The Accurate Hose Model for VPN Provisioning

Dênio Mariz, Judith Kelner, Djamel Sadok, Carlos A. Kamienski

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Cidade Universitária, Recife/PE

{denio,jk,jamel,cak}@gprt.ufpe.br

Abstract. *This work considers the problem of VPN provisioning, focusing on the process of dimensioning the links belonging to the route that connects VPN endpoints, according to the specified traffic demands. To accomplish this, we propose a new model, the Accurate Hose, which can take into consideration a complete or partial traffic matrix and supports group based bandwidth requirements, while maintaining the advantage of the point-to-multipoint style of shared provisioning and flexibility already seen in the traditional Hose model. Furthermore, it is shown that the Accurate Hose model reduces VPN resource allocation when compared to Hose and that the more precise the traffic specification is, the more optimized will be the VPN provisioning achieved.*

1. Introduction

A *Private Network* (PN) is a network where existing links are used exclusively, and built using its own or third party circuits (ex: *Frame Relay* or ATM transport network). A *Virtual Private Network* (VPN) may be seen as a privately owned network while actually built over a public infrastructure, such as the Internet, or over an access provider's backbone. Usually VPNs are implemented using encrypted tunnels, therefore offering data traffic high levels of confidentiality and security as in real private networks [7][8] while maintaining scalability, accessibility and operational costs at competitive levels [1][2]. Current estimates put WAN cost savings between 20% and 47% when exchanging dedicated access links by VPNs [11] and even higher up to 60% to 80% cost gains when upgrading from dialed access to VPNs in the case of corporate access [10]. VPNs are equally attractive from the provider's point of view since, in addition to selling competitive access technology to customers, these may be used to embed some new value-added services such as security management, support, consulting, integration of new emerging multimedia services including voice over IP (VoIP), e-commerce, etc [2][6].

In summary, a VPN may be regarded as a set of geographically spread endpoints with links spanning between them in such a way that traffic originating at a given endpoint may only be accessed by other endpoints that are part of the same VPN. Its lifespan may vary from as little as few hours such as in the case of a video conference marking a special event to as long as a number of years as in the case of VPNs for Intranet traffic. Although physically a VPN shares the same network infrastructure with other VPNs' network users, there is however a natural logical separation of VPN traffic that allows it to define network level information such as its own addressing space, routing techniques etc.

Using VPNs for the establishment of advanced applications with stringent Quality of Service (QoS) requirements remains a challenging research topic to pursue. We chose to look into some the issues this problem raises in the present study. Currently a common practice is the use of Service Level Agreements (SLAs) for adequate VPN provisioning. Furthermore, the emergence of some traffic engineering technologies such as MPLS, RSVP-TE and, more recently, a combination of MPLS with BGP [9], has allowed the deployment of VPNs with explicit routing over IP networks, while ensuring some levels of QoS to end customers. However, the issue of VPN dimensioning and adequate provisioning remains open for further research when considering the optimal use of subjacent networks.

In the process of establishing a VPN over a network some important steps need to be executed. The first one is to know which are the endpoints composing the VPN, the traffic demand and the QoS requirements among the endpoints. The second step is to find a path across its endpoints while assuring that the specified QoS requirements can be respected, if any, which is known to be NP-complete problem [2][5]. In other words, it has no known algorithm that computes an optimal solution in polynomial time. To be more accurate, the problem is defined in the following way: given a network of nodes and bi-directional links, where to each link a set of attributes (e.g. bandwidth, delay) are associated and given a set of VPNs, with each VPN having a set of endpoints and constraints between these nodes that must be observed, then one needs to find a set of paths that connects the endpoints of each of the given VPNs such that the constraints are maintained and that a minimum use of the resources is made. In order to deal with this class of problems, it is possible to build some heuristics (algorithms) that, in many cases are capable to lead to a "good" solution, assuming that not always the optimal solution is reachable. These paths connecting the VPN endpoints are called a *VPN route*.

Some of such heuristics have been proposed, for example, in [1], [2] and [3], which output a *tree* as a solution for the *VPN route*.

Once one have found a valid path across the endpoints, using a suitable algorithms, the third step is to dimension each of its links over the network in such a manner that the specified traffic among endpoints can flow accordingly. Hence, we must compute the amount of resources (bandwidth) to be reserved on each link in order to admit the expected traffic. This process is defined as the computation of the VPN cost, or *VPN Provisioning*. The final step is to map the VPN route and resulting information about the resources that need to be allocated in each link into a specific technology (e.g. MPLS with BGP [9]) and actually deploy the VPN over the network.

This work focuses in the VPN Provisioning stage. It is assumed that all prior stages were already conducted and we have on hands the VPN specification (endpoints and the traffic demands among them) and the VPN tree, that is, the tree that connect the VPN endpoints. We will first explain and discuss in details an existing mathematical model used to accomplish this task, known as "*Hose*", and proposes a new model, we called "*Accurate Hose*". This supports a more flexible traffic description based on complete or partial traffic matrix, and is able to take advantage of a more detailed description to optimize VPN resource allocation.

We start by introducing some important notations used along this paper. The network used for VPN provisioning is modeled as a bi-directional graph $G = (V, E)$ where V is a set of nodes and E a set of links spanning across them. To every link (i,j) unidirectional bandwidth attributes are associated. The set $P \subseteq V$ refers to the VPN endpoints.

Whereas the notation |***S***| indicates the number of elements present in the set ***S***, ***S-{s}*** represents the remaining nodes from the subtraction of ***s*** from the set ***S***.

This rest of this paper is organized in the following way. Section 2 discusses the terminology, the main concepts behind the *Hose* model, its mathematical formulation as well as presents related work. Section 3 proposes the *Accurate Hose* model that has the added benefit of further lowering traffic provisioning costs when some prior knowledge of VPN traffic demand characteristics is explored. In section 4, the performance of the *Accurate Hose* model is evaluated showing its gain over the simple *Hose*. Section 5 concludes the paper and lists related future open research issues.

