Institut National Polytechnique de Grenoble

THESIS

for obtaining the degree of

DOCTOR OF THE INPG

Specialty: ”Computer Science: Systems and Communications”

presented and publicly discussed
by
Lina AL-CHAAL

Defensed on February 2, 2005

Title:

Dynamic and Easily Manageable Approach for Secure IP VPN Environments

CIFRE Thesis prepared in
the INRIA Rhône-Alpes and Netcelo S.A.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N^o assigned by the library

|_|_|_|_|_|_|_|_|_|_|

THESIS

for obtaining the degree of

DOCTOR OF THE INPG

Speciality: ≪ Computer Science: Systems and Communications ≫

prepared at INRIA Rhône Alpes, Planete project-team
in the executive of **The Doctoral School ≪ MATHEMATIQUE, SCIENCES ET TECHNOLOGIE DE L'INFORMATION ≫**

presented and publicly discussed

by

Lina AL-CHAAL

Defensed on February 2, 2005

Title:

Dynamic and Easily Manageable Approach for Secure IP VPN Environments

Director of thesis :
Andrzej DUDA

Committee in charge

Prof.	Jacques MOSSIERE	President
Prof.	Abdelmadjid BOUABDALLAH	Reporter
Dr.	Ahmed SERHROUCHNI	Reporter
Prof.	Andrzej DUDA	Director of thesis
Dr.	Vincent ROCA	Supervisor
Dr.	Michel HABERT	Supervisor
Dr.	Marco CARUGI	Examiner

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N°attribué par la bibliothèque

|_|_|_|_|_|_|_|_|_|_|

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité: ≪Informatique: Systèmes et Communications≫

préparée à l'INRIA Rhône Alpes, projèt Planète

dans le cadre de **l'Ecole Doctorale ≪ MATHEMATIQUE, SCIENCES ET TECHNOLOGIE DE L'INFORMATION ≫**

présentée et soutenue publiquement

par

Lina AL-CHAAL

le 2 Février 2005

Titre:

Une Approche Dynamique et Facilement Administrable pour des Environments IPVPN Sécurisés

Directeur de thèse :
Andrzej DUDA

Jury

Prof.	Jacques MOSSIERE	Président
Prof.	Abdelmadjid BOUABDALLAH	Rapporteur
Dr.	Ahmed SERHROUCHNI	Rapporteur
Prof.	Andrzej DUDA	Directeur de thèse
Dr.	Vincent ROCA	Encadrament scientifique
Dr.	Michel HABERT	Encadrament scientifique
Dr.	Marco CARUGI	Examinateur

Acknowledgment

I want to express my sincere acknowledgment to all those who have encouraged me, with their support and suggestions, all the long way of my Phd.

I would like to extend a special thanks to Prof. Xavier Rousset de Pina for encouraging me to pursue my studies in France.

Special appreciation is given to my supervisors Dr. Vincent ROCA and Dr. Michel HABERT who provided me with insights, resources, direction, and continuous feedback, which was invaluable and helped me gain much needed perspective. I would like to thank Prof. Andrzej DUDA for his advice and encouragement throughout my thesis.

All my gratitude to prof. Abdelmadjid BOUABDALLAH and Dr. Ahmed SERHROUCHNI for the time they spent reading and commenting my thesis, and for the report they wrote. And I also would like to thank Dr. Marco CARUGI and Prof. Jacques MOSSIERE for taking part in the committee and for their invaluable notes.

I would also like to thank all those who helped, advised, and gave fruitful discussions during this work. Many thanks also are to all my colleagues in Netcelo S.A. and the members of the Planete project in Inria Rhone Alpes. I also wish to express appreciation to Mr. Ciaran DEIGNAN for his support and assistance with the technologies required for implementing this study.

My sincere appreciation and gratitude to my parents for their support and encouragement during the entire period of this thesis.

Finally, my deepest gratitude goes to the departed soul of my grandfather for his unconditional love and his spiritual support throughout the last two years of his life.

Our Achievements: Published Papers and Implementations

Magazine Papers

1. Lina Al-Chaal, Vincent Roca, and Michel Habert, **Managing and Securing Web Services with VPNs**, International Journal of Web Services Research JWSR, http://www.idea-group.com/journals/details.asp?id=4138, **submitted work, after the invitation of Dr. Liang-Jie Zhang, the founding Editor-in-Chief of the JWSR**.

Conference Papers

1. Lina Al-Chaal, Vincent Roca, and Michel Habert, **Offering a multicast delivery service in a programmable secure ip vpn environment**, Fourth International Workshop on Networked Group Communication - NGC'02, Boston, USA, October, 2002.

2. Lina Al-Chaal, Vincent Roca, Ayman El-Sayed, and Michel Habert, **A VPRN Solution for Fully Secure and Efficient Group Communications**, IEEE Symposium on Computers and Communications - ISCC'2003, Kemer-Antalya, Turkey, July, 2003. An extended version is available as INRIA Research Report number RR-4799, INRIA, Rhône Alpes, France, April, 2003.

3. Lina Al-Chaal, Vincent Roca, and Michel Habert, **Managing and Securing Web Services with VPNs**, the second IEEE International Conference on Web Services - ICWS 2004, San Diego, California, USA, July, 2004.

4. Lina Al-Chaal, Vincent Roca, and Michel Habert, **De l'Utilisation des VPNs pour l'Administration et la Sécurité des Services Web**, 3ème Conférence sur la Sécurité et Architectures Réseaux - SAR 2004, La Londe, Cote d'Azur, France, June 2004.

Software (Perl/Java/C/Linux)

1. **Internet VPN Group Management Protocol IVGMP implementation** .

2. **HAVA implementation for load balancing for VPN devices**.

3. **Implementation of VPN web services**.

Abstract

The increased use of the global Internet and IP-based applications have paved the way for service providers to offer new network services to their customers. The telecommunications' world reflects and embodies a fundamental shift in how service providers do business without depending on ISP core networks that offer service oriented networks that bundles value-added services (such as packet telephony and e-commerce) on top of transport services. Also as network services become increasingly complex and network-intensive, customers want to tap the outsourcing potential of service provider services for cost savings. Thus to offer cost effective network services, service providers will be challenged to deploy their services over public networks like the Internet and heightened by changes in business customer networks. In this dissertation, we introduce a CE-based VPN approach that offers different management network services in behalf of customers. This approach shifts the management hassles from customer's side to the VPN service provider. Yet by using this approach service providers have only to care about managing customers edge devices that are the gateways to the customers' networks. This approach is a centralized solution, where everything, including VPN creation, deployment and membership management, is under the control of a single Management Operation Point (MOP). This thesis focuses on three key aspects: management, dynamism and security. We also investigate the use of our approach to offer group communication services (e.g. Multicast service) and to manage and secure web services, yet other fields of applications are possible, like the dynamic management of firewalls or VoIP.

Résumé

Dans cette thèse, nous présentons une approache dynamique et facilement administrable basée sur la technologie des réseaux privés virtuels IP, connus sous le nom "IP Virtual Private Networks" (IPVPNs). Avec cette approche les machines terminales, les serveurs et/ou des routeurs de bordure s'organisent automatiquement en une topologie de recouvrement grâce à laquelle des données sont diffusées. Cette approche est une solution centralisée, où tout, y compris la gestion des services d'adminstration comme l'adhésion au groupe et la création de topologie VPN, est de la responsabilité d'un centre d'opérations du système, appelé le "Network Operation System" (NOS).
Nous utilisons notre approache pour établir un service alternatif de communication de groupe qui déplace le support de multipoint depuis les routeurs vers les extrémités. Nous appliquons cette approache au moyen d'un nouveau protocole IVGMP (Internet Virtual Group Management Protocol) pour réaliser un VPN supportant la diffusion multipoint sur des réseaux de transport IPV4.
Nous étudions ensuite l'utilisation d'un nouveau protocole HBM qui fournit un service de distribution multipoint au niveau applicatif afin d'établir un service entièrement sécurisé mais efficace de communication de groupe entre plusieurs sites avec l'aide de notre approche VPN. Nous montrons que HBM et notre approche sont naturellement complementaires et conduisent au concept de réseau privé virtuel routé (VPRN).
La deuxième contribution concerne les services web sécurisés. Nous définissons aussi un modèle d'architecture hybride qui s'applique aux Services Web. Nous montrons comment ce modèle correspond bien à la nature dynamique des services web, offre un service d'administration facilement intégrable, et améliore grandement la sécurité des services web grâce à l'utilisation de VPNs. La troisième contribution de ce travail concerne des techniques de partage de charge et d'amélioration de performances, telle l'addition de liens virtuels redandants qui évitent la partition de la topologie en cas de panne d'une passerelle VPN.
Nous concluons ce travail avec une discussion de différents applications de notre approche IPVPN, en particulier des services qui peuvent être fournis moyennant quelques modifications.

