Virtual Partitioning by Dynamic Priorities: Fair and Efficient Resource-Sharing by Several Services

Debasis Mitra[1] and Ilze Ziedins[2]

[1] AT&T Bell Laboratories, Murray Hill, New Jersey 07974, U.S.A.
[2] Dept. of Statistics, University of Auckland, Private Bag 92019, Auckland, New Zealand

Abstract. We propose a scheme for sharing an unbuffered resource, such as bandwidth or capacity, by various services. The scheme assigns a nominal capacity to each service class and implements a form of virtual partitioning by means of state-dependent priorities. That is, instead of each class of traffic having a fixed priority, as in traditional trunk reservation schemes, the priorities depend on the state of the system. An approximate method of analysis based on fixed point equations is given. Numerical results are obtained from the approximation, exact computations and simulations. The results show that the scheme is robust, fair and efficient.

1 Introduction

This paper considers the problem of sharing an unbuffered resource, such as bandwidth or capacity between several services both "fairly and efficiently". We restrict ourselves to looking at the single link and a single rate for the services.

We assume a single link, with capacity C, has K call classes arriving at it. Calls of class k, $k = 1, 2, \ldots, K$, arrive as independent Poisson streams of rate λ_k. All call classes have exponential holding times with the same mean $1/\mu$. Without loss of generality, assume that $\mu = 1$. The problem is to guarantee a given grade of service for each class of call (the "fairness" constraint), while at the same time ensuring that the resource is not underused (the "efficiency" constraint). The fairness constraint implies that some capacity should be allocated to each call class, while the efficiency constraint would suggest that some sharing of the allocated capacity should be allowed. The policy studied in this paper is based on forecasted traffic rates. Nominal allocations of capacity for each call class are precomputed in such a way that a given grade of service can be met, that is, there is an "engineered design". The key feature of our scheme for sharing the resource is the role of dynamic state-dependent priorities for each class. That is, instead of each class of traffic having a fixed priority, as in traditional trunk reservation schemes, the priorities may depend on the state of the system, and thus will vary dynamically with changing traffic patterns.

The problem of sharing resources between competing demands has been much studied. The simplest policies, and in some sense extremal policies are complete sharing (CS) and complete partitioning (CP). Complete sharing allows all services to share the resource indiscriminately. Complete partitioning divides the resource between the services, allowing each service exclusive use of its allocated capacity. Under complete sharing, one class of service may overwhelm all the others, so that the "fairness" requirement is not met. Under complete partitioning, the resource may be underused, so that the "efficiency" requirement is not met.

A scheme that falls partway between these two is one that we refer to as physical partitioning (PP). In this scheme, as in complete partitioning, the capacity is subdivided between the competing services. Calls of class k are assigned capacity C_k, where $\sum_{k=1}^{K} C_k = C$. Thus instead of having just one trunk group, one can think of there being K trunk groups, the k^{th} having capacity C_k. Each trunk group is assigned a trunk reservation parameter, and calls of each class have priority at their allocated trunk group. A call arriving at its trunk group is accepted if there is spare capacity on it. If there is no spare capacity, then the call chooses, in accordance with some discipline, another trunk group where the amount of spare capacity exceeds the trunk reservation parameter, and is carried there. If no such trunk group exists, the call is lost. The simplest discipline picks the overflow trunk group at random. By allowing state-dependence the discipline for selecting the overflow trunk group may be generalized.

Physical partitioning is an attractive scheme, in that it offers some protection to each class, while at the same time making more efficient use of the resource, because, unlike complete partitioning, it allows some sharing. However, if we consider the sample-path behaviour we see that it has some undesirable characteristics. To see this, consider the case $K = 2$. Suppose a call of class 1 arrives. Its own trunk allocation may be occupied by predominantly class 2 calls (if, for instance, there has been a sudden burst of these), and yet it may be blocked in the second trunk group. Thus even though the number of class 1 calls in the system is low, a call of that type may be rejected. On the other hand, an arrival of class 2 that finds the trunk groups in exactly the same state will be accepted, thus leading to a state where usage is even more unfairly distributed.

