Internet Engineering Task Force
INTERNET DRAFT

Nick Duffield
Pawan Goyal
Albert Greenberg
Partho Mishra
K. K. Ramakrishnan
Jacobus E. van der Merwe
AT&T Labs - Research
Naganand Doraswamy
Shantigram Jagannath
Nortel Networks
November 1998

A Performance Oriented Service Interface for Virtual Private Networks

<draft-duffield-vpn-qos-framework-00.txt>

Status of This Memo

(Note that this ID is also available in Postscript and PDF formats)

1. Abstract

This document presents a quality of service (QoS) framework for IP based virtual private networks (VPNs). For IP based VPNs to provide a service comparable to private line networks it has to provide a number of features including closed user groups, security and performance guarantees. This document focuses primarily on the issue of performance, and provide a QoS framework which is applicable to various VPN solution that have been proposed.

2. Introduction

A Virtual Private Network service is likely to be used by customers as a replacement for networks constructed using private lines. To determine the requirements for a VPN, consider the functionality provided by private line networks:

- Closed user group: In private line networks, only the group of entities that are connected to the private line can communicate with each other. Thus, they control the group of entities that can communicate.

- Security: In private line networks, the data of a network remains inaccessible to the other networks and endpoints. Hence, even though data may not be encrypted, it remains secure, i.e., private.

- Performance: Private lines provide guaranteed bandwidth and delay characteristics. This isolates the performance seen on private networks from other flows.

- Independence of network layer: The private line network does not constrain the network layer protocol that can be employed. Furthermore, even if different private line networks share the same hardware facilities, they can use overlapping network layer addresses.

Thus, to emulate private line networks, a VPN should control the group of entities that can communicate, ensure that the communication is secure, and provide performance guarantees. Also, though in general different network layer protocols may be employed in a VPN, we restrict our attention to VPNs that employ IP as the network layer protocol. Support for private addresses is explicitly required.

A VPN, thus, can be considered to be a group of entities that communicate securely with appropriate performance guarantees. Realization of VPN requires: (1) group membership management techniques; (2) routing protocols for communication; (3) techniques for ensuring that the communication is secure; and (4) techniques for providing performance guarantees. In this draft we develop a framework for providing Quality of Service (QoS) guarantees in VPN and address some of the issues in managing group security. Other VPN drafts have focussed on other issues, especially those related to routing [1, 2, 3].

The rest of the draft is organized as follows. In Section 3, we briefly present the different models of VPN. We then present our framework for QoS in VPNs in Section 4.

3. VPN Connectivity Models

There are several models of VPN. We consider three models that are commonly articulated.

- Virtual Private Link (VPL): In this model, the physical links are replaced by virtual links. A virtual private link is a tunnel between two end points. Figure 1(a) shows a dedicated network and Figure 1(b) shows a virtual network that emulates it using Virtual Private Links. The virtual links can originate and terminate at customer edge routers. Alternatively, by employing virtual software routers [2], the virtual links can originate and terminate at provider edge routers.

  This model does not require any modification to the routing protocols. Each VPN runs an instance of a routing protocol. The routing protocols employed by each VPN can be different. This model also presents the same management interface as a network with dedicated links.

  The main differences between VPL VPN and a private line network are:

  * Since the packets of the virtual link are transported over a shared public network, the delay and loss experienced by the packets may be significantly different from that in a private line network. To guarantee the desired performance, a network has to provide performance assurances. We discuss the performance assurances and the alternative abstractions in Section 4.

  * Since the links are virtual, they may not be as secure as private lines. This drawback can be remedied using cryptography based security techniques.

  * Unlike private lines, the virtual private links may not have an associated cost with them. Hence, unlike a private line network, a VPL VPN may have significantly larger number of links and may even be fully meshed.

- Virtually Routed Network: In Virtual Private Link VPN, multiple instances of potentially different routing protocols (one for each VPN) are run across a providers backbone. If the provider manages the routing for the customer, then this may impose a significant management burden. Also, this reduces the opportunity for the provider to offer value added services. Hence, an alternative model called the Virtually Routed Network (VRN) has been proposed.

  In a VRN VPN, each customer edge router connects to one or more provider edge routers. The customer edge router conveys reachability information to the provider routers that it is connected to. A provider edge router determines the reachability information from all other provider routers that have reachability information for a given VPN. It may then disseminate this information to the customer routers. The reachability information thus gathered is used to route packets. To ensure that private addresses can be employed by members of a VPN, a provider router encapsulates packets when it forwards packets over the provider network.

  To illustrate the concept of VRN VPN further, consider a VRN VPN shown in Figure 2. Customer edge routers 1, 2, and 3 advertise

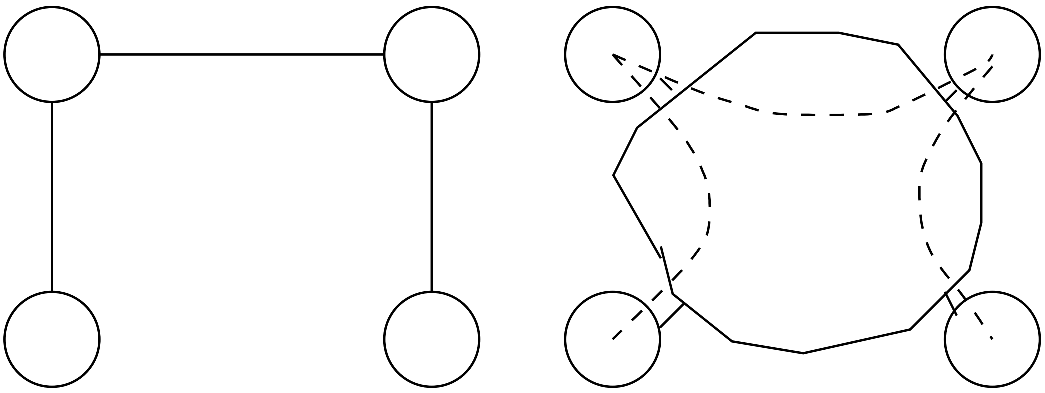

Figure 1: (a) A private line network (b) Virtual Private Link emulation of the private line network

reachability information for the hosts behind them to the provider routers. In particular, customer router 1 advertises reachability to hosts 10.127. * .* to provider routers A and B, while customer routers 2 and 3 advertise reachability information to provider routers B and C respectively. Provider routers A, B, and C disseminate the reachability information gathered from the customer routers to each other. Since customer routers 2 and 3 are connected to only one provider router, they have a default route to their respective provider routers. Hence, they do not need to learn routes from their provider routers. Consequently, provider routers B and C do not advertise routes to customer routers 2 and 3. Customer router 1, on the other hand, is connected to two provider routers. Hence, it needs to learn the routes from both the provider routers. Thus, provider routers A and B advertise routes to customer router 1.

