# Virtual Partitioning for Robust Resource Sharing: Computational Techniques for Heterogeneous Traffic*

Sem Borst, Debasis Mitra
Bell Laboratories, Lucent Technologies
P.O. Box 636
Murray Hill, NJ 07974-0636

**Abstract**

We consider Virtual Partitioning (VP), a scheme for sharing a resource among several traffic classes in an efficient, fair, and robust manner. In the preliminary design stage, each traffic class is allocated a nominal capacity, which is based on expected offered traffic and required quality-of-service. During operations, if the current capacity usage by a class exceeds its nominal allocation, then it is declared to be in 'overload', and a state-dependent trunk reservation mechanism gives it lower priority in the admission of new calls. We develop efficient computational algorithms for the case of heterogeneous traffic, and perform extensive numerical experiments to demonstrate the accuracy of the various approximations.

The performance of VP is examined and compared to that of Complete Sharing and Complete Partitioning. Particular weight is placed on robustness, meaning that traffic classes with arrival rates conforming to the design continue to receive the required quality-of-service, despite the presence of misbehaving classes with excessive arrival rates. We adopt a reward-penalty paradigm as a combined measure for efficiency and fairness, and show that not only the revenue generated by VP is extremely close to the maximum achievable value, but that the structural form of the optimal policy also closely resembles that of VP. The numerical results confirm that the scheme is efficient, fair, and very robust.

*Partial results have been reported in "Virtual Partitioning for Resource Sharing by State-Dependent Priorities: Analysis, Approximations, and Performance for Heterogeneous Traffic" [4].

# 1 Introduction

This paper considers Virtual Partitioning (VP), a scheme for sharing a resource (like a link or a buffer) among several traffic classes in an efficient, fair, and robust manner. We focus on the case of multi-rate traffic, which is motivated by the integration of multiple services with heterogeneous characteristics in emerging Broadband/ATM networks. Various numerical algorithms will be developed and evaluated. VP was originally proposed by Mitra & Ziedins [13] who considered the single-rate case only. The performance of VP is examined and compared to that of Complete Sharing and Complete Partitioning. We adopt a reward-penalty paradigm as a combined measure for efficiency and fairness, and show that not only the revenue generated by VP is extremely close to the maximum achievable value, but that the structural form of the optimal policy also closely resembles that of VP.

We consider a single link of capacity $C$, which is offered calls from $K$ classes. Class-$k$ calls arrive as a Poisson process of rate $\lambda_k$, and have independent exponentially distributed holding times with mean $1/\mu_k$. A class-$k$ call requires capacity $b_k$ for the duration of the holding time. Define $S := \{n \in \mathbb{N}^K : \sum_{l=1}^{K} b_l n_l \leq C\}$ as the set of feasible link occupancies. The capacity requirements $b_k$ may correspond to the notion of effective bandwidths (see, for instance, [7] for various related notions).

VP has the following core ideas: first, at the time of design, each class is allocated a nominal capacity, say $C_k$ to class $k$, where $\sum_{k=1}^{K} C_k \geq C$; at each point in time, classes that are using less than their nominal allocations, which classes may be termed 'underloaded', are accorded higher priority; conversely, classes that are exceeding their nominal allocations, which may be termed 'overloaded' classes, are given lower priority; finally, the priority mechanism is implemented by a variant of the classical technique of trunk reservation [1], [10]. Specifically, suppose the link occupancy is $n \in S$ when a class-$k$ call arrives. Then the call is accepted if $\sum_{l=1}^{K} b_l n_l \leq C - b_k - t_k(n_k)$, and rejected otherwise. Thus, $t_k(n_k)$ may be interpreted as the trunk reservation level operating against class-$k$ calls when there are $n_k$ such calls in progress. In traditional trunk reservation schemes, $t_k(n_k) = r_k$. In VP, trunk reservation is dynamic: $t_k(n_k) = s_k \mathrm{I}_{\{b_k n_k \leq C_k - b_k\}} + t_k \mathrm{I}_{\{b_k n_k > C_k - b_k\}}$, where $s_k \leq t_k$. That is, the trunk reservation level operating against class-$k$ calls is either $s_k$ (small) or $t_k$ (large), depending upon whether the class is currently underloaded or overloaded. We should mention that the analysis in this paper does not rely on the above particular form of the function $t_k(\cdot)$.

VP builds on the notion of 'engineered' load and design. The nominal allocations $C_k$ may be derived from traffic forecasts in conjunction with quality-of-service requirements in terms of target call blocking probabilities. The multiplexing gain allows $\sum_{k=1}^{K} C_k/C \geq 1$, where the extent of overallocation is a subject of design to which the techniques of this paper may be applied.

By giving lower priority to overloaded classes, VP protects underloaded classes against overutilization of the resource. Specifically, the dynamic trunk reservation mechanism forces overloaded classes to back off in the event that an underloaded class needs to claim its allocated share of the capacity. It is perfectly reasonable to set $C_k = 0$ for some class $k$, e.g. to provide 'best-effort' service. It is clear from the above description of the motivations for VP that an important performance measure is the probability of blocking of calls when the call class is underloaded. We present numerical results which illustrate the efficacy of VP in making this conditional probability low.

Two classical schemes for resource sharing are Complete Sharing (CS), which allows all classes to share the resource indiscriminately, and Complete Partitioning (CP), which statically divides the resource among the classes, allowing each class the exclusive use of its allocated capacity. Under CS, one class may overwhelm all the others, while under CP the resource may be underutilized. VP includes both schemes as special cases, and may be interpreted as either 'disciplined' sharing or 'flexible' partitioning.

The above qualifications refer to the extreme robustness properties of VP. By robustness we have in mind not only the ability to respond to statistical fluctuations, which are inevitable even with perfect traffic forecasts, but also adaptability in a more extreme scenario where call arrival rates are different from the expected values. Robustness is clearly an important feature of effective resource sharing. If the arrival rate of a class is below forecast, then it is desirable that the scheme acts as if the nominal allocation is reduced. Conversely, if the arrival rate is higher and the overall traffic is light, then it is desirable to effectively extend the nominal allocation. In fact, VP automatically reduces to CS, which achieves a high multiplexing gain, when the overall traffic is light. In the other extreme case of severe overloading by a subset of classes, VP behaves as CP, isolating such classes. We present numerical results in which such features are demonstrated.

Mitra & Ziedins [13] formulated a problem based on rewards and penalties, which reflect the essential role of 'engineered' load, and the nature of the implied contract, in VP. In this formulation, each call of class $k$ that is carried earns a reward $w_k$. However, a penalty $v_k$ is registered for rejecting a call of class $k$ if the class is underloaded. For example, $v_k/w_k = 2$, as in our numerical exercises. We can then seek the call admission policy that maximizes the mean revenue earned per unit time. Markov Decision Theory [20], [21] may be used to obtain the optimal policy. The details of the optimal policy are inevitably complicated. VP is advanced as a policy which, while not optimal in general, yields revenues that closely approximate the optimal policies' revenues, besides having important advantages of simplicity and robustness. We give numerical results to support this claim.

The problem of fair and efficient resource sharing has a long and rich history [5], [6], [11]. Foschini, Gopinath, & Hayes [5] consider admission control policies that induce product-form equilibrium distributions, and show that a threshold policy is optimal. Gopal & Stern [6] use Markov Decision Theory to determine threshold policies that maximize the link utilization. Kraimeche & Schwartz [11] consider a class of restricted-access policies, with the aim of minimizing blocking probabilities. The work by Ash *et al.* [2] on class-of-service routing also aims to balance fairness and efficiency. Recent work on the application of VP to robust admission control in heterogeneous ATM systems is described in [14], and to buffer management in [12]. The extension to Hierarchical VP is made in [15]. The latter work is motivated by virtual private networking, where several customers, each having various service offerings or traffic classes, share the resource, and thus there is a clear need to differentiate resource sharing between customers and between services of individual customers.

The remainder of the paper is organized as follows. In Section 2, we present a detailed model description. Sections 3 and 4 cover one- and two-dimensional approximations, respectively. A crucial assumption in our approach is also common in fixed-point approximations for loss networks. Excellent sources of information on fixed-point approximations are Kelly [9] and Ross [19]. In Section 5, we discuss the numerical results. Some concluding remarks are made in Section 6.

## 2 Model description

To recapitulate, we consider a single link of capacity $C$, which is offered calls from $K$ classes. Class-$k$ calls arrive as a Poisson process of rate $\lambda_k$, and have independent exponentially distributed holding times with mean $1/\mu_k$. A class-$k$ call requires capacity $b_k$ for the duration of the holding time. Define $S := \{n \in \mathbb{N}^K : \sum_{l=1}^{K} b_l n_l \leq C\}$ as the set of feasible link occupancies. Without loss of generality, we may assume that the link capacity $C$ and the capacity requirements $b_k$, $k = 1, \ldots, K$, are integers with greatest common divisor 1.

We are interested in determining the blocking probabilities for the above-defined class of state-dependent trunk reservation strategies. Denote by $B_k$ the blocking probability for class-$k$ calls. Denote by $\mathbf{N} = (\mathbf{N}_1, \ldots, \mathbf{N}_K)$ a stochastic vector representing the link occupancy. For any $n \in S$, denote $\pi(n) := \Pr\{\mathbf{N} = n\}$. Define the set $A_k := \{n \in S : \sum_{l=1}^{K} b_l n_l \leq C - b_k - t_k(n_k)\}$ as the acceptance region for class-$k$ calls. For any $n \in S$, denote $a_k(n) := \mathrm{I}_{\{n \in A_k\}}$, with $\mathrm{I}_{\{E\}}$ denoting the indicator function of the event $E$. By the PASTA property, $B_k = \Pr\{\mathbf{N} \notin A_k\} = \sum_{n \notin A_k} \pi(n)$. By Little's law, $B_k = 1 - \mu_k \mathrm{E}\mathbf{N}_k / \lambda_k$, with $\mathrm{E}\mathbf{N}_k = \sum_{n \in S} n_k \pi(n)$.

