VPN DESIGNER: A Tool for design of Multiservice Virtual Private Networks

Debasis Mitra, John A. Morrison, K. G. Ramakrishnan
Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974
{mitra, johnmorrison, kgr}@lucent.com

ABSTRACT

VPN DESIGNER is a software tool for the design of intranets or virtual private networks (VPNs) on a service provider's infrastructure. The design allocates bandwidth on each link of the infrastructure to the VPNs such that, when the traffic of a customer is optimally routed over its VPN, a weighted aggregate measure of carried bandwidth over the service provider's infrastructure is maximized. The network is modelled as a multirate loss network, with effective bandwidth encapsulating all packet-level behavior. Scalability of the design process, i.e., the ability to handle OC3 and higher rates, is an important contribution, and this is achieved by the systematic use of asymptotic techniques. VPN DESIGNER is written in C++ and runs on any UNIX platform.

1. INTRODUCTION

VPN DESIGNER is a software tool for the design of intranets or virtual private networks (VPNs), which have alocated bandwidth on the links of a service provider's infrastructure network. The objective of the design is to allocate bandwidth such that, when the traffic of a customer is optimally routed over its VPN, a weighted aggregate measure of carried bandwidth over the service provider's network is maximized. The aforementioned problem is a joint resource allocation and traffic routing problem.

VPN DESIGNER utilizes other major tools, notably TALISMAN [MMR96-2] which optimizes routes for given traffic data and resources. In fact, in VPN DESIGNER there is an independent TALISMAN process for each VPN, and a master process that arbitrates the bandwidth allocation among the VPNs.

Let $C_\ell^{(\Omega)}$ denote the bandwidth or capacity allocated to VPN Ω on link ℓ of the infrastructure network. Then $\sum_\Omega C_\ell^{(\Omega)} = C_\ell$, where C_ℓ is the total given capacity of link ℓ. The design variables are the quantities $\{C_\ell^{(\Omega)}\}$, the proportion of the offered traffic of each stream and VPN that is admitted to the network and the proportions that are offered to the various routes in the given admissible set of routes. The objective is to maximize network revenue W, where $W = \Sigma W^{(\Omega)}$, and $W^{(\Omega)}$ is an aggregated weighted measure of carried bandwidth on VPN Ω.

The network is modelled as a loss network, with the smallest traffic entities being flows and connections, which we refer to as calls. Calls have Poisson arrivals and random generally distributed, independent holding times and each call requires constant (effective) bandwidth on all links in its route. If the bandwidth is not available on any link, the call is lost. The notion of effective bandwidth [1,2] is used to encapsulate all packet and cell level behavior. The multiservice loss network analysis is based on fixed point equations and implied costs [3-7].

VPN DESIGNER incorporates several mathematical and algorithmic innovations:

(i) The systematic incorporation of new asymptotic theory, the Refined Uniform Asymptotic Approximations (RUAA) [8], which itself is an advance over the earlier Uniform Asymptotic Approximations [9] and gives very accurate link loss probabilities and very good sensitivities. Scaleability in design, i.e., the ability to handle OC3 and higher rates are due to this feature.

(ii) The introduction of the notion of linearized capacity costs, which allows the network-wide problem of capacity reallocations to be decomposed into per-link problems. Moreover the latter is given closed form solutions.

(iii) A flexible software which supports parallelization with significant speed-up in the computation time. VPN DESIGNER has plug-in - plug-out modules with adaptability and rapid evolutions.

For reasons of space we have had to omit many interesting mathematical and algorithmic results. Additional results are in [MMR98].

2. MODEL AND PROBLEM FORMLATION

2.1 Model

The infrastructure is a network with N nodes and L unidirectional links, with link ℓ having given capacity C_ℓ. There are S types of service, where service s required bandwidth $d_{s\ell}$ on link ℓ. It is assumed that C_ℓ and $d_{s\ell}$ are positive integers.