Now, consider the process of packet forwarding. Consider a packet with destination 10.126. * .* originating from the network connected to customer router 1. Customer router 1 utilizes the reachability information gathered from routers A and B and decides to forward the packet to router A. Router A on receipt of the packet, makes a forwarding decision based on the VPN to which this packet belongs. The forwarding decision results into the packet being forwarded to router C. To forward the packet to router C, router A encapsulates the packet with router C as being the destination. Router C on receiving a packet, determines the VPN to which the packet belongs (based on the encapsulation information) and then makes a forwarding decision that is governed by the VPN to which the packet belongs.

To realize a VRN VPN, we need:

* Group membership discovery: To disseminate the routes to the appropriate set of provider routers, we need a mechanism to discover the provider routers that have members of a VPN.

* Route dissemination: A route dissemination mechanism is required to enable the various customer sites to communicate.

* Secure (group) communication: Secure communication between the various constituent networks can be achieved by using security associations between each pair of networks. However, such an approach does not scale very well. A scalable secure group communication protocol might thus be required. No secure group communication protocols have been standardized as yet.

* QoS abstractions and techniques for their realization:

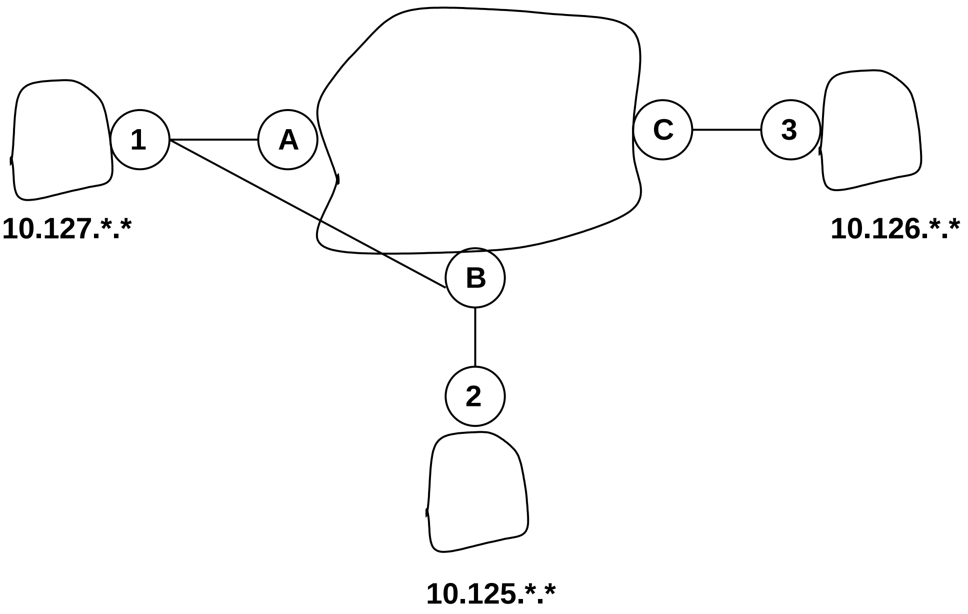

Figure 2: Virtually Routed Network VPN

- Network of Virtual Routers: In Virtual Private Link and Virtually Routed Network VPN models, the topology of the network is invisible to the customer. An alternative VPN model is to virtualize a subset of the routers in the provider network and export a virtual topology interconnecting them to a VPN customer (see Figure 3). A router is virtualized by virtualizing the control plane as well as the data plane (see ... for an example of such a virtualization). The advantage of such a VPN model is that a VPN customer can control all aspects of network operation. Though this model may be desirable for some very large VPN customers, we believe that it is not an appropriate model for most of the VPN customers.

In this document, we will focus on the QoS abstractions for the Virtual Private Link and Virtual Routed Network VPN models.

4. A Performance Oriented Service Description

One of the important requirements for IP based VPNs is to obtain differentiated and dependable Quality of Service for flows belonging to a VPN. Such functionality is crucial if VPNs implemented on IP networks are to replace the functionality provided by private

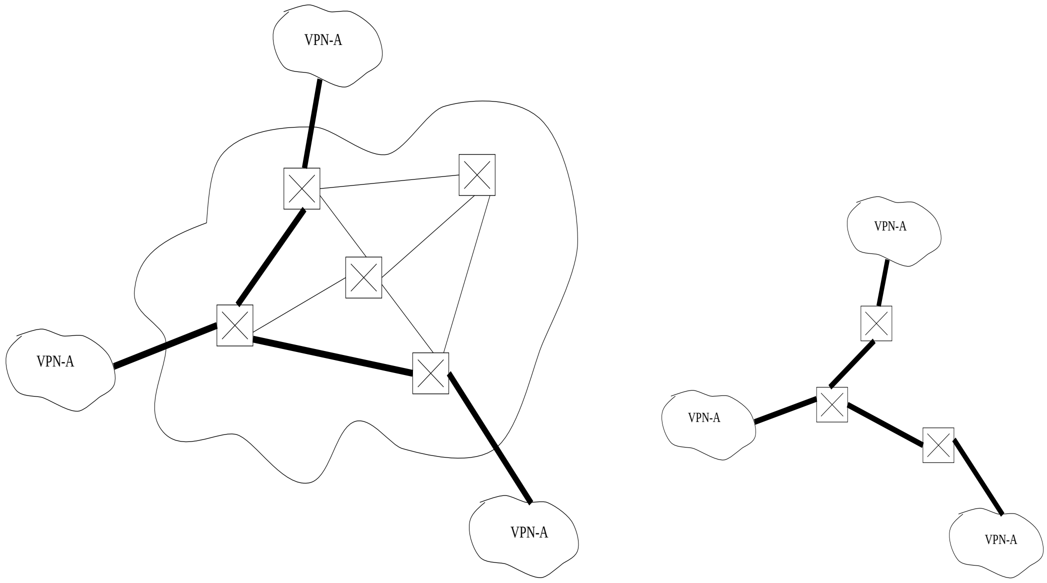

Figure 3: Network of Virtual Routers

line VPNs. It is envisaged that IP based VPNs will be capable of supporting a wide range of QoS guarantees. This can range from simple relative treatment, bandwidth commitments and/or delay assurances. Resources in the network needs to be managed to meet the QoS requirements for the VPN.

In this section two performance abstractions are defined as building blocks in the QoS framework. These performance abstractions relate to how a customer would specify or think of the performance requirements of a VPN. As such the abstractions discussed below are performance service abstractions.

In the simplest case a customer might require a pipe with certain performance guarantees between two specific VPN sites. This is analogous to specifying the capacity of a private line in conventional private line VPNs and is therefore an abstraction that will be readily understood by customers. A variety of performance guarantees can be associated with a pipe. The pipe model requires the customer to know the traffic matrix or traffic distribution between VPN sites and to translate this into a set of pipes that will meet its requirements. Such knowledge is often unknown especially for new customers and the hose performance abstraction is introduced to simplify the customer's task of specifying the performance requirements of a VPN.

