An Introduction to Virtual Private Networks

Luís Henrique Maciel Kosmalski Costa[1,2]
Luis.Costa@lip6.fr

[1] LIP6 – Université Pierre et Marie Curie
4, place Jussieu
75252 Paris Cedex 05 - France

[2] Grupo de Teleinformática e Automação – GTA
Universidade Federal do Rio de Janeiro – COPPE/EE
P.O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brasil

1 Introduction

Virtual Private Networks (VPN) are an evolving technology which is currently treated by several academic and commercial projects. VPNs are mainly used for closed group communication within a controlled and secure network environment, without needing to construct a private network. Thus a primary argument for the deployment of VPNs is the economical advantages in relation to dedicated networks.

Unfortunately, multiple definitions of a Virtual Private Network proliferate in the market place. It is difficult to give a closed definition of a VPN. Nevertheless, we find that a quite general and complete definition is provided in [1]:

"A VPN is a communications environment in which access is controlled to permit peer connections only within a defined community of interest, and is constructed through some form of partitioning of a common underlying communications medium, where this underlying communications medium provides services to the network on a non-exclusive basis".

A simpler definition is also presented:

"A VPN is a private network constructed within a public network infrastructure, such as the global Internet".

We defined that the simplest definition of a VPN derives from the main concept of VPNs, the ability to provide closed group communication. Other VPN characteristics can be derived from the *properties* associated with that communication, such as the security level and Quality of Service (QoS) guarantees.

1.1 VPN motivations

Maybe the strongest motivation of VPNs relies in the economics of communications. A collection of virtual networks implemented on top of a common physical communications infrastructure is cheaper than the construction of the equivalent collection of discrete networks, each for a single client.

Another interest of VPNs is, of course, the ability to virtually make invisible some parts of an organization's network to external observers. This aspect of communications privacy can play an important role for organizations. Depending on the level of security demanded by the organization, the simple abstraction of discretion and network obscurity of a VPN may suffice. However, for higher levels of user security a corresponding requirement for strong security of access and data transmission within the network is required, and additional mechanisms should be implemented.

The VPN abstraction is also interesting in providing closed group communication. VPNs could even be thought as support of the same functionality of multicast communication, but within a controlled and secure environment. One may also think of the use of multicast communication within a VPN, which looks like a two-level group communication. The applicability of VPNs in this area is promising.

Another important motivation of VPNs is the intrinsic ability to partition network resources. By means of resource partitioning, it is possible to provide some kind of QoS guarantees to applications. Network resources may be reserved, or provided, in a per-VPN basis. Network resources may include buffers in routers, bandwidth on links, etc., and even address space might be thought as a shared resource (depending on how VPNs are implemented, address spaces can overlap).

1.2 VPN requirements

From the definition and motivations of VPNs, one can enumerate several requirements for VPN construction [2, 3]. Any VPN has a minimal set of capabilities and may have optional features according to the requirements of the user. The main point is that for a VPN to work, it must be possible to map the user services to corresponding capabilities in the network infrastructure.

To define a VPN, an administrative mechanism is needed for designating membership in the VPN. A host may belong to one or more VPNs, as well as the global Internet. The ability to dynamically change the host membership is desirable. In the case of an enterprise, it may need to be able to signal to the provider the addition of new hosts to the VPN (e.g., a dial user). In general this should be done transparently to the VPN. However the VPN may be informed of user additions and deletions because of security issues. The set of members of a VPN may be identified: a VPN identifier is proposed in [4].

Regarding security, different capabilities may be required from the VPN depending on the user service. The minimal ability asked from a VPN is to guarantee some kind of isolation between the virtual network and the exterior world, as derived from the definition of a VPN. Depending on the level of security required, user connectivity may be defined including a variety of security mechanisms as IPSec [5]. Security may be requested discretely by end-user hosts or the VPN may enforce a mandatory security policy.

The VPN may provide QoS support. This can be done in two different ways: users may signal its QoS requirements with RSVP [6] (or other signaling method), IP precedence bits or what ever, or the VPN may internally assign QoS requirements to be mapped to the transmission infrastructure. For QoS to be guaranteed, the infrastructure may explicitly support QoS requests, or it may be projected in such a way that it consistently provides adequate QoS (with a certain level of confidence).

An enterprise may demand a specific level of availability from the service provider for the VPN service. This can also be understood as a QoS requirement and may be achieved by redundant links that are visible to, or hidden from, the user enterprise.

Other optional capabilities may also be defined, relative to addressing, load sharing, frame sequencing, MTU support, etc. [2].

