Hierarchical Virtual Partitioning: Algorithms for Virtual Private Networking

Debasis Mitra
Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974
U.S.A.

Ilze Ziedins
Dept. of Statistics
University of Auckland
Private Bag 92019
Auckland, New Zealand

ABSTRACT: Hierarchical Virtual Partitioning facilitates sharing of a resource, such as a link, by multiple customers, each with multiple service classes. Calls of each service class have distinctive bandwidth requirements, arrival rates and mean holding times. The quality of service is in terms of call blocking probabilities. The algorithms allow the desired mix of fairness, robustness and multiplexing efficiency at each of two levels, namely, customers and the various services of each customer. The scheme relies on nominal allocations of capacities to each customer and service, and priorities in the call admission process, which are implemented by dynamic trunk reservations. An approximate method of analysis based on fixed point equations is given. A reward/penalty paradigm is devised to reflect the issues in Virtual Private Networking and the performance of our algorithm compared to the optimal solution. The results show that the scheme is fair, efficient and robust.

I. Introduction

Hierarchical Virtual Partitioning (HVP) is a concept with algorithms for providing services to customers on a shared physical infrastructure. The algorithms are able to handle several such customers, each having several service offerings. The services have distinct bandwidth requirements and statistical properties, such as call arrival rates and mean call holding times, as well as quality of service requirements on the call blocking probabilities. To fix ideas, interpret a call to be either a flow or a switched virtual circuit, although other interpretations exist. The goal of the HVP algorithms is to provide customers with the quality of service that approximates as closely as desired that experienced on dedicated, private networks, while, in fact, the aggregate traffic is multiplexed on a shared network. Of course, the object of sharing is to realize multiplexing gains. Thus, the crux of the design of HVP is to strike the appropriate balance between contrary requirements. On one hand there is fairness and robustness, which stand for guaranteed performance to each well-behaved customer, independent of the behaviour of all other customers. Complete satisfaction in this respect can come only from isolating the resources for each customer, as in the Complete Partitioning (CP) paradigm of resource sharing [8; 9; 14; 23]. On the other hand there is efficiency, better resource utilization and multiplexing gain, which come from sharing resources. The well-known paradigm of Complete Sharing (CS) represents the extreme form of unrestricted sharing. Virtual Partitioning (VP) was advanced by Mitra and Ziedins [14] as an intermediate paradigm for disciplined sharing. The notion of HVP is more closely attuned to the notion of Virtual Private Networking (VPN) in which the notion of multiple customers, each with multiple service classes, is embedded. A key feature of VPN from the point of view of the resource manager is the difference in the mode of resource sharing that exists at the two levels, one at the level of customers and the other at the level of the various services of each customer. The degree of isolation, and the consequent hardness of guarantees on quality of service, between customers is naturally greater than between the various services of any customer. HVP gives a framework, which allows for the desired mix, at both levels, between fairness and robustness on one hand and efficiency on the other.

This work addresses issues at the connection, or call, layer. Issues of packet and cell level behavior, including multiplexing and related quality of service issues, are encapsulated in the capacity requirement of individual calls. The latter may correspond to the "effective bandwidth" of calls; see, for instance [1] for various notions of effective bandwidth, which in some cases is measurement-based.

We will refer to customers as superclasses and the customer's services as subclasses. A single link with bandwidth or capacity C, is shared between J superclasses, indexed by j, $1 \leq j \leq J$, the j^{th} superclass having K_j subclasses. Let class (j,k) denote subclass k in superclass j, $1 \leq k \leq K_j$, $1 \leq j \leq J$. The j^{th} superclass is allocated a nominal capacity C_j ($C_j \geq 0$), where $\sum_j C_j \geq C$. The $(j,k)^{th}$ class has nominal capacity C_{jk} ($C_{jk} \geq 0$) allocated to it, with $\sum_{k=1}^{K_j} C_{jk} \geq C_j$.

Let n_{jk} be the capacity currently occupied by calls of subclass (j,k). Let $n_j = \sum_k n_{jk}$ be the capacity currently occupied by calls of superclass j. Finally, let $n = \sum_j n_j$ be the total capacity currently occupied. The HVP control policy studied in this paper is to admit a new call of class (j,k) if

$$n \leq C - r_{jk} I\{n_{jk} > C_{jk} - d_{jk}\} - R_j I\{n_j > C_j - d_{jk}\} - d_{jk},$$

where d_{jk} is the capacity required by a call of subclass (j,k). The motivation is as follows. The nominal capacities $\{C_{jk}\}$ and $\{C_j\}$ are allocated at the time of design. At each time instant superclasses and subclasses using capacity in excess of their allocations are considered to be "overloaded", in contrast to the remaining "underloaded" classes. Arriving calls are accorded different priorities. Specifically, calls of overloaded classes are admitted only if there is unutilized reserved capacity, as in classical trunk reservation schemes [2], [11]. In contrast to the classical scheme, the trunk reservation here is dynamic and state-dependent. That is, the reservation level operating against a call of class (j,k) depends upon whether, if the call is admitted, only the subclass is overloaded, only the superclass is overloaded or both are overloaded. The corresponding reservation parameters are r_{jk}, R_j and $r_{jk}+R_j$. The role of these reservation parameters is to protect the underloaded classes. Commensurate with the need in VPN to provide greater protection to superclasses, as discussed earlier, $r_{jk} < R_k$ typically. This notion of a hierarchy of priorities to which calls of subclass (j,k) are subject, depending on the state n_{jk}, n_j and n is at the core of HVP.