The subnetworks, of which there are $\overline{\Omega}$, are indexed by Ω. The capacity allocated to subnetwork Ω on link ℓ is denoted by $C_\ell^{(\Omega)}$, with $\Sigma_{\Omega=1}^{\overline{\Omega}} C_\ell^{(\Omega)} = C_\ell$. Let σ denote a source-destination pair of nodes and let (s, σ) denote a stream of calls of service type s between source-destination σ. These quantities which are specific to subnetwork Ω are identified by the superscript (Ω). The set of admissible routes for stream $(s, \sigma)^{(\Omega)}$ is denoted by $R^{(\Omega)}(s,\sigma)$, and assumed to be given a priori and fixed. Typically route sets will depend on service type. For instance, delay sensitive traffic, like voice, will typically be carried over short routes over fewer hops in contrast to, say, data for which the routes may be much longer.

The call arrivals for stream $(s, \sigma)^{(\Omega)}$ are Poisson, with mean rate $\Lambda_{s\sigma}^{(\Omega)}$. The outcome of an independent Bernoulli trial determines if any arriving call of stream $(s,\sigma)^{(\Omega)}$ is offered to the subnetwork Ω, which is a form of admission control, and, if it is, to which route $r \in R^{(\Omega)}(s,\sigma)$ it is offered. If an arriving call in stream $(s, \sigma)^{(\Omega)}$ is offered to a route r, it may be still be blocked and lost if there is insufficient bandwidth available on any link of the route. If the call is accepted, the bandwidth on each link is simultaneously held for the duration of the call. The holding period of a call in stream $(s, \sigma)^{(\Omega)}$ is generally distributed with mean $1/\mu_{s\sigma}^{(\Omega)}$, and is independent of earlier arrival and holding times.

The call arrivals of stream $(s, \sigma)^{(\Omega)}$ for route r are Poisson, will mean rate $\lambda_{sr}^{(\Omega)}$, where

$$\sum_{r\in R^{(\Omega)}(s,\sigma)} \lambda_{sr}^{(\Omega)} \le \Lambda_{sr}^{(\Omega)} \qquad (2.1)$$

The corresponding traffic intensity is $\rho_{sr}^{(\Omega)} = \lambda_{sr}^{(\Omega)} / \mu_{s\sigma}^{(\Omega)}$, and it follows that

$$\sum_{r\in R^{(\Omega)}(s,\sigma)} \rho_{sr}^{(\Omega)} \le \overline{\rho}_{sr}^{(\Omega)}, \qquad (2.2)$$

where $\overline{\rho}_{s\sigma}^{(\Omega)} = \Lambda_{s\sigma}^{(\Omega)} / \mu_{s\sigma}^{(\Omega)}$. The probability

$$p_{s\sigma}^{(\Omega)} \triangleq \frac{1}{\Lambda_{s\sigma}^{(\Omega)}} \sum_{r\in R^{(\Omega)}(s,\sigma)} \lambda_{sr}^{(\Omega)} \qquad (2.3)$$

is a parameter of the admission control, i.e., exogenous calls of stream $(s, \sigma)^{(\Omega)}$ are offered to the network with Bernoulli probability $p_{s\sigma}^{(\Omega)}$ and dropped with probability $(1 - p_{s\sigma}^{(\Omega)})$. Similarly, the probability that a call of service type s is offered to a route r is $\lambda_{sr}^{(\Omega)} / \Lambda_{s\sigma}^{(\Omega)}$.

2.2 **Problem Formulation**

The parameters $e_{sr}^{(\Omega)}$ denote the revenue earned per carried call per unit time by calls of service type s on route r in subnetwork Ω. Also let $L_{sr}^{(\Omega)}$ be the equilibrium probability of loss on the route. Since $\rho_{sr}^{(\Omega)}(1-L_{sr}^{(\Omega)})$ is the carried load of service type s on route r, the long-run average revenue for the subnetwork is

$$W^{(\Omega)} = \sum_{(s,\sigma)^{(\Omega)}} \sum_{r\in R^{(\Omega)}(s,\sigma)} e_{sr}^{(\Omega)} \rho_{sr}^{(\Omega)} (1-L_{sr}^{(\Omega)}) \qquad (2.4)$$

The total network revenue

$$W = \sum_{\Omega=1}^{\overline{\Omega}} W^{(\Omega)} . \qquad (2.5)$$

Typically the earnings parameters $e_{sr}^{(\Omega)}$ are proportional to the bandwidth required by a representative call. However, $e_{sr}^{(\Omega)}$ can be used to tune call-level quality of service, with higher earning rate giving better service.