1.3 Administrative model

There are two basic models of administration and management of VPNS. In the first category, the end user organization designs and operates the VPN, for example with end-user access through the public dial network. In the second model, a service provider has contractual responsibility for designing and operating the VPN in response to specified user requirements. Nevertheless, VPNs are not limited to these two models. Since the concept of VPN is tied to private communication but also to resource reservation (and partitioning), one can imagine constructing VPNs within a single organization network, to ensure a certain QoS or data security to a group of users. Several variants of VPNs may be imagined [3].

This work presents an overview of the VPN research area. Section 2 briefly reviews the main technologies of constructing VPNs. Section 3 discusses the QoS issues in VPN implementations. Section 4 presents the main research projects in the area of VPNs. Section 5 presents some final remarks.

2 VPN Implementations

There are several types of VPNs, and depending on the functional requirements, there may be also multiple methods of constructing VPNs. The selection of a specific VPN implementation includes issues like the security level needed, scalability relative to growing the VPN, as well as the complexity of implementing and maintaining the VPN (how automated is VPN creation, and how dynamic is the structure of the VPN?).

We can divide the several types of VPNs according to the layer in the TCP/IP protocol stack where the VPN is implemented. In this way, we can have network-layer, link-layer and transport/application-layer VPNs, as proposed in [1].

The types of VPNs can also be divided into "peer" and "overlay" VPNs, according to the model of implementation used. In the peer model, the network layer path computation is done on a hop-by-hop basis. Each node in the data forwarding path is a peer node with the next-hop node. Traditional routed networks are examples of the peer model. Alternatively, in the overlay model the network layer forwarding is not done hop-by-hop, but via "cut-throughs" in the link-layer, communicating an entry node "directly" with an edge node on the other side of a large cloud. The overlay model is exemplified by ATM and Frame Relay implementations. In many cases this model can provide better performance, but there are scaling issues concerning its utilization when a great number of edge nodes is to be connected in a mesh fashion.

We briefly describe in the following sections the characteristics of the main VPN implementations, beginning by network-layer VPNs (route filtering and tunneling) followed by link-layer VPNs.

2.1 Route filtering

Route filtering, also referenced to as "controlled route leaking", consists simply of controlling route propagation in such a way that only certain networks receive routes for other networks which are within their own community of interest. This is a peer model, since a router within a VPN site establishes relationship with a router in the VPN provider's network, instead of edge-to-edge communication with other sites of the VPN.

While the common underlying Internet generally carries the routes for all networks connected to it, this architecture assumes that only a subset of such networks form a VPN. Given the lack of explicit knowledge of reachability to locations other than the members of the VPN, privacy of services is directly achieved by the inability of VPN members to respond to messages originating from outside the VPN.

Route filtering VPNs are conceptually simple, but present some problems and are considered as a primitive method of constructing VPNs. The use of partial routing information is very subtle to problems of misconfiguration. It supposes a common routing core, which imposes problems to VPN addressing, since the different VPN address spaces can not overlap.

There is, however, a more scalable and less suitable to human misconfiguration approach to route filtering, the use of BGP Communities [7]. The use of the BGP communities attribute allows a VPN provider to mark BGP NLRIs (Network Layer Reachability Information) with a community attribute. Configuration control allows route information to be propagated in accordance with a community profile.

2.2 Tunneling

Sending specific parts of the network traffic through a tunnel is another method of constructing VPNs. The most common mechanisms are GRE (General Route Encapsulation) [8], L2TP (Layer Two Tunneling Protocol) [9] and PPTP (Point-to-Point Tunneling Protocol) [10]. DVMRP [11] tunnels can also be used: one may even view the MBone as a kind of worldwide VPN.

Tunneling is an overlay model, thus scaling issues arise and are dependent on the utilization of point-to-point or point-to-multipoint tunnels and how these tunnels are used to construct the VPN.

2.2.1 GRE tunnels

GRE tunnels[8] are generally configured between an ingress and an egress router in the network. The packets forwarded through the tunnel are encapsulated with a GRE header and placed in the tunnel with the address of the tunnel end-point as destination. In the end-point the GRE header is popped and the packet continue to be forwarded in the original IP way. GRE tunnels are generally point-to-point but there are some vendor implementations that support point-to-multipoint configurations.

The architectural concept of tunneling is to create VPNs as a collection of tunnels across a common host network. Each tunnel connects the point of attachment to this common network to remote points from the same VPN. The routing in the VPN is isolated from routing in the common network. The several VPNs within the common host network may use the same address space without problems of overlapping. The tunnel can also encapsulate different protocol families.

However, there are problems with GRE tunnels regarding the administrative overhead and scaling to large number of tunnels. GRE tunnels must be manually configured. A complete mesh of tunnels with a large number of peering adjacencies can result in a negative effect on routing efficiency.