A hose can be thought of as a pipe into a VPN where the endpoint of the pipe is undefined, or rather defined as any other hose in the VPN. With the hose as the service interface, a customer would therefore buy (or specify) a hose into a VPN. The customer only needs to specify the performance characteristics of the traffic coming into a hose (from any other hose) and going out of a hose (to any other hose). In particular, the customer does not have to specify the traffic matrix or the spread of this traffic to other (hose) endpoints in the VPN. Connectivity to all (or a specified set of) other hose-endpoints in the VPN is implied in this abstraction.

Figure 4 illustrates a hose with O as the origin and D_1, D_2, and D_3 as the destinations. Furthermore, the maximum aggregate traffic going out from O is $5Mb/s$ and the maximum aggregate traffic coming into it is $10Mb/s$. Note that the $5Mb/s$ going out from O can go to any of the destinations. Also, the $10Mb/s$ coming into O can come in any combination from the three nodes D_1, D_2, and D_3.

The customer can "change" the traffic matrix, i.e., start sending traffic to a different hose-endpoint from what it is currently doing, without first consulting with the provider. The provider is obliged

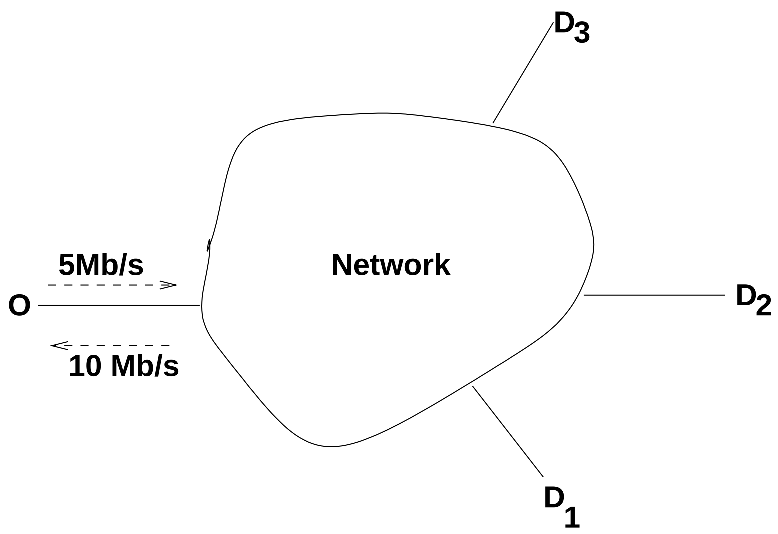

Figure 4: A hose

to keep to the agreed SLA as long as the traffic entering a hose or exiting a hose conforms to the specified profile.

An example of how a set of hoses could be specified for a VPN is shown in Figure 5. Endpoint A is a customer's central facility; endpoints X, Y and Z are branch facilities. Hoses originate at each endpoint A, X, Y, and Z. The maximum aggregate traffic outward from hose endoint A and inward to A are denoted as a_{in} and a_{out}, respectively; similar notation labels the rates associated with hoses originating from endpoints X,Y,and Z.

The hose abstraction gives the customer freedom to send traffic between each pair of endpoints, provided the aggregate traffic in and out of each of the endpoints does not exceed the respective hose capacities. When the traffic in and out of these hose endpoints satisfies this constraint, then the service provider would be expected to meet certain service level agreements (SLA), such as bounds on loss and delay. We describe the characteristics of typical SLAs below. In the context of the VPN shown in Figure 5, the hose model can accommodate a transition from a configuration in which the branch facilities only communicate with the central facility, to one in which they additionally communicate amongst each other. Whereas the hose model requires that only the aggregate traffic rates be

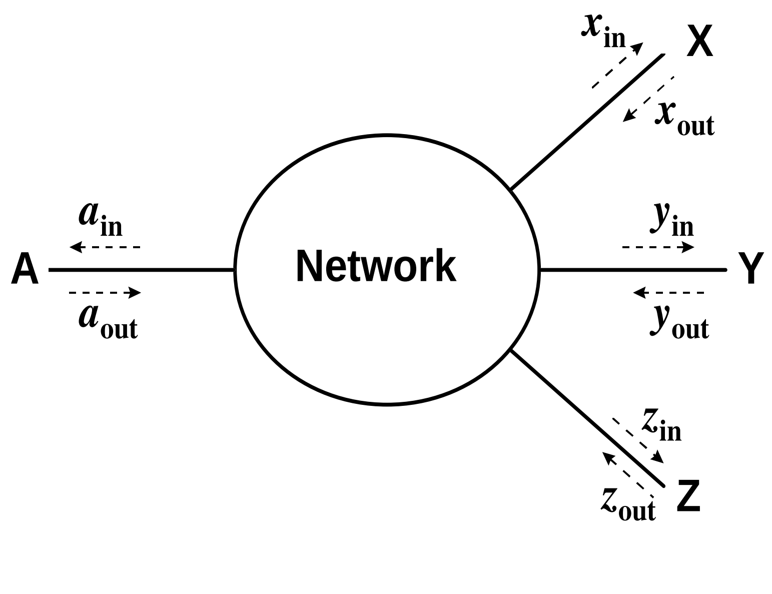

Figure 5: Hoses associated with each endpoint of a VPN

known, by contrast, a leased-line or virtual-private line service would require knowledge of the traffic rate between each pair of endpoints.

However, we envisage that partial information concerning the traffic matrix may be used at the time the hoses are provisioned in order to choose the hose capacities. Suppose, for example, it is known that each of the branch facilities X, Y and Z sends and receives to the central facility A at no more than 5Mb/sec, while each branch facility sends and receives no more than 2Mb/sec in aggregate to all the other branches. The hose capacity would then be chosen with $a_{\mathtt{in}} = a_{\mathtt{out}}$ =15Mb/s and $x_{\mathtt{in}} = x_{\mathtt{out}} = y_{\mathtt{in}} = y_{\mathtt{out}} = z_{\mathtt{in}} = z_{\mathtt{out}}$ =7Mb/sec. This gives the VPN customer the flexibility to not have to specify the capacities for communication between individual endpoints, and also allows the customer to accommodate variations in the amount of communication between the hose endpoints.

No specific VPN connectivity model between hose endpoints is assumed and the hose abstraction can be applied to any VPN connectivity model. Having this seperation between connectivity and performance simplifies the specification of the VPN from the customer's point of view in that a detailed traffic matrix is not required. Similarly from the provider's point of view the hose model allows great flexibility in how the VPN SLA may be realized in the network. Note however that when admission control is performed for a new VPN or when a new hose is added to an existing VPN, then connectivity and performance can not be considered in isolation.

Given the above discussion, the two performance service abstractions can be defined as follows:

- Pipe: A pipe provides peformance guarantees for traffic between a specific origin and destination pair. It may provide performance guarantees that are close to that of a leased line or provide weaker performance guarantees, depending on the service level agreements made with the provider.

- Hose: A hose provides performance guarantees between an origin and a set of destinations (going into the VPN) and between a node and a set of origins (coming from the VPN). A hose is characterized by:

  * The aggregate traffic from the origin to any of the destination nodes that are part of the VPN.