To overcome the undesirable behaviour that the physical partitioning scheme exhibits, we propose a scheme which we call virtual partitioning (VP) and which is implemented by means of state-dependent priorities. As with physical partitioning, the k^{th} class of call is assigned a nominal capacity C_k, where $\sum_{k=1}^{K} C_k \geq C$ and C_k is sufficiently high to ensure the desired grade of service. Let n_k denote the number of calls of class k in progress. An arriving call of class k is always accepted if $n_k < C_k$ and $\sum_{j=1}^{K} n_j < C$. If $n_k \geq C_k$, the call is accepted if $\sum_{j=1}^{K} n_j < C - r_k$, where r_k is a trunk reservation parameter chosen to protect the remaining classes against overloads of class k. Thus, depending on whether $n_k < C_k$ or $n_k \geq C_k$ the priority status of class k calls is either high or low.

In addition to overcoming the undesirable sample-path properties mentioned above, we would also expect this scheme to be very robust to changing traffic patterns, since it allows the priorities of traffic streams to vary dynamically with changing traffic patterns. We observe that in the case $K = 2$, the virtual partitioning scheme is just physical partitioning with repacking. For $K > 2$, however, that is not the case, and we would expect the blocking probabilities to decrease by virtue of the pooling of resources that occurs.

Some of the authors who have considered related problems are Foschini et al. (1981), Gopal and Stern (1983) and Kraimeche and Schwartz (1984). Foschini et al. (1981) consider only admission policies that induce product-form equilibrium distributions and show that a threshold policy is optimal. Gopal and Stern (1983) apply Markov decision theory to the analysis of threshold policies, while trying to maximise the average occupancy of a link. Kraimeche and Schwartz (1984) consider a class of restricted access policies, with the aim of minimizing blocking probabilities. The work of Ash et al. (1991) on class-of-service routing deserves particular note. Implementations aimed at balancing fairness and efficiency are given. See also the work of Mason, Dziong and Tetreault (1992). Kelly (1991) and Ross (1995) are excellent sources of information on fixed point approximations for loss networks.

There are several ways in which the problem we have outlined could be formulated. Suppose that calls of class k attract revenue w_k. Then one formulation of the problem is to maximize the revenue earned (if $w_k = 1$ for all k this is just the number of calls carried) subject to certain grade of service requirements being met.

Another formulation of the problem arose from the observation that while we would normally wish to admit as many calls as possible, we do not wish to do so if other calls are then constrained to use less than their allocated capacity and hence rejected. It is therefore natural to consider a problem formulation where some penalty is incurred when undesirable behaviour is exhibited. We add a penalty structure to the problem by assuming that a penalty v_k is paid for each call of class k that is rejected when $n_k < C_k$ (in the numerical work we set $w_k = 1$ and $v_k = 2$ for all k). We can then seek the admission policy that maximises the revenue earned per unit time. The second formulation lends itself more easily to an analysis using Markov decision processes and it is the one used in this paper.

Although in the examples discussed in this paper, we have always assumed that $\sum_{k=1}^{K} C_k = C$, there are many cases of interest where it is more natural to have a strict inequality. For instance, it may be desirable to have some capacity that is completely shared between the types of call, with no trunk reservation operating on it. Or there may be classes of traffic that have high priority and that we wish to accept under any circumstances (e.g. some kinds of overflow traffic), in which case they could have nominal capacity C. And complete sharing is a special case of virtual partitioning, obtained by setting $r_k = 0$, $(k = 1, 2, \ldots, K)$. The fixed point approximation described below can be adapted to deal with these changes.

In this paper we investigate the optimality of the suggested policy, as well as developing a fixed point approximation for the blocking probability. We give numerical examples that show that the suggested virtual partitioning policy exhibits the desired properties of fairness and efficiency, as well as being fairly robust against changes in traffic patterns.

2 Fixed Point Approximations

To study the virtual partitioning policy further, it is necessary to consider the equilibrium distribution of the process. Unlike several other policies that have been discussed in the literature, this does not induce a product-form distribution on the state-space, and finding the exact solution is a non-trivial task, even for the simplest case with $K = 2$. It was therefore natural to consider an approximation, and in particular, some form of fixed point approximation. The fixed point approximation that was developed arose from the observation that the behaviour of a single link with a virtual partitioning policy was exactly the same as that of a certain circuit switched network.