2.2.2 Virtual Private Dial Networks

There are several technologies, proprietary mechanisms as well as standard-based mechanisms, that are able to construct Virtual Private Dial Networks (VPDNs). Nevertheless, there are two technologies that seem to grow in popularity: L2TP [9] and PPTP [10] tunnels. L2TP is in fact a mechanism originated from the combination of the earlier L2F protocol specification and the PPTP protocol [1].

L2TP is a “voluntary tunnel”, or “NAS-initiated tunnel” (Network Access Server or dial access server). It means that when a user dials to a NAS, this server creates a tunnel to a predetermined end-point in the network, possibly based on a local configuration profile.

PPTP has a “voluntary tunneling” model, or client-initiated. It permits clients to configure and establish point-to-point tunnels to arbitrary located PPTP servers, without the participation of the NAS. The tunnel may terminate in any PPTP server in the network, given that this server is reachable through classical routing.

In fact, PPTP and L2TP have many similarities, the main difference between them residing in the “tunneling model” adopted. This tunneling model may be the decision factor when choosing one of the two technologies according to the needs of the application. The main issue is whether the network provider has the control of the PPP session to be established or not.

2.3 ATM and Frame Relay VPNs

To construct a conventional private data network, dedicated circuits from a public carrier are combined with additional private communications infrastructure. In general the large communications (public) infrastructure uses some kind of time-division and/or frequency-division multiplexing to create a dedicated circuit, where there is synchronization of the data clock of the sender and receiver. This clocking rate is fixed by the capacity of the dedicated circuit.

A link-layer VPN tries to provide the same functionality of these conventional private data networks, while achieving economies of scale by the use of a common switched public network infrastructure. In this way a collection of VPNs share the same infrastructure. In general these networks operate at Layer 3 or higher in the OSI layered model, while the infrastructure is represented by Frame Relay or ATM. The main difference from conventional private data networks is that there is no synchronized data clock between the sender and the receiver, nor is there a dedicated transmission path. In general, the sender has no knowledge of the available capacity of the virtual circuit, so sender and receiver may use adaptive clocking of data. The sender can adapt its transmission rate according to the requirements of the application and signaling received from the network and the receiver.

A public switched WAN (ATM or Frame Relay-based) is very flexible. The client is unaware whether the provider is actually able to guarantee the contracted service at all times under all possible conditions. It is a task of the service provider the provision of sufficient network resources and the admission control of connections in the network to guarantee a certain level of QoS. On the other hand, the client is able to transmit data at a rate higher than the contracted. The excess information units (cells) are marked (as DE - Discard Eligible in Frame Relay or by setting the CLP (Cell Loss Priority) bit in ATM). In times of congestion, these marked cells are firstly discarded. The service provider may project the network to provide a specific QoS level to the carried VPNs.

There are scaling issues with ATM and Frame Relay VPNs, related to the configuration and provisioning of new virtual circuits, as well as routing issues. A complete mesh of virtual circuits between all pair of end-points may not scale, while on the other hand a partial mesh can lead to suboptimal routing. These are scaling issues common to the overlay model found also with GRE tunnels.

2.3.1 Multi-Protocol Over ATM

The idea of constructing VPNs through the use of MPOA (Multi-Protocol Over ATM) [12] is similar to other "cut-through" methods, that is, to use a switched link-layer to enable all "Layer 3 nodes" to be a single hop away from one another.

In this model, edge routers determine the path in the ATM network since they know which egress router packets must be forwarded to. Therefore packets are forwarded on a Virtual Connection (VC) specific to an egress router. However, since egress routers cannot use ARP on the ATM cloud, an IP to ATM address resolution server is needed.

One advantage of MPOA is the use of dynamic circuits, while some other VPN implementations rely on statically configured structures. The idea is to use MPOA to control the creation of edge-to-edge VCs dynamically. There are two problems with this model. The first is the dependence on the ATM technology which poses problems with hybrid networks. Second, there are scaling issues as with other "cut-through" (and overlay) models, for example, routing sub-optimality when there is no full-mesh connectivity.

2.3.2 Multi-Protocol Label Switching

Most of the network architectures can be divided into two models: the use of network-layer routing and per-packet switching or the use of link-layer circuits and per-flow switching. MPLS (Multi-Protocol Label Switching) [13] is an hybrid architecture that mixes these two approaches. The idea of MPLS VPNs [14, 15] is to address the scaling issues presented by other models by using a single routing environment with VPN labels, as packet labels are necessary to activate per-VPN routing tables.

MPLS makes the Layer 2 switching environment visible to routing by adding Layer 3 (IP routing) functionality to ATM switches for example. However, the MPLS proposal is quite generic to support a variety of link-layer technologies and heterogeneous Layer 2 switching environments. MPLS is performed by using a label swapping forwarding structure. Packets entering the MPLS cloud are assigned a label based on local forwarding decision. Packets are then "switched" in the MPLS cloud by checking the incoming label that determines the next hop and the outgoing label, in a local label-indexed table. These tables are constructed by the use of the IP routing protocol in conjunction with a Label Distribution Protocol [16].