Our VPN DESIGN PROBLEM is

$$\max_{\{C_\ell^{(\Omega)}\},\{\rho_{sr}^{(\Omega)}\}} W \tag{2.6}$$

such that

$$\sum_{r\in R^{(\Omega)}(s,\sigma)} \rho_{sr}^{(\Omega)} \le \rho_{s\sigma}^{(\Omega)} \quad \forall (s,\sigma)^{(\Omega)}, \quad \forall \Omega$$
$$\rho_{sr}^{(\Omega)} \ge 0$$

$$\sum_{\Omega=1}^{\bar{\Omega}} C_\ell^{(\Omega)} = C_\ell \quad \forall \ell$$
$$C_\ell^{(\Omega)} \ge 0 \quad \forall \Omega, \ \forall \ell$$

Note that this is a joint resource (bandwidth) allocation and route optimization problem. The latter in isolation has been treated earlier in [10] for a fixed network. In the algorithmic development here it is simply assumed that the procedure in the earlier work is available. For the benefit of the reader we state the OPTIMUM ROUTING PROBLEM for each subnetwork Ω. The inputs to this problem are $\{ C_\ell^{(\Omega)} \}$, $\{ \bar{\rho}_{s\sigma}^{(\Omega)} \}$, $\{ e_{sr}^{(\Omega)} \}$ and $\{ R^{(\Omega)}(s,\sigma) \}$.

$$\begin{aligned} &\max \quad W^{(\Omega)} \\ &\{\rho_{sr}^{(\Omega)}\}: \qquad \forall (s,\sigma)^{(\Omega)} \\ &\left.\begin{aligned} &\sum_{r\in R^{(\Omega)}(s,\sigma)} \rho_{sr}^{(\Omega)} \le \rho_{sr}^{(\Omega)} \\ &\rho_{sr}^{(\Omega)} \ge 0 \end{aligned}\right\} \end{aligned} \tag{2.4}$$

3. **NETWORK ANALYSIS**

3.1 **Fixed Point Equations**

The FPE for the subnetwork are derived on the basis of the well known assumption of link independence. Each route r carrying traffic of service type s and using link ℓ adds load $\nu_{s\ell;r}$ on link ℓ which, by assumption, is Poisson, with a rate which is reduced by independent thinning by all other links in the route. By summing $\nu_{s\ell;r}$ over all routes r which use link ℓ we obtain $\nu_{s\ell}$, the total thinned load of service s offered to link ℓ. Now let $B_{s\ell}$ denote the loss probability of service s on the link ℓ. This quantity is computed in VPN DESIGNER from $\nu_\ell = \{\nu_{s\ell}\}$ by either the Kaufman-Roberts recursion [18], [19], with complexity $O(C_\ell)$, or if $C_\ell >> 1$ and $\nu_\ell = O(C_\ell)$, by RUAA [16], with complexity O(1). In any case, we let $\phi_{s\ell}$ denote this mapping . The complementary part of the FPE gives the thinned offered load at each link as a function of the link loss probabilities. This mapping is denoted by ψ. The complete subnetwork FPE are:

$$\begin{aligned} B_{s\ell} &= \phi_{s\ell}(\nu_\ell), \ \ (s=1,2,..,S;\ \ell = 1,2,\ldots,L), \\ \nu &= \psi(B), \end{aligned} \tag{3.1}$$

where

$$\nu = \{\nu_\ell\} \quad \text{and} \quad B = \{B_{s\ell}\} \ .$$

The numerical technique for solving the FPE is successive approximations. Finally, the loss probability L_{sr} of service s on route r is obtained from the solution of the FPE thus

$$L_{sr} = 1 - \prod_{\ell \in r}(1 - B_{s\ell}); \qquad r \in R(s,\sigma) \tag{3.2}$$