The notion of an engineered design also underlies HVP. The nominal allocations $\{C_{jk}\}$ and $\{C_j\}$ are derived from this design on the basis of expected loads and quality of service requirements, in terms of call blocking probabilities. The multiplexing gain allows $\sum_k C_{jk}/C_j \geq 1$ and $\sum_j C_j/C \geq 1$, where the extent of overallocation is a topic for design to which the techniques of this paper may be applied. Note that "best-effort" quality of service for subclass (j,k) suggests that the nominal capacity $C_{jk}=0$.

The classical paradigms of CS and CP extended to the hierarchical framework may be obtained as special cases of HVP. Complete Sharing is obtained by setting the trunk reservation parameters $\{r_{jk}\}$ and $\{R_j\}$ to zero. The scheme in which there is CP between superclasses and CS between the subclasses of each superclass is obtained by making the reservation parameters $\{R_j\}$ large, while setting the reservation parameters $\{r_{jk}\}$ to zero. Finally, CP at all levels is obtained by making all the reservation parameters large.

For evaluating the performance of HVP we make the following assumptions. Calls of subclass (j,k) arrive at the link as a Poisson process of rate λ_{jk}, independently of all other arrival streams, and request capacity d_{jk}. The call holding times are assumed to be independent and exponentially distributed with mean μ_{jk} for calls of subclass (j,k).

An important component of this paper is concerned with a reward/penalty paradigm, which has been devised to reflect the essential tradeoffs of VPN. The reward aspect, which is fairly standard, assigns for each call of subclass (j,k) that is carried a reward that may depend on (j,k), say w_{jk}. The non-standard aspect assigns a penalty for rejecting a call when either the subclass or the superclass of the call is underloaded. Specifically, let U and $\mathcal{U}$ denote sets associated with underloaded subclasses and superclasses:

$$\begin{aligned} U &= \{(j,k) | n_{jk} \leq C_{jk} - d_{jk}\} \\ \mathcal{U} &= \{(j,k) | n_j \leq C_j - d_{jk}\}. \end{aligned}$$

The complementary sets are identified by the superscript C. We propose the following assignment of varying penalties depending on the underload/overload status of the class of the blocked call.

Underload/Overload states	penalty for rejecting class (j,k) call
$(j,k) \in U$ and $(j,k) \in \mathcal{U}$	$V_j + v_{jk}$
$(j,k) \in U^C$ and $(j,k) \in \mathcal{U}$	V_j
$(j,k) \in U$ and $(j,k) \in \mathcal{U}^{\mathcal{C}}$	v_{jk}
$(j,k) \in U^C$ and $(j,k) \in \mathcal{U}^{\mathcal{C}}$	0

To correspond to the previously stated goals of VPN, the penalty incurred from blocking calls belonging to an underloaded superclass will be greater than that incurred if just the subclass is underloaded. That is, $V_j > v_{jk}$.

Given such a reward/penalty structure, the goal is to obtain the call admission policy which maximizes the revenue earned per unit time. The techniques of Markov Decision Theory [22], [23] may be used to obtain such a policy. Both the process of obtaining the optimal policy and its detailed structure are complicated. HVP is advanced as a policy which, while not optimal in general, yields performance that closely approximates the optimal policy's performance, besides having advantages of simplicity and robustness. We give results which support this claim.

Robustness is clearly a crucial aspect of effective resource sharing. We have in mind not only the ability to respond to statistical fluctuations in call arrivals, which are inevitable even when call arrival rates correspond exactly to expectations, but also adaptability in cases where the rates are different from those assumed in the engineered design. Note that the reward/penalty paradigm does not address this higher level issue of robustness. In its performance, HVP automatically behaves like CS when the overall traffic is light. If, on the other hand, there is overloading by a subset of superclasses and subclasses, then HVP behaves like CP at the appropriate level by effectively isolating the rogue classes. We give results which reflect this feature.

Another aspect of robustness of HVP concerns the sensitivity to the reservation parameters $\{r_{jk}\}$ and $\{R_j\}$. Our results show that HVP is quite robust in this respect. This is not surprising, given prior results on the relative flatness of performance with respect to variations in the reservation parameters in classical trunk reservation [18].

The problem of fair and efficient resource sharing has a long history. Foschini, Gopinath and Hayes [7] consider admission control policies which induce product-form equilibrium distributions, and show that a threshold policy is optimal. Gopal and Stern [8] use Markov Decision Theory to determine threshold policies that maximize the link utilization. Kraimeche and Schwartz [12] consider a class of restricted-access policies which aim to reduce blocking probabilities. The recent important work on Link Sharing by Floyd and Jacobson [6] has motivations in common with this work, except that the framework here is that of calls and loss models, whereas in [6] it is queueing models. The work of Ash et. al. [3] on class-of-routing is also aimed at balancing fairness and efficiency. Borst and Mitra [5] develop computational algorithms for analyzing heterogeneous traffic classes in VP. A key assumption in our analytic approximation to HVP is a certain link independence, which is common to fixed point approximations for loss networks. Excellent sources of information on fixed-point approximations are Kelly [10] and Ross [21]. Recent applications of VP to admission control and buffer management are reported in [16] and [13], respectively. Additional information related to the present work may be found in [15].

The paper is organized as follows. In section 2 we develop fixed point approximations for HVP. In section 3 we investigate the accuracy of the fixed point approximation in the single rate case. Section 4 investigates the properties of fairness and robustness. In section 5 we comment briefly on the efficiency of HVP when compared to the optimal call control policy for the reward/penalty paradigm.

II. Fixed point approximations

In this section we discuss fixed point approximations for HVP on a single link, which extends the technique in Mitra and Ziedins [14] for VP. The latter does not incorporate the idea of superclasses (equivalently, the nonhierarchical version can be thought of as having just a single superclass). The approximation arises from the observation that the behaviour of a single link under HVP can be emulated by a multilink circuit-switched network.