This lightweight encapsulation based on the use of labels together with the notion of boundary-determined transit paths provides some of the mechanisms needed to support VPNs. MPLS VPNs are formed by the constrained distribution of routing information (a way to form VPNs and control connectivity between them), the use of VPN Identifiers (that concatenated with IP addresses provides unique addresses) and the use of MPLS to forward packets along the constructed routes [17].

In [18], a method for implementing MPLS VPNs is proposed that uses the BGP protocol for distributing routing information. This method views the MPLS cloud (or backbone) as a set of P and PE routers (service Provider and service Provider Edge routers). PE routers are connected to CE devices (Customer Edge devices - often a switch or router). CE devices connect the MPLS backbone to the site network. Each PE router maintains a forwarding routing table per site; packets are forwarded only to sites that have at least one VPN common with the originating site. Since it is desirable that the address spaces of (non-intersecting) different VPNs can overlap, the routing protocol has to distribute multiple routes to the same destination (address prefix in the case of BGP). The proposal in [18] is to use a new address family (named VPN-IPv4), since the

BGP Multiprotocol Extensions [19] allow BGP to carry routes from multiple address families. Figure 1 shows an overview of the BGP/MPLS VPN framework.

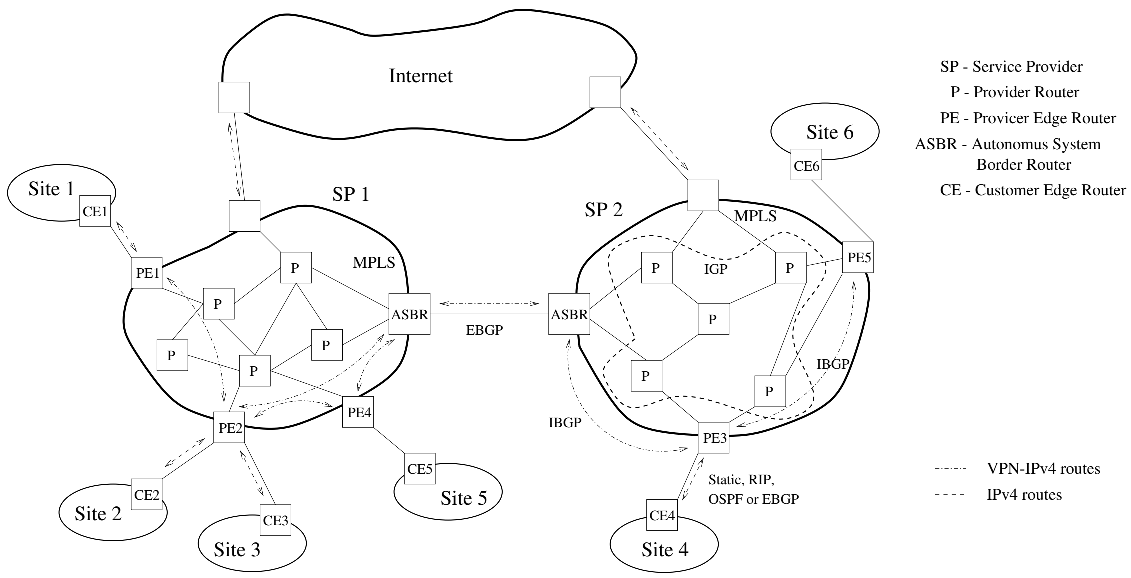

Figure 1: BGP/MPLS VPN architecture.

Scalability issues are considered in the architecture. P routers do not store any VPN routes, by means of a two-level labeling scheme in MPLS [18]. PE routers maintain one routing table per site and store routes just for VPNs that are present in the sites directly connected to it. In figure 1, router PE1 stores just one routing table for site 1 while router PE2 stores two routing tables, one associated with the Site 2 the other for Site 3. Routing table for Site 1 stores routes just for sites which have VPNs in common with Site 1. In this way the communication between two sites that have no VPN in common is prevented and on the other hand VPNs that have no site in common may have overlapping address spaces. Each PE routing table is marked with the associated VPN (or VPNs), and the PE router distributes these routers just for other PE and ASBR (Autonomous System Border Routers) routers that have sites within common VPNs connected to it. Thus VPN-IPv4 routes are exchanged just between PE routers, they are not propagated to the customer sites neither to the Internet backbone.

BGP route reflectors, if used, can be partitioned among VPNs so that each partition carries routes for only a subset of the VPNs supported by the service provider. The same applies to ASBRs, present in the case where VPNs span more than one service provider. No component has to store routes for all VPNs, which indeed improves scalability.