The link occupancy constitutes an irreducible Markov process with the feasible region $S$ as the state space. The stationary distribution is uniquely determined by the global balance equations

$$[\sum_{k=1}^{K} \lambda_k a_k(n) + \sum_{k=1}^{K} n_k \mu_k] \pi(n) = \tag{1}$$

$$\sum_{k=1}^{K} \lambda_k a_k(n - e_k) \pi(n - e_k) \mathrm{I}_{\{n_k > 0\}} + \sum_{k=1}^{K} (n_k + 1) \mu_k \pi(n + e_k) \mathrm{I}_{\{\sum b_l n_l + b_k \leq C\}},$$

for all $n \in S$, with $e_k$ denoting the $k$-th unit vector, supplemented with the normalization condition $\sum_{n \in S} \pi(n) = 1$.

In the absence of a product form, calculating the link occupancy distribution involves numerically solving the balance equations, which is prohibitively demanding for all but the smallest link capacities. In view of the computational complexity of an exact approach, we develop in the next two sections methods for approximating the blocking probabilities.

# 3 One-dimensional approximations

We now construct approximations for the blocking probabilities. We intend to avoid solving exactly for the joint link occupancy distribution. Instead, we will be solving approximately for the marginal distribution of each of the $\mathbf{N}_k$'s.

### 3.1 Mitra-Ziedins Approach

Denote $\pi_k(i) := \Pr\{\mathbf{N}_k = i\}$ for $i = 0, \ldots, \lfloor C/b_k \rfloor$. Denote by $\mathbf{N}_0 := \sum_{k=1}^{K} b_k \mathbf{N}_k$ a stochastic variable representing the total amount of capacity in use. Denote $\pi_0(h) := \Pr\{\mathbf{N}_0 = h\}$ for $h = 0, \ldots, C$. With some minor abuse of notation, define $a_k(n_k, \sum_{l=1}^{K} b_l n_l) := a_k(n)$, reflecting the fact that $a_k(n)$ depends on $n$ only through $n_k$, $\sum_{l=1}^{K} b_l n_l$.

Summing the balance equations (1) over $\{n : n_k = i\}$ yields

$$[p_k^+(i) + p_k^-(i)]\pi_k(i) = p_k^+(i-1)\pi_k(i-1)\mathrm{I}_{\{i>0\}} + p_k^-(i+1)\pi_k(i+1)\mathrm{I}_{\{i<\lfloor C/b_k \rfloor\}}, \tag{2}$$

for all $i = 0, \ldots, \lfloor C/b_k \rfloor$, with $p_k^-(i) := i\mu_k$ and

$$p_k^+(i) := \lambda_k \sum_{n:\, n_k = i} a_k(n)\pi(n)/\pi_k(i) = \lambda_k \sum_{h=0}^{C} a_k(i,h)\Pr\{\mathbf{N}_0 = h \mid \mathbf{N}_k = i\}. \tag{3}$$

Thus, the number of class-$k$ calls, although *not* constituting a birth-death process, *does* have the stationary distribution of such a process, with birth and death rates $p_k^+(\cdot)$ and $p_k^-(\cdot)$, respectively. The quantities $p_k^+(i)$ and $p_k^-(i)$ may be interpreted as the intensities with which class-$k$ calls are admitted and completed, respectively, when there are $i$ such calls in progress.

Summing (2) over $\{i, \ldots, \lfloor C/b_k \rfloor\}$ yields $p_k^+(i-1)\pi_k(i-1)\mathrm{I}_{\{i>0\}} = p_k^-(i)\pi_k(i)$ for all $i = 0, \ldots, \lfloor C/b_k \rfloor$. Thus, the probabilities $\pi_k(\cdot)$ may be recursively computed from

$$\pi_k(i) = \frac{1}{G_k} \prod_{j=1}^{i} \frac{p_k^+(j-1)}{p_k^-(j)}, \qquad i = 0, \ldots, \lfloor C/b_k \rfloor, \tag{4}$$

with $G_k$ representing the normalization constant. The death rates $p_k^-(\cdot)$ are specified explicitly, while the birth rates $p_k^+(\cdot)$ in (3) are expressed in terms of the conditional distribution of $\mathbf{N}_0 \mid \mathbf{N}_k$. We make the following assumption, which is similar to the link independence assumption in Mitra & Ziedins [13]:

**Assumption 3.1**

$$\Pr\{\mathbf{N}_k = i, \mathbf{N}_0 = h\} \approx \Pr\{\mathbf{N}_k = i\}\Pr\{\mathbf{N}_0 = h\}, \tag{5}$$

for all $i = 0, \ldots, \lfloor C/b_k \rfloor$, $h = 0, \ldots, C$.

□

**Remark 3.1**

There are various refinements to Assumption 3.1 conceivable, e.g., taking into account the fact that $\Pr\{\mathbf{N}_0 \geq b_k\mathbf{N}_k\} = 1$. Limited numerical experience suggests that these refinements do not significantly improve the accuracy of the resulting approximations.

□

Substituting (5) into (3),

$$p_k^+(i) \approx \lambda_k \sum_{h=0}^{C} a_k(i,h)\Pr\{\mathbf{N}_0 = h\} = \lambda_k \sum_{h=0}^{C} a_k(i,h)\pi_0(h). \tag{6}$$

The corresponding approximation for the blocking probability for class-$k$ calls is

$$B_k = \sum_{i,h:\, a_k(i,h)=0} \Pr\{\mathbf{N}_k = i, \mathbf{N}_0 = h\} \approx \sum_{i,h:\, a_k(i,h)=0} \pi_k(i)\pi_0(h). \tag{7}$$

Summarizing so far, equation (7) expresses the blocking probabilities $B_k$ in terms of the probabilities $\pi_k(\cdot)$ and $\pi_0(\cdot)$. Equations (4) and (6) relate the probabilities $\pi_k(\cdot)$ to the probabilities $\pi_0(\cdot)$. So next we need to determine the probabilities $\pi_0(\cdot)$.

Summing the balance equations (1) over $\{n : \sum_{k=1}^{K} b_k n_k = h\}$ yields

$$[\sum_{k=1}^{K} q_k^+(h) + \sum_{k=1}^{K} q_k^-(h)]\pi_0(h) = \tag{8}$$

$$\sum_{k=1}^{K} q_k^+(h-b_k)\pi_0(h-b_k)\mathrm{I}_{\{h \geq b_k\}} + \sum_{k=1}^{K} q_k^-(h+b_k)\pi_0(h+b_k)\mathrm{I}_{\{h \leq C-b_k\}},$$

for all $h = 0, \ldots, C$, with

$$q_k^+(h) := \lambda_k \sum_{n:\, \sum b_l n_l = h} a_k(n)\pi(n)/\pi_0(h) = \lambda_k \sum_{i=0}^{\lfloor C/b_k \rfloor} a_k(i,h)\Pr\{\mathbf{N}_k = i \mid \mathbf{N}_0 = h\}, \tag{9}$$

and

$$q_k^-(h) := \mu_k \sum_{n:\sum b_l n_l = h} n_k \pi(n)/\pi_0(h) = \mu_k \mathrm{E}(\mathbf{N}_k \mid \mathbf{N}_0 = h). \tag{10}$$

Thus, the total amount of capacity in use, although *not* constituting a Markov process, *does* have the stationary distribution of such a process, with transitions from state $h$ to states $h + b_k$ and $h - b_k$ at rates $q_k^+(h)$ and $q_k^-(h)$, respectively. The quantities $q_k^+(h)$ and $q_k^-(h)$ may be interpreted as the intensities with which class-$k$ calls are admitted and completed, respectively, when the total amount of capacity in use is $h$.

Consistent with invoking Assumption 3.1 to determine the rates $p_k^+(\cdot)$ in (3), we also assume independence to approximate the rates $q_k^+(\cdot)$ in (9). Substituting (5) into (9),

$$q_k^+(h) \approx \lambda_k \sum_{i=0}^{\lfloor C/b_k \rfloor} a_k(i,h)\mathrm{Pr}\{\mathbf{N}_k = i\} = \lambda_k \sum_{i=0}^{\lfloor C/b_k \rfloor} a_k(i,h)\pi_k(i). \tag{11}$$

What remains is determining the rates $q_k^-(\cdot)$ in (10), which amounts to finding the conditional mean $\mathrm{E}(\mathbf{N}_k \mid \mathbf{N}_0 = h)$. Below, in Sections 3.1.1 and 3.1.2, we consider two different approaches to this problem. Before doing that, we should observe that the above framework, which is based on the independence Assumption 3.1 and the coupled computations of the probabilities $\pi_k(\cdot)$ and $\pi_0(\cdot)$, is similar to the framework of '$\alpha$-$\beta$-networks' in [13]. In [13], only the single-rate case was considered, i.e., $b_k = 1$, $\mu_k = 1$, for all $k = 1, \ldots, K$, in which case equations (8) reduce to

$$[q_0^+(h) + q_0^-(h)]\pi_0(h) = q_0^+(h-1)\pi_0(h-1)\mathrm{I}_{\{h>0\}} + q_0^-(h+1)\pi_0(h+1)\mathrm{I}_{\{h<C\}}, \tag{12}$$

for all $h = 0, \ldots, C$, with $q_0^+(h) := \sum_{k=1}^{K} q_k^+(h)$ and $q_0^-(h) := \sum_{k=1}^{K} q_k^-(h) = h$. Summing (12) over $\{h, \ldots, C\}$ yields

$$q_0^+(h-1)\pi_0(h-1)\mathrm{I}_{\{h>0\}} = q_0^-(h)\pi_0(h), \tag{13}$$

for all $h = 0, \ldots, C$. Thus, the probabilities $\pi_0(\cdot)$ may be recursively computed from

$$\pi_0(h) = \frac{1}{G_0} \prod_{j=1}^{h} \frac{q_0^+(j-1)}{q_0^-(j)}, \qquad h = 0, \ldots, C,$$

with $G$ representing the normalization constant.