3.2 **Single Link Asymptotics**

Just here we focus on the use of the RUAA for a single link and suppress the link index ℓ. The reader is referred to [3] or [9] for detailed information on the UAA, and to [8] for the RUAA. Here it will suffice to give an expression for the capacity gradient, which is needed later. The link capacity C is large, the traffic intensities $\{\nu_s\}$ are also large, O(C), and each call requires capacity d_s for service s. Let

$$F(z) \underline{\underline{\Delta}} \sum_s \nu_s (z^{ds} - 1) - C \log z \qquad (z > 0) \tag{3.3}$$

This function is strictly convex with an unique positive z which minimizes F(z), and is readily calculated by bisection. Now let the loss probability $B_s = \mathbb{L}_s(d,\nu,C)$. Then the capacity gradient

$$\frac{\partial \mathbb{L}_s}{\partial C} \sim -\sum_{i=1}^{3} A_s^{(i)} \sigma^{(i)} \tag{3.4}$$

$$\sigma^{(1)} = \frac{\log z^*}{(z^*-1)}, \quad \sigma^{(2)} = -\left[\frac{(z^* \log z^* + 1 - z^*)}{(z^*-1)^2} + \sigma^{(1)}\right] \tag{3.5}$$

and

$$\sigma^{(3)} = \frac{[2(z^*)^2 \log z^* + 2z^*(1-z^*) - (z^*-1)^2]}{(z^*-1)^3} - 2\sigma^{(2)} \tag{3.6}$$

3.3 **RUAA in Implied Costs**

We consider the implication of the RUAA in the calculations for the implied costs. The implied costs $c_{s\ell}$ $1 \le C \le S$, $1 \le \ell \le L$, are important in both optimizations of routing and capacity allocation [10], [11-14]. We have been able to show [ ] that the implied costs are given asymptotically by

$$c_{t\ell} \sim \sum_{i=1}^{3} a_{t\ell}^{(i)} \xi_\ell^{(i)} , \tag{3.7}$$

where

$$\xi_\ell^{(i)} = \sum_{\sigma,s} \sum_{r \in R(s,\sigma):\ell \in r} A_{s\ell}^{(i)} \nu_{s\ell;r} \left(e_{sr} - \sum_{k \in r-\{\ell\}} \sum_{j=1}^{3} a_{sk}^{(j)} \xi_k^{(j)} \right) . \tag{3.8}$$

Equations 3.8 is a complete system of equations in $\{\xi_\ell^{(i)}\}$. The parameter $\{A_{s\ell}^{(i)}\}$, $\{\nu_{s\ell;r}\}$ and $\{a_{sk}^{(j)}\}$ are all obtained from the solution of the FPE (3.1). The complexity of solving the system of 3L linear equations (3.8) is $O(27\,L^3)$, whereas the complexity of solving the system of SL linear equations (3.4) directly for the implied costs is $O(S^3L^3)$. Independence from the number of services S constitutes a major gain if S is large.

3.4 **Revenue Sensitivity to Link Capacity**

We consider the implication of the RUAA for the sensitivity of the subnetwork revenue to the link capacities. We have been able to show [ ] that the corresponding gradient of the subnetwork revenue with respect to the capacity of link ℓ is

$$\frac{\partial W}{\partial C_\ell} = -\sum_{\sigma,s} \sum_{r\in R(s,\sigma):\ell\in r} \frac{\partial \mathbb{L}_s}{\partial C_\ell} \nu_{s\ell;r}\left(e_{sr} - \sum_{k\in r-\{\ell\}} c_{sk}\right) \tag{3.9}$$

Hence, from (3.4), (3.7) and (3.8), we obtain

$$\frac{\partial W}{\partial C_\ell} \sim \omega_\ell \tag{3.10}$$

where

$$\omega_\ell \underline{\underline{\Delta}} \sum_{i=1}^{3} \sigma_\ell^{(i)} \xi_\ell^{(i)} . \tag{3.11}$$

Here $\sigma_\ell^{(i)}$, i = 1, 2, 3, are given by (3.5) and (3.6), with z^* replaced by z^*_ℓ. Also, $\xi_\ell^{(i)}$ are obtained by solving the system of 3L linear equations in (3.8).

3.5 **Optimal Capacity Reallocations**

We consider optimal capacity reallocations based on a linearization of the total network revenue and the linearized capacity costs. After each such reallocation, the optimal routing problem is solved for each newly resized subnetwork.