2. The *Hose* Model for VPN Provisioning

There are basically two ways in which to provision QoS in the VPN context: using the *pipe* or *Hose* models [1][2][4][5]. Under the *pipe* model, a VPN costumer specifies the QoS requirements between each pair of its VPN endpoints. This requires prior knowledge of the entire end-to-end traffic matrix. The *Hose* model, as originally described by Duffield e*t al.* in [1], characterizes a traffic aggregation originated at an endpoint and spanning towards all the other VPN endpoints. This way, a *Hose* supports a traffic style that is different from the traditional point-to-point as seen in the previous pipe model. It allows VPNs to take advantage of the point-to-multipoint style of shared provisioning with added flexibility and simplicity. Kumar *et al*. later introduced in [2] new mathematical and more rigorous notations for computing VPN cost with the *Hose* model. In their work, Kumar et al. also suggested the use of a tree (a graph with no cycles) to interconnect VPN endpoints (also called the *VPN route*), and proposed some algorithms for calculating the VPN route.

The argument favoring the use a tree as the VPN route is based on some interesting properties that these structures represent. First, the use of trees to connect three or more points results in a link being shared by different pairs of points. Second, trees are scalable from the point of view of routing and path backup in the face of failures [5], which is very important for the practical deployment of VPNs. Last, the lack of loops simplifies the algorithms used for path building and management.

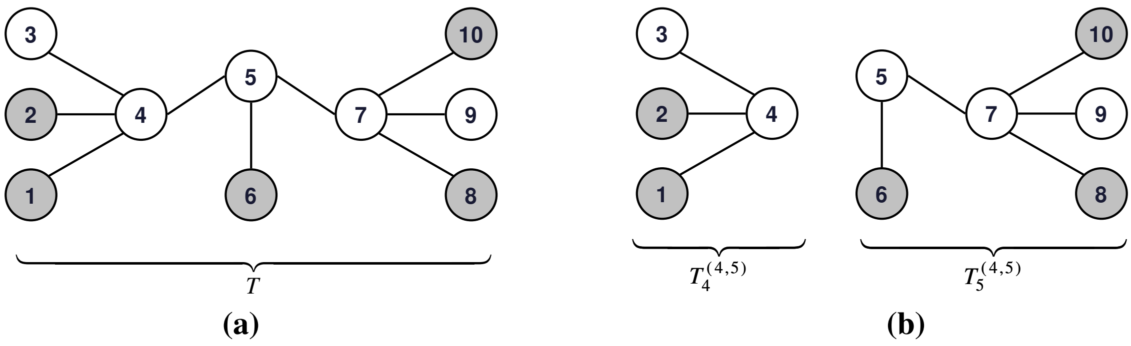


Figure 1 – A tree T and two Trees $T_4^{(4,5)}$ and $T_5^{(4,5)}$ resulting from the removal of the link (4, 5) from T.

A VPN specification using the *Hose* model consists of two parts: a) a set of nodes $P \subseteq V$, corresponding to the VPN endpoints; and b) for each endpoint $p \in P$, we associate two traffic attributes B_p^{in} and B_p^{out}, that are the ingress aggregated traffic (arriving at ***p***) and the egress aggregated traffic (leaving ***p***), respectively with regard to all the other VPN endpoints at any instant. Based on this specification, a VPN could be provisioned to guaranty the indicated traffic requirements.

The cost of a VPN to be provisioned using the *Hose* model may be defined as in the following. Consider a tree $\boldsymbol{T}$ and a link $\boldsymbol{(i, j) \in T}$ (see Figure 1(a)). Note that the removal of this link $\boldsymbol{(i, j)}$ from $\boldsymbol{T}$ produces two disjoint components: one on the side of $\boldsymbol{i}$ and the other one on that of $\boldsymbol{j}$. Let us denote $T_i^{(i,j)}$ the component on the side of node $\boldsymbol{i}$ resulting from the removal of link $\boldsymbol{(i,j)}$ from tree $\boldsymbol{T}$ and $T_j^{(i,j)}$ the component on the side of node $\boldsymbol{j}$ resulting from the removal of node $\boldsymbol{(i, j)}$ from tree $\boldsymbol{T}$. For example, Figure 1(b) shows two components $T_4^{(4,5)}$ e $T_5^{(4,5)}$, as a result of removing $\boldsymbol{(4, 5)}$ from $\boldsymbol{T}$.

We also denote $P_i^{(i,j)}$ and $P_j^{(i,j)}$ the sets of VPN endpoints contained in the trees $T_i^{(i,j)}$ and $T_j^{(i,j)}$, respectively. For example, from Figure 1(b), we have $P_4^{(4,5)} = \{1, 2\}$ and $P_5^{(4,5)} = \{6, 8, 10\}$.

Considering the link $\boldsymbol{(i, j)}$ connecting endpoints from $P_i^{(i,j)}$ and $P_j^{(i,j)}$, the egress aggregated traffic $\Psi_i^{out}(i, j)$ that should flow in $\boldsymbol{(i, j)}$ from $\boldsymbol{i}$ to $\boldsymbol{j}$ is given by:

$$\Psi_i^{out}(i, j) = \sum_{p \in P_i^{(i,j)}} B_p^{out} \tag{1}$$

Similarly the aggregated ingress traffic $\Psi_j^{in}(i, j)$ of endpoints $P_j^{(i,j)}$, in other words the traffic they can receive, is limited by:

$$\Psi_j^{in}(i, j) = \sum_{p \in P_j^{(i,j)}} B_p^{in} \tag{2}$$

Since it is not necessary to send to endpoints $P_j^{(i,j)}$ more traffic than they can receive, the total traffic $\boldsymbol{C_T(i, j)}$ crossing $\boldsymbol{(i, j)}$ should be the minimum of the egress traffic aggregated at $P_i^{(i,j)}$ and ingress traffic aggregated at $P_j^{(i,j)}$. Hence:

$$C_T(i, j) = \min\left\{\Psi_i^{out}(i, j), \Psi_j^{in}(i, j)\right\} \text{, or}$$

$$C_T(i, j) = \min\left\{\sum_{p \in P_i^{(i,j)}} B_p^{out}, \sum_{p \in P_j^{(i,j)}} B_p^{in}\right\} \tag{3}$$