Acronyms

AAA	:	Authentication, Authorization and Accounting
AES	:	Advanced Encryption Standard
AH	:	Authentication Header
BGP	:	Border Gateway Protocol
CA	:	Certificate Authority
CE	:	Customer Equipment
CM	:	Centralized Multicast
CPE	:	Customer Premises Equipment
DES	:	Data Encryption Standard
DIN	:	Distributed Intelligent Network
DMZ	:	Demilitarized Zone
DVMRP	:	Distance Vector Multicast Routing Protocol
ED	:	Edge Device
EAP	:	Extensible Authentication Protocol
EGP	:	Exterior Gateway Protocol
ESP	:	Encapsulating Security Payload
ESP	:	Enterprise Service Provider
FAI	:	Fournisseur d'Accès Internet
FEC	:	Forward Error Correction
GCS	:	Group Communication Services
GRE	:	Generic Routing Encapsulation protocol
HBM	:	Host Based Multicast
HTTP	:	Hypride Text Transport Protocol
ICMP	:	Internet Control Message Protocol
IGMP	:	Internet Group Management Protocol
IGP	:	Interior Gateway Protocol
IKE	:	Internet Key Exchange
IP	:	Internet Protocol
IPsec	:	Internet Protocol security
ISAKMP	:	Internet Security Association and Key Management Protocol
ISP	:	Internet Service Provider
IVGMP	:	Internet VPN Group Management Protocol
IPLS	:	IP LAN Service
IPv4	:	Internet Protocol version 4
IPv6	:	Internet Protocol version 6
L2F	:	Layer 2 Forwarding
L2TP	:	Layer 2 Tunneling Protocol

L2VPN	:	Layer 2-based VPN
L3VPN	:	Layer 3-based VPN
LAN	:	Local Area Network
MBone	:	Multicast backBone
MOP	:	Management Operation Point
MPLS	:	Multi-Protocol Label Switching
MPPE	:	Microsoft's Point-to-Point Encryption
MS-CHAP	:	Microsoft Challenge Handshake Authentication Protocol
MSEC	:	Multicast SECurity
NAS	:	Network Access Server
NAT	:	Network Address Translation
NOS	:	Network Operation System
NSP	:	Network Service Provider
OSI	:	Open Systems Interconnection Reference Model
OSPF	:	Open Shortest Path First
P	:	Provider router
PE	:	Provider Edge
PGP	:	Pretty Good Privacy
PIM	:	Protocol Independent Multicast
PIM-DM	:	Protocol Independent Multicast-Dense Mode
PIM-SM	:	Protocol Independent Multicast-Sparse Mode
PIM-SSM	:	Protocol Independent Multicast-Source Specific Mode
POP	:	Point Of Presence
PPP	:	Point-to-Point Protocol
PPTP	:	Point-to-Point Tunneling Protocol
PPVPN	:	Provider Provisioned Virtual Private Networks
QoS	:	Quality of Service
RAS	:	Remote Access Server
RDP	:	Remote Desktop Protocol
RFC	:	Request for Comment
RP	:	Rendez-vous Point
RTT	:	Round-Trip Time
RTP	:	Real-Time Transport Protocol
SA	:	Security Associations
SDP	:	Session Description Protocol
SIP	:	Session Initiation Protocol
SLA	:	Service-Level Agreement
SNMP	:	Simple Network Management protocol
SOAP	:	Simple Object Access Protocol
SP	:	Service Provider
SSL	:	Secure Socket Layer
TBCP	:	Tree Based Control Protocol

TCP	:	Transmission Control Protocol
TTL	:	Time To Live
UDP	:	User Datagram Protocol
VC	:	Virtual Circuit
VC	:	Virtual Channel
VNOC	:	Virtual Network Operation Center
VPDN	:	Virtual Private Dial-up Network
VPLS	:	Virtual Private LAN Service
VPN	:	Virtual Private Network
VPRN	:	Virtual Private Routed Network
VPWS	:	Virtual Private Wire Service
VRF	:	VPN Routing and Forwarding
VSI	:	VPN Switching Instance
WAN	:	Wide Area Network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

We start this chapter by giving a quick glance on the history that leads to the design of the Virtual Private Networks, or VPNs. We define them and introduce their benefits in general. We summarize the management obstacles that most current VPN models have to face, and the essential points for a well-managed system. These considerations lead us to quickly introduce the need for a well-managed dynamic system over IPVPN. Finally we introduce the goals and the organization of this thesis.

1.1 VPN Definition

The idea behind VPNs is not new [50], but VPNs went a step further by offering the opportunity to create dynamic links over different transmission media [38]. In addition to offering a better flexibility and scalability, a VPN provides a service functionally equivalent to a private network, using the shared resources of a public network. Typically a VPN uses an existing infrastructure to establish secure communications between trusted associates in a very cost effective way. This infrastructure can be either an IP backbone from a Network Service Provider (NSP), or the whole Internet. In places, the infrastructure may be frame relay or ATM carrying IP. In other words, a VPN simulates the behavior of dedicated WAN links over leased lines.

1.1.1 A Bit of History

At the beginning, corporations first leased multiple circuits between geographically dispersed sites to extend their private networks. *Physical circuits*, leased from carriers, connected pairs of sites to create a point-to-point private communications infrastructure. Then the *circuit-switched private networks* have been used for communicating between distant sites. The circuit costs included a one-time setup charge and a periodic recurring charge based on the bandwidth. Because providers supplied fixed bandwidth, customers had exclusive access to leased bandwidth, whether or not they actually used it.

The VPN notion has been around as soon as one started to talk about Virtual Connections/Channels/Circuits, or VCs. The concept behind the VCs is the logical separation of customer communication channels thanks to a virtual connection. The provider can then multiplex many distinct point-to-point VCs over a single physical infrastructure.

The *Frame Relay and ATM technology* are the most important approaches using the VCs. They allowed providers to sell less expensive private network services through economies of scale. Both Frame Relay and ATM protocols, classified as Level 2 protocols, provided remote-site point-to-point connectivity without the need for dedicated bandwidth between sites, since VCs were overlaid on the physical infrastructure.

Just in the last decade the *Virtual Private Network VPN* technology started to overwhelm the landscape of the telecommunication world. VPN services were typically not sensitive to distance charges and were much less expensive than leasing dedicated circuits. The use of VPNs enabled enterprises to use the Internet infrastructure to deploy their own private networks. Thus VPNs have become an essential part of business communications between employees, customers, and enterprise partners spread over the Internet, since VPNs enable the creation of secure connections to protect private data and resources when they travel over an untrusted network like the public Internet [18].

The Virtual Private Dial Networks (VPDNs) were the first VPN deployed over IP networks. They targeted “road warriors” who required secure access to their home network servers. In this kind of VPNs, an end user establishes a PPP session to the Internet Service Provider (ISP) gateway, and then a compulsory tunnel is established between the ISP gateway and the corporate gateway. A voluntary tunnel can also be established directly from the end user to the corporate gateway.

The IPsec VPNs were the second and the most popular VPNs providing data integrity, authentication and privacy when communicating across the Internet. We will discuss this kind of VPNs more in details in the remainder of this document.

1.1.2 A Simple Example to Better Understand VPNs

To understand the idea behind VPN let’s give an analog from our real life by considering each LAN as an island. Imagine that you live on an island in a huge ocean. There are thousands of other islands all around you, some very close and others farther away. The normal way to travel is to take a ferry from your island to whichever island you wish to visit. Of course, traveling on a ferry means that you have almost no privacy. Anything you do can be seen by someone else.