Taking note of (3.10), a linearized extrapolation of the revenue for the network is

$$W(C-n) - W(C) \approx -\sum_{\ell=1}^{L} \Delta W_\ell \tag{3.12}$$

where $\Delta W_\ell \underline{\underline{\Delta}} \sum_{\Omega=1}^{\overline{\Omega}} \omega_\ell^{(\Omega)} n_\ell^{(\Omega)}$. Here $n_\ell^{(\Omega)}$ $(\Omega = 1,2,\ldots,\overline{\Omega})$ are candidates for modifying the capacity allocation to subnetwork Ω on link ℓ. We proceed to select $\{n_\ell^{(\Omega)}\}$ such that we ΔW_ℓ is minimized for ℓ = 1, 2, …, L.

$$\min \quad \Delta W_\ell$$

$$\{n_\ell^{(\Omega)}\}: \sum_{\Omega} n_\ell^{(\Omega)} = 0 \qquad (3.13)$$

$$\overline{n}_\ell^{(\Omega)} \le n_\ell^{(\Omega)} \le \overline{\overline{n}}_\ell^{(\Omega)}$$

In (3.13) the last constraint ensures that the new capacity allocations are nonnegative, do not individually exceed any link capacity and also that the variables are not so large that the approximation in (3.12) is invalid.

We have been able to obtain a closed form solution to (3.13). The solution has a simple structure. The decision variables corresponding to subnetworks with small (large) linearized capacity costs are selected to equal upper (lower) bounds in the second constraint, with the exception of at most one variable which is such that the first constraint in (3.13) is satisfied.

4. ALGORITHMS AND SOFTWARE

4.1. Pseudocode of the Algorithm

To recapitulate, the problem in (3.13) of optimal selection of **n** such that W(**C** - **n**) is maximized, which we refer to as the LINEARIZED PROBLEM OF OPTIMAL CAPACITY REALLOCATIONS, has an explicit solution (see [10) for each link ℓ. Then $(C_\ell^{(\Omega)} - n_\ell^{(\Omega)})$ is the new allocation of bandwidth to subnetwork Ω on link ℓ. Recall that for each newly sized subnetwork the OPTIMUM ROUTING PROBLEM has to be solved, and on completion the linearized capacity costs need to be calculated, as described in Sec. 3. At the highest level, VPN DESIGNER algorithm has the structure.

While (not_converged) {

- solve optimal routing

problem for each VPN given current, allocation, to obtain capacity gradients;

- solve a capacity reallocation

problem, to reallocate link bandwidths

}

In the first step, we solve $\overline{\Omega}$ optimal routing problems, one for each subnetwork; obtaining capacity gradients. In the second step, the linearized capacity reallocation problem is solved to get new allocations $C_\ell^{(\Omega)}$. The expanded pseudocode is given below.

Pseudocode of the Algorithm for the VPN DESIGN PROBLEM

input: $\{C_\ell\}$, $\{\overline{\rho}_{s\sigma}^{(\Omega)}\}$, $\{e_{sr}^{(\Omega)}\}$, $\{R_{(s,\sigma)}^{(\Omega)}\}$

output: $\{C_\ell^{(\Omega)}\}$, $\{W^{(\Omega)}\}$

1. Initialize: $k \leftarrow 0$

   initial allocations $\{C_\ell^{(\Omega)} : \sum_\Omega C_\ell^{(\Omega)} = C_\ell, \forall \ell\}$

   initial hypercubes $H_\ell(\overline{n}_\ell, \overline{\overline{n}}_\ell) = \{\overline{n}_\ell^{(\Omega)} \le n_\ell \le \overline{\overline{n}}_\ell^{(\Omega)}, \forall \ell\}$

   hypercube threshold H_T

   revenue threshold W_T

   $W_{prev} = 0$

   not_converged $\leftarrow$ true

2. While (not_converged) do

   2.1 for $\Omega = 1, 2, \ldots, \overline{\Omega}$ {

   solve OPTIMUM ROUTING PROBLEM (see (2.7))