Consider a network consisting of $K+1$ links, as illustrated in Figure 1. The k^{th} link, $1 \leq k \leq K$ has capacity $C - r_k$, and the $K+1^{st}$ has capacity C. We call the $K+1^{st}$ link the α link, and the k^{th} link, β_k, for $1 \leq k \leq K$. Let n_α denote the number of calls in progress on link α, and n_k denote the number of calls in progress on link β_k. Calls of class k arrive at Poisson rate λ_k, and request a unit of capacity simultaneously from link α and link β_k. They are accepted if *either* $n_k < C_k$ and $n_\alpha < C$ *or* if $n_k \geq C_k$ and $n_\alpha < C - r_k$. Note that we are implicitly assuming here that $C - r_k > C_k$. The sequence of acceptance/rejection decisions for this network is indistinguishable from that of the single link with virtual partitioning, that is, if both start with the same numbers of type k calls in progress, and are offered the same sequence of arrivals, then both will accept and reject the same calls. The capacity of link β_k was set equal to $C - r_k$ because under virtual partitioning there can never be more than $C - r_k$ calls of class k in the system. We will refer to the characterization of the virtual partitioning policy in this form as the alpha/beta network.

For the alpha/beta network there is a natural fixed point approximation that can be constructed by assuming link independence. Let $\pi_k = (\pi_k(0), \ldots, \pi_k(C - r_k))$ be the marginal distribution for link β_k, and $\pi_\alpha = (\pi_\alpha(0), \ldots, \pi_\alpha(C))$ the marginal distribution for link α. Define $B_\alpha = \pi_\alpha(C)$, $b_\alpha^{(k)} = \sum_{i=C-r_k}^{C} \pi_\alpha(i)$, $B_k = \pi_k(C - r_k)$ and, finally, $b_k = \sum_{i=C_k}^{C-r_k} \pi_k(i)$. Then, if we assume link independence, the state-dependent transition rates on link β_k are given by

$$q_k(i, i+1) = \lambda_k[(1 - b_\alpha^{(k)}) + (b_\alpha^{(k)} - B_\alpha) I_{\{i < C_k\}}]$$

for $0 \leq i < C - r_k$ and $q_k(i, i-1) = i$ for $1 \leq i \leq C - r_k$. The transition rates on link α are given by

$$q_\alpha(i, i+1) = \sum_{k=1}^{K} \lambda_k[(1 - b_k) + b_k I_{\{i < C - r_k\}}]$$

for $0 \leq i < C$ and $q_\alpha(i, i-1) = i$ for $0 < i \leq C$. The upward rate for, link α, for instance, is given by the observation that a call of class k will always be offered

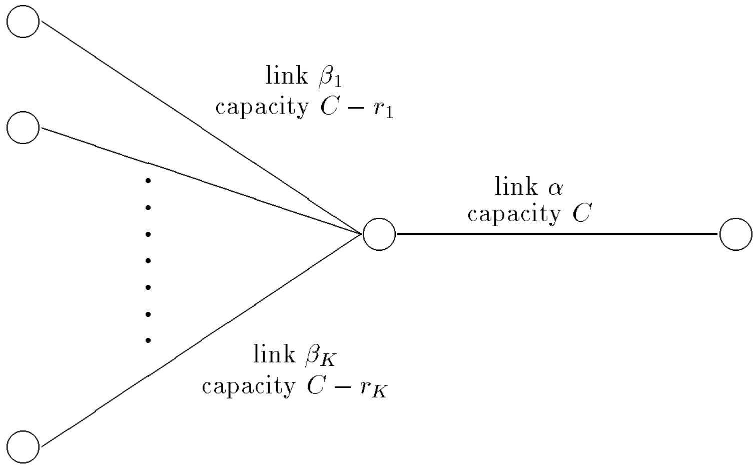


Fig. 1. The alpha/beta network

to link α if $n_k < C_k$, and this happens with probability $1 - b_k$. On the other hand, if $C_k \leq n_k$ (which happens with probability b_k), the call is offered to link α only if $n_\alpha < C - r_k$. Now sum over all possible classes k, $1 \leq k \leq K$.