Consider a network consisting of $1 + J + \sum_j K_j$ links, as illustrated in Figure 1. Links are of three types – α, β or γ. There is a single α-link, with capacity C – we denote it simply by α. *All* arriving calls request capacity from this link. Let n be the occupancy on this link – it is exactly the total occupancy of the single link. The network has J β-links attached to the α link - denote the j^{th} β-link by β_j, $1 \leq j \leq J$. Link β_j has capacity $C - R_j$, $1 \leq j \leq J$. All arriving calls in superclass j request capacity from link β_j. We let n_j denote its occupancy – it corresponds to the occupancy of calls from superclass j on the single link. Finally, link β_j has K_j γ-links attached to it. The $(j,k)^{th}$ γ-link has capacity $C - R_j - r_{jk}$, and we denote the link by γ_{jk}. All calls from subclass (j,k) request capacity on link γ_{jk}. Let n_{jk} denote the occupancy of γ_{jk} – this corresponds to the occupancy of calls from subclass (j,k) on the single link.

Calls from subclass (j,k) arrive at this network as a Poisson process of rate λ_{jk}, independently of all other arrival streams, and request capacity d_{jk} simultaneously from links α, β_j and γ_{jk}. They are accepted if any of the following conditions holds:-

1. $n_{jk} \leq C_{jk} - d_{jk}$, $n_j \leq C_j - d_{jk}$, $n \leq C - d_{jk}$
2. $n_{jk} \leq C_{jk} - d_{jk}$, $n_j > C_j - d_{jk}$, $n \leq C - R_j - d_{jk}$
3. $n_{jk} > C_{jk} - d_{jk}$, $n_j \leq C_j - d_{jk}$, $n \leq C - r_{jk} - d_{jk}$
4. $n_{jk} > C_{jk} - d_{jk}$, $n_j > C_j - d_{jk}$, $n \leq C - R_j - r_{jk} - d_{jk}$.

The sequence of acceptance/rejection decisions for this network is therefore indistinguishable from that of the single link with HVP, that is, if both start with the same numbers of class (j,k) calls in progress, $1 \leq k \leq K_j, 1 \leq j \leq J$, and are offered the same sequence of arrivals, then both will accept and reject the same calls. The capacity of link β_j is set equal to $C - R_j$ because under HVP there can never be more than $C - R_j$ calls of superclass j in the system. Similarly, there can never be more than $C - R_j - r_{jk}$ calls of subclass (j,k) in the system under the HVP policy. We refer to the characterization of the HVP policy in this form as the alpha/beta/gamma network.

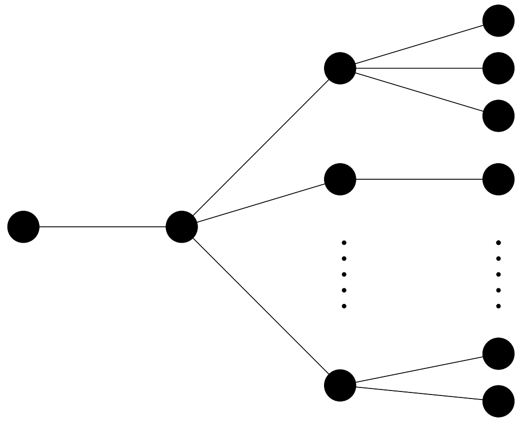

Fig. 1. The alpha/beta/gamma network

The simplest approximation assumes that all links block independently of one another. Under this assumption, the equilibrium distribution is found for each link separately and the probability that a call of class (j,k) is lost is then given by

$$\begin{aligned} L_{j,k} &= Pr(n_{jk} < C_{jk}, n_j < C_j, n = C) \\ &+ Pr(n_{jk} < C_{jk}, n_j \geq C_j, n \geq C - R_j) \\ &+ Pr(n_{jk} \geq C_{jk}, n_j < C_j, n \geq C - r_{jk}) \\ &+ Pr(n_{jk} \geq C_{jk}, n_j \geq C_j, n \geq C - R_j - r_{jk}). \end{aligned}$$

Finding the equilibrium distribution for the individual links is nontrivial in the presence of multi-rate traffic. Various approximations have been suggested for single links with multi-rate traffic, some of which can be modified to apply here. Kaufman [9] and Roberts [19] develop an exact recursion for the multi-rate case when there are no admission controls. Roberts [20] and Bean et. al. [4] give approximations for links with trunk reservation. Borst and Mitra [5] compare these approaches for Virtual Partitioning, as well as considering two-dimensional approximations.

We outline here the approximation proposed in Bean et. al. [4]. Let $\pi_\alpha = (\pi_\alpha(0), \ldots, \pi_\alpha(C))$ be the marginal distribution for link α, $\pi_j = (\pi_j(0), \ldots, \pi_j(C - R_j))$ the marginal distribution for link β_j, and $\pi_{j,k} = (\pi_{j,k}(0), \ldots, \pi_{j,k}(C - R_j - r_{jk}))$ the marginal distribution for link $\gamma_{j,k}$. With L_{jk} as defined above, note that the expected number of class (j,k) calls in progress in equilibrium is then given by

$$\bar{n}_{jk} = \frac{\lambda_{jk}}{\mu_{jk}}(1 - L_{jk})$$

and the expected amount of capacity occupied by class (j,k) calls is $\bar{n}_{jk} d_{jk}$.

Consider first the $\gamma_{j,k}$ link. This has only one type of call passing through it, and the state-space is $S_{j,k} = \{0, d_{jk}, 2d_{jk}, \ldots, d_{jk}[C - R_j - r_{jk}/d_{jk}]\}$. Only the transition rates $q_{j,k}(n_{jk}, n_{jk} \pm d_{jk})$ are non-zero and obtained as in [14] by invoking link independence. The equilibrium distribution $\pi_{j,k}(id_{j,k})$ is that of a birth and death process on $S_{j,k}$ and is explicit.

