906, 1994.

[17] D. Mitra, M.I. Reiman, and J. Wang. Robust admission control for heterogeneous ATM systems with both cell and call QoS requirements. In *Proceedings of ITC-15*, pages 1421–1432, Washington, D.C., 1997.

[18] D. Mitra and I. Ziedins. Virtual partitioning by dynamic priorities: Fair and efficient resource-sharing by several services. In B. Plattner, editor, *Broadband Communications:* Proceedings of 1996 International Zurich Seminar on Digital Communications, pages 173–185. Springer, 1996.

[19] M.I. Reiman and A. Shwartz. Call admission: a new approach to quality of service. submitted for publication, 1997.

[20] M.I. Reiman, J. Wang, and D. Mitra. Dynamic call admission control of an ATM multiplexer with on/off sources. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 1382–1388, 1995.

[21] J.W. Roberts (Ed.). *COST 224: Performance evaluation and design of multiservice networks.* ECSC-EEC-EAEC, Brussels, 1992.

[22] S. M. Ross. *Applied Probability Models with Optimization Applications.* Holden-Day, San Francisco, 1970.

[23] H. Saito. Call admission control in an ATM network using upper bound of cell loss probability. *IEEE Trans. on Communications*, 40:1512–1521, 1992.

[24] A. Shwartz and A. Weiss. *Large Deviation for Performance Analysis.* Chapman & Hall, London, 1995.

APPENDIX

Here we give some details of the asymptotic analysis in Section 6. There are three parts to the Appendix. First, we prove (25), secondly we make use of (26) to prove (27) and, finally, we establish (29).

To prove (25) for $\mathcal{K} > C$ we note that from the expression in (21),

$$\text{Num} = \sum_{n=C+1}^{\mathcal{K}} \frac{(\lambda p)^n}{n!}(n-C)\sum_{k=0}^{\mathcal{K}-n}\frac{\{\lambda(1-p)\}^k}{k!}\,. \tag{A1}$$

Now, we may use Euler's representation for $n!$ to obtain

$$\begin{aligned}\sum_{k=0}^{\mathcal{K}-n}\frac{\{\lambda(1-p)\}^k}{k!} &= \int_0^\infty e^{-u}\sum_{k=0}^{\mathcal{K}-n}\frac{\{\lambda(1-p)\}^k}{k!}\frac{u^{\mathcal{K}-n-k}}{(\mathcal{K}-n-k)!}\,du \\ &= \frac{1}{(\mathcal{K}-n)!}\int_0^\infty e^{-u}\sum_{k=0}^{\mathcal{K}-n}\binom{\mathcal{K}-n}{k}\{\lambda(1-p)\}^k u^{\mathcal{K}-n-k}\,du \\ &= \frac{1}{(\mathcal{K}-n)!}\int_0^\infty e^{-u}[\lambda(1-p)+u]^{\mathcal{K}-n}\,du\,. \end{aligned}\tag{A2}$$

Substituting (A2) in (A1),

$$\begin{aligned}\text{Num} &= \sum_{n=C+1}^{\mathcal{K}}\frac{(\lambda p)^n}{n!}\frac{(n-C)}{(\mathcal{K}-n)!}\int_0^\infty e^{-u}[\lambda(1-p)+u]^{\mathcal{K}-n}\,du \\ &= \sum_{l=0}^{\mathcal{K}-C}\frac{l(\lambda p)^{C+l}}{(C+l)!(\mathcal{K}-C-l)!}\int_0^\infty e^{-u}[\lambda(1-p)+u]^{\mathcal{K}-C-l}\,du\,,\end{aligned}\tag{A3}$$

which is (25).

To prove (27), we make use of (26) in (A3) to obtain

$$\begin{aligned}\text{Num} &\sim \frac{1}{C!}\sum_{l=0}^{\mathcal{K}-C}\frac{l(\lambda p)^{C+l}}{C^l(\mathcal{K}-C-l)!}\int_0^\infty e^{-u}[\lambda(1-p)+u]^{\mathcal{K}-C-l}\,du \\ &= \frac{(\lambda p)^C}{C!}\int_0^\infty e^{-u}\left[\sum_{l=0}^{\mathcal{K}-C}\frac{l\rho^l[\lambda(1-p)+u]^{\mathcal{K}-C-l}}{(\mathcal{K}-C-l)!}\right]du\,,\end{aligned}\tag{A4}$$

where the definition of ρ in (23) has been used. Now introduce Euler's representation of $l!$:

$$\begin{aligned}\text{Num} &\sim \frac{(\lambda p)^C}{C!(\mathcal{K}-C)!}\int_0^\infty e^{-u}\int_0^\infty e^{-v}\sum_{l=0}^{\mathcal{K}-C} l\binom{\mathcal{K}-C}{l}(\rho v)^l[\lambda(1-p)+u]^{\mathcal{K}-C-l}\,dv\,du \\ &= \frac{(\lambda p)^C}{C!(\mathcal{K}-C)!} \\ &\quad\cdot\left[(\mathcal{K}-C)\int_0^\infty\int_0^\infty e^{-(u+v)}\rho v\sum_{l=1}^{\mathcal{K}-C}\binom{\mathcal{K}-C-1}{l-1}(\rho v)^{l-1}[\lambda(1-p)+u]^{\mathcal{K}-C-l}\,dv\,du\right] \\ &= \frac{(\lambda p)^C\rho}{C!(\mathcal{K}-C)!}\left[(\mathcal{K}-C)\int_0^\infty\int_0^\infty e^{-(u+v)}v[\lambda(1-p)+u+\rho v]^{\mathcal{K}-C-1}\,dv\,du\right]\,.\end{aligned}\tag{A5}$$

The expression in outer brackets is observed to be $\partial I/\partial \rho$, where I is defined in (28). Hence (27) is proven.

To prove (29) note that from the definition of I in (28),

$$I = \{\lambda(1-p)\}^{\mathcal{K}-C} \int_0^\infty \int_0^\infty e^{-(u+v)} H(u,v;\mathcal{K}) du\, dv\,, \tag{A6}$$

where we let $\mathcal{K}$ be the surrogate for the large parameters $(\lambda, \mathcal{K}, C)$, and

$$H(u,v;\mathcal{K}) \triangleq \exp\left[(\mathcal{K}-C)\log\left\{1+\frac{u}{\lambda(1-p)}+\frac{\rho v}{\lambda(1-p)}\right\}\right]\,. \tag{A7}$$

Now,

$$H(u,v;\mathcal{K}) = \left[\exp\left\{\frac{(\mathcal{K}-C)u}{\lambda(1-p)}+\frac{(\mathcal{K}-C)\rho v}{\lambda(1-p)}\right\}\right]\left[1+O\left(\frac{(u+\rho v)^2}{\mathcal{K}}\right)\right]\,. \tag{A8}$$

Hence, noting that $(\mathcal{K}-C)/\{\lambda(1-p)\} \sim \alpha$ from the scaling,

$$\begin{aligned}
&\int_0^\infty \int_0^\infty e^{-(u+v)} H(u,v;\mathcal{K}) du\, dv \\
&\quad = \int_0^\infty \int_0^\infty e^{-(u+v)} e^{\alpha u+\rho\alpha v}\left[1+O\left(\frac{(u+\rho v)^2}{\mathcal{K}}\right)\right] du\, dv \\
&\quad = \frac{1}{1-\alpha}\cdot\frac{1}{1-\rho\alpha}\left[1+O\left(\frac{1}{\mathcal{K}}\right)\right]\,.
\end{aligned} \tag{A9}$$

The above together with (A6) gives (29).