Let’s say that each island represents a private LAN and the ocean is the Internet. Traveling by ferry is like connecting to a Web server or other device through the Internet. You have no control over the wires and routers that make up the Internet, just like you have no control over the other people on the ferry. This leaves you susceptible to security issues if you are trying to connect between two private networks using a public resource.

Continuing with our analogy, your island decides to build a bridge to another island so that there is easier, more secure and direct way for people to travel between the two. It is expensive to build and maintain the bridge, even though the island you are connecting with is very close. But the need for a reliable, secure path is so great that you do it anyway. Your island would like to connect to a second island that is much farther away but decides that the cost are simply too much to bear.

This is very much like having a leased line. The bridges (leased lines) are separate from the ocean (Internet), yet are able to connect the islands (LANs). Many companies have chosen this route because of the need for security and reliability in connecting their remote offices. However, if the offices are very far apart, the cost can be prohibitively high just like trying to build a bridge that spans a great distance.

So how does VPN fit in? Using our analogy, we could give each inhabitant of our islands a small submarine. Let's assume that your submarine has some amazing properties:

- It's fast.
- It's easy to take with you wherever you go.
- It's able to completely hide you from any other boats or submarines.
- It's dependable.
- It costs little to add additional submarines to your fleet once the first is purchased.

In our analogy, each person having a submarine is like a remote user having access to the company's private network. Although they are traveling in the ocean along with other traffic, the inhabitants of our two islands could travel back and forth whenever they wanted to with privacy and security. That's essentially how a VPN works. Each remote member of your network can communicate in a secure and reliable manner using the Internet as the medium to connect to the private LAN. A VPN can grow to accommodate more users and different locations much easier than a leased line. In fact, scalability is a major advantage that VPNs have over typical leased lines. Unlike with leased lines, where the cost increases in proportion to the distances involved, the geographic locations of each office matter little in the creation of a VPN.

1.1.3 VPN Benefits

We could summarize the benefits of using VPNs as follows:

- Lower costs: combining Internet, Intranet and Extranet connectivity through the same VPN solution reduces the cost and complexity of managing multiple networks. There is no need to train or employ skilled security specialists or incur expensive operational costs.
- Extended geographic connectivity: a VPN connects remote workers to central resources, making it easier than ever to set up widely distributed global operations.
- Increased return on investment: an effective security solution significantly reduces threats and consequently decreases downtime and lost business.
- Remarkably secure: data that is sent over VPNs is confidential, requiring authorization to be received or replayed. Users can authenticate packets to establish the validity of the information, and the integrity of the data is usually guaranteed.

- Flexible: new users and sites within a VPN can be complex and time consuming.
- Easily scalable: a VPN allows customers to utilize the remote access infrastructure within ISPs. Therefore, companies can add a virtually unlimited amount of capacity without adding significant infrastructure.

1.2 VPN Management Requirements

With all networks, the complexity and the cost of management tasks increase with the number of devices, and this is also true with VPNs, perhaps even more. Each site connected to a VPN must have a gear that secures traffic before it crosses the Internet or some other public IP network that is being used as the VPN backbone. Because VPNs are so often used for remote access, as the number of VPN devices that need to be managed can range up into the tens of thousands.

Managing large corporate VPNs requires adding multiple shifts of highly trained staff that would be difficult to retain given the high demand for these skilled workers. While each element must be configured and reconfigured as a VPN grows and changes, a good management platform will mask that level of detail from the user.

Element management is not only time-consuming, but it also invites misconfigurations. The more individual commands administrators have to enter, the more likely they are to make mistakes. As more sites are added to a network, the number of relationships that need to be established among them blossoms. As such associations grow, the number of policies for a VPN can spiral out of control.

For these reasons we noticed the need for a *central policy management server* that can enable VPN policies to be set once and distributed to VPN servers at all of the other sites connected to the VPN. In this way, servers can also be configured by groups and subgroups. When a new site is added, the configuration of its server and the configuration adjustments the other servers require should be generated by the platform. So the management system or platform should not only set policies, but they should also translate them into commands to reconfigure the VPN gateways.

In general, a good management system could be evaluated regarding the following points:

- Policy: the management system should automatically convert policy changes into configuration changes that must be made at each node of the VPN.
- Configuration: the management system should automatically send configuration changes to VPN sites after these changes have been made on a central management station.
- Deployment: there should be a mechanism to distribute, configure and update software on remote PCs without someone manually setting up each machine. Updates and policy changes should be pushed automatically on remote machines.
- Full knowledge of VPN topologies: the system should be best-fit and support for tailored VPN solutions (e.g. fully meshed and partially meshed VPNs). It should also create VPN maps that depict the types of connections that have been authorized by policy settings and should pull data from existing databases to reduce the amount of time it takes to define user groups and authorizations.
- Quick reaction: the system should react quickly for any changement in the VPN topologies and in case a failure requires the deployment of new configuration policies. It should also identify the presence, location and cause of service degradations in dynamic, global networks consisting of shared resources that impact multiple customers and services.

- Monitoring: constant monitoring of the global security environment to keep pace with the emergence of new vulnerabilities and ever more sophisticated threats.

From the previous evaluating points, we derive the need for having a *dynamic management system*. Integrating a dynamic access capability into the management system will come up with a robust solution to deploy and control access privileges for large and diverse end-user populations, such as employees, corporate partners and customers. It will also provide flexibility in defining rich authentication and authorization policies for diverse user groups.
The dynamic feature of management systems minimizes the amount of time needed to spend on security management. And VPN tunnels can be created between sites by using dynamic routing to communicate network topology and link state information. As a result, enterprises do not have to worry about revising policies every time there is a change in the network and can feel confident that the VPN connection will be able to survive a failure.

1.3 Standards and Reference Models

We referred to the following standards and reference models when we built our VPN model and to the previous points mentioned in section 1.2.

1.3.1 Distributed Intelligent Network (DIN) model

The DIN model ([57] , [58]) moves network services closer to users. The model's core is a simple and reliable transport service. The management operations in this model are highly centralized with local service enforcement while wide area connections are mixed but primarily broadband. The DIN represents major structural changes in enterprise networks such as the elimination of the aggregation level in LAN hierarchy.
The benefits of DIN are (1) a lower Total Cost of Ownership (TCO) (2) and a more responsive network aligned to business needs. What is driving DIN is a complex set of competitive forces, convergence and most of all the need for business IT to be agile or adaptive to changes.
This model introduces three components:

- Logical network: for communications between entities such as users (laptops) and applications (corporate sites). This network is an overlay network over physical access and interconnection networks. Within this network there are a set of management services such as configuration, supervision, monitoring, quality of service, high availability, IP telephony and end to end Service-Level Agreement (SLA) control.

- Physical networks: there are the interconnection networks for carrying data in behalf of users.

- Information system: including laptops (personal sites) and Gateways (network sites). These equipments bring information services close to the users.

The logical network is an abstraction level between the information system and physical networks, and it creates a complete independence between them. We adopted the DIN model in our architecture since it fits well with the nature of a VPN environment where a VPN could form the logical network that separates the physical networks from the information system and emulates whatever local or wide area network connectivity customers desire.

1.3.2 Concept of tunnel broker

The Tunnel Broker idea [30] is an approach based on the provision of a dedicated server, called Tunnel Broker, that automatically manages tunnel requests coming from the users. The motivation is to reduce the management complexity of tunnels configurations. In particular, it is largely used to offer an automatic way for isolated IPv6/IPv4 host to interactively set up an IPv6-in-IPv4 tunnel to an IPv6-only network. The tunnel broker is closely associated with both a tunnel server, and a DNS service.
On a request from an isolated host, the tunnel broker assigns an IPv6 address to the isolated host from its address space, updates the DNS automatically, sends a configuration order to the tunnel server, and sends back a script to the requester. The tunnel server establishes a tunnel from the IPv6-only network to the requesting host, and executes the script on the requesting host to establish the tunnel in the reverse way. This mechanism allows people to try out IPv6 without any need of special or dedicated routing infrastructure.