2.4 Virtual Local Area Networks

Virtual Local Area Networks (VLAN) [20] may be viewed as a particular type of VPNs. VLANs were developed as an alternative solution to using routers to contain broadcast traffic. The VLAN technology was recently standardized by the IEEE (standard IEEE 802.1q).

A LAN can be defined as a broadcast domain and a LAN segment as a collision domain. LAN segments are connected through bridges or switches, elements that do not forward collisions (as opposite to hubs and repeaters). LANs are connected through routers, elements that do not propagate broadcasts. VLANs allow a network manager to logically segment a LAN into different broadcast domains without routers. Bridging software defines which workstations belong to the broadcast domain and routers are only used to communicate between two VLANs.

VLANs can be constructed by means of explicit tagging or implicit tagging. In explicit tagging a VLAN identifier is added to the data (in [20], the tag headers for Ethernet, Token Ring and FDDI are presented). In implicit tagging data is not marked: the VLAN from which the data came is defined based on information like the port on which the data arrived. This kind of approach is less flexible, especially relating to host mobility.

Membership may be defined by the port (e.g. of a bridge), by MAC address, protocol type, IP subnet address, or even by the application type. Any combination of these methods supposes that the bridges have a coherent database with the VLAN membership.

The main arguments for the deployment of VLANs relate to performance gains especially when the network traffic consists of a high percentage of broadcasts and multicasts. There is also an economical reason for using VLANs: using bridges instead of routers to implement broadcast domains is likely to be cheaper.

2.5 Transport and Application Layer VPNs

Although there is no theoretical impeachment to the deployment of transport and application-layer VPNs, this is not very common. The most common method of implementing virtualization is to use encryption services at either layer. For example, encrypted e-mail transactions and authenticated DNS (Domain Name System) are cited in [1].

The Transport Layer Security protocol (TLS) [21] should also be mentioned. It was recently published as a Request for Comments by the IETF. The main idea of TLS is to provide privacy and data integrity between two communicating applications.

3 QoS Provision in VPNs

The definition of a VPN as a "private network" leads directly to performance issues, in other words, how one can associate some QoS characteristics to a specific VPN. There are several works in progress that address these QoS requirements in VPNs [22, 23, 24, 25].

A QoS framework for IP-based VPNs is presented in [22], the following reviews its main concepts. As VPNs are supposed to replace networks constructed with private lines, the same kind of service offered by these networks would be expected from VPNs. Therefore, in addition to closed user group communication and secure data transmission, VPNs have to offer performance guarantees in terms of delay and bandwidth for example.

Two connectivity models are identified. The first is the Virtual Private Link, where the VPN is constructed from a set of pipes. The tunneling techniques presented earlier fit into this model. The second is the Virtually Routed Network. Customer routers connect to provider routers and traffic isolation between VPNs is achieved by some kind of encapsulation within the provider network. Techniques like MPLS VPNs fit into this model.

A performance oriented service description is defined by means of two performance service abstractions. In the first one, a customer asks the service provider a pipe between two VPN sites with certain QoS requirements. To construct a VPN, a customer has to know the traffic matrix of the network to demand the pipes between all pair of sites in the network.

The "hose" performance abstraction is introduced to reduce the information that the customer has to pass to the provider. In this abstraction, a VPN end-point has to inform the "outgoing" and “incoming” maximum data rates that this site is expected to send and receive from **all** other VPN end-points (optionally a set of them). A hose can be thought as a pipe where one end-point is defined and the other end-point may be any other hose in the VPN. The customer does not have to provide a complete traffic matrix nor the spread of this traffic to other end-points in the VPN. A hose is exemplified in figure 2. The user site just inform its outgoing and incoming data rates, and the hose end-points. The contract defines that $x + y + z = 5Mb/s$, but it does not define the exact values of x, y and z, as would be the case if the pipe abstraction was used.

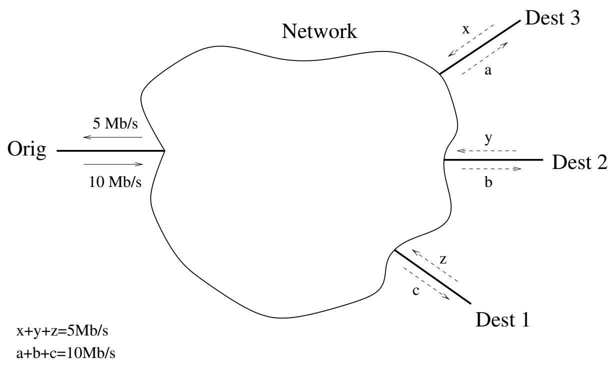

Figure 2: A hose.