   input: $\{C_\ell\}$, $\{\overline{\rho}_{s\sigma}^{(\Omega)}\}$, $\{e_{sr}^{(\Omega)}\}$, $\{R_{(s,\sigma)}^{(\Omega)}\}$

   output: $\{\rho_{sr}^{(\Omega)}\}$, $W^{(\Omega)}$

   compute linearized capacity costs $\{\omega_\ell^{(\Omega)}, \forall \ell\}$

   }

   2.2 test for convergence

$W = \Sigma_{\Omega} W^{(\Omega)}$

if $(W - W_{prev})/W \le W_T$ and $|H(\bar{n}, \bar{\bar{n}})| \le H_T$

not_converged ← false

else $W_{prev} \leftarrow W$

2.3 if not_converged

for $\ell = 1, 2, \ldots, L$

solve the LINEARIZED PROBLEM OF OPTIMAL CAPACITY REALLOCATIONS

input: $\{C_{\ell}^{(\Omega)}\}$, $\{\omega_{\ell}^{(\Omega)}\}$

4.2 Software Architecture of VPN DESIGNER

The software architecture of VPN DESIGNER is shown in Fig. 4.1. Each circle in the figure is either one unix process or a collection of unix processes. The user describes the infrastructure topology, graphically using a TCL/TK GUI, or textually using a symbolic non-procedure language. The language, VPN DESIGNER modeling language (VML, for short) has a simple syntax

entity-keyword entity-numeric id entity-symbolic-id entity-descriptors;

where the entity is nodes, edges, or subnetworks.

The language statements for nodes and edges are identical to the statements of TALISMAN modelling language, TML [M1196-2]. The subnetwork entity statement has the syntax

subnetwork numeric-id symbolic-id tml-file 0 or 1;

tml-file is the name of the file that describes the subnetwork's demand, services and other parameters in the TML language.

VML assumes that a corresponding algorithm specification file also exists with the same stub as the tml file, i.e., same prefix but with the suffix of ·spcs. The 0 or 1 flag indicates exclusion or not, for the current run, 0 implying non-exclusion and 1 implying exclusion.

VPN DESIGNER parser process parses the VML file (either directly input, or generated through a TCL/TK GUI process), and generates a binary object file. To complete the description of the VPN DESIGNER problem, the user needs to specify how the customer subnetwork nodes and edges are mapped to the infrastructure. This specification is given in a subnetwork Map file. Observe that the VPN DESIGNER has *no knowledge* of the topology, the entity symbols that the customer uses to describe the subnetwork, the traffic matrix and service structure. This "information hiding and data transparency" is crucial and makes VPN DESIGNER portable to the web environment, parallelizable, modular, and extendable. Moreover, this data transparency enables the customers to preserve the security and proprietary nature of their network; each customer will simply describe the subnetwork himself and provide a map file to map the subnetwork to the infrastructure. The modular VPN DESIGNER structure has far reaching implications which we will exploit in future versions of VPN DESIGNER. The map file is an ASCII file. Each statement in the file, terminated by an *end* keyword, describes one mapping of a single infrastructure linu. The syntax of the statement is shown in Fig. 4.2.