Once we have established how the individual cost of each link is computed, we define the total tree cost $\boldsymbol{C_T}$, as the sum of $\boldsymbol{C_T(i, j)}$ for all $\boldsymbol{(i, j) \in T}$:

$$C_T = \sum_{(i,j) \in T} C_T(i, j) \tag{4}$$

3. The *Accurate Hose* Model

The original *Hose* model presents a serious practical limitation: both ingress and egress aggregated traffic requirements are specified for a given endpoint relatively to *all* the other endpoints. This makes it impossible to establish, separately, requirements from an endpoint to another one or another group of endpoints, something that clearly may be of practical use in VPNs. In order to overcome this drawback, a new model named *Accurate Hose* is presented, whose mathematical model was formulated to support group based bandwidth requirements while maintaining the benefits of the point-to-multipoint style of shared provisioning and flexibility seen in *Hose*.

Using the *Accurate Hose* it is possible establish the cost of a VPN tree based on a complete or partial traffic matrix. The complete Traffic Matrix ***P*** shows the aggregated amount of traffic between all individual elements of ***P***. The process of deriving accurate individual traffic demands between endpoints from ***P*** is a complex task [12] because it reflects the total knowledge about traffic between each pair of endpoints of a VPN. A partial traffic matrix is a traffic matrix where its rows not always have a value for each column. Instead, it has a value meaning a traffic demand to be distributed to some columns (a group of endpoints), without any precision.

Often, the total traffic matrix is unknown and it is only possible to have a partial matrix instead [13]. The *Accurate Hose* model takes into consideration complete or partial traffic matrix information, and the more the matrix is complete the more optimized will be the VPN provisioning achieved by the *Accurate Hose*.

We show that the *Accurate Hose* model reduces VPN resource allocation. For example, consider the network illustrated in Figure 2(a), where endpoints ***A***, ***B***, ***C***, ***D*** and ***E*** of a given VPN are connected by a tree (darker lines). Endpoints ***A*** and ***B*** have traffic demands 10Mb/s to each other (ingress and egress) and 4Mb/s to the remaining endpoints. Endpoints ***D*** and ***E*** have the same configuration. We say that ***A*** and ***B*** and points ***D*** and ***E*** form two *demand groups*, say $g_1=\{A,B\}$ and $g_2=\{D,E\}$. In a real scenario, these endpoint groups may represent regional company offices, or groups of distributed data servers (web servers, database servers), surrounded with regional customers (a formal definition for demand group is presented later).

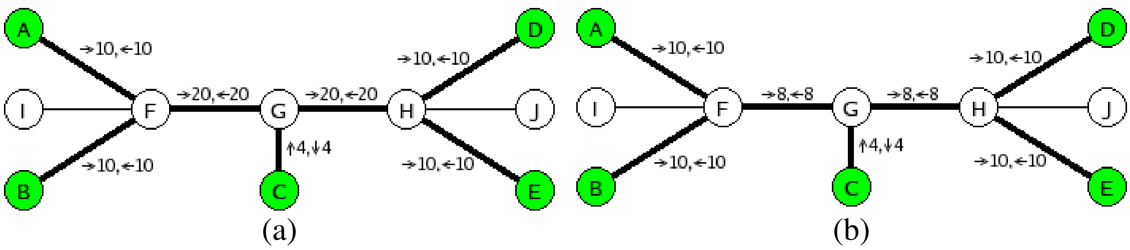


Figure 2 –VPN Cost with Demand Groups Computed (a) using the *Hose* model (cost=168) and (b) using *Accurate Hose* model (cost=120).

This type of traffic specification (detailed and group based) cannot be handled by the *Hose* model which provisions links according to the highest observed demand in each endpoint (in our example, 10Mb/s for ***A***,***B***,***D***,***E*** and 4Mb/s for ***C***). On the other hand the *Accurate Hose* model is capable of considering differentiated demands and allocating only what is really necessary at each link, hence "confining" the local traffic of demand groups and reducing VPN provisioning costs in these cases. The minimum bandwidth values needed for all the links using both models are shown in Figure 2(a) and Figure 2(b), respectively. The total VPN cost was 168 Mb/s for the *Hose* model as opposed to 120 Mb/s when using the *Accurate Hose*. Such a difference is due to the fact that the cost is computed, in the case of the *Accurate Hose*, considering only traffic that actually flows through it. Hence, in calculating the cost of link ***(F,G)***, for example, the traffic of 10Mb/s between endpoints ***A*** and ***B*** is not considered, as we will show later.

3.1. Computing VPN cost using *Accurate Hose*

Before that the mathematical model for *Accurate Hose* be present in details, some definitions are first introduced.

Definition 1: A ***demand group*** $\pi^p \subseteq P-\{p\}$ for an endpoint ***p*** of a VPN ***P*** is a set of endpoints that excludes ***p*** and for which ***p*** have a traffic requirement (ingress or egress). Each endpoint $p \in P$ can have as many as ***n*** demand groups, where (1 ≤ ***n*** < |*P*|) and we denote π_i^p as the *i*-th demand group of ***p***. Furthermore, all demand groups are disjoint and complimentary, that is $\bigcap_{i=1,\dots,n} \pi_i^p = \varnothing$ and $\bigcup_{i=1,\dots,n} \pi_i^p = P-\{p\}$. In other words, an endpoint ***q*** cannot participate in more than one demand group of ***p***; and the union of demand groups for any endpoint ***p*** must include all the other endpoints $P-\{p\}$.