Under the assumption of link independence, the marginal distributions π_k and π_α are those of truncated birth and death processes with state-spaces $\{0, 1, \ldots, C - r_k\}$ and $\{0, 1, \ldots, C\}$, respectively, and transition rates as given above. Thus

$$\pi_\alpha(i) = \pi_\alpha(0) \prod_{j=0}^{i-1} \frac{q_\alpha(j, j+1)}{q_\alpha(j+1, j)}$$

for $1 \leq i \leq C$, where $\pi_\alpha(0)$ is a normalizing constant. Similarly,

$$\pi_k(i) = \pi_k(0) \prod_{j=0}^{i-1} \frac{q_k(j, j+1)}{q_k(j+1, j)}$$

for $1 \leq i \leq C - r_k$, where $\pi_k(0)$ is a normalizing constant.

Using this approximation the probability that a call of class k is lost is given by

$$\begin{aligned} L_k &= Pr(n_k < C_k \text{ and } n_\alpha = C) + Pr(n_k \geq C_k \text{ and } n_\alpha \geq C - r_k) \\ &= Pr(n_k < C_k) Pr(n_\alpha = C) + Pr(n_k \geq C_k) Pr(n_\alpha \geq C - r_k) \\ &= (1 - b_k) B_\alpha + b_k b_\alpha^{(k)} \ . \end{aligned}$$

The earnings per unit time in the virtual partitioning scheme is given by

$$E = \sum_{k=1}^{K} \lambda_k w_k - E_L \tag{1}$$

where

$$\begin{aligned} E_L = \sum_{k=1}^{K} \lambda_k \ \ & [(w_k + v_k)\ \Pr(n_k < C_k \text{ and } n_\alpha = C) \\ & + w_k\ \Pr(n_k \geq C_k \text{ and } n_\alpha \geq C - r_k)] \end{aligned} \tag{2}$$

The first term in (1) is the potential revenue and the second term, E_L, represents lost earnings. Lost earnings has two components: penalties incurred from blockage when the number of calls of the blocked call's class is below allocated capacity; and potential revenue from calls blocked because either there is no capacity or available capacity is reserved to forestall future penalties.

In the fixed point approximation,

$$E = \sum_{k=1}^{K} \lambda_k [w_k(1 - b_k b_\alpha^{(k)}) - (w_k + v_k)((1 - b_k) B_\alpha)] \tag{3}$$

To implement the approximation, it is necessary to use an iterative procedure, starting with some estimate of the marginal distributions π_k, $1 \leq k \leq K$, and π_α; using these to calculate the transition rates for the single link distributions given above; recalculating estimates for the marginal distributions using these transition rates; and continuing until the procedure converges.

3 Numerical Results

The aim of virtual partitioning is to protect traffic streams against one another. We examined the extent to which this was achieved by looking at two examples, one with two classes of traffic, and the other with four. In both examples the arrival rate for class 1 traffic was increased, while holding the arrival rate for the remaining classes constant. We also examined the adequacy of the fixed point approximation developed in Section 2 for these two examples.

The tools developed for evaluating performance were:

(i) the exact numerical solution of the equilibrium equations for a Markov chain using Gauss-Seidel iteration,
(ii) a discrete-event simulator, and
(iii) a fixed-point solver.

The first example had $K = 2$, $C = 120$, $C_1 = 80$, and $C_2 = 40$. The second had $K = 4$, $C = 240$, $C_k = 60$, $1 \leq k \leq 4$. We set $w_k = 1$ and $v_k = 2$ for all k.

3.1 Accuracy of Fixed Point Approximations

Gauss-Seidel iteration was used to find the equilibrium distribution for the case $K = 2$. In the standard example, even with a favourably chosen initial distribution, the number of iterations necessary to obtain convergence to 10^{-7} in the L_∞ norm was 100 or 200. In higher dimensions, the number of iterations necessary would be even greater, and this underscores the importance of developing an adequate approximation for the blocking probabilities.

The numerical results are presented in Tables 1 and 2, and indicate that the approximation suggested above performs reasonably well in many cases of interest, and aids in assessment of qualitative behaviour. When compared with the exact distribution, the percentage error of the blocking increased as the trunk reservation increased. The error is around 5-6%, and occasionally as high as 15%.