If the single-rate assumption does not apply, i.e., it is not the case that $b_k = 1$, $\mu_k = 1$ for all $k = 1, \ldots, K$, then the above simplification in determining the probabilities $\pi_0(\cdot)$ from equations (8) does not arise. Note that solving equations (8) amounts to analyzing a single link, which is offered traffic with different capacity requirements $b_k$, different mean holding times $1/\mu_k$, and state-dependent arrival rates $q_k^+(\cdot)$. Below we employ two relatively simple approaches to this problem, which agree with (13) in case $b_k = 1$, $\mu_k = 1$ for all $k = 1, \ldots, K$.

### 3.1.1 Bean-Gibbens-Zachary Approximation

The first approach approximates the conditional mean $\mathrm{E}(\mathbf{N}_k \mid \mathbf{N}_0 = h)$. If we simply use Assumption 3.1, then we obtain

$$\mathrm{E}(\mathbf{N}_k \mid \mathbf{N}_0 = h) \approx \mathrm{E}\mathbf{N}_k, \tag{14}$$

which is called the *constant* death-rate approximation in Bean *et al.* [3], who consider the standard trunk reservation strategy. A slight refinement is

$$\mathrm{E}(\mathbf{N}_k \mid \mathbf{N}_0 = h) \approx \frac{h\mathrm{E}\mathbf{N}_k}{\sum\limits_{m=1}^{K} b_m \mathrm{E}\mathbf{N}_m}, \tag{15}$$

which is referred to as the *variable* death-rate approximation in [3]. By definition, $\mathrm{E}\mathbf{N}_k = \sum\limits_{i=0}^{\lfloor C/b_k \rfloor} i\pi_k(i)$. Alternatively, we could obtain $\mathrm{E}\mathbf{N}_k$ from (7) using the relationship $\mathrm{E}\mathbf{N}_k = \lambda_k(1 - B_k)/\mu_k$. Similarly, we could also derive $B_k$ from these two formulas instead of (7).

### 3.1.2 Roberts' Approximation

The second approach eliminates the rates $q_k^-(\cdot)$ from equations (8). If $b_k = b$, $\mu_k = \mu$ for all $k = 1, \ldots, K$, then equations (8) reduce to

$$[q_0^+(h) + q_0^-(h)]\pi_0(h) = q_0^+(h-b)\pi_0(h-b)\mathrm{I}_{\{h \geq b\}} + q_0^-(h+b)\pi_0(h+b)\mathrm{I}_{\{h \leq C-b\}}, \tag{16}$$

for all $h = 0, b, 2b, \ldots, \lfloor C/b \rfloor b$, with $q_0^+(h) := \sum\limits_{k=1}^{K} q_k^+(h)$ and $q_0^-(h) := \sum\limits_{k=1}^{K} q_k^-(h) = h\mu/b$. Summing (16) over $\{h, h+b, h+2b, \ldots, \lfloor C/b \rfloor b\}$ yields $h\pi_0(h) = \sum\limits_{k=1}^{K} \frac{b}{\mu} q_k^+(h-b)\pi_0(h-b)$. We now assume that, even if it is not the case that $b_k = b$, $\mu_k = \mu$ for all $k = 1, \ldots, K$,

$$h\pi_0(h) \approx \sum_{k=1}^{K} \frac{b_k}{\mu_k} q_k^+(h-b_k)\pi_0(h-b_k), \tag{17}$$

which is similar to the approximation in Roberts [18] for the standard trunk reservation strategy.

In conclusion, equations (4) determines the probabilities $\pi_k(\cdot)$ up to the birth rates $p_k^+(\cdot)$. Equation (6) expresses the birth rates $p_k^+(\cdot)$ in terms of the probabilities $\pi_0(\cdot)$. In the approach of section 3.1.1, equations (8) relate the probabilities $\pi_0(\cdot)$ to the transition rates $q_k^+(\cdot)$ and $q_k^-(\cdot)$. Equations (14) or (15) relate the transition rates $q_k^-(\cdot)$ to the probabilities $\pi_k(\cdot)$. Equation (11) expresses the rates $q_k^+(\cdot)$ in terms of the probabilities $\pi_k(\cdot)$. In the approach of section 3.1.2, equations (17) relate the probabilities $\pi_0(\cdot)$ to the rates $q_k^+(\cdot)$. We have thus obtained a set of fixed-point equations for each of the two approaches.

### 3.2 Convolutional Approach

Considering the difficulties in determining the probabilities $\pi_0(\cdot)$ from equations (8), we could instead determine the distribution of $\mathbf{N}_0$ directly from the distribution of the $\mathbf{N}_k$'s, using the fact that $\mathbf{N}_0 = \sum_{k=1}^{K} b_k \mathbf{N}_k$:

**Assumption 3.2**

$$\Pr\{\mathbf{N}_0 = h\} \approx \frac{\sum_{n:\sum b_l n_l = h} \prod_{k=1}^{K} \Pr\{\mathbf{N}_k = n_k\}}{\sum_{n:\sum b_l n_l \leq C} \prod_{k=1}^{K} \Pr\{\mathbf{N}_k = n_k\}}, \tag{18}$$

for all $h = 0, \ldots, C$.

□

Thus, equation (18) represents the distribution of $\mathbf{N}_0$ as the convolution of the distribution of the $\mathbf{N}_k$'s, which leads to a set of fixed-point equations. Sophisticated techniques for implementing convolutions exist, such as the Fast Fourier Transform (see, for instance, [16]), and these may be exploited to gain relief in an otherwise difficult computational task.

Remember that we needed to determine the distribution of $\mathbf{N}_0$, because we used the independence Assumption 3.1 to relate the rates $p_k^+(\cdot)$ to the probabilities $\pi_0(\cdot)$. Instead of computing the distribution of $\mathbf{N}_0$ from the distribution of the $\mathbf{N}_k$'s, we could also use the probabilities $\pi_k(\cdot)$ to determine the rates $p_k^+(\cdot)$ directly:

**Assumption 3.3**

$$\Pr\{\mathbf{N} = n\} \approx \frac{\prod_{k=1}^{K} \Pr\{\mathbf{N}_k = n_k\}}{\sum_{n:\sum b_l n_l \leq C} \prod_{k=1}^{K} \Pr\{\mathbf{N}_k = n_k\}}, \tag{19}$$

for all $n \in S$.

$\square$

Note that (19) allows us to compute the rates $p_k^+(\cdot)$ in (3) from the probabilities $\pi_k(\cdot)$, which again yields a set of fixed-point equations.

# 4 Two-dimensional approximations

Instead of the marginal distribution of each of the $\mathbf{N}_k$'s, we now consider the joint distribution of each of the pairs $(\mathbf{N}_k, \mathbf{N}_0)$. Denote $\pi_k(i,h) := \Pr\{\mathbf{N}_k = i, \mathbf{N}_0 = h\}$ for $i = 0, \ldots, \lfloor C/b_k \rfloor$, $h = ib_k, \ldots, C$. Note that the blocking probability for class-$k$ calls may be obtained from

$$B_k = \sum_{i,h:a_k(i,h)=0} \pi_k(i,h). \tag{20}$$

Summing the balance equations (1) over $\{n : n_k = i, n_0 = h\}$ yields

$$[\sum_{l=1}^{K} r_{k,l}^+(i,h) + \sum_{l=1}^{K} r_{k,l}^-(i,h)]\pi_k(i,h) = \tag{21}$$

$$r_{k,k}^+(i-1, h-b_k)\pi_k(i-1, h-b_k)\mathrm{I}_{\{i>0, h\geq b_k\}} +$$

$$\sum_{l\neq k} r_{k,l}^+(i, h-b_l)\pi_k(i, h-b_l)\mathrm{I}_{\{h\geq b_l\}} +$$

$$r_{k,k}^-(i+1, h+b_k)\pi_k(i+1, h+b_k)\mathrm{I}_{\{i<\lfloor C/b_k\rfloor, h\leq C-b_k\}} +$$

$$\sum_{l\neq k} r_{k,l}^-(i, h+b_l)\pi_k(i, h+b_l), \mathrm{I}_{\{h\leq C-b_l\}}$$

for all $i = 0, \ldots, \lfloor C/b_k \rfloor$, $h = ib_k, \ldots, C$, with $r^+_{k,k}(i,h) := \lambda_k a_k(i,h)$, $r^-_{k,k}(i,h) := i\mu_k$,

$$\begin{aligned} r^+_{k,l}(i,h) &:= \lambda_l \sum_{n:\, n_k = i, n_0 = h} a_l(n)\pi(n)/\pi_k(i,h) \\ &= \lambda_l \sum_{j=0}^{\lfloor C/b_l \rfloor} a_l(j,h)\Pr\{\mathbf{N}_l = j \mid \mathbf{N}_k = i, \mathbf{N}_0 = h\}, \end{aligned} \tag{22}$$

and

$$r^-_{k,l}(i,h) := \mu_l \sum_{n:\, n_k = i, n_0 = h} \pi(n) n_l / \pi_k(i,h) = \mu_l \mathrm{E}(\mathbf{N}_l \mid \mathbf{N}_k = i, \mathbf{N}_0 = h). \tag{23}$$

The transition rates $r^+_{k,k}(\cdot,\cdot)$ and $r^-_{k,k}(\cdot,\cdot)$ are specified explicitly, while the transition rates $r^+_{k,l}(\cdot,\cdot)$ and $r^-_{k,l}(\cdot,\cdot)$ for $l \neq k$ in (22) and (23) are expressed in terms of the conditional distribution of $\mathbf{N}_l \mid (\mathbf{N}_k, \mathbf{N}_0)$. We make the following independence assumption:

**Assumption 4.1**

$$\Pr\{\mathbf{N}_l = j, \mathbf{N}_k = i, \mathbf{N}_0 = h\} \approx \Pr\{\mathbf{N}_l = j \mid \mathbf{N}_0 = h\}\Pr\{\mathbf{N}_k = i, \mathbf{N}_0 = h\}, \qquad l \neq k, \tag{24}$$

for all $j = 0, \ldots, \lfloor C/b_l \rfloor$, $i = 0, \ldots, \lfloor C/b_k \rfloor$, $h = ib_k + jb_l$.