  * The aggregate traffic from all the other nodes in the VPN to a particular sink node in the VPN.

A hose provides performance guarantees based on such aggregate traffic specifications.

The hose performance abstraction might more naturally fit the Virtually Routed VPN model. Similarly, the pipe performance abstraction will more naturally fit the Virtual Private Link VPN connectivity model. At the same time it should be emphasized that these performance abstractions are not tightly coupled to any connectivity models and indeed both performance abstractions may be utilized in a single VPN. (For example the hose model may be used for a VPN as a whole while a pipe may be specified between the two sites of some mirrored application.) In the remainder of this section the discussion will focus mainly on the hose performance model as applied to a virtually routed VPN connectivity model.

A full specification of traffic between all the endpoints in a general network involves specifying the contents of the full traffic matrix (the traffic between each source and destination). However, with the hose performance abstraction, the contents of the traffic matrix that needs to be specified becomes significantly simpler. The customer needs to only specify the sum of the rows (the total traffic generated from a source hose endpoint) and the sum of the columns (the total traffic generated to a destination hose endpoint). Thus, producing an accurate traffic specification for the simplified hose model primarily requires an understanding of the capacity of the ''pipe'' that the customer has into the provider's network.

However, specifying the traffic specification for even this simplified case might however be a difficult process. The traffic specification provided by the customer is therefore expected to vary from very simple to very complex depending on the needs and sophistication of the customer. The performance guarantees that the customer can expect might be coupled to the detail of its traffic specification. A reasonable service offering would indeed be for customers to start off with a fairly simple specification and to then refine this specification based on operational experience and operator feedback.

A customer might request hose capacities based on an estimate of perceived needs and choose a SLA based on tariffs. (For example higher bandwidth into a central offic where the internal Web server of the company is hosted.) Coming up with this initial estimate of the VPN specification might be a service provided by the provider to its potential customers. Based on operating experience with other customers the provider might be able to suggest initial access capacities and an SLA to the customer. The customer can then monitor the performance of the VPN in terms of loss and delay to determine whether it satisfies the needs of its applications and can then negotiate a different capacity and the corresponding SLA with the provider. Again the provider might be in a position to monitor the traffic of a specific VPN in its network and provide the customer with feedback regarding traffic load and potential performance problems.

The capability of a provider to monitor and analyze the traffic load on a VPN might indeed be used as a mechanism to establish initial hose characteristics for a new VPN. As part of its VPN service offering a provider might offer an initial VPN characterization phase. The basic idea is that the customer would specify little more than the sites that it wants to be connected. During the characterization phase the operator would then analyze the traffic carried by the VPN. During this phase the SLA for the VPN might be undefined or might be defined as some best-effort service. Typically, however, a provider will act conservatively and over provision in terms of the resouces it allocates for the VPN during this phase. At the end of this initial phase the provider can then present the customer with a breakdown of the traffic charateristics of of the VPN and a number of price/performance options (SLAs) of how the provider can subsequently carry the VPN traffic. In practice, a procedure such as this will be constrained by the ease with which the capacity of the customer access links can be changed. It might not be practical to give a customer access to the VPN at very high capacity during the characterization phase, only to then scale it down to some very modest access rate afterwards. Again, based on operating experience a provider might be able to come up with a reasonable starting points for access capacity. It is expected that customers will be better able to predict their future traffic needs if they are provided with a clear picture of the current situation as described here. Service level agreements following the characterization phase might be based on the current traffic load with provisions made for expected gradual growth as well as expected drastic traffic changes that the customer might foresee (or protect against).

4.1. Service Level Agreements

The nature of the service level agreement between a customer and a service provider is driven by the traffic characteristics and QoS requirements of the (customer) applications that make use of the VPN. For example, an IP voice VPN service might request a pipe or hose with very tight bounds on the per-packet loss rates and delay while a data-only VPN service might have relatively lax loss and delay requirements.

A common way of viewing the SLA's is that the customer and network agree on the loss-utilization and delay-utilization curves associated with a pipe or a hose - see Figure 6 (1). The ''utilization'' is measured with respect to the bandwidth that is requested for each pipe or hose. In principle, the utilization can be greater than 1, to allow a customer to get more bandwidth than was initially requested, albeit with the possibility of higher packet loss and delays. A customer might also be allowed to specify distinct SLA's for different time intervals, possibly to allow for time-of-day variations.

The traffic characteristics can be specified at the entry point for the VPN as a whole, i.e., treating the VPN as a single hose. Alternatively, different traffic characteristics at a single VPN entry point can be specified as different hoses with different characteristics. In the most general case the traffic for each hose in a VPN will be specified for each direction from the origin, and would be specified over a set of time intervals. For each time interval, the customer could specify the traffic that can be sent/received in that interval on that hose.

The Service Level Agreements are such that the network in turn will provide a loss-utilization and delay-utilization curve as a function of the arrival rate into the hose, for each time interval over which the traffic is specified. The utilization is measured with respect to the rate that had been requested over the relevant time interval. We allow ''utilization'' to be greater than 1 in the QoS specification, so that there is some flexibility for over-booking of the capacity that the different hoses of a VPN use. Figure 6 illustrates the nature of the loss and delay curves that the network may guarantee.

The hose model does not explictly require the concepts of policing or shaping. A network may choose any technique as long as it ensures that the loss and delay curves are satisfied.

4.2. QoS Support within VPNs

A Virtual Private Network is likely to want to support multiple classes of traffic. This could be to differentiate the types of data traffic that is carried within the VPN among the hose endpoints. The different classes of traffic could also be used to allow for

----

1. This kind of SLA has been proposed by the Automotive Network Exchange (ANX) specification from the Automotive Industry Action Group [4]

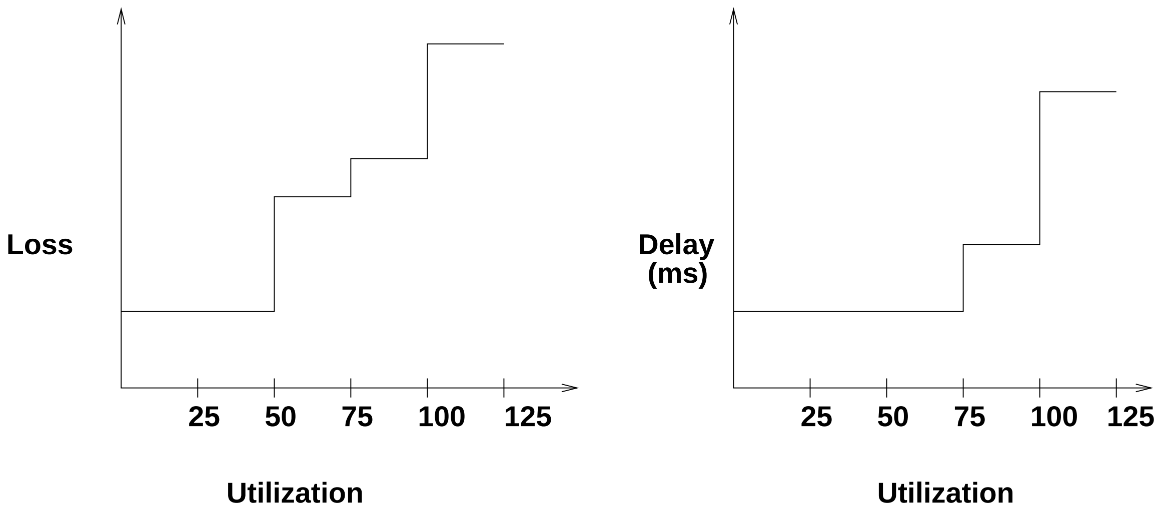

Figure 6: Example loss-utilization and delay-utilization curves

differentiation between the different types of traffic that the customer requires the VPN to carry, such as data vs. voice or other delay and loss sensitive media streams. As such, we believe there is a need to support different QoS classes within a VPN.