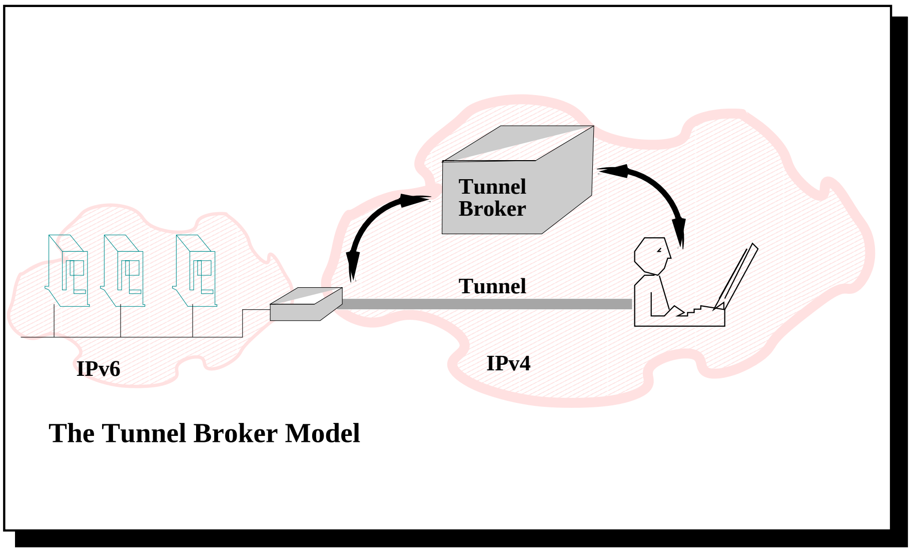


Figure 1.1. The Tunnel Broker Model

In our context, most of VPN networks are built using IPsec tunnels over the existing IPv4 infrastructure. So we adopted the concept of tunnel broker to configure and maintain those tunnels in a large scale environment where administration issues become more difficult and delicate. The concept of tunnel broker seems to be useful to stimulate the growth of VPN that provides connectivity between groups of users over the Internet. To avoid a fastidious and complex management of tunnels, we have chosen the tunnel broker model to implement a carrier-class tunnel management system transparent and automatic designed from scratch to support tens of thousands of users.

1.3.3 Policy based management

Policies can be defined as the plans of an organization to achieve its objectives. A policy is a persistent specification of an objective to be achieved or a set of actions to be performed in the future or as an on-going regular activity. The primary policy management architectural

components are the Policy Decision Point (PDP), Policy Enforcement Points (PEPs), and the policy repository, typically stored in a directory like LDAP repository.
To take its decisions, the PDP taps the repository for the rules and conditions established by the network manager and decides, based on current network conditions, how traffic and access rights should be enforced. Once the PDP has made a decision on how to treat network traffic, it communicates this behavior to the PEPs (routers, switches, and gateways) via management protocols such as SSH or Open Policy Service Protocol (COPS). Each PEP has separate network interfaces, enabling network managers to set a fine granularity of policy management rules or to create batches of rules that apply to particular applications, users, or locations.

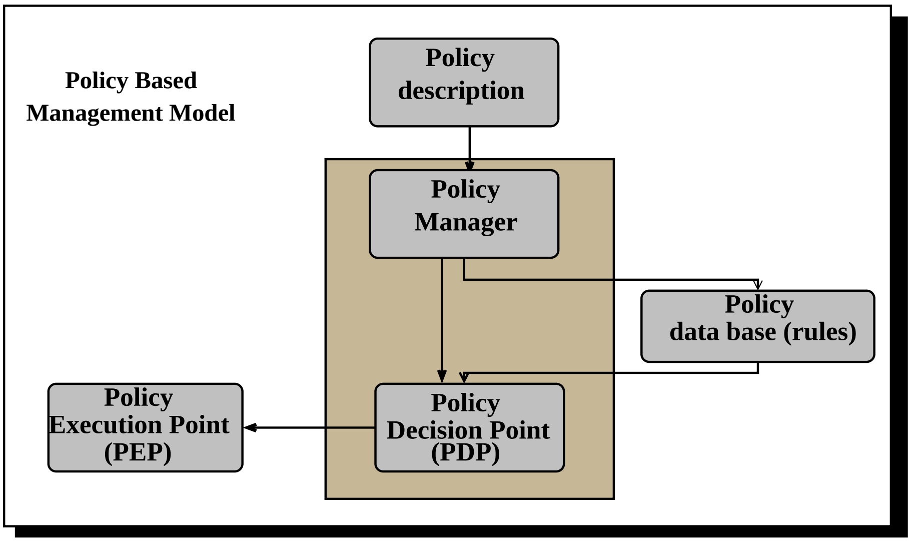


Figure 1.2. The Policy Based Management

Regarding our architecture we adopted the PDP as the central point of our system, while the PEPs represent the VPN sites. Later, we will discuss the whole components of our system more in details.

1.4 Goals of the Thesis

As we have seen, the deployment of services over the Internet is still far behind expectations. The most important obstacle is the *lack of good management model* that works transparently in behalf of clients to offer the services they need. Another obstacle is the *lack of security* over the Internet especially when it comes to secure communications between two peers that do not know each other before, but require a mutual authentication. Meanwhile there is a huge need for a management system that reacts immediately to customer changements without any need of human intervention, which is always a problem due to the risk of misconfiguration or to the delay before answering to the client needs. Therefore we introduce a well-managed system based on IPVPN, that offers security services in behalf of its customers. Our proposal also has a dynamic feature that enables VPNs to be created or changed easily, with minimal impacts to individual sites. We implemented our proposal using different programmable languages (C, Java, Perl and Shell scripts). There are various domains in which our proposal can be applied, but this thesis will focus on the following two aspects: group communication services and web services.

1.5 Organization of the Document

The rest of the document is organized into as follows:

- Chapter 2 introduces and discusses several models for building VPNs. We classify the models into several categories: based on routing information exchanged between customers equipments and providers equipments (overlay and peer-to-peer VPNs), or depending on the management's way of the VPN equipments (CE-based or Network based VPNs).

- Chapter 3 describes our dynamic CE-based VPN proposal which is a centralized solution, where everything, including clients membership management and topology creation, is under the control of a single Management Operation Point (MOP).

- In chapter 4 we describe how to adapt our proposal to offer group communication services. We describe our new protocol, Internet VPN Group Management Protocol (IVGMP), as an alternative to traditional multicast routing protocols, that forwards multicast traffic over secure VPN topology.

- In chapter 5, we investigate the use of an application level multicast protocol called Host Based Multicast (HBM) to build a fully secure and efficient group communication service between several sites. We showed that HBM and our VPN approach naturally fit with one-another and lead to the concept of Virtual Private Routing Network (VPRN). This chapter explains how the scalability can be largely improved by this new proposal.

- In chapter 6, we discuss the use of our proposal to manage and secure web services in a simple and dynamic way that fits well with the dynamic nature of web services. We show how there is a mutual need between the VPN technology and the web service technology. And we also discuss some design and performance aspects.

- Chapter 7 focuses on the problem of high availability of CE devices into VPN sites. We show how our proposal and its dynamic aspect play an essential role to rapidly overcome failure problems and to naturally offer a load balancing when the number of VPNs to which a VPN site belongs to, increases.

- Finally, we discuss some key points and conclude this thesis in chapter 8.

Part I

Overview of VPN Technologies and Models

Chapter 2

Survey of VPN technologies and their Management

Contents

In this chapter, we introduce the role of the VPN technology in the telecommunication world as well as the different VPN models that exist. We compare these technologies and highlight in particular the management challenges they raised.

2.1 Introduction

Today's service providers are experiencing an increasing demand for IP services due to the success of the Internet. More and more companies want to outsource Internet, intranet and extranet services, managed network services, and content-related services such as Web hosting, mail service, and secure remote access. As a result, the Service Provider's profitability in the telecommunication world is defined by their ability to rapidly introduce new services tailored to the specific needs of their customers.