The analysis for link β_j is more complicated and we give details for it. The state-space is $S_j = \{0, 1, \ldots, C - R_j\}$. For $1 \le k \le K_j$, let

$$\begin{aligned} &A_j(n_j; j,k) = I_{\{n_j \le C_j - d_{jk}\}} \\ &\cdot [P(n_{jk} \le C_{jk} - d_{jk})P(n \le C - d_{jk}) \\ &\quad + P(n_{jk} > C_{jk} - d_{jk})P(n \le C - r_{jk} - d_{jk})] \\ &\quad + I_{\{n_j > C_j - d_{jk}\}} \\ &\cdot [P(n_{jk} \le C_{jk} - d_{jk})P(n \le C - R_j - d_{jk}) \\ &\quad + P(n_{jk} > C_{jk} - d_{jk})P(n \le C - R_j - r_{jk} - d_{jk})]. \end{aligned}$$

This is an approximation to the probability that a call of subclass (j,k) is accepted on link j when it is in state n_j. Then the approximation to the equilibrium distribution for the β_j link, as given in [4], satisfies the following balance equations for $0 \le n_j \le C - R_j$:

$$\begin{aligned} &\pi_j(n_j) \sum_k \left(\lambda_{jk} A_j(n_j; j,k) + n_j \frac{\bar{n}_{jk}}{\sum_k \bar{n}_{jk} d_{jk}} \mu_{jk} \right) \\ &= \sum_k \pi_j(n_j - d_{jk}) \lambda_{jk} A_j(n_j - d_{jk}; j,k) \\ &\quad + \sum_k \pi_j(n_j + d_{jk})(n_j + d_{jk}) \frac{\bar{n}_{jk}}{\sum_k \bar{n}_{jk} d_{jk}} \mu_{jk}, \end{aligned}$$

where we assume $\pi_j(n_j + d_{jk}) = 0$ if $n_j + d_{jk} > C - R_j$ and $\pi_j(n_j - d_{jk}) = 0$ if $n_j - d_{jk} < 0$.

As with link β_j, the probability that a call of class (j,k) is accepted if the alpha link is in state n can be approximated using link independence. The approximation to the equilibrium distribution for the α link is obtained by solving the balance equations.

To implement the approximation, it is necessary to use an iterative procedure, starting with some estimate of the marginal distributions π_α, π_j, $1 \le j \le J$, and $\pi_{j,k}$, $1 \le j \le J$, $1 \le k \le K_j$; 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.

Other quantities of interest can now be calculated, using the equilibrium distributions obtained from the fixed point calculations. The probability that a call of superclass j is blocked is given by

$$L_j = \sum_{k=1}^{K_j} \lambda_{jk} L_{j,k} \left[\sum_{k=1}^{K_j} \lambda_{jk} \right]^{-1}$$

where the blocking probabilities for the subclasses are weighted by their arrival rates.

The earnings per unit time in HVP are given by

$$E = \sum_{j=1}^{J} \sum_{k=1}^{K_j} \lambda_{jk} w_{jk} - E_L \tag{1}$$

where

$$\begin{aligned} E_L = &\sum_{j=1}^{J} \sum_{k=1}^{K_j} \lambda_{jk} \\ &\cdot [(w_{jk} + v_{jk} + V_j) Pr(n_{jk} < C_{jk}, n_j < C_j, n = C) \\ &\quad + (w_{jk} + v_{jk}) Pr(n_{jk} < C_{jk}, n_j \ge C_j, n \ge C - R_j) \\ &\quad + (w_{jk} + V_j) Pr(n_{jk} \ge C_{jk}, n_j < C_j, n \ge C - r_{jk}) \\ &\quad + w_{jk} Pr(n_{jk} \ge C_{jk}, n_j \ge C_j, n \ge C - R_j - r_{jk})]. \end{aligned}$$

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 (whether subclass or superclass) 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. Again, the probabilities can be approximated by assuming link independence, and then using the appropriate marginal distributions.

Of considerable importance is the probability that a call of class (j,k) is lost, given that it is underloaded, which is given by

$$\begin{aligned} &L_{j,k|n_{jk} < C_{jk}} \\ &= Pr(n_j < C_j) Pr(n = C) + Pr(n_j \ge C_j) Pr(n \ge C - R_j) \end{aligned}$$

if we assume link independence. A similar approximation can be made for the probability that a call in superclass j is lost, given that that superclass is underloaded.

III. Accuracy of fixed point approximation to HVP for single-rate traffic

In this section we investigate the accuracy of the fixed point approximation when the link only carries single-rate traffic, i.e. $d_{jk} = 1$ for all j,k. If in addition we have $\mu_{jk} = \mu_j$ for all k then the occupancy of each link in the alpha/beta/gamma network, considered in isolation from all the others, is just a birth and death process and the global equations given for the equilibrium distribution in the previous section simplify considerably. We have, for instance,

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

for $1 \le i \le C - R_j$, where $\pi_j(0)$ is a normalizing constant, and the transition rates are those given in the preceding section.

We examine the accuracy of the fixed point approximation by studying the example below. Variants of this example will be used in succeeding sections.

Example 1. The example has two superclasses ($J = 2$), with the first superclass having 3 subclasses ($K_1 = 3$) and the second superclass 2 subclasses ($K_2 = 2$). The capacity of the link is $C = 120$, and the nominal capacities for the two superclasses are $C_1 = 80$ and $C_2 = 40$. The subclasses within each superclass all have the same properties – this is so that qualitative properties of the algorithm will be more clearly visible. Within the first superclass the arrival rate for each subclass is $\lambda_{1k} = 25.0$ and nominal capacity is $C_{1k} = 30$. The subclasses within the second superclass have arrival rates $\lambda_{2k} = 17.0$ and nominal capacities $C_{2k} = 22$. We assume that all calls are worth the same, and set $w_{jk} = 1.0$ for all j and k. We also assume that $v_{jk} = 2.0$ is the same for all subclasses, and that $V_j = 4.0$ is the same for all the superclasses.