□

Substituting (24) into (22),

$$\begin{aligned} r^+_{k,l}(i,h) &\approx \lambda_l \sum_{j=0}^{\lfloor C/b_l \rfloor} a_l(j,h)\Pr\{\mathbf{N}_l = j \mid \mathbf{N}_0 = h\}, \\ &= \lambda_l \left( \sum_{j=0}^{\lfloor C/b_l \rfloor} a_l(j,h)\pi_l(j,h) \right) \Bigg/ \left( \sum_{j=0}^{\lfloor C/b_l \rfloor} \pi_l(j,h) \right). \end{aligned} \tag{25}$$

What remains is determining the rates $r^-_{k,l}(i,h)$ in (23), which amounts to finding the conditional mean $\mathrm{E}(\mathbf{N}_l \mid \mathbf{N}_k = i, \mathbf{N}_0 = h)$. If we simply use Assumption 4.1, then we obtain

$$\begin{aligned} \mathrm{E}(\mathbf{N}_l \mid \mathbf{N}_k = i, \mathbf{N}_0 = h) &\approx \mathrm{E}(\mathbf{N}_l \mid \mathbf{N}_0 = h) \\ &= \left( \sum_{j=0}^{\lfloor C/b_l \rfloor} j\pi_l(j,h) \right) \Bigg/ \left( \sum_{j=0}^{\lfloor C/b_l \rfloor} \pi_l(j,h) \right). \end{aligned} \tag{26}$$

A slight refinement is

$$\begin{aligned} \mathrm{E}(\mathbf{N}_l \mid \mathbf{N}_k = i, \mathbf{N}_0 = h) &\approx \frac{(h - ib_k)\mathrm{E}(\mathbf{N}_l \mid \mathbf{N}_0 = h)}{\sum_{m \neq k} b_m \mathrm{E}(\mathbf{N}_m \mid \mathbf{N}_0 = h)} \\ &= \frac{(h - ib_k)\left( \sum_{j=0}^{\lfloor C/b_l \rfloor} j\pi_l(j,h) \right) \Big/ \left( \sum_{j=0}^{\lfloor C/b_l \rfloor} \pi_l(j,h) \right)}{\sum_{m \neq k} b_m \left( \sum_{j=0}^{\lfloor C/b_m \rfloor} j\pi_m(j,h) \right) \Big/ \left( \sum_{j=0}^{\lfloor C/b_m \rfloor} \pi_m(j,h) \right)}. \end{aligned} \tag{27}$$

In conclusion, equations (21) determine the probabilities $\pi_k(\cdot,\cdot)$ up to the transition rates $r^+_{k,l}(\cdot,\cdot)$ and $r^-_{k,l}(\cdot,\cdot)$ for $l \neq k$. Conversely, equations (25), and (26) or (27) relate the transition rates $r^+_{k,l}(\cdot,\cdot)$ and $r^-_{k,l}(\cdot,\cdot)$ for $l \neq k$ to the probabilities $\pi_k(\cdot,\cdot)$. We have thus obtained a set of fixed-point equations.

# 5 Numerical experiments

We now describe the numerical experiments that we have conducted. First, the accuracies of the various one-dimensional approximations are studied. Then, the two-dimensional approximation is evaluated. Next, we compare the performance of Virtual Partitioning to that of Complete Sharing and Complete Partitioning. Finally, we present results which compare the revenues generated by VP to that of the optimal call admission policy for the problem with rewards and penalties.

We consider models with two call classes: class-1 calls have mean holding times $1/\mu_1 = 1/2$, and capacity requirements $b_1 = 1$; class-2 calls have longer mean holding times $1/\mu_2 = 1$, and larger capacity requirements $b_2 = 2$. Note that two-class models seem to be the most challenging ones to the various independence assumptions, since each of the individual classes accounts for a substantial portion of the total amount of capacity in use.

We consider two different settings for the nominal allocations: (I). $(C_1, C_2) = (1, 2)C/3$, i.e., the greater portion of the capacity is nominally allocated to the longer and larger class-2 calls; (II). $(C_1, C_2) = (2, 1)C/3$. As mentioned earlier, VP builds on the notion of 'engineered' load and design. The nominal allocations may be derived from traffic forecasts in conjunction with quality-of-service requirements in terms of target call blocking probabilities. Taking the inverse route here, we determine the 'engineered' load in both cases from $\lambda^*_k/\mu_k = C_k/b_k - \sqrt{C_k/b_k}$, $k = 1, 2$, and consider various loading scenarios $(\sigma_1, \sigma_2)$, with $\sigma_k := \lambda_k/\lambda^*_k$. The more than proportional increase in the 'engineered' load with growing capacity accounts for scaling efficiencies. We always take trunk reservation levels $s_1 = s_2 = 0$ and $t_1 = t_2 = t$ for various values of $t$.

### 5.1 Accuracy of One-Dimensional Approximations

We have examined three different one-dimensional approximations. In each of the three approximations, we use equation (4) to compute the probabilities $\pi_k(\cdot)$ from the rates $p_k^+(\cdot)$ and $p_k^-(\cdot)$. In the first two approximations, we employ equation (6) to derive the rates $p_k^+(\cdot)$ from the probabilities $\pi_0(\cdot)$. In the first approximation (Bean-Gibbens-Zachary), we use equations (8) to calculate the probabilities $\pi_0(\cdot)$ from the rates $q_k^+(\cdot)$ and $q_k^-(\cdot)$, and employ equation (15) to obtain the rates $q_k^-(\cdot)$. In the second approximation (Roberts'), we use equations (17) to compute the probabilities $\pi_0(\cdot)$ from the rates $q_k^+(\cdot)$. In each of the first two approximations, we invoke equation (11) to derive the rates $q_k^+(\cdot)$ from the probabilities $\pi_k(\cdot)$. In the third approximation (convolutional approach), we invoke equation (19) to relate the rates $p_k^+(\cdot)$ directly to the probabilities $\pi_k(\cdot)$.

We consider the two-class models described above. We take the link capacity to be $C = 1500$ and set the trunk reservation level to $t = 10$. The results are shown in Tables 5.1 and 5.2 below. The columns 'exact' contain the blocking probabilities $(B_1, B_2)$ obtained from solving the balance equations (1). We have solved the equations using Gauss-Seidel iteration, preceded by Gaussian elimination on a subset of the state space to obtain a good starting solution. The columns 'BGZ', 'Roberts', and 'Convolution' show the results obtained from the corresponding approximations mentioned in the above paragraph. The respective sets of fixed-point equations were solved through numerical iteration. Although convergence is difficult to establish, we did not encounter any problems in that respect. The rows correspond to various loading scenarios relative to the target arrival rates $(\lambda_1^*, \lambda_2^*)$.

| | exact | | BGZ | | Roberts | | Convolution | |
|---|---|---|---|---|---|---|---|---|
| $(\sigma_1, \sigma_2)$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| (0.80, 1.00) | .00000 | .00015 | .00001 | .00006 | .00001 | .00009 | .00001 | .00030 |
| (0.80, 1.10) | .00000 | .01178 | .00003 | .01116 | .00000 | .01093 | .00081 | .01552 |
| (0.80, 1.20) | .00001 | .06008 | .00003 | .06059 | .00000 | .05881 | .00402 | .06535 |
| (0.90, 0.90) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00001 |
| (0.90, 1.00) | .00002 | .00218 | .00023 | .00117 | .00026 | .00139 | .00020 | .00349 |
| (0.90, 1.10) | .00007 | .03343 | .00044 | .03335 | .00015 | .03176 | .00231 | .03898 |
| (0.90, 1.20) | .00011 | .09428 | .00266 | .11816 | .00021 | .10164 | .00679 | .09825 |
| (1.00, 0.80) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| (1.00, 0.90) | .00007 | .00003 | .00003 | .00002 | .00007 | .00004 | .00027 | .00008 |
| (1.00, 1.00) | .00395 | .00885 | .00463 | .00629 | .00516 | .00704 | .00760 | .01065 |
| (1.00, 1.10) | .00956 | .05881 | .01430 | .06256 | .01166 | .05909 | .01262 | .06321 |
| (1.00, 1.20) | .01141 | .12528 | .03118 | .13887 | .01872 | .13323 | .01545 | .12940 |
| (1.10, 0.80) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00001 | .00000 |
| (1.10, 0.90) | .00201 | .00008 | .00146 | .00002 | .00259 | .00014 | .00464 | .00078 |
| (1.10, 1.00) | .03261 | .01198 | .03692 | .00788 | .04030 | .01033 | .04796 | .01451 |
| (1.10, 1.10) | .05874 | .06739 | .06585 | .07058 | .06541 | .07015 | .06585 | .06847 |
| (1.20, 0.80) | .00006 | .00000 | .00005 | .00000 | .00012 | .00000 | .00028 | .00004 |
| (1.20, 0.90) | .01342 | .00015 | .01150 | .00007 | .01614 | .00059 | .02125 | .00409 |
| (1.20, 1.00) | .08292 | .01267 | .08869 | .00958 | .09209 | .01247 | .09206 | .02238 |