There are multiple ways in which VPN QoS may be managed in the network. It is envisaged that there will be some QoS requirements for the VPN as a whole and that on some cases this will suffice. In other cases there will be a requirement for different QoS guarantees within a particular VPN. There are at least two ways in which the latter more general requirement can be achieved:

- Resources are managed on a VPN specific basis. All of the different flows associated with different QoS's within a VPN have their resources allocated from the resources specific to that VPN.

- Resources are managed on an individual QoS basis. Thus, the traffic associated with a VPN for a specific QoS would use the share of resources allocated for the QoS.

The level of multiplexing gain that can be obtained may be different with each of these alternate models. We believe that the relative ease of pricing or tariffing the service from a provider's perspective may also be different based on the choice of these models being offered to a customer.

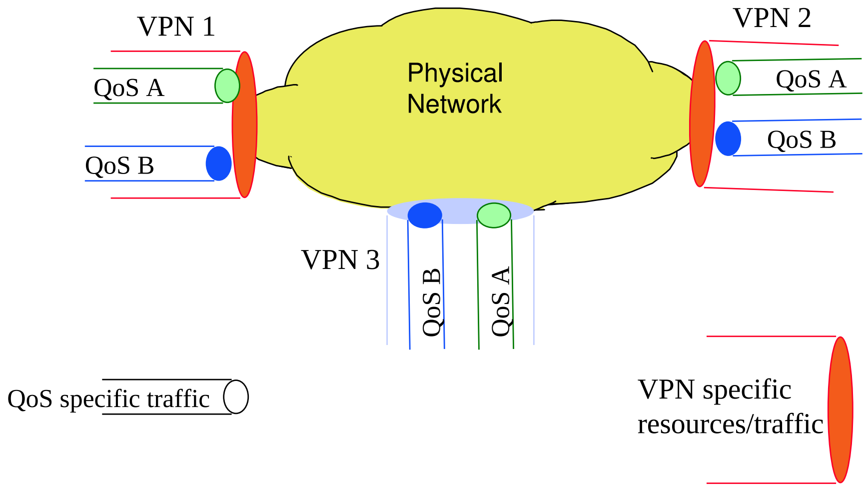

Figure 7: Model A for supporting VPN QoS: Resources allocated on a VPN by VPN basis

In the first model, Model A, shown in Figure 7, the resources associated within each individual VPN are managed locally, possibly by the customer. The traffic that has a specific QoS needs to use a share of the resources for that VPN. The performance assurances/Service Level Assurances (SLAs) would be for the overall VPN, rather than for each specific QoS within the VPN. It may be up to the customer of the VPN to ensure that the resources allocated for the VPN are shared suitably across the QoS classes within the VPN.

We see a few alternative means of implementing this model. Two alternatives, especially from the perspective of scheduling are as follows:

- Schedule only at the edges, Option 1: From a scheduling perspective, one possible way of achieving the objective is to schedule only at the edges. This technique assumes that congestion occurs primarily at the access point. Hence, different QoS can be provided by appropriately scheduling access to the VPN hose provided by the network. This fits in with the philosophy stated above which says that resources for a VPN are managed by the customer. From a QoS management perspective, the most stringent QoS across all the classes in that VPN hose has to govern the quality of service specification given to the network (and the corresponding SLA) for the VPN.

- Mark packets and schedule within the core (hierarchical scheduling), Option 2: In this technique, the network provides a single hose for the VPN, but allows the owner of the VPN to control the scheduling of resources allocated to that VPN hose by cooperating with the network for serving the different QoS classes. An end point marks packets with an identifier for the individual QoS. Whenever a scheduling decision for a QoS class within the VPN hose has to be made (i.e., a packet from the VPN is transmitted or a packet from the VPN has to be dropped), the QoS identifier is employed to make the appropriate decision. The interpretation of the QoS identifiers may be standardized or programmable on a per VPN basis. The understanding of the QoS identifiers have to be communicated at least at the time the VPN is configured with the network, so that the relative importance of the QoS class is incorporated in the scheduling by the network. However, the advantage is that this form of scheduling can be customized on a VPN by VPN basis. This approach is a little more flexible than the previous approach, because the network doesn't strictly manage resources solely on the basis of a VPN. Thus, the resources available for a QoS class of a VPN do not have to be available from within the resources allocated to that VPN only. However, the network has to still ensure the SLAs for the VPN, and needs to ensure the isolation of traffic from one VPN to another, when needed.

Queueing for these two options for Model A, would likely be based on the tuple (VPN_i, QoS_j).

With Option 1 of Model A, one may think of adopting different scheduling approaches at different points: Scheduling such as priority, weighted fair queueing (WFQ) or a combination of priority and WFQ may be used at the edge of the network. The core may use a simpler fair allocation policy for each VPN. We believe that the network would only be required to meet the loss-load curve SLA for the VPN.

The second model, Model B, shown in Figure 8, is to have resources managed for the QoS class, across all the VPNs. Thus we use hoses with different QoS guarantees for traffic for each QoS class. The traffic specification is thus made for each of the QoS classes. In addition, there may be a "broad" traffic specification for an individual VPN. The network manages the traffic across all tuples of (QoS classes, VPNs). However, the resources that a particular flow of a VPN for a given QoS class gets is not exclusively from the resources "allocated" for that VPN. The resources are allocated on