Since this system is too large to permit the calculation of exact values for the blocking probabilities, the fixed point approximation is compared with simulation results, shown below with 95% confidence intervals.

Tables 1–3 give the overall blocking probability, the blocking probability for calls from superclass 1, and the blocking probability for class (1,1) calls conditional on that subclass being underloaded. In each case we give simulation results with 95% confidence intervals and below them, the figure given by the fixed point approximation.

We see that the fixed point approximation consistently underestimates the blocking probabilities when reservation parameters are included. In Table 1 the error increases with the size of the reservation parameters and is about 14% when $R_j = r_{jk} = 3$ for all j and k. The same is true for Table 2, with an error of around 15% when $R_j = r_{jk} = 3$. In both tables the error increases markedly with R_j and less so as the r_{jk} increase. Table 3, giving the conditional blocking probabilities displays somewhat different behaviour. Here the fixed point approximation consistently *overestimates* the conditional blocking probabilities. Rather surprisingly, the error *decreases* as the R_j increase, while increasing in the r_{jk}. In all three tables, the fixed point approximation gives a good idea of the qualitative behaviour of the blocking probabilities.

In Table 4 we examine the lost revenue for a range of reservation parameters. The lost revenue is given for $0 \le R_j \le 4$, $0 \le r_{jk} \le 3$. The penalty structure we have chosen gives the optimal choice of parameters as $R_j = 2$, $r_{jk} = 1$. Note

	r_{jk}			
	0	1	2	3
$R_j = 0$	.02463±.0003 .0246	.0255±.0004 .0250	.0268±.0004 .0258	.0284±.0004 .0269
$R_j = 1$	.0264±.0003 .0252	.0274±.0003 .0257	.0287±.0004 .0265	.0303±.0004 .0275
$R_j = 2$	.0287±.0004 .0262	.0294±.0004 .0267	.0310±.0004 .0275	.0325±.0004 .0285
$R_j = 3$	.0311±.0004 .0276	.0320±.0004 .0281	.0332±.0004 .0288	.0346±.0004 .0298

Table 1. Overall blocking probability. Simulation results and fixed point approximation.

	r_{jk}			
	0	1	2	3
$R_j = 0$	.0246±.0003 .0246	.0262±.0004 .02531	.0279±.0004 .02631	.0296±.0004 .0276
$R_j = 1$	.0292±.0004 .0259	.0301±.0004 .0266	.0313±.0004 .0275	.0328±.0004 .0287
$R_j = 2$	.0319±.0004 .0275	.0328±.0004 .0281	.0340±.0004 .0291	.0353±.0004 .0301
$R_j = 3$	.0344±.0004 .0293	.0353±.0004 .0299	.0364±.0004 .0307	.0376±.0004 .0318

Table 2. $L_1 = L_{1,1}$, the blocking probability for class 1 calls. Simulation results and fixed point approximation.

	r_{jk}			
	0	1	2	3
$R_j = 0$	.0193±.0004 .0246	.0146±.0002 .0213	.0119±.0002 .0187	.0100±.0003 .0166
$R_j = 1$	.0230±.0003 .0259	.0184±.0003 .0227	.0153±.0002 .0199	.0132±.0002 .0176
$R_j = 2$	.0252±.0004 .0275	.0211±.0004 .0247	.0182±.0003 .0219	.0160±.0004 .0194
$R_j = 3$	.0274±.0005 .0293	.0236±.0004 .0268	.0211±.0004 .0243	.0190±.0003 .0218

Table 3. $L_{1,1|n_{11}<C_{11}}$, the conditional blocking probability for subclass (1,1) calls. Simulation results and fixed point approximation.

that there is a range of parameters that give very similar performance. It is a feature of trunk reservation policies in general that they are robust to changes in arrival patterns (we shall see this further below), and a complementary result is that the response surface is relatively flat, as observed here.

	r_{jk}			
	0	1	2	3
$R_j = 0$	10.70±.14	9.37±.16	8.89±.16	8.89±.14
$R_j = 1$	8.43±.14	7.97±.13	8.01±.12	8.30±.11
$R_j = 2$	8.04±.14	7.80±.12	8.04±.09	8.47±.09
$R_j = 3$	8.31±.12	8.15±.12	8.41±.11	8.92±.11
$R_j = 4$	8.80±.12	8.70±.13	8.95±.13	9.46±.13
$R_j = 5$	9.36±.12	9.30±.13	9.57±.14	10.05±.13

Table 4. Lost revenue for various choices of reservation parameter. Simulation results.

IV. Fairness and robustness

In this section we examine the effects of increasing arrival rates on blocking probabilities under HVP. We consider two examples, the first a single-rate example, and the second a multi-rate example. The results displayed in this section are estimates from simulations, with accompanying 95% confidence intervals.