In the hose abstraction, the connectivity between all endpoints (or a specified set) is supposed. The hose abstraction gives the customer freedom to send traffic between each pair of end-points provided that the traffic in and out of the end-points does not exceed the agreed capacities. In these conditions the service provider is expected to meet certain Service Level Agreements (SLA), such as bounded delay and packet losses. Therefore problem is to the service provider to guarantee these SLAs under all possible conditions all the time. Another problem is related to the Virtual Private Line service, where the transmission rate between each pair of nodes is supposed to be known. The Virtually Routed Network model fits better to this abstraction.

Regarding QoS provision there are several possibilities. One may define a single set of QoS requirement for the VPN. The VPN construction reflects these requirements, possibly with bandwidth reservation by the service provider. However we may think also of different QoS levels within the same VPN. There are at least two ways to achieve this more general model:

- Model A - To manage the resources on a VPN-specific basis. Resources for different flows, with different QoS requirements, are allocated from the resources reserved to the specific VPN. Here there are two possibilities: 1 - to do scheduling only at the edges, with different scheduling services for different QoS flows (it is possibly done by the customer) or 2 - to mark packets at the network entry, and in this case the network serves differently different packet types. In any case resources are managed and reserved for a specific VPN.

- Model B - To manage resources on a QoS-basis, across all VPNs. The traffic specification is thus made for each of the QoS classes (a VPN with three classes of QoS has to provide three traffic specifications). The network threats differently each of the classes for a specific VPN, or in other words, the network manages the traffic across all tuples (QoS class, VPNs). In this technique, each origin of the VPN has a set of hoses with different QoS characteristics.

3.1 Implementing QoS

3.1.1 Integrated Services

One of the possibilities to implement QoS within a VPN is to use the Integrated Services (IntServ) [26] framework. Either the Guaranteed Service or the Controlled Load Service can be used to support the resource management for a VPN, given its QoS requirements. In the IntServ framework, there is a signaling protocol (Resource Reservation Protocol - RSVP) [6] that is used to make reservations between two end-points.

In the hose model, the dynamic reservation mechanism provided by RSVP can be used to implement reservations between any two end-points in a dynamic fashion. It remains to the service provider to guarantee the SLAs given that the hose specification is respected. The capacity negotiated for the hose plays a role when there are several pipes reserved for a customer.

The difference between the hose model and the IntServ framework is that in the first the reservation is made for a pipe, that is, for all traffic between a pair of nodes; in IntServ the reservation is made on a per-flow basis. A pipe can potentially carry several flows of the same VPN. A resource aggregation might be used in this case.

3.1.2 Differentiated Services

In the Differentiated Services (DiffServ) [27] model scheduling decisions at routers are based in the DS byte of the IP header. The values of DS define different per-hop behaviors (PHB). These DiffServ PHBs may be used with additional signaling and/or provisioning to support QoS in VPNs.

When a customer constructs a VPN, it passes to the network a set of end-points to be connected along with the traffic characteristics and QoS profile of the specific hose/pipe. The network has to check if there are available resources to meet the requirements associated with this hose/pipe. In the case of a pipe, resources have to be available across a path between two end-points. In the case of a hose, it is necessary to make such a check between all VPN access points. In this case one should assume a particular traffic matrix. Defining such a traffic matrix is a problem of the service provider, considering the dynamic characteristic of a hose. The bandwidth check may be made by a bandwidth broker (aware of the network topology) or by a hop-by-hop signaling mechanism, that in this case is just for admission control. This approach differs from the IntServ model: there is no creation of packet classifiers or scheduling state at intermediate routers.

If resources are managed in a VPN-basis with scheduling made by the customer (model A - option 1), the QoS required from the network to a specific VPN would have to be the more stringent QoS requirement of that VPN. Model A option 2 would be difficult to implement since the DiffServ routers do not know how to distinguish traffic from different VPNs by default.

Model B, where resources are partitioned on a per-QoS basis, can be implemented by marking packets at the access point based on its traffic class. The admission control would be made in a per-class basis. In fact, the construction of the resource-management models proposed in [22] should be better studied since one may imagine many variations of them.

3.1.3 MPLS

MPLS seems to be a technology well suited to support QoS in VPNs, especially considering the hose abstraction and a virtual routed connectivity. A full mesh can be realized in MPLS by means of a sink tree (in fact, a Label Switched Path (LSP) tree) to each hose end-point from all other end-points. The performance model may be realized by dynamically changing the resources allocated to each LSP tree.

Using a sink tree makes it easy to the provider to check if the committed resources for a hose are respected, since the reservation requests flowing towards the exit point will be merged and only succeed if the total falls in the range for the hose exit.

Another possibility is to use the IntServ model at the customer network. In this case the RSVP requests from the customer site are merged and translated to reservation requests in the MPLS network, in a way similar to the pure IntServ realization.

The traffic management and service categories that MPLS will support are still in discussion [13]. The generic MPLS encapsulation initially supported a 3-bit CoS (Class of Service) field, but it is not clear that these bits will remain for CoS use since in the latest specification they are identified as experimental.