The approximation was also compared with simulation results for the second example with $K = 4$. Initially it was assumed that $\lambda_k = 55$, $r_k = r$, $1 \le k \le 4$, and then λ_1 was increased in 10% increments, up to a level of 82.5. Again the percentage error increased with the trunk reservation parameter. With $r = 2$ the percentage error was no greater than 6.8%, and mostly around 5% or less. With $r = 5$, the percentage error in blocking for calls of class 1 was less than 7%, whereas for the other classes it could be as high as 15%.

Table 1a. Accuracy of fixed point, compared with exact calculations $K = 2$, $C = 120$, $C_1 = 80$, $C_2 = 40$, $\lambda_1 = 73$, $\lambda_2 = 37$

	Blocking class 1		Blocking class 2		Lost revenue
(r_1, r_2)	Exact	Fixed point	Exact	Fixed point	Exact
(0,0)	.02781		.02781		5.93
(1,1)	.02882	.02728	.03069	.03070	4.58
(2,2)	.02971	.02738	.03553	.03373	4.13
(2,3)	.02873	.02570	.04064	.03900	3.97
(3,2)	.03185	.02956	.03489	.03152	4.22
(3,3)	.03096	.02792	.04001	.03688	4.06
(3,4)	.03041	.02648	.04433	.04203	4.04
(4,3)	.03279	.03016	.03970	.03488	4.18
(4,4)	.03232	.02877	.04402	.04014	4.15
(5,5)	.03368	.02983	.04773	.04347	4.31
(2,4)	.02814	.02440	.04497	.04390	3.94

Table 1b. Accuracy of fixed point, compared with exact calculations $K = 2$, $C = 120$, $C_1 = 80$, $C_2 = 40$, $\lambda_1 = 80.3$, $\lambda_2 = 37.0$

	Blocking class 1		Blocking class 2		Lost revenue
(r_1, r_2)	Exact	Fixed point	Exact	Fixed point	Exact
(0,0)	.0568	.0568	.0568	.0568	12.03
(2,2)	.0677	.0628	.0513	.0540	8.32
(2,3)	.0667	.0606	.0563	.0606	8.02
(3,3)	.0707	.0658	.0549	.0552	8.17
(3,4)	.0701	.0640	.0589	.0612	8.06
(4,4)	.0735	.0687	.0583	.0571	8.28
(5,5)	.0760	.0715	.0613	.0594	8.48
(2,4)	.0661	.0591	.0602	.0659	7.92
(2,5)	.0657	.0580	.0633	.0702	7.91
(3,5)	.0698	.0627	.0621	.0660	8.05

Table 1c. Accuracy of fixed point, compared with exact calculations $K = 2$, $C = 120$, $C_1 = 80$, $C_2 = 40$, $\lambda_1 = 87.6$, $\lambda_2 = 37.0$

	Blocking class 1		Blocking class 2		Lost revenue
(r_1, r_2)	Exact	Fixed point	Exact	Fixed point	Exact
(0,0)	.0924	.0924	.0924	.0924	19.52
(2,2)	.1154	.1089	.0619	.0693	13.64
(3,3)	.1198	.1143	.0641	.0673	13.40
(4,4)	.1236	.1190	.0666	.0675	13.54
(2,4)	.1140	.1053	.0691	.0808	13.10
(2,5)	.1137	.1044	.0714	.0843	13.05
(2,6)	.1135	.1039	.0732	.0870	13.05
(1,5)	.1056	.0961	.0784	.0949	13.65

Table 2. Accuracy of fixed point, comparison with simulation results (with 95% confidence intervals) $K = 4$, $C_k = 60$, $1 \leq k \leq 4$, $\lambda_2 = \lambda_3 = \lambda_4 = 55.0$, $r_1 = r_2 = r_3 = r_4 = 2$

λ_1	Blocking class 1		Blocking classes 2,3,4	
	Simulation	Fixed point	Simulation	Fixed point
55.0	$.0128 \pm .0004$	.0124	$.0128 \pm .0004$	.0124
60.5	$.0266 \pm .0007$	.0273	$.0191 \pm .0004$	.0182
66.0	$.0475 \pm .0010$	.0491	$.0256 \pm .0005$	.0241
71.5	$.0715 \pm .0009$	.0752	$.0318 \pm .0005$	.0300
77.0	$.0993 \pm .0015$	.1030	$.0388 \pm .0006$	.0362
82.5	$.1269 \pm .0019$	.1320	$.0451 \pm .0006$	.0425