Example 2. The first example is the single-rate example taken from the previous section, with $R_j = 2$, $r_{jk} = 1$, $j = 1, 2$, $1 \le k \le K_j$. We begin by taking increasing λ_{11}, with increments of 2.5, that is, 10%. All other arrival rates remain fixed with $\lambda_{1k} = 25.0$, $k = 1, 2$ and $\lambda_{2k} = 17.0$ for $k = 1, 2$. Table 5 gives blocking probabilities and conditional blocking probabilities for various call types. We

λ_{11}	L_1	$L_{1,1}$	$L_{1,1\|n_{11}<30}$	$L_{1,2}$
25.0	.0328±.0004	.0330±.0007	.0211±.0004	.0326±.0003
27.5	.0448±.0005	.0492±.0008	.0266±.0007	.0421±.0003
30.0	.0585±.0007	.0683±.0008	.0314±.0010	.0524±.0005
32.5	.0734±.0009	.0898±.0009	.0363±.0014	.0626±.0008
35.0	.0893±.0008	.1124±.0008	.0413±.0010	.0731±.0010
37.5	.1060±.0009	.1360±.0010	.0463±.0016	.0832±.0013
40.0	.1231±.0009	.1596±.0011	.0518±.0015	.0936±.0011

λ_{11}	$L_{1,2\|n_{12}<30}$	L_2	$L_{2,1\|n_{21}<22}$	E_L
25.0	.0213±.0009	.0228±.0003	.0124±.0005	7.80±.12
27.5	.0301±.0006	.0262±.0003	.0147±.0007	10.36±.12
30.0	.0402±.0005	.0292±.0007	.0171±.0009	13.13±.16
32.5	.0504±.0008	.0317±.0010	.0191±.0010	16.00±.24
35.0	.0613±.0012	.0340±.0009	.0207±.0008	19.02±.23
37.5	.0722±.0015	.0359±.0007	.0223±.0012	22.17±.28
40.0	.0836±.0012	.0375±.0010	.0234±.0014	25.43±.29

Table 5. Example 2. Effect of increasing λ_{11} for $R_j = 2$, $r_{jk} = 1$.

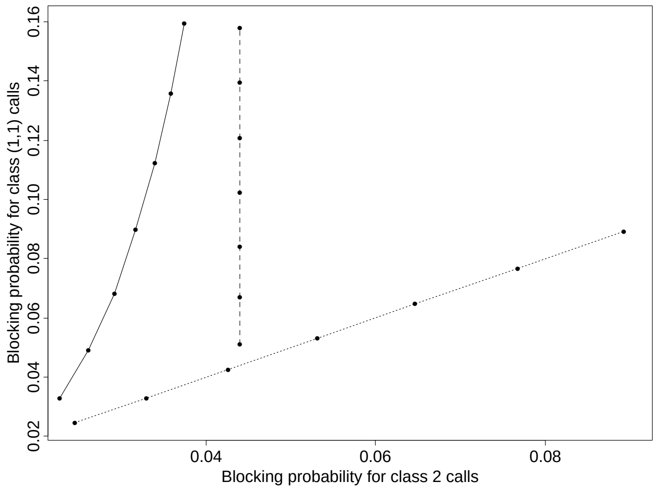

Fig. 2. Blocking probabilities as λ_{11} increases.
——— VP, ... CS, - - - - - semi-CS.

note that, by symmetry, $L_{1,2} = L_{1,3}$ for all choices of λ_{11}, and $L_{2,1} = L_{2,2} = L_2$.

In this example HVP demonstrates the desired properties of fairness and robustness. As λ_{11} increases, the blocking probability for calls of class (1,1) increases from .033 to .123 ± .001. We note that the blocking probability conditional on the nominal allocation not being exceeded is much lower, and increases from .021 to .052 ± .002. The remaining subclasses within superclass 1 are protected to some extent from this increase – their total blocking probability increases from .033 to .094 ± .001, while the conditional blocking probability increases from .021 to .084 ± .001. The subclasses within superclass 2 are even more effectively protected against the increasing arrival rate of class (1,1) – their total blocking probability increases from .023 to .038 ± .001, while the conditional blocking probabilities increase from .012±.001 to .023±.001. These blocking probabilities should be compared with those obtained under two commonly used policies. Under Complete Sharing (CS), when all streams have access to all the capacity, blocking for all streams rises from .024 to .089. We see that HVP thus provides considerable protection against the rogue arrival stream. Another commonly used policy is Complete Partitioning (CP). For this example, since $\sum C_{jk} > C$, we consider a hybrid policy of Complete Sharing within superclasses (semi-CS). That is, superclass 1 has capacity 80 allocated to it, which is not shared with superclass 2, and similarly, superclass 2 has capacity 40 allocated to it, which is not shared with superclass 1. Within each superclass, all subclasses have access to all the capacity. In this case, the blocking probability for superclass 2 calls is .044, while that of superclass 1 calls increases from .051 to .158. It is interesting that these blocking probabilities are higher than the blocking probabilities under HVP for *all* classes of call. Figure 2 plots the blocking probability of class (1,1) calls against class 2 calls for the three policies.

An important point is that, for greater robustness, it may be desirable to choose reservation parameters that are larger than the optimal choice for revenue maximization at the engineered arrival rates. We illustrate this in Table 6, which has $R_j = r_{jk} = 2$ for all j and k, but otherwise has parameters equal to those in Table 5. The main features of this table, when compared with Table 5, are striking. Increasing the r_{jk} by 1 lowers the conditional blocking probabilities for all classes. This decrease is most marked at the higher arrival rates, for classes other than $(1,1)$, when it is as high as 25%. And while the overall blocking for superclass 1 has increased by about 3% for all arrival rates, this increase is targeted to the rogue subclass. When $\lambda_{11} = 40$, the blocking probability for subclass (1,1) is about 16% greater than in Table 5, whereas the blocking probability for the other subclasses in superclass 1 has *decreased* by about 16%. Meanwhile the increase in lost revenue never exceeds 3%. Thus a small increase in the reservation parameters can give a significant increase in robustness.