**Table 5.1**: Blocking probabilities for $C = 1500$, $(C_1, C_2) = (500, 1000)$, $t = 10$.

| | exact | | BGZ | | Roberts | | Convolution | |
|---|---|---|---|---|---|---|---|---|
| $(\sigma_1, \sigma_2)$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| (0.80, 1.00) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| (0.80, 1.10) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| (0.80, 1.20) | .00000 | .00009 | .00000 | .00004 | .00000 | .00007 | .00001 | .00017 |
| (0.90, 0.90) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| (0.90, 1.00) | .00000 | .00009 | .00000 | .00002 | .00001 | .00004 | .00001 | .00017 |
| (0.90, 1.10) | .00003 | .00242 | .00005 | .00167 | .00002 | .00183 | .00022 | .00375 |
| (0.90, 1.20) | .00020 | .01725 | .00023 | .01636 | .00003 | .01501 | .00141 | .02135 |
| (1.00, 0.80) | .00000 | .00000 | .00001 | .00000 | .00002 | .00001 | .00013 | .00002 |
| (1.00, 0.90) | .00080 | .00031 | .00046 | .00020 | .00068 | .00035 | .00248 | .00068 |
| (1.00, 1.00) | .00422 | .00934 | .00401 | .00615 | .00445 | .00691 | .00713 | .01191 |
| (1.00, 1.10) | .00745 | .04587 | .00841 | .04314 | .00760 | .04193 | .00947 | .05187 |
| (1.00, 1.20) | .00900 | .09981 | .01329 | .11001 | .00885 | .09982 | .01188 | .10469 |
| (1.10, 0.80) | .00780 | .00000 | .00710 | .00000 | .00879 | .00001 | .01560 | .00232 |
| (1.10, 0.90) | .02749 | .00072 | .02695 | .00019 | .02981 | .00033 | .03944 | .00687 |
| (1.10, 1.00) | .05195 | .01629 | .05982 | .01229 | .06104 | .01440 | .06394 | .01988 |
| (1.10, 1.10) | .06491 | .06509 | .07094 | .07082 | .07061 | .06993 | .07024 | .06671 |
| (1.20, 0.80) | .05648 | .00000 | .05613 | .00000 | .05832 | .00007 | .06672 | .01240 |
| (1.20, 0.90) | .09105 | .00079 | .09683 | .00044 | .09888 | .00079 | .09770 | .02138 |
| (1.20, 1.00) | .12131 | .01719 | .13949 | .01979 | .13941 | .02383 | .12911 | .03509 |

**Table 5.2**: Blocking probabilities for $C = 1500$, $(C_1, C_2) = (1000, 500)$, $t = 10$.

Tables 5.1 and 5.2 show that the proposed one-dimensional approximations perform quite well, even for two-class systems. As mentioned earlier, the accuracy is likely to further improve as the number of classes increases. Remember that solving the equations (8) is substantially more demanding than using the recursion (17) to determine the probabilities $\pi_0(\cdot)$. The two corresponding approximations perform very similarly though, suggesting that the higher complexity of the BGZ approximation does not render greater accuracy than the Roberts' approximation. Both these approximations tend to slightly outperform the convolutional approximation (19). Apparently, coupling the computation of the probabilities $\pi_k(\cdot)$ with that of the probabilities $\pi_0(\cdot)$ improves the accuracy, despite the complications that arise in the multi-rate case.

### 5.2 Accuracy of Two-Dimensional Approximations

We have also investigated the two-dimensional approximation. We consider the two-class models described above. We take the link capacity to be $C = 150$ and set the trunk reservation level to $t = 5$. Again, two-class models seem to be the most challenging ones to the independence Assumption 4.1. In practice, however, it would not really make sense to use the approximation for two-class models, since repeatedly computing the joint distribution of $(\mathbf{N}_1, \mathbf{N}_0)$ and $(\mathbf{N}_2, \mathbf{N}_0)$ is computationally more demanding than 'simply' calculating the joint distribution of $(\mathbf{N}_1, \mathbf{N}_2)$ only once.

The results are shown in Tables 5.3 and 5.4 below. The columns 'exact' contain the blocking probabilities $(B_1, B_2)$ obtained from solving the balance equations (1). The columns 'approximation' give the results obtained from equations (20), (21), (23), (25), and (27).

| | exact | | approximation | |
|---|---|---|---|---|
| $(\sigma_1, \sigma_2)$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| (0.80, 1.00) | .00072 | .01553 | .00069 | .01551 |
| (0.80, 1.10) | .00126 | .03894 | .00126 | .03891 |
| (0.80, 1.20) | .00180 | .07330 | .00187 | .07324 |
| (0.90, 0.90) | .00173 | .00671 | .00160 | .00674 |
| (0.90, 1.00) | .00382 | .02268 | .00362 | .02271 |
| (0.90, 1.10) | .00615 | .05170 | .00605 | .05167 |
| (0.90, 1.20) | .00818 | .09070 | .00837 | .09059 |
| (1.00, 0.80) | .00191 | .00169 | .00179 | .00176 |
| (1.00, 0.90) | .00612 | .00917 | .00573 | .00941 |
| (1.00, 1.00) | .01258 | .02878 | .01189 | .02918 |
| (1.00, 1.10) | .01921 | .06185 | .01857 | .06225 |
| (1.00, 1.20) | .02459 | .10401 | .02448 | .10424 |
| (1.10, 0.80) | .00526 | .00216 | .00504 | .00236 |
| (1.10, 0.90) | .01519 | .01101 | .01438 | .01170 |
| (1.10, 1.00) | .02903 | .03294 | .02742 | .03420 |
| (1.10, 1.10) | .04230 | .06844 | .04039 | .06992 |
| (1.20, 0.80) | .05260 | .03557 | .04976 | .03798 |
| (1.20, 0.90) | .02955 | .01232 | .02821 | .01358 |
| (1.20, 1.00) | .05260 | .03557 | .04976 | .03798 |

**Table 5.3**: Blocking probabilities for $C = 150$, $(C_1, C_2) = (50, 100)$, $t = 5$.

| | exact | | approximation | |
|---|---|---|---|---|
| $(\sigma_1, \sigma_2)$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| (0.80, 1.00) | .00020 | .00342 | .00021 | .00338 |
| (0.80, 1.10) | .00037 | .00809 | .00038 | .00803 |
| (0.80, 1.20) | .00064 | .01622 | .00066 | .01616 |
| (0.90, 0.90) | .00162 | .00556 | .00163 | .00534 |
| (0.90, 1.00) | .00254 | .01354 | .00261 | .01317 |
| (0.90, 1.10) | .00358 | .02685 | .00375 | .02631 |
| (0.90, 1.20) | .00469 | .04558 | .00500 | .04491 |
| (1.00, 0.80) | .00659 | .00438 | .00653 | .00425 |
| (1.00, 0.90) | .01047 | .01273 | .01039 | .01243 |
| (1.00, 1.00) | .01451 | .02812 | .01453 | .02758 |
| (1.00, 1.10) | .01821 | .05080 | .01847 | .04991 |
| (1.00, 1.20) | .02134 | .07930 | .02201 | .07800 |
| (1.10, 0.80) | .02433 | .00670 | .02413 | .00695 |
| (1.10, 0.90) | .03411 | .01848 | .03369 | .01907 |
| (1.10, 1.00) | .04321 | .03892 | .04267 | .03980 |
| (1.10, 1.10) | .05077 | .06740 | .05034 | .06826 |
| (1.20, 0.80) | .08576 | .04463 | .08432 | .04811 |
| (1.20, 0.90) | .07228 | .02168 | .07136 | .02387 |
| (1.20, 1.00) | .08576 | .04463 | .08432 | .04811 |

**Table 5.4**: Blocking probabilities for $C = 150$, $(C_1, C_2) = (100, 50)$, $t = 5$.

Tables 5.3 and 5.4 show that the two-dimensional approximation performs extremely well, even for two-class systems. As mentioned earlier, the accuracy is likely to further improve as the number of classes increases.

A comparison with the one-dimensional approximations shows a trade-off between complexity and accuracy. Results for $C = 150$ not shown here indicate that the one-dimensional approximations are typically less accurate. In terms of computational effort, however, the two-dimensional approximations are significantly more demanding. In most cases of practical interest, we would not expect the greater accuracy of the two-dimensional approximations to warrant the higher complexity.