Definition 2: A ***Requirement value*** $Q_p^{\pi_i}(\alpha)$ is a non-negative value representing the requirement from an endpoint ***p*** to a demand group π_i^p with respect to the QoS parameter α. In this paper, we consider $\alpha \in \{B^{in}, B^{out}\}$, where $\boldsymbol{B^{in}}$ and $\boldsymbol{B^{out}}$ represent the aggregated ingress and egress endpoint traffic, respectively.

Given the above definitions, a VPN specification using the *Accurate Hose* model consists of the following components:

a) A set of endpoints ***P*** $\subseteq$ ***V***;
b) For each endpoint ***p∈P*** and $\alpha \in \{B^{in}, B^{out}\}$, a list $\Pi_p^\alpha = \{\pi_1^p, \pi_2^p, \dots, \pi_n^p\}$ of *demand groups* of ***p***.
c) For each demand group π_i^p of ***p***, a set $Q_p(\alpha) = \{Q_p^{\pi_i}(\alpha) : 0 < i < |P|\}$, where $\alpha \in \{B^{in}, B^{out}\}$ is a specific QoS parameter and $Q_p^{\pi_i}(\alpha)$ are positive values indicating the *requirement values* for endpoint ***p*** related to a set of other endpoints $\pi_i \subseteq$ ***P***, while taking into consideration QoS parameter α.

Note that the demand groups and their respective requirements values can be supplied independently for each $\alpha \in \{B^{in}, B^{out}\}$, allowing complete flexibility and independence of traffic specification for ingress and egress traffic.

In order to identify which element from the set $Q_p(\alpha)$ represents a requirement value from an endpoint ***p*** related to another one ***q***, with QoS parameter α, we define $\delta_p^G(\alpha)$ as:

$$\delta_p^G(\alpha) = \begin{cases} Q_p^{\pi_i}(\alpha) & \text{if there is a } \pi_i^p \text{ such that } q \in \pi_i^p\text{. In this} \\ & \text{case, update } G = G - \pi_i^p \\ 0 & \text{elsewhere} \end{cases} \tag{5}$$

Assuming that a solution to the problem of interconnecting ***P*** endpoints is given by tree ***T***, where ***P*** $\subseteq$ ***T***, and considering the link ***(i, j)*** from ***T*** connecting endpoints $P_i^{(i,j)}$ to $P_j^{(i,j)}$, the aggregated egress traffic $\Phi_i^{out}(i,j)$ that should flow across ***(i, j)*** from ***i*** to ***j*** is given by:

$$\Phi_i^{out}(i,j) = \sum_{p \in P_i^{(i,j)}} \delta_p^{P_j^{(i,j)}} (B^{out}) \tag{6}$$

Meanwhile, the aggregated ingress traffic $\Phi_j^{in}(i,j)$ of points $P_j^{(i,j)}$, in other words what these endpoints may receive, is limited by:

$$\Phi_j^{in}(i,j) = \sum_{p \in P_j^{(i,j)}} \delta_p^{P_i^{(i,j)}} (B^{in}) \tag{7}$$

Therefore, the total traffic $C_T^*(i,j)$ that could flow ***(i, j)***, from ***i*** to ***j***, *under the Accurate Hose mode*, will be given by:

$$C_T^*(i,j) = \min\left\{\Phi_i^{out}(i,j),\ \Phi_j^{in}(i,j)\right\} \tag{8}$$

Once we know how to estimate the cost of each individual link, we define C_T^*, the total cost of ***T***, as:

$$C_T^* = \sum_{(i,j) \in T} C_T^*(i,j) \tag{9}$$

Note that link costs are differentiated according to traffic direction.

Based on the formulations above, the use of the *Accurate Hose* model for representing VPN costs yields a cost that is at most equal to that of the *Hose* model, but, depending on topology and traffic matrix, it can obtain gains in resource allocation for the VPN provisioning process.

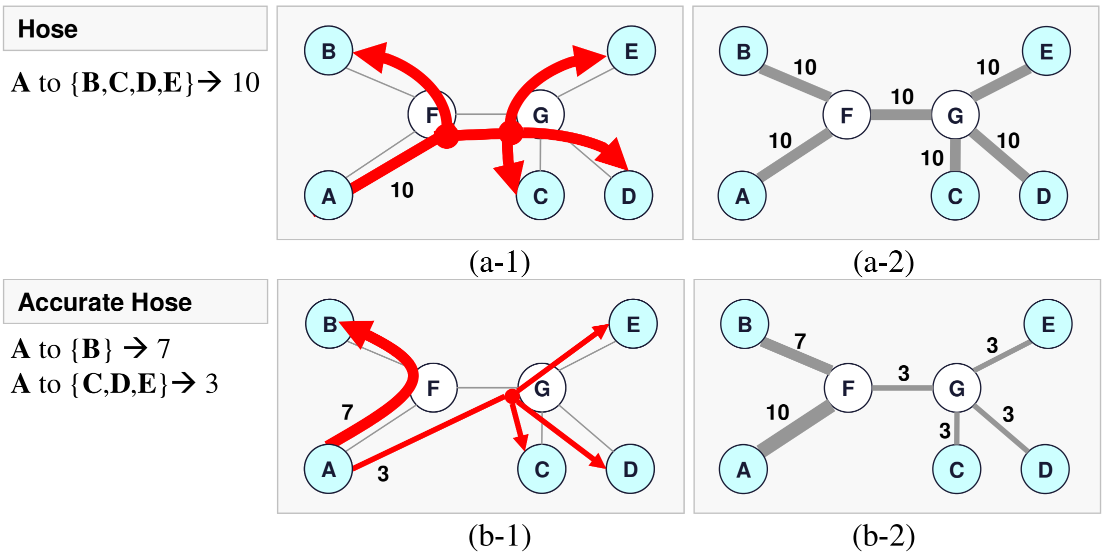


Figure 3 – How the models dimension the links for the traffic flow: *Hose* (a) and *Accurate Hose* (b).