3.2 Fairness and Robustness

Figures 2 and 3 give changes in the blocking probabilities as class 1 traffic increases its arrival rate. The data is drawn from the tables given in the section above. So for the case $K = 2$, the blocking probabilities are exact, whereas for the example with $K = 4$, the blocking probabilities are based on simulation results. For the latter example, therefore, a confidence interval for the blocking probabilities is indicated on the graph.

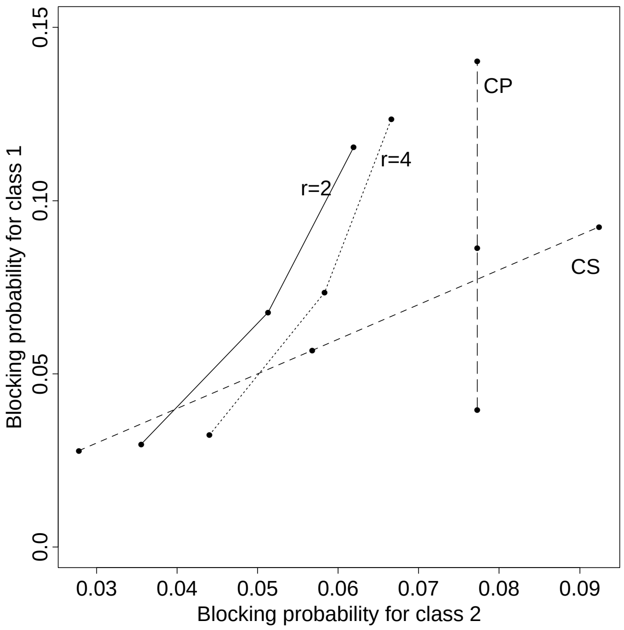


Fig. 2. Blocking probabilities for class 1 vs. class 2 as class 1 arrival rate increases. $K = 2$; $C_1 = 80$, $C_2 = 40$; $\lambda_2 = 37.0$; $\lambda_1 = 73.0,\ 80.3,\ 87.6$

Four cases are compared – complete sharing (CS), complete partitioning (CP), and virtual partitioning with two choices of trunk reservation parameter (the reservation parameter is the same for all traffic classes).

For both examples we see that the blocking probability for all classes increases as the arrival rate for class 1 increases, except, of course, for complete partitioning, where it remains unchanged. However, virtual partitioning with $r > 0$ does have the desired property of substantially reducing the effect of this increase on those classes with constant arrival rate.

We now examine the change in lost revenue as λ_1 increases for the example with $K = 4$. In Table 3 we give, for each value of λ_1, the choice of r, r^*, which

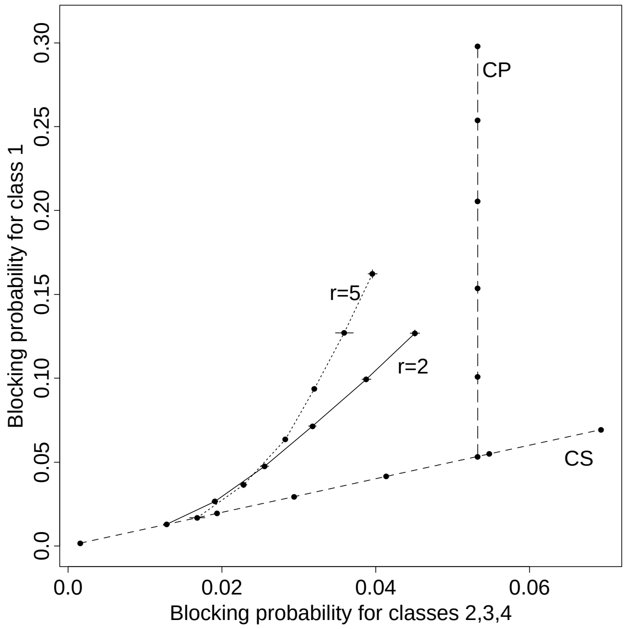


Fig. 3. Blocking probabilities for class 1 vs. others as class 1 arrival rate increases. $K = 4$; $C_1 = C_2 = C_3 = C_4 = 60$; $\lambda_k = 55.0$, $k = 2, 3, 4$; $\lambda_1 = 55.0,\ 60.5,\ 66.0,\ 71.5,\ 77.0,\ 82.5$

minimizes the lost revenue per unit time (r^* in each case is assumed to be the same for all classes), and the lost revenue when $r = r^*$. We also give, for each arrival rate, the lost revenue per unit time when $r = 2$ and $r = 5$.