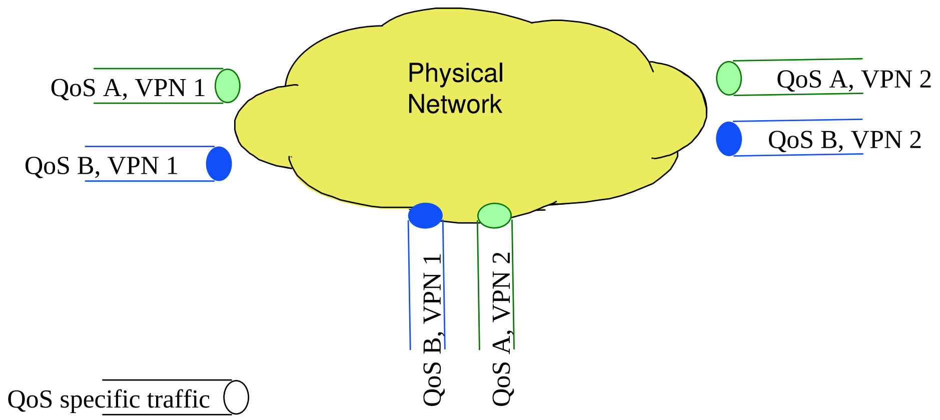

Figure 8: Model B for supporting VPN QoS: Resources allocated for a QoS class, potentially across all VPNs Further, policing and admission control may be done for an individual VPN.

an individual QoS class basis, from the perspective of the customer. There is the potential for over-subscription of an individual VPNs resources, by the customer. In this technique, each origin of the VPN has a set of hoses with different QoS characteristics. A packet with a given QoS requirement is mapped onto the appropriate hose. Queueing for this option would likely be based on just the QoS class, QoS_j.

It is our opinion that the Option 2 in Model A and Model B are preferable over the Option 1 of Model A, especially from the perspective of obtaining a higher multiplexing gain. However, the choice between the two Models, A and B is unclear. We propose to examine the performance implications between the two choices. The two approaches can be evaluated using the following metrics:

- Ease of specification: A customer may know the aggregate traffic requirements but may not precisely know the requirements of each class of traffic. Hence, it may be easier for a customer to specify a single VPN hose rather than several hoses. This suggests the use of Model A.

- Ease of Provider Offerings: A provider may find it easier to offer a VPN hose of a certain capacity and SLA, from the point of view of pricing, marketing and other potentially non-technical reasons. Using Option 1 of Model A appears attractive for this reason.

- Trust of Customer With Model A, the customer is trusted at some level to ensure that the traffic of the different QoS classes are scheduled appropriately. This is to ensure that the QoS characteristics of the class are met. There is more potential for non-technical issues being raised when the desired QoS are not met, even though the overall SLA for the VPN hose is met.

- Multiplexing gains: Another argument that has been made is that it seems that the multiplexing gains will reduce with the hierarchical scheduling approach (Model A). This is because:

  * The QoS requirements of the single hose will be governed by the applications with the most stringent requirements. Note that in either model, the traffic specification for each VPN hose and for each QoS class have to be known to almost the same level of detail.

  * The multiplexing gain across applications with similar QoS requirements is reduced.

- Protection: Hierarchical scheduling offers greater protection. Thus, using Option 2 of Model A appears attractive.

- Implementation cost: Hierarchical scheduling (Option 2 in Model A) requires that the identity of a hose (VPN) be known within the network. Furthermore, Model A also assumes a queue per hose. Thus, it may have higher implementation cost. Model B appears to offer better scalability in terms of forwarding. However, it moves the complexity of the admission control and resource management functions into the network.

A challenge is to come up with a way of providing heterogeneous QoS which has most of the desirable features of the both approaches.

5. Implementation of Hoses

There are a range of possibilities for the implementation of hoses in the provider's network.

The first example is for a particular hose to be implemented using a set of "pipes" between the various hose end-points in the network.

Each of these pipes offer both a connectivity abstraction between hose end-points as well as the underlying performance abstraction on top of which the hose performance abstraction is constructed.

A hose may also be thought of, straightforwardly, as a source tree (as in a source-based multicast tree) between an originating hose end-point and all of the destination hose end-points. The source tree would be only for the purposes of modeling the performance between the hose end-points, but not for the forwarding of data sent from a hose end-point to another hose end-point. The capacity of the branches of the tree may be setup in a way that is commensurate with the required capacity in the provider's network to carry the traffic between hose end-points. However, the data is sent as unicast packets between an originating hose end-point and a destination hose end-point. The interconnection amongst all of the hoses of a VPN would be achieved using a mesh of source trees, each originating at a distinct hose end-point for the VPN.

A set of hoses could also be modeled using a set of sink trees, each of the sink points being the destination hose end-point of the VPN. All the source hose end-points of the VPN are the leaves for each of the sink trees. This is a model that may be suitable when implementing hoses within the MPLS framework.

5.1. Implementing Hoses in an Integrated Services Framework

One possible way of providing Quality of Service for flows belonging to a VPN is using the IntServ framework of services. Either the Guaranteed Service [5] or Controlled Load Service [6] could be used to provide the resource management needed for a given VPN, based on the level of commitment desired for the VPN.

We assume that the connectivity for the purposes of this discussion exists between all the hose points of the VPN. In a pure IP environment, this implies that we have the ability to at least setup IP in IP tunnels between all the hose points.

Within the IntServ framework, a signaling protocol is used to allocate resources between selected hose points on an as needed basis. Signaling to establish or change the reservations between hoses is done by nodes within the provider network, however, the events that trigger such signaling might be signaling messages from the customer network. With the hose model the SLA with the provider allows the customer to send traffic between the originating hose and any other hose. A customer can therefore establish an IntServ reservation between any two end-points in its VPN. The signaling used for such a reservation (e.g., RSVP) might then trigger signaling in the provider network to change the reservations associated with the hose in question. Note that there is no need to negotiate with the service provider each time the capacity needed between two end-points points changes, but interpreting these customer signaling messages might be a convenient way for the provider to efficiently manage the resources associated with a VPN. When the customer has multiple pipes for which resources are requested, then the constraint that the capacity negotiated with the service provider for the hose plays a role. As described earlier, the constraint of the sums of the rows and the sums of the columns being below the pre-defined value has to be satisfied at all times. The service level agreement that the customer has with the service provider applies across all the pipes emanating from a hose point on an aggregate basis.

The main distinction in the manner we use the Integrated Services model is that the resources that are nailed up between hose points is for all the flows (where an individual flow is between a (source,destination) IP address and port number pair) for the VPN arriving at the hose point and destinated to a particular hose point. In the conventional way that the Integrated Services model is used, each individual end-to-end flow has a reservation associated with it. On the other hand, the model we propose here for using the IntServ model is for reservations to be associated with the pipe while in the provider network. The pipe potentially carries several flows over the pipe associated with the VPN. This requires us to find ways for selected routers to handle the reservation requests from individual flows. At the network edge, where the hose point enters the provider's network, all the flows belonging to the VPN are aggregated and associated with the pipe. The reservation request through the core would then be for allocating resources for the ‘‘pipe’’. This could be done either through aggregation of reservations for a set of source and destination addresses, or having the addresses of the peer hose points of the pipe in an encapsulation header of the reservation request.

The soft-state approach of RSVP is convenient for signaling because we can simply age out the resources allocated for a pipe when it is no longer required. When the customer is not using the pipe to carry traffic at some time, then the RSVP messages are not being sent (hence the reservations are not being refreshed). Therefore, the resources are automatically withdrawn. This is a convenient way of managing resources for the pipe. There may be other, potentially better ways of achieving this.