Fortunately the evolution of telecommunication networks toward the current broadband networks made feasible the support of those services, at least in terms of offered bandwidth. The main issue that remains is to resolve the deployment, provision and control of network services.

Lately in the last decade, service providers have considered the VPN technology that offers the promising solution to managing network services in a transparent way to clients. This VPN technology plays a significant role to achieve management goals and deliver guaranteed levels of performance, reliability, leading-edge management and monitoring tools. However the VPN management issues are still in their early stages.

In this chapter, we describe the different VPN technologies currently in use. We compare them, considering their functionalities and application fields. Then we focus on the VPN management challenges and what we get by applying such technologies from the management's viewpoint. Later, in the next chapter, we will introduce our own VPN approach, see how it compares with other technologies, and the benefits it offers regarding the management issues.

2.2 ISP Network Components Terminology

In the remaining of this document, we will use the following definitions [10], that define an Internet Service Providers (ISP)'s network components:

- *Internet Service Provider* (ISP): the ISP is a company that provides dial-up or direct access to the Internet for a fee. It is the company that provides the gateway between a customer and the Internet.

- *Service Provider* (SP): the SP is a company that generates revenues for the services delivered to their customers over a network, typically the Internet. The price for such services must include the actual delivery of the service including accommodations for the cost of the infrastructure to deliver the service (e.g. hardware, software, data center, labor, IP). Since most of SPs provide connections to the Internet, they are also known as ISPs.

- *Customer Edge/Equipment* (CE): the CE device is located at the customer's side. It has an access connection with the provider's edge router, PE. Note that sometimes in the VPN terminology, we distinguish between two types of customer edge devices: the CE device and the CPE device. In the first case, the CE device is owned by the customer, but it is managed by a service provider. On the opposite, the CPE device is both provided and managed by a service provider. In this document, we will use CE to refer to both types of devices, unless otherwise mentioned.

- *Provider Edge* (PE): the PE device links several CEs to the provider network. The PE forwards traffic from CEs to other provider's routers P, depending on routing information within the PEs. There is routing information for all VPNs used by the CEs connected to that PE.

- *Provider router* (P): the P router is located at the provider's network and used to interconnect the PE routers. There is no direct connection between a CE and a P router.

- *Edge Device* (ED): the edge device may be either a PE or a CE, depending on the service demarcation point between the provider and the customer.

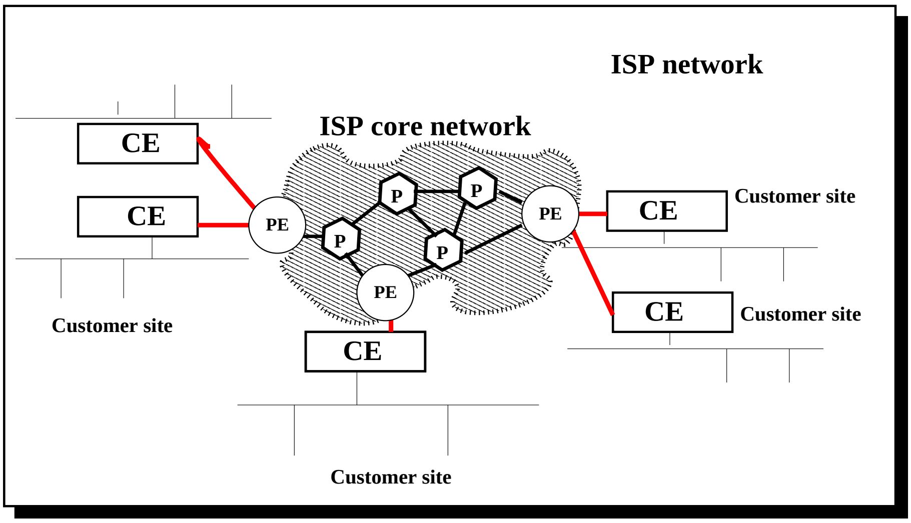


Figure 2.1. ISP Network and Elements

To avoid ambiguity, we use the term 'client site' or simply 'site' to refer to the customer network which can range from a simple personal computer (e.g. PC, laptop, PDA or mobile phone) to complex networks settled behind a CE device.

In ISP based VPNs, the category of VPNs that we are the most interested in, a client site is connected via a Customer Edge (CE) to the ISP core network. The CE is an IP router that provides a client access to the ISP network over a data link to one or more Provider Edge (PE) routers. A PE router maintains VPN routing information for each VPN and forwards the VPN traffic to a Provider (P) router. This latter could be any router in the ISP network that does not attach to CE devices.

2.3 VPN Technology in the Telecommunication World

2.3.1 VPN Working Groups

A few years ago many IETF working groups have defined and specified new VPN solutions and requirements like L3VPN, L2VPN, PPVPN and PWE3. All working groups address the same problem space, the provider provisioned virtual private networking for the end customers. They address a wide range of services without creating new protocols, but by providing new extensions to the existing ones. Here are briefly the goals of the respective working groups:

Provider Provisioned IPVPN (PPVPN) WG:
PPVPN is responsible for defining and specifying a limited number of sets of solutions for supporting provider-provisioned virtual private networks (PPVPNs). Precisely, the goals of this WG are to:

- Define a checklist of SP requirements for security, privacy, scalability and manageability
- Produce a small number of approaches, that are based on collections of individual already existing technologies, to foster the interoperability among implementations of a specific approach
- Identify gaps and shortcomings in individual approaches with regards to the SP requirements.
- Focus on at least three specific approaches including: BGP-VPNs (e.g. RFC 2547), virtual routers and port-based VPNs (i.e. where the SP provides a Layer 2 interface, such as Frame Relay or ATM, to the VPN customer, while using IP-based mechanisms in the provider infrastructure to improve scalability and configurability over traditional L2 networks)

Layer 3 VPN (L3VPN) WG:
The L3VPN goal is to define and specify a limited number of solutions for supporting provider-provisioned L3VPNs. L3VPNs basically assume that an ”‘IP in IP”’ tunneling scheme is used. To summarize, L3VPN:

- Is responsible for the standardizations: BGP/MPLS IPVPNs (based on RFC 2547), IPVPNs using Virtual Routers and CE-based VPNs using IPsec.
- Addresses the deployment of some services (e.g. Multicast, IPv6) within VPN environments.
- Addresses the deployment of VPN services in different scenarios.

Layer 2 VPN (L2VPN) WG:
The L2VPN goal is to define and specify a limited number of solutions for supporting provider-provisioned L2VPNs. L2VPNs are independent of the network protocol used by the customer. In other words, the VPN is not limited to carrying IP traffic. We will examine the use of L2VPNs more in details later. To summarize, L2VPN:

- Is responsible for the standardizations: VPLS, VPWS, IP only L2VPN.
- Works on the following items: discovering the PEs and the topology of the required connectivity, managing signaling aspects during the setup and maintenance of L2VPN circuits, defining MIBs and specifying Operation, Administration and Maintenance (OAM) extensions.

Pseudo Wire Edge to Edge Emulation (PWE3) WG:
In a L2VPN, the access at the CE looks like a private, point-to-point wire. For this reason, the initiative has been renamed Pseudo Wire Emulation End to End, or PWE3. PWE3 WG is primarily targeted at using tunnels or ”pseudo-wires” to provide a point to point type of service where the incoming encapsulation is tunneled across a provider infrastructure. Before PWE3, service providers that wanted to offer different services (such as Internet, Private Lines, Frame Relay, etc.) typically had to build multiple transport infrastructures. With PWE3 these services can be provided using a single MPLS-based transport infrastructure. The goals of this WG are:

- To develop standards for the encapsulation and service emulation of pseudo-wires.

- To pursue standardization of the framework and the service-specific techniques for pseudo-wires.
- To define network management information needed for tunnel operation and management.
- To specify the security mechanisms to be used to protect the control of the pseudo-wires.

The efforts of these working groups have led to the development of a (partly) new set of concepts to describe the VPN services offered that we are going to develop in the remainder of this chapter.