λ_{11}	L_1	$L_{1,1}$	$L_{1,1\|n_{11}<30}$	$L_{1,2}$
25.0	.0340±.0004	.0342±.0006	.0182±.0003	.0338±.0002
27.5	.0462±.0005	.0537±.0008	.0237±.0008	.0418±.0003
30.0	.0602±.0007	.0770±.0007	.0286±.0010	.0502±.0005
32.5	.0755±.0009	.1030±.0008	.0339±.0010	.0578±.0007
35.0	.0918±.0009	.1301±.0010	.0393±.0010	.0651±.0006
37.5	.1088±.0010	.1580±.0011	.0451±.0011	.0718±.0009
40.0	.1263±.0008	.1856±.0013	.0520±.0019	.0787±.0009

λ_{11}	$L_{1,2\|n_{12}<30}$	L_2	$L_{2,1\|n_{21}<22}$	E_L
25.0	.0183±.0010	.0245±.0004	.0106±.0004	8.04±.09
27.5	.0255±.0007	.0280±.0005	.0128±.0006	10.64±.15
30.0	.0333±.0004	.0309±.0006	.0146±.0008	13.44±.17
32.5	.0406±.0005	.0335±.0009	.0162±.0008	16.38±.23
35.0	.0478±.0007	.0357±.0008	.0175±.0008	19.44±.23
37.5	.0551±.0010	.0375±.0007	.0187±.0008	22.62±.29
40.0	.0624±.0011	.0390±.0010	.0195±.0011	25.90±.29

Table 6. Example 2. Effect of increasing λ_{11} for $R_j = r_{jk} = 2$.

Example 3. The second example is a multi-rate example related to the previous one. We have $J = 2$, $K_1 = 3$, $K_2 = 2$. But now we set $d_{jk} = k$, $j = 1, 2$, $1 \le k \le K_j$.

λ_{11}	L_1	$L_{1,1}$	$L_{1,1\|n_{11}<30}$	$L_{1,3}$	$L_{1,3\|n_{13}<30}$
25.0	.0620	.0296	.0170	.0956	.0470
27.5	.1034	.0569	.0297	.1512	.0872
30.0	.1496	.0918	.0447	.2087	.1330
32.5	.1961	.1304	.0608	.2632	.1811
35.0	.2404	.1700	.0770	.3127	.2272
37.5	.2817	.2090	.0931	.3567	.2692

λ_{11}	L_2	$L_{2,1}$	$L_{2,1\|n_{21}<22}$	$L_{2,2}$	$L_{2,2\|n_{22}<22}$	E_L
25.0	.0441	.0287	.0142	.0596	.0221	14.10
27.5	.0553	.0357	.0188	.0749	.0302	23.48
30.0	.0638	.0408	.0221	.0867	.0388	34.54
32.5	.0704	.0444	.0245	.0963	.0452	46.70
35.0	.0754	.0475	.0266	.1032	.0512	59.46
37.5	.0795	.0498	.0283	.1091	.0571	72.56

Table 7. Example 3. Increasing the call arrival rate in a multi-rate example.

Initially we suppose that the departure rate is proportional to the bandwidth, with $\mu_{jk} = d_{jk}$, and that the arrival rates are, just as before $\lambda_{1k} = 25.0$, $k = 1, 2, 3$ and $\lambda_{2k} = 17.0$, $k = 1, 2$. We also keep the same rewards and penalties that we had in the previous example, with $w_{jk} = 1.0$ for all j and k, $v_{jk} = 2.0$ for all j and k, and $V_j = 4.0$ for $j = 1, 2$. The choice of bandwidths and departure rates means that it is reasonable to assume each call is worth the same amount. It is then also easier to compare performance with the single-rate example we have already given. We set the reservation parameters to be $R_1 = 2$, $R_2 = 3$, $r_{1k} = 2$, $1 \le k \le 3$ and $r_{2k} = 3$, $k = 1, 2$. These parameters were chosen by first of all finding the optimal policy with R_j constant for $j = 1, 2$ and r_{jk} constant for all j and k, (it set $R_j = 2$ and $r_{jk} = 2$) and then examining policies in its neighbourhood. It is interesting to note that the improvement gained by the latter step is not great - it amounts to about 2%. This again is due to the robustness of generalized trunk reservation policies. For this example we will increase the arrival rate for all subclasses within superclass in increments of 2.5. This is exactly what was done in the second half of the previous example. The simulation results can be found in Table 7. The 95% confidence intervals for the blocking probabilities in this table were all less than .004 in width, and those for E_L were all narrower than 0.7.

Again, the conclusions from this example are very similar to those of the previous example. One feature of interest here is that with this choice of trunk reservation parameter, those call classes with higher bandwidth requirements fare worse, in terms of blocking probabilities. If the aim were to equalize blocking probabilities then the reservation parameters would need to be changed.

V. Optimality

In this section we examine the form of the optimal policy maximizing the expected revenue per unit time under the penalty structure that we outlined in section 1. We will compare it with HVP. The optimal policy can be found by formulating the problem as a Markov Decision Process and then numerically by using value iteration (see e.g. Ross [22] or Tijms [23]). Since the state space is finite, the optimal policy is stationary, and can therefore be simply described as an acceptance region for each class of call. In order to clarify the exposition we give a quite small and simple example which exhibits features that are commonly found.

Example 4. The example discussed in this section has $C = 21$, $J = 2$, $K_1 = 1$ and $K_2 = 2$, with $\lambda_{11} = \lambda_{21} = \lambda_{22} = 7.0$ and $C_{11} = C_{21} = C_{22} = 7$, $C_1 = 7$, $C_2 = 14$. That is, there are two superclasses, the first of which only has one subclass, and the second two subclasses. We assume $d_{jk} = 1$ for all j and k, so we are only considering single-rate traffic. The cost and penalty structure is kept simple with $V_1 = 2.0, v_{11} = 0.0, V_2 = 1.0, v_{21} = v_{22} = 1.0$.