### 5.3 Impact of Trunk Reservation Parameters

We now illustrate the effect of the trunk reservation parameters on the blocking probabilities. We consider the two-class models described above. We take the link capacity to be $C = 1500$, and let the loading be $(\sigma_1, \sigma_2) = (1.00, 1.20)$. The results are shown in Tables 5.5 and 5.6 below. The columns 'VP' contain the standard blocking probabilities, while the columns 'Conditional' give the blocking probabilities conditioned on the corresponding class being underloaded. These were both obtained by solving the respective sets of balance equations (1). The rows correspond to increasing trunk reservation levels. Notice that for $t = 0$ Virtual Partitioning reduces to Complete Sharing.

| | VP | | Conditional | |
|---|---|---|---|---|
| $t$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| 0 | .05170 | .10105 | .05145 | .01250 |
| 1 | .02781 | .11220 | .02518 | .01302 |
| 2 | .01859 | .11691 | .01282 | .00630 |
| 3 | .01448 | .11938 | .00662 | .00437 |
| 4 | .01271 | .12083 | .00346 | .00213 |
| 5 | .01182 | .12189 | .00182 | .00136 |
| 6 | .01150 | .12270 | .00097 | .00070 |
| 7 | .01132 | .12342 | .00051 | .00044 |
| 8 | .01133 | .12407 | .00027 | .00024 |
| 9 | .01133 | .12469 | .00014 | .00014 |
| 10 | .01141 | .12528 | .00008 | .00008 |

**Table 5.5**: Blocking probabilities for $C = 150$, $(C_1, C_2) = (50, 100)$.

| | VP | | Conditional | |
|---|---|---|---|---|
| $t$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| 0 | .02741 | .05442 | .02698 | .01067 |
| 1 | .02089 | .06782 | .01972 | .00748 |
| 2 | .01683 | .07658 | .01461 | .00178 |
| 3 | .01413 | .08278 | .01085 | .00098 |
| 4 | .01238 | .08718 | .00812 | .00027 |
| 5 | .01119 | .09050 | .00610 | .00013 |
| 6 | .01040 | .09307 | .00462 | .00004 |
| 7 | .00983 | .09518 | .00351 | .00002 |
| 8 | .00945 | .09693 | .00267 | .00000 |
| 9 | .00917 | .09846 | .00205 | .00000 |
| 10 | .00900 | .09981 | .00157 | .00000 |

**Table 5.6**: Blocking probabilities for $C = 150$, $(C_1, C_2) = (100, 50)$.

Tables 5.7 and 5.8 indicate that the (unconditional) blocking probabilities quickly converge as the trunk reservation parameters are increased. This confirms the common fact that even moderate levels of trunk reservation are extremely effective in achieving the desired blocking performance. Further observe that the conditional blocking probabilities seem to decrease to zero geometrically fast.

### 5.4 Comparisons of VP, CP, and CS

We now compare the performance of Virtual Partitioning to that of Complete Sharing and Complete Partitioning. In CP, we assume that the amount of capacity exclusively available for class-$k$ calls equals $C_k$, the nominal allocation in VP.

We consider the two-class models described above. We take the link capacity to be $C = 1500$, and set the trunk reservation level in VP to $t = 10$. The results are shown in Tables 5.7 and 5.8 below. The rows correspond again to the various loading scenarios relative to the target arrival rates $(\lambda_1^*, \lambda_2^*)$. The columns 'VP' contain the standard blocking probabilities, while the columns 'Conditional' give the blocking probabilities conditioned on the corresponding class being underloaded.

The columns 'CS' show the blocking probabilities under CS. These were computed from the Kaufman-Roberts recursion [8], [17]

$$\pi_0(j) = \sum_{k=1}^{2} \frac{\lambda_k b_k}{j \mu_k} \pi_0(j - b_k) \mathrm{I}_{\{j \geq b_k\}}, \qquad j = 1, \ldots, C,$$

together with $\sum_{j=0}^{C} \pi_0(j) = 1$, and

$$B_k = \sum_{j=C-b_k+1}^{C} \pi_0(j), \qquad k = 1, 2.$$

The columns 'CP' give the blocking probabilities under CP with parameters $(C_1, C_2)$. These were calculated from the recursion

$$\pi_k(i) = \frac{\lambda_k}{\mu_k} \pi_k(i-1), \qquad i = 1, \ldots, \lfloor C_k/b_k \rfloor,$$

together with $\sum_{i=0}^{\lfloor C_k/b_k \rfloor} \pi_k(i) = 1$, and

$$B_k = \pi_k(\lfloor C_k/b_k \rfloor), \qquad k = 1, 2.$$

| | VP | | Conditional | | CS | | CP | |
|---|---|---|---|---|---|---|---|---|
| $(\sigma_1, \sigma_2)$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| (0.80, 1.00) | .00000 | .00015 | .00000 | .00000 | .00003 | .00006 | .00000 | .01250 |
| (0.80, 1.10) | .00000 | .01178 | .00000 | .00000 | .00351 | .00712 | .00000 | .06908 |
| (0.80, 1.20) | .00001 | .06008 | .00001 | .00000 | .02261 | .04497 | .00000 | .13780 |
| (0.90, 0.90) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00008 | .00008 |
| (0.90, 1.00) | .00002 | .00218 | .00000 | .00000 | .00052 | .00106 | .00008 | .01250 |
| (0.90, 1.10) | .00007 | .03343 | .00001 | .00000 | .01117 | .02243 | .00008 | .06908 |
| (0.90, 1.20) | .00011 | .09428 | .00004 | .00000 | .03659 | .07214 | .00008 | .13780 |
| (1.00, 0.80) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .01250 | .00000 |
| (1.00, 0.90) | .00007 | .00003 | .00000 | .00000 | .00002 | .00005 | .01250 | .00008 |
| (1.00, 1.00) | .00395 | .00885 | .00000 | .00002 | .00344 | .00698 | .01250 | .01250 |
| (1.00, 1.10) | .00956 | .05881 | .00003 | .00006 | .02316 | .04607 | .01250 | .06908 |
| (1.00, 1.20) | .01141 | .12528 | .00008 | .00008 | .05170 | .10105 | .01250 | .13780 |
| (1.10, 0.80) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .06908 | .00000 |
| (1.10, 0.90) | .00201 | .00008 | .00000 | .00001 | .00047 | .00096 | .06908 | .00008 |
| (1.10, 1.00) | .03261 | .01198 | .00001 | .00025 | .01130 | .02270 | .06908 | .01250 |
| (1.10, 1.10) | .05874 | .06739 | .00003 | .00041 | .03768 | .07425 | .06908 | .06908 |
| (1.20, 0.80) | .00006 | .00000 | .00000 | .00008 | .00002 | .00004 | .13780 | .00000 |
| (1.20, 0.90) | .01342 | .00015 | .00000 | .00008 | .00336 | .00683 | .13780 | .00008 |
| (1.20, 1.00) | .08292 | .01267 | .00001 | .00071 | .02376 | .04725 | .13780 | .01250 |

**Table 5.7**: Blocking probabilities for $C = 1500$, $(C_1, C_2) = (500, 1000)$.

| | VP | | Conditional | | CS | | CP | |
|---|---|---|---|---|---|---|---|---|
| $(\sigma_1, \sigma_2)$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ | $B_1$ | $B_2$ |
| (0.80, 1.00) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .01747 |
| (0.80, 1.10) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .06783 |
| (0.80, 1.20) | .00000 | .00009 | .00000 | .00000 | .00001 | .00002 | .00000 | .13098 |
| (0.90, 0.90) | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00000 | .00081 |
| (0.90, 1.00) | .00000 | .00009 | .00000 | .00000 | .00001 | .00002 | .00000 | .01747 |
| (0.90, 1.10) | .00003 | .00242 | .00003 | .00000 | .00033 | .00069 | .00000 | .06783 |
| (0.90, 1.20) | .00020 | .01725 | .00020 | .00000 | .00315 | .00641 | .00000 | .13098 |
| (1.00, 0.80) | .00000 | .00000 | .00000 | .00000 | .00001 | .00002 | .00892 | .00000 |
| (1.00, 0.90) | .00080 | .00031 | .00000 | .00000 | .00035 | .00072 | .00892 | .00081 |
| (1.00, 1.00) | .00422 | .00934 | .00011 | .00000 | .00343 | .00696 | .00892 | .01747 |
| (1.00, 1.10) | .00745 | .04587 | .00065 | .00000 | .01277 | .02565 | .00892 | .06783 |
| (1.00, 1.20) | .00900 | .09981 | .00157 | .00000 | .02741 | .05442 | .00892 | .13098 |
| (1.10, 0.80) | .00780 | .00000 | .00000 | .00000 | .00373 | .00758 | .07192 | .00000 |
| (1.10, 0.90) | .02749 | .00072 | .00004 | .00000 | .01388 | .02786 | .07192 | .00081 |
| (1.10, 1.00) | .05195 | .01629 | .00028 | .00000 | .02947 | .05845 | .07192 | .01747 |
| (1.10, 1.10) | .06491 | .06509 | .00074 | .00000 | .04719 | .09255 | .07192 | .06783 |
| (1.20, 0.80) | .05648 | .00000 | .00002 | .00000 | .03175 | .06288 | .14437 | .00000 |
| (1.20, 0.90) | .09105 | .00079 | .00012 | .00000 | .05036 | .09861 | .14437 | .00081 |
| (1.20, 1.00) | .12131 | .01719 | .00030 | .00001 | .06904 | .13374 | .14437 | .01747 |

**Table 5.8**: Blocking probabilities for $C = 1500$, $(C_1, C_2) = (1000, 500)$.

Tables 5.7 and 5.8 confirm that Virtual Partitioning is efficient, fair, and extremely robust with regard to traffic perturbations. The state-dependent trunk reservation in VP allows overloaded classes only to borrow the excess capacity of underloaded classes. Thus, VP protects underloaded classes, without wasting any capacity that may be used by overloaded classes.

The above characteristics are clearly reflected in the blocking performance. Because it does not leave any capacity unused, VP ensures that the total amount of blocking is at most marginally higher than in CS, and hence the link utilization is very similar. In CP, the total amount of blocking may be dramatically higher, in particular when one class is overloaded while the other class is underloaded. Unlike CP, VP makes the excess capacity of underloaded classes available to overloaded classes, thus avoiding underutilization of the link.

At the same time, since it guarantees the availability of capacity up to the nominal allocation, VP ensures that none of the classes experiences higher blocking than in CP, or at most by a negligible amount in case of a severely overloaded system. In CS, the individual blocking may be radically higher. Unlike CS, VP does not allow overloaded classes to claim any additional capacity, unless left unused by underloaded classes, thus protecting conforming class from being overwhelmed by misbehaving classes.