In order to better understand the mathematical model presented, consider the example shown in Figure 3, which presents a network with VPN endpoints ***A*** to ***E***. If the traffic demand from ***A*** to all the other endpoints is known to be 10 units, the *Hose* model understands that the traffic will flow as shown in Figure 3(a-1) and provisions (dimension) the links as shown in Figure 3(a-2), with a total of 60 units. On the other hand, if there is a more precise knowledge about the traffic distribution, say that demand from ***A*** is not expected to be 10 to *all* the endpoints but 7 only from ***A*** to ***B*** and 3 to the remain endpoints, for example, the *Accurate Hose* is able to understand that the traffic flow can be as shown in Figure 3(b-1) and provision the links as shown in Figure 3(b-2), with a total of 29 units.

Next, two theorems are presented in order to formally demonstrate that the mathematical model presented for *Accurate Hose* yields to provisioning VPNs with lesser cost, and that it can be used as a general model when compared to *Hose*.

Theorem 1. The cost computed for a tree ***T*** connecting the points ***P*** using the *Accurate Hose* is always lesser than or equal to that cost computed by the *Hose* for the same tree.

♦ Proof: One have to proof that $C_T^* \le C_T$, or $C_T^*(i, j) \le C_T(i, j)$. Substituting equations (3) and (8) we have to proof that

$$\min\left\{\sum_{p\in P_i^{(i,j)}} \delta_p^{P_j^{(i,j)}}(B^{out}),\ \sum_{p\in P_j^{(i,j)}} \delta_p^{P_i^{(i,j)}}(B^{in})\right\} \le \min\left\{\sum_{p\in P_i^{(i,j)}} B_p^{out}, \sum_{p\in P_j^{(i,j)}} B_p^{in}\right\}.$$

Wherefore, it is sufficient to proof that both expressions [a] e [b] bellow are true.

[a] $\sum_{p\in P_i^{(i,j)}} \delta_p^{P_j^{(i,j)}}(B^{out}) \le \sum_{p\in P_i^{(i,j)}} B_p^{out}$

[b] $\sum_{p\in P_j^{(i,j)}} \delta_p^{P_i^{(i,j)}}(B^{in}) \le \sum_{p\in P_j^{(i,j)}} B_p^{in}$

Now, since all terms in [a] and [b] are sums (∑) of positive values, it will be equivalent if both expressions [c] and [d] bellow are proven to be true for every endpoint ***p*** and all possible link ***(i,j)***.

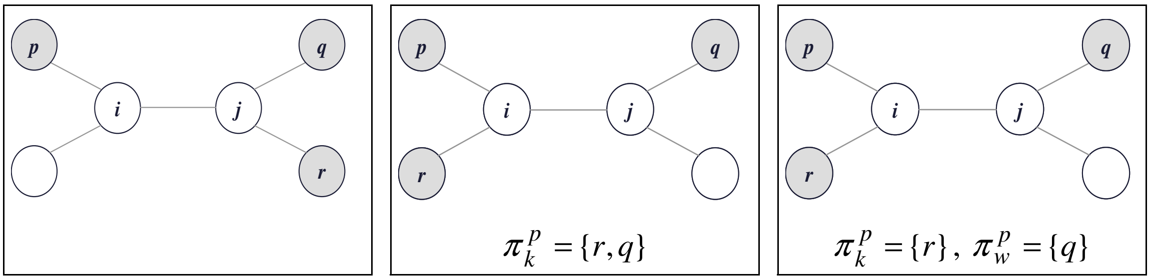
[c] $\delta_p^{P_j^{(i,j)}}(B^{out}) \le B_p^{out}$

[d] $\delta_p^{P_i^{(i,j)}}(B^{in}) \le B_p^{in}$

It is intended to prove that an expression $\boldsymbol{x \le y}$ is true showing that $\boldsymbol{x > y}$ never occurs. We start showing that (c) is true and extend the same considerations to statement [d]. Note that $\delta_p^G(B^{out})$ returns the sum of egress aggregated traffic from ***p*** to all elements of the specific set ***G***, where ***G***= $P_j^{(i,j)}$, that is, the endpoints in the side ***j*** of the link ***(i,j)***.

First of all, we recall that B_p^{out} is the sum of the egress aggregated traffic from ***p*** to ***all*** the other points and B_p^{in} is the sum of the ingress aggregated traffic arriving in ***p*** from ***all*** the other points. Hence, the relations $\delta_p^{P_j^{(i,j)}}(B^{out}) > B_p^{out}$ and $\delta_p^{P_i^{(i,j)}}(B^{in}) > B_p^{in}$ could never be verified true.

Now, consider Figure 4(a), Figure 4(b) and the Figure 4(c), where ***p***, ***q*** and ***r*** are any endpoints chosen and ***(i,j)*** is any link under analysis.



(a) (b) (c)

Figure 4 – (a) A VPN tree with endpoints *p*, *q* and *r* ; (b) a specific case where *r* and *q* are into the same demand group of *p* ; (c) a specific case where *r* and *q* are in different demand groups of *p*.

By the definition of function $\delta_p^G(\alpha)$, where in the case (c) we have $G = P_j^{(i,j)}$ and $\alpha = B^{out}$, the aggregate traffic from ***p*** to any other point, say ***r***, contributes to the cost of link ***(i,j)*** only in two situations:

❶ if all other endpoints $P - \{p\}$ are on the same side ***j*** of link ***(i,j)***, as shown in Figure 4(a), *i.e.* $P_j^{(i,j)} = P - \{p\}$; or

❷ if there is a set *S* with at least one endpoint belonging to the same demand group of any endpoint in the side ***j*** of link ***(i,j)***. In other words, if there exists a set $S = \{s : q \in P_j^{(i,j)},\ s, q \in \pi_i^p\}$, $S \subset P - \{p\}$. An example, it is the specific case shown in Figure 4(b).

When ❶ or ❷ occurs the expression $\delta_p^{P_j^{(i,j)}}(B^{out}) = B_p^{out}$ is verified true and, in this case, the cost computed by the *Accurate Hose* is equal to that computed by *Hose*.