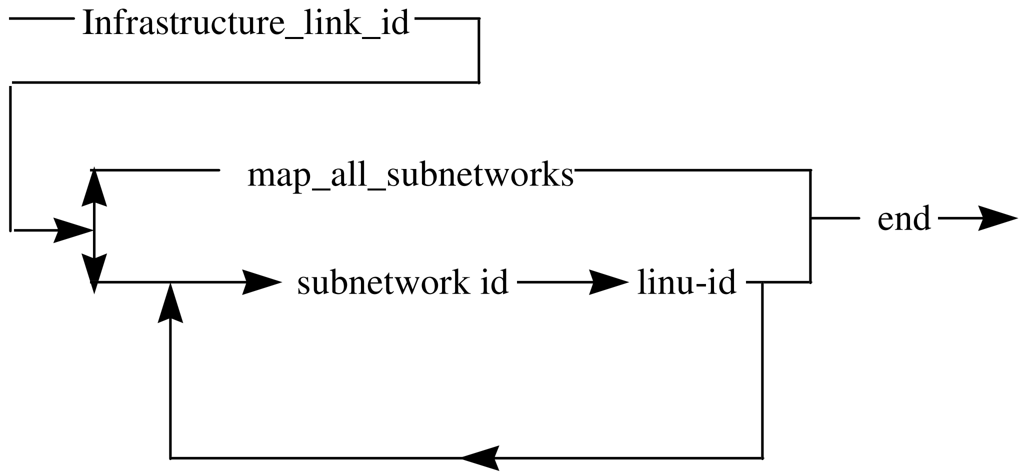

Figure 4.2. Syntax of map file statements

As an example of the map file statement consider the infrastructure network shown in Fig. 5.1. The map file for the VPN DESIGNER problem will include the statements

san-diego-palo-alto

west-corp	main-main-linu
midwest-corp	SD-SF linu
east-corp	west-trunu
south-corp	SD-Silicon-Valley-

end

Argonne-houston map-all-subnetworks

Fig. 4.3. Snapshot of a map file for the VPN DESIGNER problem described in Sec. 5.

In Fig. 4.3, map file states that the subnetwork TML files have their own entity names for the San-diego - palo alto linu, and have an *identical* name Argonne-houston for that linu in all the subnetworks.

The third and final input file that the user specifies is the algorithmic parameter specification file, which provides the convergence thresholds W_T, H_T, and other parameters. The object module describing the backbone topology, is the subnetwork map file and the algorithms specs file are passed to the VPN DESIGNER optimizer process. This process is the “master” process. After an initial allocation of link bandwidths to various subnetworks, VPN DESIGNER optimizer process creates $\overline{\Omega}$ TALISMAN master processes and passes them the current allocation C_{ℓ}^{Ω}, along with their corresponding TML and algorithmic specification file.

The TALISMAN master processes, in turn create their own process hierarchy, as described in [M11-2] to accomplish the optimal routing for each subnetwork. After termination, the TALISMAN master process passes b___ the linearized capacity costs to the VPN DESIGNER optimizer process and the iterations continue. All interprocess communication is through binary/ascii files.

VPN DESIGNER Software is entirely written in C++ language with template classes defining various entities of the problem. It currently runs on any UNIX platform.

4.4 **Parallelization**

The software architecture shown in Fig. 4.1 provides a natural process-level macro parallelism among the TALISMAN processes. VPN DESIGNER optimizer process creates $\overline{\Omega}$ TALISMAN master processes and waits for all of them to finish. The TALISMAN processes execute “in parallel” and solve the $\overline{\Omega}$ optimal routing problem. Special care has been taken in the software to remove resource contention and name-space conflicts among TALISMAN processes imposed by the hardwave and the operating system. Since no communication takes place between TALISMAN processes, VPN DESIGNER software parallelizes in either a message passing architecture or a shared memory architecture, with very little overhead.

4.5 **Warm Start**

We restart the TALISMAN processes using the optimal routing solution obtained in the previous iteration. This "warm start" considerably reduces the TALISMAN iterations to converge to an optimal solution. The reason warm starts speed up the iterations is because of the fact that the capacity changes for each subnetwork are gradual and incremental (as ensured by a small hypercube), thus the optimal routing for the current iteration does not change significantly.

5. COMPUTATIONAL RESULTS

We briefly describe the results of our computational experiments based on VPN DESIGNER. The infrastructure network, shown in Figure 5.1, has 8 (N) nodes and 24 OC3 links connecting 10 pairs of nodes in both directions (L = 20). There are 6 (=S) services, with effective bandwidth of individual calls being 16, 48, 64, 96, 384 and 640 Kbps over all links.

These are 4 (= $\overline{\Omega}$) VPNs for the West, Mid West, East and South Corporation. These are all national customers with traffic from every node to every node, for each of the services with the traffic patterns for individual corporations exhibiting some preponderance of traffic to and from the nodes in the geographical areas corresponding to their names. For details on the traffic, revenues etc., see [10]. The admissible routes are such that these are at most 4 hops in the route.

Table I gives a summary of the offered and carried bandwidth over the VPNs for the optimized design. Of particular importance is the absence of significant variability in the blocking for the VPNs. This is consistent with the global load balancing in the design.