4 VPN Projects

There are some projects currently in progress that threat the problem of QoS provisioning for VPNs. We make a brief review of the most important ones.

Cisco Systems [23] proposes a QoS-VPN architecture that is based on some features of a previous QoS-provision product. The proposal includes packet classification (using a committed access rate), bandwidth management (including policing, shaping and bandwidth allocation), congestion avoidance and continuity of packet priority over Layers 2 and 3 by means of MPLS.

Packet classification is made (by now) using the ToS (Type of Service) field of the IP header and packets can be marked following different criteria such as IP addresses, TCP/UDP port numbers, etc. Bandwidth allocation is realized by WFQ, in one of two ways, class-based or flow-based. Congestion avoidance is implemented by WRED. These components have to be combined to provide specific QoS guarantees to VPNs. A VPN-specific "signaling facility" is not present.

Other projects are concerned also with the VPN construction by the user. These projects try to automate the generation of VPNs and some of them address QoS issues. It is the case of the VNS (Quality of Service Virtual Network Service) [24] project.

VNS proposes an architecture where the users will be provided an application that enables them to construct virtual networks. VNS is supposed to be responsible for partitioning the network resources (bandwidth, router buffers, etc.) and allocating them to individual virtual networks. From the point of view of QoS provision, some characteristics were defined: routers should be able to identify flows and virtual networks. A packet scheduling algorithm (HFSC - Hierarchical Fair Service Curve) at routers is responsible for delivering link sharing and supporting real-time service. The link layer is supposed to be ATM, and each virtual-link of the network is associated with a virtual circuit. In this way the bandwidth of virtual links may be specified.

The Supranet concept is presented in [25]. A Supranet is a virtual network that has some characteristics or properties. The approach employed to provide QoS is to construct VPNs on top of a Integrated Services network. A Supranet is a network-layer VPN constructed by means of tunneling. The proposed architecture has a double-stack layered structure, with one stack for real-time services and the other for non real-time ones.

There are also other projects that are mainly concerned with automating the creation of virtual networks, such as the Genesis Kernel [28] and the X-Bone proposal [29]. In the Genesis architecture, programmable networks are envisioned as capable of creating "child" virtual networks. This is a new paradigm for automating the creation, deployment and management of the network architectures. One of the main motivations of the project is the emergence of open programmable networks.

The X-Bone is a system for the automated deployment of overlay networks, that may be viewed as virtual networks constructed by means of encapsulation tunnels. The idea is to automate the creation of overlay backbones such as the 6-Bone and to facilitate testing of new protocols by creating an isolated environment. The X-Bone uses a graphical interface and multipoint control channel to manage overlay deployment at the IP layer, much like teleconference sessions are controlled via the session directory (sd) tool. The X-Bone software has three basic components. The Overlay Manager is responsible for running the algorithms that acquire resources, configure them and manage them as a coherent overlay. Resource Daemons are processes running at or near resources such as routers and links, they are responsible for keeping information about the state of that resource and coordinating its sharing by multiple overlays. A Multicast Control Protocol uses a two-layer IP system similar to that used by sd/sdr. A common announcement channel (the first layer) is used for resource requests. Once the resources for an overlay are acquired, a control channel (second layer) is created for this specific overlay. This control channel reaches only that set of resources.

5 Final Considerations

The VPN-concept is quite general and to give an exact definition of a VPN is not easy. However one is able to identify some characteristics that any "VPN-service" should provide: closed user group communication and data security. Economies of scale appears as a strong argument for the deployment of most of the VPN solutions, provided by the VPN ability to share a common underlying infrastructure in a transparent manner.

VPNs may be implemented by a large variety of technologies, ranging from dial-lines to ATM or MPLS links, using multiple mechanisms to support data security as well as some level of QoS guarantees. Some of the protocols and technologies that are likely to be used to implement VPNs are in the area of current research, this is the case of the IPSec protocol and the MPLS technology. Thus we may expect several new implementation approaches in the near future.

Especially regarding QoS provision in VPNs, there is a lot of work to be done as some of the possible support-technologies of VPNs are still under discussion, likely MPLS and the IntServ and DiffServ architectures.

References

[1] P. Ferguson and G. Huston, *What is a VPN.* Cisco Systems and Telstra Internet, Apr. 1998. White paper available at http://www.employees.org/ ferguson/.

[2] H. C. Berkowitz, *Requirements Taxonomy for Virtual Private Networks.* Netarchitecture Mentoring, Feb. 1999. Internet-draft.

[3] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis, *A Framework for IP Based Virtual Private Networks.* Shasta Networks, Telia Finland, Bell Labs ans Ascend Communications, Feb. 1999. Internet-draft.