2.3.2 Tunneling Mechanisms

Tunneling Definition

The process of transferring data in a VPN for one network over another network is called Tunneling. The data to be transferred can be frames (or packets) of another protocol. The tunneling protocol encapsulates the frame in an additional header, instead of sending frame as it is produced by an originating node. The additional header is required for providing the routing information to the encapsulated payload to traverse the intermediate internetwork. The encapsulated packets are then routed between tunnel endpoints over the internetwork. The logical path through which the encapsulated packets travel through the internetwork is called a tunnel. Once these encapsulated packets reach the destination they are decapsulated to get the original data.

Briefly, the basic idea behind network tunneling is that you can take non-routable data packets and encapsulate them inside routable packets for a transmission over the Internet. At the destination the encapsulation is stripped off and the original data enters the private network as if it had come from a local source.

Tunneling makes the routed network totally transparent to users. Tunnels are used for many services, like multicasting and mobile IP, and has amazing implications for VPNs. For example, you can place a packet that uses a protocol not supported on the Internet (e.g. NetBeui) inside an IP packet and send it safely over the Internet. Or you can place a packet that uses a private (non-routable) IP address inside a packet that uses a globally unique IP address to extend a private network over the Internet.

Tunnels can be static or dynamic. Static tunnels are of little use for VPNs, because they reserve bandwidth even if they are not used. The second type does not require bandwidth reservation since they are set up as needed and torn down when they are no longer needed.

Tunneling Requirements

Tunneling requires three different protocols:

- Carrier protocol: the protocol used by the underlying network where the information travels
- Encapsulating protocol (referred later by tunneling protocol): the protocol (GRE, IPsec, SSL, PPTP, L2TP, etc.) that is wrapped around the original data
- Passenger protocol: the original protocol (IPX, NetBeui, IP) being carried

Tunneling Protocols

For a tunnel to be established, both the tunnel client and tunnel server must use the same tunneling protocol (or encapsulating protocol), like IPsec, L2TP, or PPTP [53]. Lately, a new kind of VPN, based on the Secure Sockets Layer SSL protocol, has emerged as the leading solution for remote access.
The tunneling functionality can be based on:

- Layer 2 tunneling protocol: this corresponds to the data-link layer and uses frames as their unit of exchange. PPTP and L2TP both operate at this layer and both encapsulate the payload in a PPP frame to be sent across an internetwork.

- Layer 3 tunneling protocol: this corresponds to the network layer and uses packets. The IPsec tunnel mode operates at this layer and encapsulates the packets in an additional header before sending them across an IP network.

- Layer 5 protocol: this corresponds to the session layer. SSL is a higher-layer security protocol that sits close to the application. It uses a public key to encrypt data that is transfered over the SSL connection.

Here are the descriptions of the most current tunneling protocols in use:

2.3.3 IPsec VPN Tunneling Mechanism

IPsec Definition

The Internet Protocol Security (IPsec) provides several mechanisms that aim at securing traffic on the network layer. It is a set of IETF open standards that provides cryptographic-based protection mechanisms for IP packets [49]. IPsec has the advantages of offering advanced cryptographic services and proves to be the best security protocol for LAN-to-LAN VPNs, while other security protocols work better for Host-to-Host connections [37]. Besides IPsec is now well known and integrated in many operating systems (e.g. FreeS/Wan for the Linux OS [36]).

IPsec Communication Channels

An IPsec VPN generally consists of two communication channels between the endpoint hosts:

- a *key-exchange channel* over which authentication and encryption key information is passed. The key-exchange channel is a classical UDP connection to and from port 500; and

- one or more *data channel(s)* over which private traffic is carried. The data channels carrying the traffic between the client and server use IP protocol number 50 (ESP).

IPsec Modes: Transport and Tunnel

IPsec supports two encryption modes: Transport and Tunnel. Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. At the receiving side, an IPsec-compliant device decrypts each packet.
IPsec offers the following security features:

- Packet integrity: packets are protected so that any alterations during their transmission could be detected.
- Packet confidentiality: packets are encrypted before being transmitted over the network so that only authorized entities can read them.
- Packet origin authentication: packets are protected to ensure that they are indeed from the claimed sender whose IP address is contained in the IP header.
- Protection against replay: packets are protected from being captured and resent later on.

These features are provided by IPsec through two types of security services: *Authentication Header (AH)*, which enables end-user authentication, and *Encapsulating Security Payload (ESP)*, which supports both end-user authentication and data encryption.

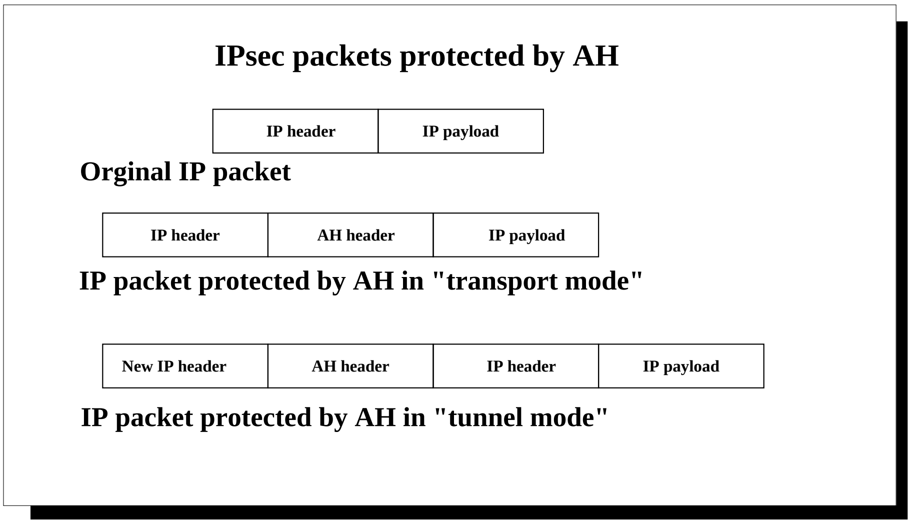

IPsec packets protected by AH
IP header
IP payload
Orginal IP packet
IP header
AH header
IP payload
IP packet protected by AH in "transport mode"
New IP header
AH header
IP header
IP payload
IP packet protected by AH in "tunnel mode"

Figure 2.2. AH Location in IPsec Packet

The IPsec modes discussed above are closely related to the function of these two core protocols. The choice of a mode does not affect the way by which each protocol generates its header, but rather, changes what specific parts of the IP datagram are protected and how the headers are arranged to accomplish this. In essence, the mode really describes, not prescribes how AH or ESP do their thing. It is used as the basis for defining other constructs, such as Security Associations (SAs).

Before either AH or ESP can be used, however, it is necessary for the two devices to exchange the secret that the security protocols themselves will use. The primary support protocol used for this purpose in IPsec is called Internet Key Exchange (IKE). IPsec parameters are communicated and negotiated between network devices in accordance with IKE. IKE involves the process of choosing the hashing and encryption methods and transferring key sets [1]. Note that for long period, Internet Security Association and Key Management Protocol/Oakley (ISAKMP/Oakley) was used to refer to IKE version 1, now IKE simply refers to both versions of IKE. More details about IKE will be given later on.

The primary strengths of IPsec-based VPN are:

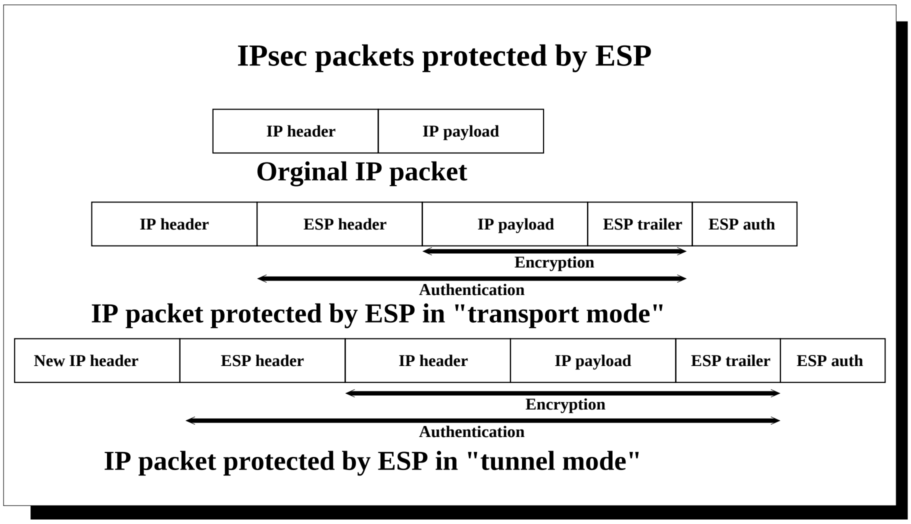


Figure 2.3. ESP Location in IPsec Packet

- *Security*: IPsec helps ensure data privacy with a flexible suite of encryption and tunneling mechanisms that protect packets as they travel over the network. Users are authenticated with digital certificates or pre-shared keys. Packets that do not conform to the security policy are dropped.