However, if none of situations ❶ or ❷ occurs, that is, if there exists at least one endpoint ***r*** that is not in the same side ***j*** of link ***(i,j)*** (or $r \notin P_j^{(i,j)}$) and not belonging to the same demand group of any endpoint in the same side ***j***, in other words, $\forall q \in P_j^{(i,j)} \exists r : r \notin P_j^{(i,j)}, q \in \pi_i^p, r \notin \pi_i^p$ (as shown in Figure 4(c)) then, the function $\delta_p^{P_j^{(i,j)}}(B^{out})$ won't add the traffic from ***p*** to the demand group containing ***r***. This way the expression $\delta_p^{P_j^{(i,j)}}(B^{out}) < B_p^{out}$ is verified true, proving the truth of [c], as required to the entire proof. Also note that is straightforward to apply the same considerations made for $\alpha = B^{out}$ also to $\alpha = B^{in}$, that is, the case [d]. Since the statements [a] and [b] are based on [c] and [d], respectively, they are also true, which proves the theorem. ♦

Theorem 2. The *Hose* model is a specific case of the *Accurate Hose* model.

♦ Proof: A VPN traffic specification using *Accurate Hose* can be expressed using *Hose* when each endpoint ***p*** has only one demand group π_1^p comprising all the other endpoints $P-\{p\}$ and the values $Q_p^{\pi_1}(B^{in}) = B_p^{in}$ for the ingress traffic, and $Q_p^{\pi_1}(B^{out}) = B_p^{out}$ for the egress traffic. This way, the cost computed is the same using both models, that is, $C_T^*(i,j) = C_T(i,j)$. To prove this it is sufficient to expand the equation (8) for $C_T^*(i,j)$ and the equation (4) for $C_T(i,j)$ to

$$\min\left\{\sum_{p\in P_i^{(i,j)}} \delta_p^{P_j^{(i,j)}}(B^{out}),\ \sum_{p\in P_j^{(i,j)}} \delta_p^{P_i^{(i,j)}}(B^{in})\right\} = \min\left\{\sum_{p\in P_i^{(i,j)}} B_p^{out}, \sum_{p\in P_j^{(i,j)}} B_p^{in}\right\}$$

Note that, since there is only one demand group for the equivalent description using *Accurate Hose*, the function $\delta_p^G(B^{in})$ always returns $\delta_p^G(B^{in}) = Q_p^{\pi_1}(B^{in}) = B_p^{in}$ and the function $\delta_p^G(B^{out})$ always returns $\delta_p^G(B^{out}) = Q_p^{\pi_1}(B^{out}) = B_p^{out}$, where $G = P_j^{(i,j)}$. Hence, any specification using the *Hose* model can be expressed using the *Accurate Hose* model without loss of semantic, with the same cost for the correspondent VPN. Thus, we conclude that the *Hose* is a particular case of *Accurate Hose*. ♦

3.2. The complexity of *Accurate Hose*

In terms of storage the *Hose Model* has complexity ***O(***$2|P|$***)***, since it needs to store two arrays for traffic demands B_p^{out} and B_p^{in} for each endpoint. For the *Accurate Hose* model, an additional matrix having a variable number of columns for each row, depending on how the demand groups are formed is stored. Thus, the storage complexity for *Accurate Hose* is ***O(***$2|P|(n+C)$***)***, where $|P|$ is the number of endpoints, ***n*** is the number of existing demand groups and C is a constant which takes into consideration an extra data structure to control the storage of the demand groups.

The computational complexity for the *Hose* model is ***O(***$\lg(|P|)$***)***, assuming that the traffic demands are stored into a binary tree. The analysis of the computational complexity of accurate model, is basically an analysis of the computational cost held to find a *requirement value* (see Definition 1) for a specific demand group, since this is the core of the function $\delta_p^D(\alpha)$, described by (5), which is used to compute C_T^*, the cost of the VPN. A detailed analysis culminates to $O\left(\frac{|P|-1}{n}\lg^2 D\right)$, where |P| is the number of endpoints, ***n*** is the number of existing demand groups and ***D*** is the average size of the demand groups. The complete analysis is not shown here due to the space constraints, but it is shown in details in [3].

4. Evaluation of the *Accurate Hose* Model

As earlier shown in section 3, the *Accurate Hose* model, proposed in this work, makes additional intelligent use of more detailed traffic specifications between VPN endpoints. In order to evaluate its performance, we conducted two different experiments using a diverse set of topologies, modeled from real commercial and research networks such as AT&T [16][17], GÉANT [15] and RNP2 [14], and also a fixed topology of the Manhattan type with 500 nodes. Due to space restrictions, only the results for AT&T

and Manhattan topologies are shown in this paper. Before computing the cost of the VPN using *Accurate Hose* or *Hose* models, it is necessary to calculate the VPN tree connecting the VPN endpoints. The results shown in this paper were obtained using the algorithm COMPUTETREESYMMETRIC presented in [2]. However, 11 more other algorithms, mainly proposed in [1], [2] and [3], were tested over a variety of networks and the results were similar to those presented here.

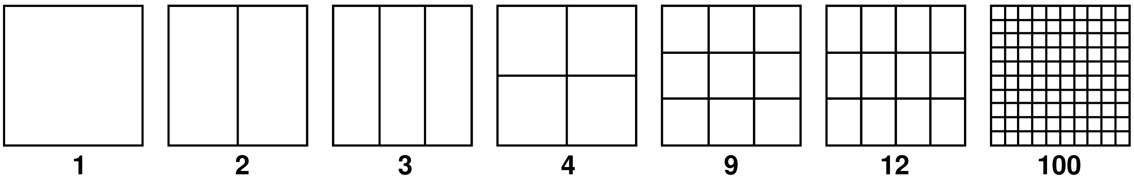


Figure 5 – Demand Groups Formed by endpoints in equal size regions

In the first experiment, VPNs are created over the networks in such a manner that its endpoints form demand groups (see Definition 1, section 3) into geographic regions over the topologies. The matrix representing the traffic distribution between the endpoints was established in a way to create ***n*** demand groups, hence dividing the geographical area into ***n*** equal size regions, as shown in Figure 5. Endpoints positioned in a common region form a demand group and a traffic matrix is built so that the internal traffic volume (within the group) is at least five times bigger than the external traffic (traffic between groups). In the simulated scenarios, ***n*** has been varied from 1 to 100, where for each ***n***, 300 random VPN replications were analyzed, each VPN with size equal to 20% of the network (in number of nodes).