Table 3. Lost revenue $K = 4$; $C_k = 60$, $1 \le k \le 4$; $\lambda_2 = \lambda_3 = \lambda_4 = 55.0$

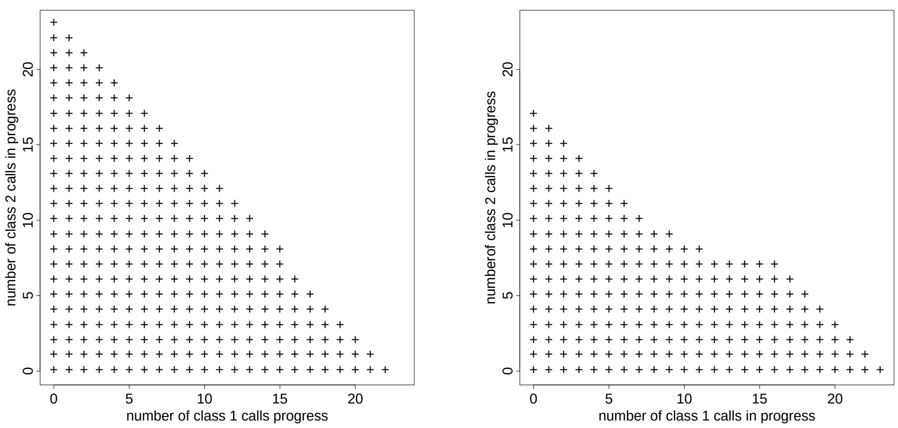
λ_1	55.0	60.5	66.0	71.5	77.0	82.5
r^*	5	6	7	7	7	7
Lost revenue						
$r = r^*$	4.453	18.22	38.18	60.97	85.53	111.3
$r = 2$	5.063	21.34	44.99	71.68	99.70	128.4
$r = 5$	4.453	18.30	38.54	61.71	86.46	112.2

We see that lost revenue is relatively insensitive to the choice of trunk reservation parameter under the engineered loading. However, it appears to be important not to have too low a trunk reservation parameter, particularly in heavily overloaded conditions.

3.3 Optimality

When a penalty structure is attached to the problem, it is possible to find the optimal policy maximising the expected reward per unit time. It can be found numerically by using value iteration (see e.g. Ross (1983) or Tijms(1986)). Since the state space is finite, the optimal policy will be stationary, and can thus be simply described as an acceptance region for each class of service.

The example discussed in this section has $K = 2$, $C = 24$, $C_1 = 16$, $C_2 = 8$, $\lambda_1 = 20$, $\lambda_2 = 6$, $w_1 = w_2 = 1$ and $v_1 = v_2 = 2$. The policy that maximizes the expected return per unit time is given in Figure 4.



Optimal acceptance region for class 1. Optimal acceptance region for class 2.

Fig. 4. Policy that maximizes the expected return per unit time. $K = 2$, $C = 24$, $C_1 = 16$, $C_2 = 8$, $\lambda_1 = 20$, $\lambda_2 = 6$, $w_1 = w_2 = 1$ and $v_1 = v_2 = 2$

We note that the optimal policy, even though it is not exactly virtual partitioning, is nevertheless of a very similar form. The policy is monotone, in that if a call of class i is rejected in state (i, j), (where $i + j < C$), it is also rejected in states $(i + 1, j)$ and $(i, j + 1)$. Furthermore, if a call of class 1 is rejected in state (i, j), then it is rejected in state $(i + 1, j - 1)$; and similarly, if a call of class 2 is rejected in state (i, j) it is also rejected in state $(i - 1, j + 1)$. All the examples that were investigated had monotone optimal policies.