Table I: Summary Statistics for the Optimized Design

VPN	West	Mid West	East	South	Total
Offered Bandwidth	18484	17269	23167	15758	74679
Carried Bandwidth	17904	16739	22530	15230	72400
Blocking (%)	3.14	3.07	2.75	3.35	3.05

We compare the above results with that obtained by the compete sharing design in which the VPN structure is eliminated and the aggregate traffic is optimally routed. The latter solution is obtained from TALISMAN. The total Carried Bandwidth in the Complete Sharing Solution is 73,795, which gives 1.2% blocking. Hence about 1.9% incremental blocking is due to partitioning resources among the VPNs.

In Table II we consider the Palo Alto-Argonne link and give the capacity allocations to the VPNs in the optimized design.

Table II: Palo Alto - Argonne Link (capacity = OC3)

VPN	West	Mid West	East	South
Relative Capacity Allocation to VPN (%)	32.9	25.0	24.9	17.1
Relative Utilization in VPN by service 1	1.0	0.3	2.1	1.8
“ 2	2.7	0.1	6.2	5.7
“ 3	7.3	0.6	16.7	14.7
“ 4	4.6	0.3	12.4	10.8
“ 5	21.7	23.2	14.0	38.7
“ 6	62.7	75.5	48.6	28.3
Total Utilization in VPN (%)	82.6	79.2	83.4	80.5

Also given in the table are the utilizations by each of the six services. Quite remarkable is the lack of significant variability in the utilization of the link by the VPNs, possibly due to effective global load balancing.

Only 10 outer iterations were necessary to meet the convergence threshold of 10^{-3} amd Table III gives the total carried bandwidth after even numbers of iterations. Particularly noteworthy is the goodness of the initial design which, is a VPN design based on the Complete Sharing solution.

Table III: Convergence in the Design Process

Iterations	0 (initial)	2	4	6	8	10 (converged)
Carried Bandwidth	72042	72207	72308	72359	72384	72400

REFERENCES

[1] *IEEE J. Select. Areas Commun.*, **13**(6-7), Aug.-Sept. 1995.

[2] M. Schwartz, *Broadband Integrated Networks*, Prentice Hall, New York, 1996.

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

[4] F. P. Kelly, "Routing in circuit-switched networks: optimization, shadow prices and decentralization", *Adv. Appl. Prob*, **20**, pp. 112-144, 1988.

[5] F. P. Kelly, "Fixed point models of loss networks", *J. Austr. Math. Soc.*, Ser. B, **31**, pp. 204-218, 1989.

[6] K. W. Ross, *Multirate Loss Models for Broadband Telecommunications*, Springer, New York, 1995.

[7] A. Farago, S. Blaabjerg, L. Ast, G. Gordos, and T. Henk, "A new degree of freedom in ATM network dimensioning: optimizing the logical configuration", *IEEE J. Select. Areas Commun.*, **13**(7), pp. 1199-1206, 1995.

[8] J. A. Morrison, K. G. Ramakrishnan, and D. Mitra, "Refined asymptotic approximations to loss probabilities and their sensitivities in shared unbuffered resources", submitted to *SIAM J. Appl. Math.*, Aug. 1997. Paper located in http://cm.bell-labs.com/who/mitra.

[9] D. Mitra and J. A. Morrison, "Erlang capacity and uniform approximations for shared unbuffered resources", *IEEE/ACM Trans. Networking*, **2**, pp. 558-570, 1994.

[10] D. Mitra, J. A. Morrison and K. G. Ramakrishnan, "TALISMAN: An integrated set of tools for ATM network design and optimization", Proc. NETWORKS 96, pp.

[11] D. Mitra, J. A. Morrison and K. G. Ramakrishnan, “Multiservice virtual private networks: The joint resource allocation and routing problem”, Bell Labs report, 1998, submitted for publication.

[12] J. S. Kaufman, “Blocking in a shared resource environment”, *IEEE Trans. Commun.*, **COM-29**, pp. 1474-1481, 1981.

[13] J. W. Roberts, “Teletraffic models for the Telecom 1 integrated services network”, *Proc. International Teletraffic Congress - 10*, Session 1.1, paper #2, 1983.