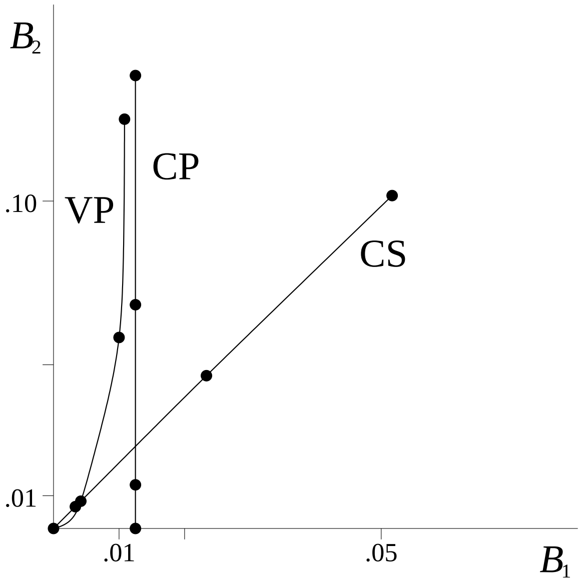


**Figure 5.1**: Blocking probabilities for $C = 1500$, $(C_1, C_2) = (500, 1000)$, $t = 10$.

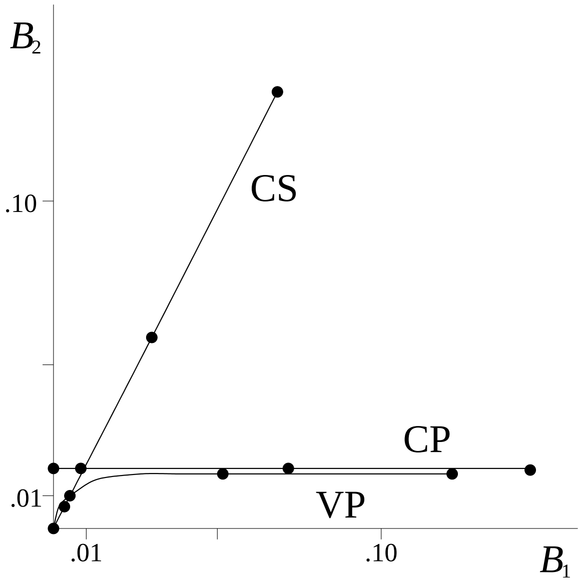


**Figure 5.2**: Blocking probabilities for $C = 1500$, $(C_1, C_2) = (1000, 500)$, $t = 10$.

The robustness properties of VP are illustrated in Figures 5.1 and 5.2 above. In Figure 5.1, the blocking probabilities are plotted for fixed $\sigma_1 = 1.00$, while $\sigma_2$ is incremented from 0.80 to 1.20 in steps of size 0.10. In Figure 5.2, the blocking probabilities are shown for fixed $\sigma_2 = 1.00$, while $\sigma_1$ is increased from 0.80 to 1.20. In a light-traffic scenario where one class is underloaded while the other is nominally loaded, VP achieves a similar high multiplexing gain as CS by allowing the nominally-loaded class to borrow excess capacity from the underloaded class, whereas CP fails to do so. In a heavy-traffic scenario where one class is overloaded while the other is nominally loaded, VP has a similar isolating effect as CP, whereas CS fails to prevent the overloaded class from taking away capacity from the nominally-loaded class. In conclusion, over the whole range, both classes experience lower blocking in VP than in CP, without suffering indiscriminately under overload conditions as in CS.

### 5.5 Revenue Optimality

We now consider the behavior of VP and the optimal call admission policy for the problem with rewards and penalties as described in the introduction. Each call of class $k$ that is carried provides a reward of $w_k$, and additionally each call of class $k$ which is blocked when the class is underloaded generates a penalty of $v_k$. Here, $w_1 = 1$, $w_2 = 4$ and $v_1 = 2$, $v_2 = 8$. Notice that the revenue $w_k$ is proportional to $b_k/\mu_k$, and that the penalty $v_k$ is also proportional to $w_k$. We take the link capacity to be $C = 150$, and set the trunk reservation level in VP to $t = 5$. Tables 5.9 and 5.10 give the mean revenue rate obtained by VP and the optimal call admission policy, which have been obtained using value iteration [20], [21], for various loadings. The results show that VP generates revenues quite close to the optimal values. In fact, the maximum difference is less than 0.68%. This is even more remarkable considering that the VP scheme is fixed in all cases, whereas the optimal admission policy is allowed to differ, depending on the particular loading scenario.

| $(\sigma_1, \sigma_2)$ | VP | Optimal | % |
|---|---|---|---|
| (0.80, 1.00) | 260.125 | 261.597 | 0.563 |
| (0.80, 1.10) | 267.719 | 269.504 | 0.663 |
| (0.80, 1.20) | 272.754 | 274.610 | 0.676 |
| (0.90, 0.90) | 255.960 | 256.556 | 0.232 |
| (0.90, 1.00) | 265.458 | 266.391 | 0.350 |
| (0.90, 1.10) | 271.695 | 272.746 | 0.385 |
| (0.90, 1.20) | 275.809 | 276.921 | 0.402 |
| (1.00, 0.80) | 250.117 | 250.253 | 0.054 |
| (1.00, 0.90) | 261.968 | 262.293 | 0.124 |
| (1.00, 1.00) | 269.819 | 270.385 | 0.209 |
| (1.00, 1.10) | 274.886 | 275.679 | 0.288 |
| (1.00, 1.20) | 278.222 | 279.166 | 0.338 |
| (1.10, 0.80) | 256.886 | 256.946 | 0.023 |
| (1.10, 0.90) | 266.691 | 266.952 | 0.098 |
| (1.10, 1.00) | 272.974 | 273.593 | 0.226 |
| (1.10, 1.10) | 277.020 | 278.046 | 0.369 |
| (1.20, 0.80) | 262.457 | 262.496 | 0.015 |
| (1.20, 0.90) | 270.218 | 270.587 | 0.137 |
| (1.20, 1.00) | 275.066 | 276.020 | 0.338 |

**Table 5.9**: Mean revenue rate for $C = 150$, $(C_1, C_2) = (50, 100)$.

| $(\sigma_1, \sigma_2)$ | VP | Optimal | % |
|---|---|---|---|
| (0.80, 1.00) | 249.765 | 250.130 | 0.146 |
| (0.80, 1.10) | 255.748 | 256.136 | 0.151 |
| (0.80, 1.20) | 260.544 | 260.901 | 0.137 |
| (0.90, 0.90) | 257.630 | 257.941 | 0.121 |
| (0.90, 1.00) | 262.606 | 262.785 | 0.068 |
| (0.90, 1.10) | 266.339 | 266.462 | 0.046 |
| (0.90, 1.20) | 269.093 | 269.194 | 0.038 |
| (1.00, 0.80) | 264.381 | 265.232 | 0.321 |
| (1.00, 0.90) | 268.702 | 269.338 | 0.236 |
| (1.00, 1.00) | 271.881 | 272.414 | 0.196 |
| (1.00, 1.10) | 274.110 | 274.716 | 0.221 |
| (1.00, 1.20) | 275.610 | 276.476 | 0.313 |
| (1.10, 0.80) | 273.047 | 274.228 | 0.431 |
| (1.10, 0.90) | 275.873 | 276.772 | 0.325 |
| (1.10, 1.00) | 277.919 | 278.834 | 0.328 |
| (1.10, 1.10) | 279.344 | 280.481 | 0.405 |
| (1.20, 0.80) | 278.338 | 279.575 | 0.443 |
| (1.20, 0.90) | 280.246 | 281.278 | 0.367 |
| (1.20, 1.00) | 281.699 | 282.784 | 0.384 |

**Table 5.10**: Mean revenue rate for $C = 150$, $(C_1, C_2) = (100, 50)$.

The detailed acceptance policies for the particular loading scenario $(\sigma_1, \sigma_2) = (1.10, 1.10)$ are shown in Figures 5.3 and 5.4 below. In these figures, a '1' indicates that only class-1 calls are admitted, a '2' indicates that only class-2 calls are admitted, an 'x' means that calls from both classes are accepted, and a '0' means that calls from both classes are rejected. Observe that the boundary to the admission regions is almost completely linear, and that the structure of the optimal admission policy closely resembles that of VP.