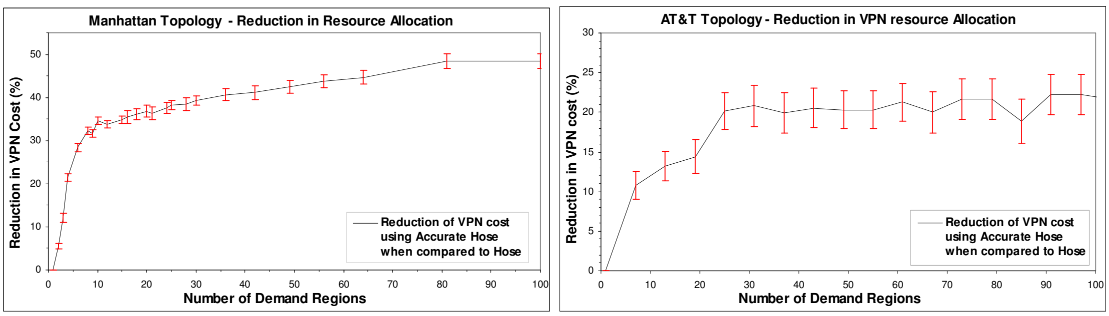


Figure 6 – Provisioning gain of the *Accurate Hose* model over the *Hose* model for Manhattan (left) and AT&T (right) topologies.

Figure 6 shows the average curve with an asymptotic confidence level of 99% for Manhattan (left) and AT&T (right). The values represent a reduction of the VPN costs obtained in the case of the *Accurate Hose* model as compared to the *Hose* model considering the number of demand groups shown over the horizontal axis. With only one demand group, no cost difference is noted between the two approaches once all endpoints have the same traffic demands and, therefore, there is no additional information on the traffic to be considered by the *Accurate Hose* model at this stage. As the number of demand groups increases, the *Accurate Hose* model outperforms the *Hose* model by lowering its VPNs costs. Its gain varies between 0% and 34% with 1 to 10 demand groups and 34% to 46% with 12 to 100 groups in the Manhattan scenario. For AT&T topology, the results also vary with the number of demand groups, but the obtained gain reaches 13% with 12 demand groups, up to 20% with 25 demand groups and up to 22% with 26 demand groups. The smoother curve from Manhattan results can be explained by its regular topology and shorter links, yielding to a major number of connected endpoints within demand regions, while in the AT&T topology links are longer, resulting in the opposite.

Despite Manhattan being an 'unreal' topology, its use is important to grasp the asymptotic behavior of both models. The results for real topologies, however, are of more practical interest and show that *Accurate Hose* reduces the resource allocation allowing lower utilization and more VPN allocation over the network.

The *Accurate Hose* model, on the other hand, presents a higher computational cost than that of the *Hose*, as we have shown in section 3.2. We observed in this experiment that, on average, the additional computational cost of *Accurate Hose* was around 49%, with values ranging from 0% to 84%. This is due basically to the use of more complex data structures for the storage and search over a larger demand set related to each endpoint.

Since *Accurate Hose* permits a more detailed traffic specification between endpoints, we conducted a second experiment to detect the relationship between the precision of traffic matrix and the gain obtained by *Accurate Hose*.

We define $A_p(\alpha)$, the *precision of a traffic specification* for an endpoint ***p*** with relation to the other endpoints $P-\{p\}$, as:

$$A_p(\alpha) = \frac{n-1}{|P|-2} \qquad (10)$$

where ***n*** is the number of demand groups for ***p***, $|P|$ is the number of VPN endpoints and $\alpha \in \{B^{in}, B^{out}\}$ (we can have different precision for ingress and egress traffic). Using this definition, we have $0 \leq A_p(\alpha) \leq 1$, and the precision is a scale indicating how precise is the traffic specification. We have $A_p(\alpha)=0$ when an aggregate traffic is indicated for all other endpoints (only one demand group, as in *Hose* model) and $A_p(\alpha)=1$ when a specific traffic demand is specified separately for all other endpoints (a complete traffic matrix with |P|-1 demand groups is used).

Since $A_p(\alpha)$ gives the precision for a specific endpoint, we define $A(\alpha)$, the precision of the entire traffic matrix as the mean of each $A_p(\alpha)$. Hence,

$$A(\alpha) = \frac{\sum_{p \in P} A_p(\alpha)}{|P|} \qquad (11)$$

The experiment consists of varying the precision from 0 to 1 and, as a function of that precision, establishing the quantity and the size of the demand groups for each endpoint ***p***. Then, each demand group is formed by endpoints randomly chosen from the remained endpoints $P-\{p\}$. The traffic demand from ***p*** for each of its demand groups is a random variable uniformly distributed in [1,10]. The matrices for ingress and egress traffic are generated independently. Since the *Hose* model is unable to deal with traffic matrices, we generated a vector of aggregate traffic instead.