[4] B. A. Fox and B. Gleeson, *Virtual Private Networks Identifier.* Lucent Technologies and Shasta Networks, Feb. 1999. Internet-draft.

[5] S. Kent and R. Atkinson, *RFC2401: Security Architecture for the Internet Protocol*. Network Working Group, Nov. 1998.

[6] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, *RFC2205: Resource Reservation Protocol (RSVP) - Version 1 Functional Specification*. Network Working Group, Sept. 1997.

[7] R. Chandra, P. Traina, and T. Li, *RFC1997: BGP Communities Attribute*. Network Working Group, Aug. 1996.

[8] S. Hanks, T. Li, D. Farinacci, and P. Traina, *RFC1701: Generic Routing Encapsulation (GRE)*. Network Working Group, Oct. 1994.

[9] W. M. Townsley, A. Valencia, A. Rubens, G. S. Pall, G. Zorn, and B. Palter, *Layer Two Tunneling Protocol (L2TP)*. Cisco Systems, Ascend Communications, Microsoft Corporation, and Redback Networks, Feb. 1999. Internet-draft.

[10] K. Hamzeh, G. S. Pall, W. Verthein, J. Taarud, W. A. Little, and G. Zorn, *Point-to-Point Tunneling Protocol (PPTP)*. Ascend Communications, Microsoft Corporation, 3Com, Copper Mountain Networks and ECI Telematics, Feb. 1999. Internet-draft.

[11] D. Waitzman, C. Partridge, and S. Deering, *RFC1075: Distance Vector Multicast Routing Protocol*. Network Working Group, Nov. 1988.

[12] J. Heinanen, *RFC1483: Multiprotocol Encapsulation over ATM Adaptation Layer 5*. Network Working Group, July 1993.

[13] E. C. Rosen, A. Viswanathan, and R. Callon, *Multiprotocol Label Switching Architecture*. Cisco Systems, Lucent Technologies and IronBridge Networks, July 1998. Internet-draft.

[14] J. Heinanen and E. Rosen, *VPN Support with MPLS*. Telia Finland and Cisco Systems, Mar. 1998. Internet-draft.

[15] J. Heinanen, B. Gleeson, and A. Lin, *MPLS Mappings of Generic VPN Mechanisms*. Telia Finland and Shasta Networks, Aug. 1998. Internet-draft.

[16] E. Gray, Z. Wang, and G. Armitage, *Generic Label Distribution Protocol Specification*. Lucent Technologies, Nov. 1997. Internet-draft.

[17] D. Jamieson, B. Jamoussi, G. Wright, and P. Beaubien, *MPLS VPN Architecture*. Nortel, Aug. 1998. Internet-draft.

[18] E. C. Rosen and Y. Rekther, *BGP/MPLS VPNs*. Cisco Systems, Dec. 1998. Internet-draft.

[19] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, *RFC2283: Multiprotocol Extensions for BGP-4*. Network Working Group, Feb. 1998.

[20] S. Varadarajan, *Virtual Local Area Networks*. Ohio State University, Aug. 1997. Tech. report available at http://www.cis.ohio-state.edu/ jain/cis788-97/virtual_lans/index.htm.

[21] T. Dierks and C. Allen, *RFC2246: The TLS Protocol Version 1.0*. Network Working Group, Jan. 1999.

[22] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, J. E. van der Merwe, N. Doraswamy, and S. Jagannath, *A Performance Oriented Service Interface for Virtual Private Networks*. AT&T Labs and Nortel Networks, Nov. 1998. Internet-draft.

[23] C. Systems, *Quality of Service for Virtual Private Networks*. White Paper available at http://www.cisco.com/warp/public/779/largeent/vpne/qsvpn_wp.htm.

[24] L. K. Lim, H. Zhang, and A. Mankin, *Implementing Quality of Service Virtual Network Service (VNS) on CAIRN*. project site: http://www.cs.cmu.edu/ hzhang/VNS.

[25] L. Delgrossi and D. Ferrari, "A virtual network service for integrated-services internetworks," in *7th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, May 1997.

[26] R. Braden, D. Clark, and S. Shenker, *RFC1633: Integrated Services in the Internet Architecture: An Overview*. Network Working Group, June 1994.

[27] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *RFC2475: An Architecture for Differentiated Services*. Network Working Group, Dec. 1998.

[28] A. T. Campbell, H. G. D. Meer, M. E. Kounavis, K. Miki, J. Vicente, and D. A. Villela, "The genesis kernel: A virtual network operating system for spawning network architectures," in *IEEE International Conference on Open Architectures and Network Programming - OPENARCH99*, Mar. 1999.

[29] J. Touch and S. Hotz, "The X-bone," in *IEEE GLOBECOM'98 - Internet Mini-Conference*, 1998.