In Table 4 we present numerical results on the performances of the optimal policy and the VP scheme in the example considered above (see Fig. 4). We observe that there is little to be gained over VP schemes by using a strictly optimal policy. The difference in lost revenue between the optimal policy and

Table 4. Blocking probabilities and lost revenue in the virtual partitioning scheme and the optimal policy for the example in Fig. 4.

		Blocking probability		Lost revenue
		class 1	class 2	
VP	$(r_1, r_2) = (1, 4)$	.2311	.1335	5.7992
	$= (2, 4)$	.2559	.1171	5.9164
	$= (1, 5)$	.2310	.1357	5.7989
	$= (2, 5)$	.2558	.1221	5.9163
	$= (1, 6)$	.2309	.1369	5.8011
	$= (2, 6)$	.2557	.1234	5.9189
Optimal policy		.2311	.1340	5.7984

VP with $(r_1, r_2) = (1, 5)$ is .0005, or about 0.01%. We also observe that, as before, the VP scheme is robust in that performance is close to optimal for variations in the trunk reservation parameters.

In contrast to the relatively high Poisson arrival rates, λ_1 and λ_2, and the correspondingly high blocking probabilities in the example above, we now present results for an example in which $\lambda_1 = 12$ and $\lambda_2 = 5$, while C, C_1 and C_2 are as before. In this case our results show that the optimal policy is exactly VP with $r_1 = 1$ and $r_2 = 2$. Also, in this case,

$$\begin{aligned}\text{Blocking probability} &= 0.0310 \ \text{ for class 1 calls}\\ &= 0.0318 \ \text{ for class 2 calls}\\ \text{Lost revenue} \qquad\quad &= 0.5924.\end{aligned}$$

In fact, in all the examples we have studied in which the arrival rates and the blocking probabilities are low, the optimal policy is VP.

The above investigations on robustness have been for fixed traffic patterns. To investigate further the adequacy of the VP scheme when traffic patterns vary dynamically we also considered the model described above, but with Markov modulated Poisson arrival streams, rather than fixed arrival rates. We found that a virtual partitioning policy was very close to optimal (or even optimal) in many situations, particularly in the two extreme cases where arrival rates modulated either very slowly or very rapidly. This corresponds to the modulation varying on a different time-scale than the arrivals and departures for the system. We do not give details of this here.

4 Conclusions

We have proposed a robust, fair and efficient scheme for sharing an unbuffered resource, such as bandwidth or capacity, by various services. The scheme is a form of virtual partitioning which is implemented by state-dependent priorities. The numerical results show that the scheme is robust, fair and efficient.

Future work will address services with multiple rates. The motivation for the extension comes from multiple services offered on ATM networks. The notion of effective bandwidth translates to a specific rate or bandwidth requirement for each traffic source type. The central ideas presented here are amenable to generalizations in the multi-rate framework.

Acknowledgements

Ilze Ziedins would like to thank A.T.& T. Bell Labs for support while working on this project.

References

Ash, G.R., Chen, J.S., Frey, A.E., Huang, B.D.: Real-time network routing in an integrated network. Proceedings of the International Teletraffic Congress (ITC-13), Copenhagen (1991).

Foschini, G.J., Gopinath, B., Hayes, J.F.: Optimum allocation of servers to two types of competing customers. IEEE Trans. Comm. **COM-29** (1981) 1051–1055.

Gopal, I.S., Stern, T.E.: Optimal call blocking policies in an integrated services environment. Proceedings of the Conference on Information Sciences and Systems, Johns Hopkins University (1983) 383–388.

Kelly, F.P.: Loss networks. Ann. Appl. Prob. **1** (1991) 319–378.

Kraimeche, B., Schwartz, M.: Traffic access control strategies in integrated digital networks. Proceedings of Infocom '84 (1984) 230–235.

Mason, L., Dziong, Z., Tetreault, N.: Fair-efficient call admission control policies for broadband networks. Proceedings of Globecom '92 (1992).

Ross, K.W.: Multirate Loss Models for Broadband Telecommunications Networks. Springer (1995).

Ross, S.M.: Introduction to Stochastic Dynamic Programming. Academic Press (1983).

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