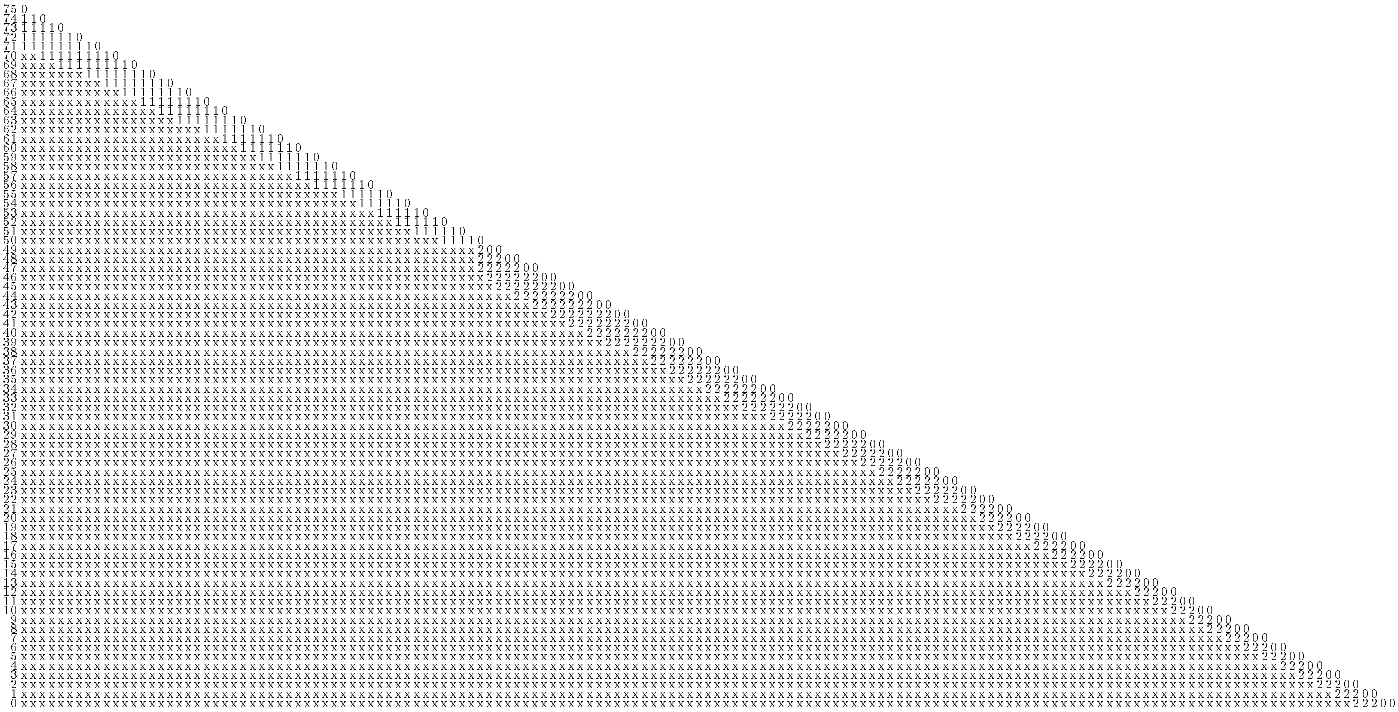

**Figure 5.3**: Optimal policy for $C = 150$, $(C_1, C_2) = (50, 100)$, $(\sigma_1, \sigma_2) = (1.10, 1.10)$.

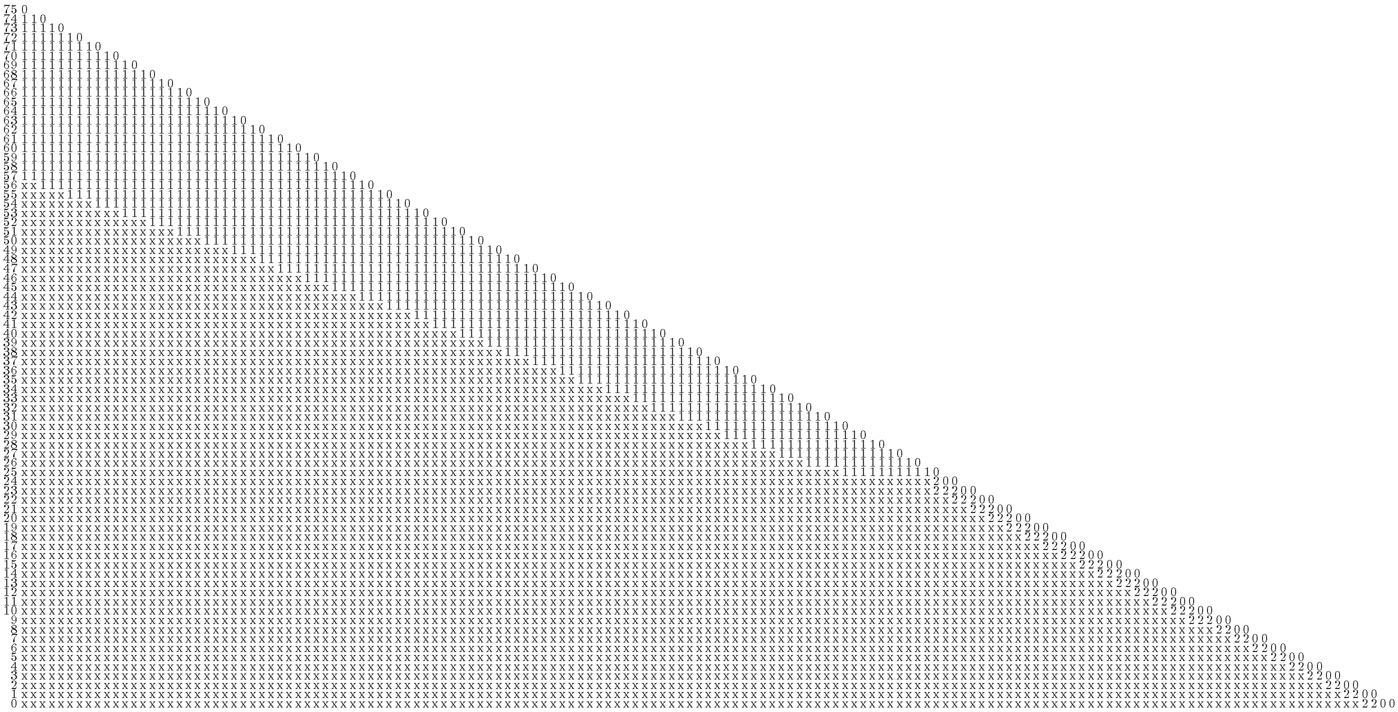

**Figure 5.4**: Optimal policy for $C = 150$, $(C_1, C_2) = (100, 50)$, $(\sigma_1, \sigma_2) = (1.10, 1.10)$.

## 6 Conclusion

We have considered Virtual Partitioning, a scheme for sharing a resource among several traffic classes in an efficient, fair, and robust manner. The scheme uses a state-dependent trunk reservation mechanism to give lower priority to misbehaving classes which violate their predetermined nominal capacity allocation, thus protecting the conforming classes.

We developed efficient computational algorithms for the case of heterogeneous traffic, and performed extensive numerical experiments to demonstrate the accuracy of the various approximations. We also examined the performance of VP and compared it to that of Complete Sharing and Complete Partitioning. We adopted a reward-penalty paradigm as a combined measure for efficiency and fairness, and showed that not only the revenue generated by VP is extremely close to the maximum achievable value, but that the structural form of the optimal policy also closely resembles that of VP. The numerical results confirm that the scheme is efficient, fair, and very robust.

## References

[1] Akinpelu, J.M. (1984). The overload performance of engineered networks with nonhierarchical and hierarchical routing. *AT&T Bell Labs Techn. J.* **63**, 1261-1281.

[2] Ash, G.R., Chen, J.S., Frey, A.E., Huang, B.D. (1991). Real-time network routing in an integrated network. In: *Teletraffic and Datatraffic in a Period of Change, Proc. ITC-13*, eds. A. Jensen, V.B. Iversen (North-Holland, Amsterdam), 187-194.

[3] Bean, N.G., Gibbens, R.J., Zachary, S. (1995). Asymptotic analysis of single resource loss systems in heavy traffic, with applications to integrated networks. *Adv. Appl. Prob.* **27**, 273-292.

[4] Borst, S.C, Mitra, D. (1997). Virtual partitioning for resource sharing by state-dependent priorities: analysis, approximations, and performance for heterogeneous traffic. In: *Teletraffic Contributions for the Information Age, Proc. ITC-15*, eds. V. Ramaswami, P.E. Wirth (North-Holland, Amsterdam), 1457-1468.

[5] Foschini, G.J., Gopinath, B., Hayes, J.F. (1981). Optimum allocation of servers to two types of competing customers. IEEE *Trans. Commun.* **29**, 1051-1055.

[6] Gopal, I.S., Stern, T.E. (1983). Optimal call blocking policies in an integrated services environment. In: *Proc. Conf. Information Sciences and Systems*, 383-388.

[7] Special Issue on Advances in the Fundamentals of Networking, Part I (1995). IEEE *J. Sel. Areas Commun.* **13**.

[8] Kaufman, J.S. (1981). Blocking in a shared resource environment. IEEE *Trans. Commun.* **29**, 1471-1481.

[9] Kelly, F.P. (1991). Loss networks. *Ann. Appl. Prob.* **1**, 319-378.

[10] Key, P.B. (1990). Optimal control and trunk reservation in loss networks. *Prob. Eng. Inf. Sc.* **4**, 203-242.

[11] Kraimeche, B., Schwartz, M. (1984). Traffic access control strategies in integrated digital networks. In: *Proc. Infocom '84*, 230-235.

[12] Kumaran, K., Mitra, D. (1998). Performance and fluid simulations of a novel shared buffer management system. To appear in: *Proc. Infocom '98*.

[13] Mitra, D., Ziedins, I. (1996). Virtual partitioning by dynamic priorities: fair and efficient resource sharing by several services. In: *Broadband Communications, Proc. 1996 Int. Zürich Seminar Digital Commun.*, ed. B. Plattner (Springer, Berlin), 173-185.

[14] Mitra, D., Reiman, M.I., Wang, J. (1997). Robust admission control for heterogeneous ATM systems with both cell and call QoS requirements. In: *Teletraffic Contributions for the Information Age, Proc. ITC-15*, eds. V. Ramaswami, P.E. Wirth (North-Holland, Amsterdam), 1421-1432.

[15] Mitra, D., Ziedins, I. (1997). Hierarchical virtual partitioning: algorithms for virtual private networking. In: *Proc. Globecom '97*, 1784-1791.

[16] Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T. (1986). *Numerical Recipes, The Art of Scientific Computing* (Cambridge University Press).

[17] Roberts, J.W. (1981). A service system with heterogeneous service requirements. In: *Performance of Data Communications Systems and Their Applications*, ed. G. Pujolle (North-Holland, Amsterdam), 423-431.

[18] Roberts, J.W. (1983). Teletraffic models for the Telecom 1 integrated services network. In: *Proc. ITC-10*, session 1.1, paper #2.

[19] Ross, K.W. (1995). *Multirate Loss Models for Broadband Telecommunication Networks* (Springer, Berlin).

[20] Ross, S.M. (1983). *Introduction to Stochastic Dynamic Programming* (Academic Press).

[21] Tijms, H.C. (1986). *Stochastic Modelling and Analysis: A Computational Approach* (Wiley).