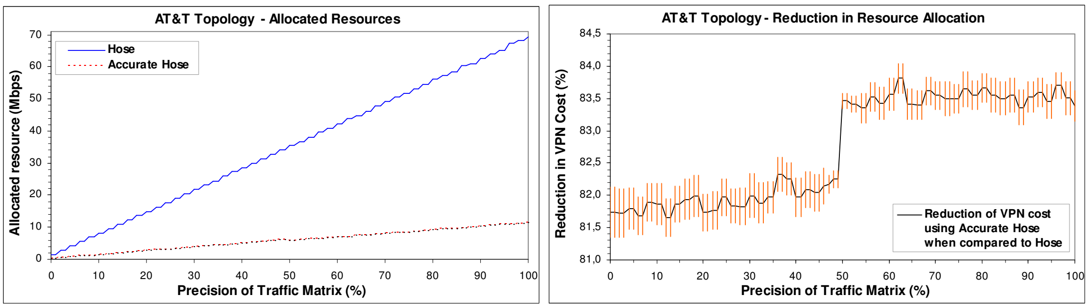


Figure 7 – Allocated resources as a function of the precision of the traffic matrix

Figure 7(left) shows the VPN cost (vertical axis) as computed by *Accurate Hose* and *Hose* as a function of the precision of traffic matrix (horizontal axis). When the precision is zero, both models compute the same VPN cost, what is expected. However, as the precision grows, the *Accurate Hose* is able to compute lesser VPN cost. Figure 7 (right) shows the reduction reached between the computed cost by each model, measured as percentiles. The results show that, for the particular scenario evaluated, *Accurate Hose* can obtain a reduction of ~82% to ~84% when compared do *Hose*.

5. Conclusions

In this work we proposed the *Accurate Hose* model, which supports a group based bandwidth requirement specification (something that the *Hose* model is unable to deal with) while maintaining the advantage of the point-to-multipoint style of shared provisioning and flexibility seen in *Hose*. Furthermore, a mathematical model for *Accurate Hose* was formulated to make further smart use of more detailed traffic specifications between VPN endpoints. We shown that *Accurate Hose* model takes into consideration complete or partial traffic matrix information, and the more the matrix is more complete the more optimized will be the VPN provisioning achieved by the *Accurate Hose*. This also leads to a greater computational cost for the *Accurate Hose* when compared to *Hose*, although this can be seen as a reasonable tradeoff for obtained gains.

Actually, based on a set of experiments conducted, it was identified that some factors have influence over the economy reached by *Accurate Hose*. The first one is number of demand groups in the traffic specification. The second factor is the variability of the traffic matrix, that is, the standard deviation of the values of traffic demand for the demand groups: the higher is the variability, the higher is the gain of reached by the *Accurate Hose*. The last factor is the topology and the distribution of the VPN endpoints. This is because the nodes degree, the distance of the links and the way the topology is organized have influence on the VPN route, and therefore, in the way the links are shared.

Determining a theoretical bound for the gain of this model over the *Hose* model is the object of current study.

6. References

[1] N.G. Duffield, P. Goyal, A.G. Greenberg, P.P. Mishra, K.K. Ramakrishnan , J. E. van der Merwe, *A flexible model for resource management in virtual private networks*, Proceedings SIGCOMM'99, Comp.Comm. Review, Vol 29, No 4, Oct 1999, pp 95-108.

[2] Amit Kumar, Rajeev Rastogi, Avi Silberschatz, Bulent Yener, *Algorithms for Provisioning Virtual Private Networks in the Hose Model*. IEEE/ACM Transactions of Networking, Vol 10 no. 4, Aug 2002.

[3] Sousa, Dênio M. T. *Algoritmos para Aprovisionamento de Redes Privadas Virtuais Baseadas em QoS usando o Modelo Hose*. Ph.D thesis (in portuguese). Centre of Informatics UFPE, Pernambuco, Brazil, April 2004.

[4] R. Cohen and G. Kaempfer, *On the cost of virtual private networks*, IEEE/ACM Transactions on Networking, vol. 8, no. 6, pp. 775-784, December 2000.

[5] Giuseppe F. Italiano, Rajeev Rastogi, Bülent Yener. *Restoration Algorithms for Virtual Private Networks in the Hose Model*. In Proceedings of IEEE INFOCOM, 2002.

[6] Ayhan Erdoğan, Dz. Yzb; *Virtual Private Networks (VPNs): A Survey*. Institute of Naval Sciences and Engineering, Naval Academy , Tuzla, Istanbul, 1999.

[7] Maughan D., Schertler M., Schneider M., J. Turner. *Internet Security Association and Key Management Protocol (ISAKMP)*. RFC 2408, IETF, 1998.

[8] S. Kent and R. Atkinson., *IP Encapsulating Security Payload*. RFC 2406, Internet Engineering Task Force, November 1998.

[9] Rosen E. C., Rekhter Y.; *BGP/MPLS IP VPNs*, IETF draft-ietf-l3vpn-rfc2547bis-03.txt, October 2004, work in progress.

[10] Günter. M. *Virtual Private Networks over the Internet*. White Paper, IBM Research 1998.

[11] Infonetics Research. *Virtual Private Networks: A Partnership Between Service Providers and Network Managers*. White Paper. October 1998.

[12] Alberto Medina, Nina Taft, Kave Salamatian, Supratik Bhattacharyya, Christophe Diot. *Traffic matrix estimation: Existing techniques compared and new directions*. In Proceedings of ACM SIGCOMM, pages 161-174, August 2002.

[13] Anja Feldmann, Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True. *Deriving traffic demands for operational IP networks: Methodology and experience*. IEEE/ACM Transactions on Networking, vol.9, 2001.

[14] *Mapa do Backbone RNP2* (in portuguese). URL <http://www.rnp.br/*backbone*>. Accessed in Dec 3, 2004..

[15] *GÉANT Topology Map - October 2004*. <http://www.dante.net/upload/pdf/Topology_Oct_2004.pdf >. Accessed in Dec 3, 2004.

[16] *RocketFuel Maps - AT&T(US) Network Map*. URL <http://www.cs.washington.edu/research/networking/rocketfuel/interactive/7018us.html>. Accessed in Dec 3, 2004.

[17] *CAIDA - Cooperative Association for Internet Data Analysis. AT&T Wordnet* (updated 08-jul-2000). URL <http://www.caida.org/tools/visualization/mapnet/*Backbones*/>. Accessed in Dec 3, 2004.