In the data path of the routers of the core network, all the flows belonging to this VPN are associated with the resources of the pipe. This implies that the packet classifier in the router has to have this capability. There may be many ways of doing this within the existing framework of IP. The classifier for each of the private links setup for the VPN could use prefixes for the source and destination addresses, if addresses are aggregateable. Or, an encapsulation header with the addresses of the hose end-points (source and destination addresses) could be used to identify the flow for the VPN. However, this implies that the end-systems generating traffic between two hose end-points belong distinctly to a given VPN. When this is not the case, a further identification of which VPN the two end-systems belong to is needed. This may result in our considering the use of a VPN identifier, which currently does not exist.

The IntServ model allows scheduling of packets in nodes based on an arbitrary subset of header fields. As such an implementation of the hose model using IntServ would be able to realize all the performance abstraction models of the previous section. In particular, scheduling based on both the QoS class and the VPN identified by a particular flow in the provider network will allow the most general option (Option 2) of model A to be implemented.

The Intserv model of Guaranteed Service can be used to set up a strict Constant Bit Rate Service Pipe, to model for example the service a user gets from a Virtual Private Link, in terms of the SLAs. But now it is not just for a single pipe to a specific hose point, but a more flexibile hose that gives the same performance for conforming traffic sent by the user. Using the guaranteed service, the SLA could be one where the Loss vs. Utilization curve is such that there is very little loss (and is flat) until we reach the point where the allocated capacity of the hose is reached. Thus, the way the pipe resources allocated have to be judiciously set to ensure that the customer is able to send up to the capacity of the hose. After that point, the loss is likely to be much higher - however, the SLA could be written in such a way that the loss after that point is undefined. In effect, the user sends traffic above the hose capacity on a truly best-effort basis. We anticipate that there is value in providing such a service in some cases, where applications are elastic, at least to a certain extent.

The Controlled Load Service could also be used. However, the SLAs that are applicable are likely to be much weaker, both from the point of loss and delay. We could explore this avenue in more detail in future discussions.

The SLAs that can be provided within the IntServ framework is likely to be somewhat less stringent than the SLAs that we have in Figure refqos-illustration. Further, this depends on the service that we have - whether it is Guaranteed Service or Controlled Load Service.

5.2. Realizing hoses in a Differentiated Services framework

In this section, we describe how some of the VPN QoS abstractions described in this section can be realized using the IETF diff-serv framework.

The diff-serv model assumes that scheduling decisions at routers are based on the value of the DS byte of the IP header [7]. The diff-serv working group is standardizing on a small set of DS values that imply a particular per-hop (packet handling) behavior (PHB) [8]. For example, packets marked as requiring Expedited Forwarding (EF) PHB will be forwarded with very little queueing delay at intermediate routers. These diff-serv PHB's can be used in combination with additional signaling/provisioning and traffic policing mechanisms to support some of our VPN service models.

For example, consider Model A, Option 1 in which a customer buys either a pipe or a hose from the network and implements scheduling at the VPN access point. This could be implemented in a diff-serv framework in the following way.

When the customer requests the VPN service, it specifies a set of VPN access points along with the traffic and QoS profile associated with the hose/pipe. The network then verifies that adequate bandwidth is available to meet the quantitative QoS bounds associated with the hose/pipe. In the case of a pipe, it is necessary to ensure that adequate bandwidth is available at every hop along the path between the two end points of the pipe. In the case of a hose, it is necessary to perform this check pairwise between all the VPN access points (assuming a particular traffic matrix). The bandwidth check could be implemented by either contacting a bandwidth broker that is aware of the network topology and available capacity at each hop, or by using a hop-by-hop signaling mechanism (2). For a hose, the VPN traffic matrix may change over time. Hence, the service provider would need to monitor the traffic matrix and dynamically renegotiate the bandwidth required for the VPN.

In this model, the QoS required for the pipe/hose would equal the most stringent QoS across all the traffic classes that are being multiplexed. Therefore, all packets for that VPN would be marked to require EF treatment through the core network. Policing at the

----

2. Unlike in the int-serv model described previously, the use of hop by hop signaling is used only for admission control and does not instantiate any packet classifiers or scheduling state at intermediate routers.

access points would ensure that the traffic conformed to the traffic parameters associated with the pipe or the hose.

Model B, can be implemented by marking packets at the access point based on the traffic class associated with the packet. For example, packets require bounded delay forwarding would be marked as requiring EF service. The admission control check would need to be done on a per-class basis but would use the same mechanisms, i.e. contacting a bandwidth broker or hop by hop signaling.

Model A, Option 2 is difficult to implement with vanilla diff-serv because it is not possible for a router to distinguish between packets belonging to different VPNs. Therefore it is not possible to provide different PHB's to packets from different VPN's.

5.3. Realizing hoses in an MPLS environment

This section considers how the hose performance abstraction might be implemented by means of MPLS. In particular, in this discussion a hose performance abstraction and virtually routed connectivity abstraction as defined in Section 3 is assumed. Full mesh connectivity in an MPLS environment can be provided by creating a sink tree (or label switched path (LSP) tree) to each hose endpoint, from all other hose endpoints. The hose performance model can be realised by dynamically changing the resources associated with such an LSP sink tree.

For example, a provider might use traffic measurements to determine the actual spread of traffic from a hose entry point to several hose exit points and signal on each LSP involved to reserve the required resources. From the point of view of a hose entry point, where such measurements might be done, signalling to change reservations will have to be done on each LSP originating from the entry point. The signaling to create LSPs and to reserve reservations on such paths might be combined in a single protocol.

An alternative would be where the customer network is an IntServ environment and MPLS is only used in the provider network. In this case RSVP requests from the customer site can be merged and translated into reservation requests in the MPLS network in a similar fashion to the pure IntServ realization described in Section 5.1.

Using a sink tree in this fashion to realize the hose model means that it is very simple to ensure that a hose exit point is not over commited. Reservation requests flowing towards the exit point will be merged and will only succeed if the total falls within the specified range for the hose exit.

The details of traffic management in MPLS and specifically the service catogories that MPLS will support is still under consideration. It is expected however that MPLS will be able to realize at least two of the QoS models defined in this section. Some consideration has to be given as to how far MPLS is deployed, i.e., whether the CPE router is MPLS capable or whether MPLS only starts in the provider network. In the following the assumption is that MPLS is only used in the provider network.

To support option 1 of model A, a single LSP sink tree will be created from all hose ingress points to all other hose egress points. The QoS of each sink tree will be the determined by the most stringent QoS traffic requirement to be carried by the hose.

Similarly support for model B will simply mean an LSP sink tree will be created for each QoS class in the VPN.