- *Ease of deployment*: IPsec enables a fast and easy deployment because it can be deployed across any existing IP network with little or no change to the existing IP network infrastructure.

- *Network Extension*: an IPsec VPN can greatly increase the service provider's reach because it can take advantage of the Internet. Therefore, the service provider does not need to invest in infrastructure outside of its existing footprint.

Encryption Mechanisms

Encryption is the process of taking all the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode. Most computer encryption systems belong to the following two categories:

- Symmetric algorithms (AKA shared-key encryption): in symmetric algorithms, both parties share the same key for encryption and decryption. The basic use of a symmetric-key requires that one knows in advance which computers will be talking to each other so that one can install an appropriate key on each one. To provide privacy using symmetric algorithms, this key needs to be kept secret. Symmetric algorithms have the advantage of not consuming too much computing power. A few well-known examples are: DES, Triple-DES (3DES), IDEA, CAST5, BLOWFISH, TWOFISH.

- Asymmetric algorithms (AKA public-key encryption): asymmetric algorithm uses a pair of keys: a private key and a public key. The private key is known only to you, while the public key is given to anybody who wants to communicate securely with you. Most encryption algorithms use a public key to encrypt data. If the sender wants to send an encrypted message to the receiver, the sender would use the receiver's public key to encrypt the data. When the receiver gets the encrypted message, the receiver would use his or her private key to decrypt the message. The private key can't be reconstructed from the public key. The idea of asymmetric algorithms was first published 1976 by Diffie and Hellmann. As the private key does not have to be shared, the risk of getting known is much smaller. Every user only needs to keep one secret key and a collection of public keys that only need to be protected against being changed. While with symmetric keys, every pair of users would need to have an own shared secret key. Well-known asymmetric algorithms are RSA, DSA, ELGAMAL.

Asymmetric algorithms seem to be ideally suited for real-world use. However, asymmetric algorithms are much slower than symmetric ones. Therefore, in many applications, a combination of both is being used. The asymmetric keys are used for authentication and after this has been successfully done, one or more symmetric keys are generated and exchanged using the asymmetric encryption. For instance, the sending computer can encrypt the document with a symmetric key, and then encrypt the symmetric key with the public key of the receiving computer. The receiving computer then uses its private key to decode the symmetric key, and then uses the symmetric key to decode the document. The sending computer insures that his message will be only seen by the computer who only owns the private key, without requiring that a secret key be exchanged beforehand. This way the advantages of both algorithms can be used. Typical examples are the RSA/IDEA combination of PGP2 or the DSA/BLOWFISH used by GnuPG.

Internet Key Exchange IKE

IKE provides a secure method of exchanging keys and negotiating protocols and encryption algorithms to use. The information negotiated IKE is stored in a Security Association(SA). The SA is like a contract laying out the rules of a connection for the duration of the SA. It describes how two or more entities will use security services to communicate securely. An SA is assigned a 32-bit number that, when used in conjunction with the destination IP address, uniquely identifies the SA. This number is called the Security Parameters Index or SPI. Briefly IKE: (1) creates the first secure and authenticated channel between two communicating entities (IKE SA). (2) creates and manages child SAs (IPSEC SA).

To tie this all together, let's look at an example. User A wants to send data to User B. User A's router (router_A) has a security policy applied with a rule that says all traffic to User B needs to be encrypted. User B's router (router_B) will be the other end of an IPsec tunnel. Router_A checks to see if an IPsec SA exists between itself and router_B. If no SA exists, router_A requests an IPsec SA from IKE. If an IKE SA exists, an IPsec SA is issued. If an IKE SA does not exist, one has to be negotiated first, with the routers exchanging information signed by a third-party CA that both routers trust. Once the IKE SA is agreed upon by the routers, an IPsec SA can be issued, and secure, encrypted communications can begin. This process is transparent to both User A and User B.

To clarify the things, let's go through the steps for setting up an IPsec connection. Basically, the steps are as follows: Set up an IKE SA. Agree upon the terms of communication and encryption algorithm. Create an IPsec SA. Start sending data. Once the IPsec client on a IPsec peer gets any interesting traffic to treat, it checks if any secure channel is already established to the traffic destination. If not the IPsec client starts negotiating an IKE phase 1 exchange. The main purpose of IKE *Phase 1* is to authenticate the IPsec peers and to set

up a secure channel between the two IKE peers, called the IKE SA to enable IKE phase 2 exchanges. The Diffie-Hellman key agreement is always performed in this phase.
IKE phase 1 occurs in one of the two modes: main mode or aggressive mode.
Main Mode: Main mode has three two-way exchanges between the initiator and the receiver.

- First exchange: the algorithms and hashes used to secure the IKE communications are agreed upon in matching IKE SAs in each peer.
- Second exchange: it uses a Diffie-Hellman exchange to generate shared secret keying material used to generate shared secret keys and to pass noncesrandom numbers sent to the other party and then signed and returned to prove their identity.
- Third exchange: it verifies the other side's identity. The identity value is the IPsec peer's IP address in encrypted form. This part is encrypted with the Diffie-Hellman value agreed upon in the second pair of messages, each side reveals its identity and proves it knows the relevant secret (for example, private key or pre-shared secret key).

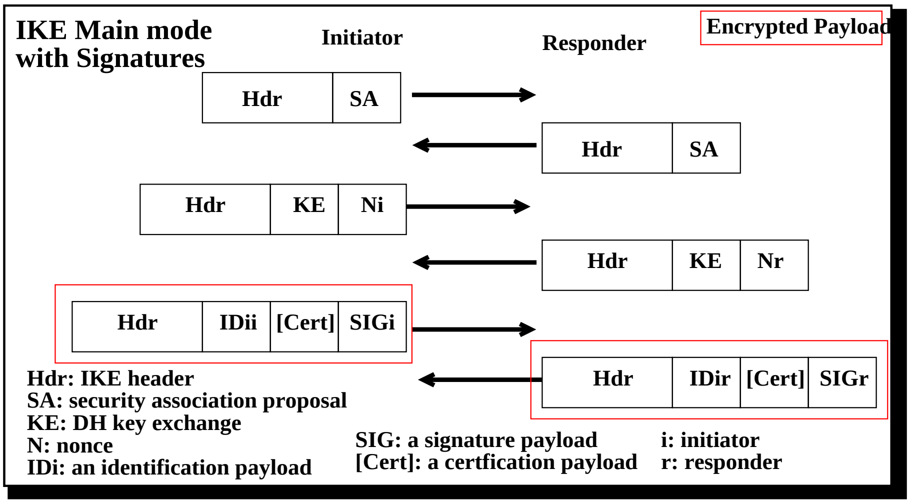


Figure 2.4. IKE Main Mode with Signatures

The main outcomes with the main mode are the matching IKE SAs between peers to provide a protected pipe for subsequent protected exchanges between the IKE peers. The IKE SA specifies values for the IKE exchange: the authentication method used, the encryption and hash algorithms, the Diffie-Hellman group used, the lifetime of the IKE SA in seconds or kilobytes, and the shared secret key values for the encryption algorithms. The IKE SA in each peer is bi-directional.
Aggressive Mode: In aggressive mode there are only three messages. The first two messages consist of a Diffie-Hellman exchange to establish a session key, and in the second and third messages each side proves they know both the Diffie-Hellman value and their secret. The fast setup feature of the aggressive mode, compared to the main mode, has a price: both sides exchange information before there is a secure channel, and it is possible to sniff the wire and discover who formed the new SA.
Several methods to authenticate IPsec peers are widely used in IKE Phase 1:

- Pre-shared keys: a shared secret key is distributed out-of-band to the peers. The peers use this information and nonce parameters (a nonce value is a random number generated by the peer) to create a hash that is used to authenticate messages.
- Digital signatures (RSA or DSS): certificates of the peers are exchanged in the last two messages and hashes are calculated over these certificates to authenticate each other.
- RSA public key encryption: each negotiating party has a public-private key-pair that they use to encrypt/decrypt messages starting from the third message. Nonces exchanged are secured through this and the parties calculate hashes over these nonces to authenticate each other.
- Revised RSA public key encryption: this method is similar to the prior method, but reduces the number of public key operations from four to two and, instead, incorporates two symmetric-key operations.