We note first that with this choice of parameters, the optimal policy for calls of classes $(2, 1)$ and $(2, 2)$ will be the same. The optimal policy computed by value iteration is simply described. Calls of class $(1, 1)$ are accepted either if $n_1 \le 6$ and $n < C$ *or* if $n < C - 3 = 18$. Calls of class $(2, 1)$ are accepted if $n_{21} < 7$ and $n < C$ *or* if $n_2 < 14$ and $n < C$ *or* sometimes if $n_{21} \ge 7$ and $n_2 \ge 14$, in which case a variable trunk reservation parameter is applied. In Figure 6 we illustrate the acceptance region when $n_1 = 0$. We note

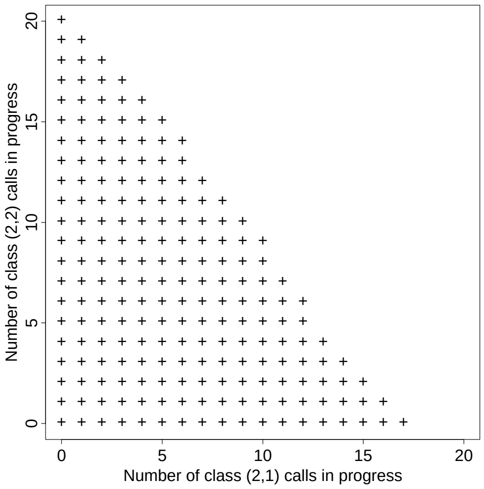

Fig. 3. Optimal acceptance region for class (2,1) when $n_1 = 0$.

that in the region where $n_{21} \ge 7$ and $n_2 \ge 14$ the trunk reservation parameter applied against class (2,1) increases as n_{21} increases and as n_2 increases - that is, it is optimal to use trunk reservation parameters that increase with the degree of overloading that subclass $(2, 1)$ is experiencing. Thus the optimal policy is not far removed from a HVP policy. We found that particularly at low arrival rates, the optimal policy is sometimes exactly HVP.

The expected lost revenue under the optimal policy is computed to be 4.49. It is instructive to compare the return from the optimal policy with the closest HVP policy. Such HVP policies will have $R_1 + r_{11} = 3$ – since there is only one subclass in superclass 1, any choice of non-negative R_1 and r_{11} such that $R_1 + r_{11} = 3$ will induce the same acceptance region. In addition we have $r_{21} = r_{22} = 0$. Finally

R_2	1	2	3
E_L	4.790±.031	4.937±.025	5.096 ± .024

Table 8. Example 4. Expected lost revenue.

R_2 could be equal to either 1, 2 or 3. Table 8 gives estimates of the expected lost revenue, E_L, for the three choices of R_2. All three choices of R_2 give values of E_L that are close to that of the optimal policy. Again, this is a consequence of the relatively flat response surface in the region of the optimal policy. Choosing $R_1 = 1$ gives the lowest lost revenue, and the increase over the optimal policy is 6.6%, which corresponds to a decrease in the expected revenue earned per unit time of only 1.8%.

VI. Conclusions

We have proposed a scheme for sharing a resource by multiple customers, each having multiple service classes, which are characterized by distinctive bandwidth requirement, call arrival rate and call holding times. The scheme, HVP, implements a hierarchy of state-dependent priorities. The numerical results show that the scheme is robust, fair and efficient.

Future work will extend the application of the concepts of HVP from a single link to a network. A key step is the calculation of the nominal capacities on each link of the network. This is a problem in network design, which needs to be addressed separately. For work in this direction, see [17]. The merger of the resource-sharing ideas of HVP with the output of network design has the potential of yielding quite attractive VPN implementations.

Acknowledgements

Ilze Ziedins would like to thank A.T.& T. Bell Labs and Lucent Technologies for support while working on this project. Some of the writing up for this project was completed while she was visiting the University of Adelaide with the support of an ARC Grant.

References

[1] Special Issue on Advances in the Fundamentals of Networking. Part I. *IEEE J. Selected Areas in Communications* **13**. Aug. 1995.

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

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

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

[5] Borst, S. and 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, ed. V. Ramaswami and P. E. Wirth, Elsevier, 1457–1468.

[6] Floyd, S. and Jacobson, V. (1995) Link-sharing and resource management models for packet networks. *IEEE/ACM Trans. Networking* **3**(4), 365-386.

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

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

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

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

[11] Key, P. (1990) Optimal control and trunk reservation in loss networks. *Prob. in the Engineering and Informational Sciences* **4**, 203-242.

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

[13] Kumaran, K. and Mitra, D. (1997) Performance and fluid simulations of a novel shared buffer management system. Submitted to *INFOCOM 98*.

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

[15] Mitra, D. and Ziedins, I. (1997) Hierarchical virtual partitioning: Algorithms for virtual private networking. *Bell Labs Tech. J.* **2**(2), 68–81.

[16] Mitra, D., Reiman, M. I. and 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, ed. V. Ramaswami and P. E. Wirth, Elsevier, 1421–1432.

[17] Mitra, D., Morrison, J.A. and Ramakrishnan, K.G. (1996) ATM network design and optimization: a multirate loss network framework. *IEEE/ACM Trans. Networking* **4**(4), 531–543.

[18] Reiman, M. (1989) Asymptotically optimal trunk reservation for large trunk groups. *Proc. 28th Conference on Decision and Control, IEEE, Tampas*, 2536-2541.

[19] 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.

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

[21] Ross, K. (1995) *Multirate Loss Models for Broadband Telecommunications Networks.* Springer.

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

[23] Tijms, H.C. (1986) *Stochastic modelling and analysis: a computational approach.* Wiley.