Support for option 2 of model A requires some form of hierarchical scheduling in the network based on both the VPN identifier and the QoS marking of individual packets. It might therefore not be possible to implement this option on some technologies on which MPLS will be implemented. Specifically, both ATM and Frame Relay have no support for class of service (CoS) indication in packet headers and will therefore not be able to support such hierarchical scheduling within an LSP. The "generic" MPLS encapsulation initially supported a 3 bit CoS field which could have been used to support this model. It is not clear however that these bits will remain for CoS use and in the latest ID they are identified as "experimental" [Rosen].

6. Security Requirements

Security services such as confidentiality, authenticity, and integrity are an integral part of VPNs. As the security of the public Internet infrastrucutre is always in question for many organizations, it becomes imperative to provide reliable security services in a VPN. A given VPN should be able to choose a subset of these features as needed.

We describe the security issues and requirements that arise in the context of the hose model we describe in this draft. However, we recognize that there would be a need to maintain distinct security associations between hose end-points as well as between end-systems.

Security features can be guaranteed in various ways. These can range from simple source address verification to full blown security based on cryptography protocols. The following are some very high level requirements.

- The appropriate level of security should be configurable on a per-VPN basis.

- The security mechanisms should work in presence of dynamic VPN membership. Different VPNs may have different requirements in terms of group dynamics. Some VPNs may want to ensure that at any given time only the members of the group at that time are able to decrypt the group's communication. Other VPNs may not care if an entity that was initially member of the VPN but has subsequently dropped off, is able to decrypt the group's communications. Such policies should be configurable for each VPN.

6.1. Security Model in the Context of Hoses

The draft [1] discusses various aspects of security for a VPN. IPSec is recommended for securing the packets flowing between the edges. However, most of the mechanisms described assume edge to edge model for security. A limitation of using IPSec in the edge to edge model is its scalability. The number of security associations increases at a rate of O(N**2), when N is the number of edges participating in a particular VPN. This problem becomes pronounced when an edge router belongs to multiple VPN's and has to manage a large number of security associations. This involves both storage and protocol overhead for negotiating and maintaining the security associations.

The scalability problems could be addressed by using a group security model for VPNs. We can model the hose end-points as being part of the same group. In this model, the members of the group (VPN) share the same keys. IPSec is still the underlying mechanism to provide security. However, all the members belonging to a particular VPN share the same keys. The edge routers would encapsulate all of the traffic between hose end-points appropriately. Instead of using a <src, dst, spi> tuple (spi is the IPsec security parameter index) to identify the security associations used to provide security services, a unique ID, VPN-ID, could be used for all of the communication between the hose end-points of a given VPN. In this context, the ''spi'' could be interpreted as the VPN-ID. The encapsulating header would also allow us to identify the source and destination hose end-points, which has the added benefit of providing the necessary information for resource management.

The group security model proposed here is functionally similar to multicast security where a VPN-ID is used instead of a multicast address. We would need a method of identifying the members of the group (i.e., the current set of hose end-points that are part of a given VPN.) The membership may be determined through simple configuration as a starting point. However, the group membership model for VPN's is simpler than multicast security model because of the following reasons.

- The edge routers (hose end-points) joining and leaving a VPN will not be as dynamic as the hosts joining and leaving a multicast group. This allows us to make certain design decisions such as centralized control.

- The key management issues such as rekeying, key distribution is simpler as the number of edge routers belonging to a VPN is bounded.

-

6.2. Issues

The group security model that we are proposing has a set of issues associated with it which should be the subject of further discussion.

- The model works well when the group membership changes are relatively slow, and infrequent. For the case where users dialup to join the VPN, the model of security described here may be limited. If the router to which the user dials-in has to dynamically decide to participate in a given VPN or not based on the user that has dialled-in, our security model has to be enhanced to support such dynamic changes in the group membership.

- Inter-ISP security issues are not addressed here and should be a subject for futher study. Our model assumes a single administrative domain.

- This model does not provide non-repudiation services. There may be a limited benefit of providing non-repudiation service at the network layer.

- As the model assumes shared keys, it is possible for one edge router to spoof the other. We do not have the ability to authenticate the source of each encapsulated packet as being from a particular edge router (hose endpoint). This is not an issue as long as all the network edge routers are in the same adminstrative domain and are trusted. The issue of inter-ISP security may not allow us to make this assumption, and should be a subject for futher study.

7. Summary

This document presented a QoS framework for IP based VPNs which will be applicable to various VPN connectivity models that have been proposed elsewhere. Addressing the performance aspects of IP based VPNs will be crucial to enable IP based VPNs to be offered as an alternative for private line VPNs. The emphasis of the proposed framework is performance service abstractions which simplifies the task of the customer in terms of specifying the QoS requirements of the VPN. It is expected that in addition to service differentiation per VPN there will be a requirement for a variety of performance guarantees within a VPN. Various models of how this can be achieved were presented. Finally, potential realizations of the framework were briefly discussed for three different technologies.

References

[1] B. Gleeson, A. Lin, J. Heinanen, and G. Armitage, ‘‘A framework for ip based virtual private networks.’’ draft-gleeson-vpn-framework-00.txt, September 1998. Work in progress.

[2] K. Muthukrishnan and A. Malis, ‘‘Core ip vpn architecture.’’ draft-muthukrishnan-corevpn-arch-00.txt, October 1998. Work in progress.

[3] D. Jamieson, B. Jamoussi, G. Wright, and P. Beaubien, ‘‘Mpls vpn architecture.’’ draft-jamieson-mpls-vpn-00.txt, August 1998. Work in progress.

[4] ANX, ‘‘Anx release 1 draft document publication.’’ Automotive Industry Action Group, 26200 Lahser Road, Suite 200, Southfield, Michigan 48034, 1997.

[5] S. Shenker, C. Partridge, and R. Guerin, ‘‘Specification of guaranteed quality of service.’’ RFC 2212, September 1997.

[6] J. Wroclawski, ‘‘Specification of the controlled-load network element service.’’ RFC 2211, September 1997.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, ‘‘An architecture for differentiated services.’’ draft-ietf-diffserv-arch-02.txt, October 1998. Work in progress.

[8] K. Nichols, S. Blake, F. Baker, and D. L. Black, ‘‘Definition of the differentiated services field (ds field) in the ipv4 and ipv6

headers.'' draft-ietf-diffserv-header-04.txt, October 1998. Work in progress.

Authors' Addresses

AT&T Labs. Research
180 Park Avenue, Florham Park, N.J. 07932

Nick Duffield
duffield@research.att.com
Phone:+1 973 360 8726

Pawan Goyal
goyal@research.att.com
Phone:+1 973 360 7036

Albert Greenberg
albert@research.att.com
Phone:+1 973 360 8730

Partho Mishra
partho@research.att.com
Phone:+1 973 360 8783

K. K. Ramakrishnan
kkrama@research.att.com
Phone:+1 973 360 8766

Jacobus E. van der Merwe
kobus@research.att.com
Phone:+1 973 360 8225

Nortel Networks
3 Federal St, BL3-03
Billerica, MA 01821

Naganand Doraswamy
naganand@baynetworks.com

Shantigram Jagannath
jagan@baynetworks.com