In *Phase 2*, also called as the "Quick Mode," is used to establish the IPSec SA and to generate new keying material.The sender offers one or more transform sets that are used to specify an allowed combination of transforms with their respective settings. The sender also indicates the data flow to which the transform set is to be applied. The sender must offer at least one transform set. The receiver then sends back a single transform set, which indicates the mutually agreed-upon transforms and algorithms for this particular IPsec session. A full Diffie-Hellman key exchange may be done to provide perfect forward secrecy (PFS), otherwise the keys are derived from the phase 1 keying material. The basic quick-mode message exchanges are illustrated in 2.5.

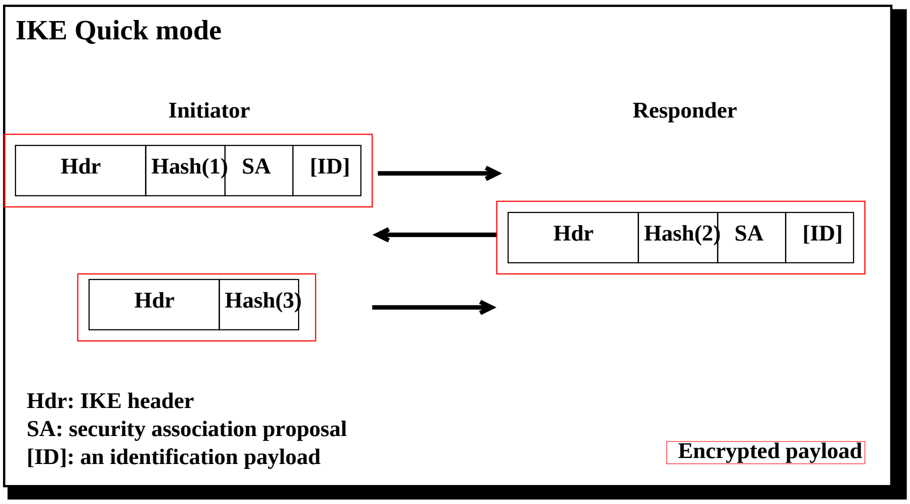


Figure 2.5. IKE Quick Mode

The quick mode is also used to renegotiate a new IPsec SA when the IPsec SA lifetime expires. Base quick mode is used to refresh the keying material used to create the shared secret key based on the keying material derived from the Diffie-Hellman exchange in phase 1.

To perform peer authentication at the end of phase 1, a variety of authentication methods to verify the identity of valid users can be implemented with IPsec, including shared secret, token cards, and digital certificates. For a large extranet implementation, the easiest method is Public Key Infrastructure (PKI) using digital certificates.

Certificate Authorities and Digital Certificates

The key exchange can get quite complicated and the distribution of keys in a public key scheme requires some trust. If the infrastructure is untrusted and control is questionable (such as on the Internet), distribution of keys is troublesome.
In fact, several mechanisms exist to verify whether the public key that the sender gave the receiver actually belongs to the sender and was not obtained elsewhere.
Before investigating the Certificate Authority (CA) which is the most powerful insurance mechanism used to ensure the legitimacy of public keys, let's review the following definitions:

- Electronic Signatures are data attached to other data for authentication purposes.

- Digital Signatures are electronic signatures linked to the signed data in a way that tampering is noticed and that the sender can be identified unequivocally. To create a digital signature, a transmitter creates a hash of the message and then uses an exclusively transmitter-owned private key to encrypt the hash. This is the digital signature and it is attached to the real message. The private key has a matching public key that the receiver can use to verify the signature. The receiver uses the same hash function to create a hash of the real message, and then takes the public key to the transmitter, decrypts the digital signature, and compares hashes. A trustworthy institution (i.e., a Trust Center or a Certificate Authority) assigns this pair of keys to a particular person.

- Digital Certificate is simply a collection of information to which a digital signature is attached. This information contains:

  - Information of the entity (person, company, or so on) whose certificate it is. This entity is referred to as the certificate subject, or owner. These information include the following attributes (or a subset): the entity's name, organizational unit, organization, city or locality, state or province, and country code.
  - The owner's public key.
  - The digital signature of the CA or the issuer. This is used to vouch for the fact that the enclosed public key is the actual public key of the owner.
  - Information for the CA or the issuer.

  The standard digital certificate format is defined in the X.509 specification.

VeriSign, Entrust, and Netscape are examples of companies that are providing digital certificates. A client registers with a certificate authority; after the CA verifies the customer's credentials, a certificate is issued. A CA performs the following main functions:

1. It issues a pair of keys for a customer guaranteeing the uniqueness of the pair.

2. It certifies a customer's public key.

3. It publishes a customer's certificate.

4. It Issues Certificate Revocation Lists (CRLs).

The CA acts as the agent of trust in the key management system the Public Key Infrastructure (PKI). The PKI uses to secure exchange of digital signatures, encrypted documents, authentication and authorization of many communication partners in open networks. A PKI can be used to manage IKE's authentication keys, but IKE does not specify its interaction with a PKI. In addition to the CA, the PKI has three other parts: The Registration Authority (RA) which is responsible for recording and verifying all information the CA needs, The directory service which publishes certificates and a CRL or make an on-line certificate available and finally a special service the Timestamping that confirms the receipt of digital documents at a specific point in time.

2.3.4 SSL VPN Tunneling Mechanism

Secure Sockets Layer (SSL) VPNs rely on a commonly used protocol for managing the security of data transmissions on the Internet. SSL VPNs use a program layer located between the Internet's Hypertext Transfer Protocol (HTTP) and TCP protocols for communication [71] [12].
SSL VPNs have many benefits:

- They are easy to deploy and easy to maintain.
- They need fewer firewalls changes since SSL firewalls are generally kept open.
- SSL VPNs are the ideal solution for corporations whose employees are often on the go, since they can be extended to reach remote accesses.
- SSL support is included with most web browsers (Mozilla, Netscape, IE), as well as with most web server products.
- SSL VPNs deliver user-level authentication, ensuring that only authorized users have access to the specific resources as allowed by the security policy in use. SSL uses the public-and-private key encryption system from RSA, which also includes the use of a digital certificate.

The essential limit of SSL VPN is that it provides access only to web applications, while it fails to address access services to other client/server applications.
We can distinguish three types of SSL VPN technologies;

- *Application layer proxies*: this is the simplest form of SSL VPNs because they rely on the SSL functionality used by existing applications. Because of this, application layer proxies have the least application support. Generally, they only support Email and Web-based traffic. To use email, your administrator would configure the SSL functionality in your email client and proxy all email traffic via the gateway. One of the advantages of application layer proxies is that they do not have to get a special program or support, and operate with nearly all operating systems and web browsers.
- *Protocol redirector*: the protocol redirector approach is more flexible than the application layer proxies, but it requires a kind of program or support. Protocol redirectors work by downloading a mini support from the gateway, which installs locally and redirects traffic. For instance, if a connection is made from an application that does not use SSL, the connection is captured below the IP layer and encapsulated within an SSL tunnel. Once the traffic reaches the SSL gateway, it is decrypted and then proxied to the original destination. This would appear to be an ideal mechanism, because all

normal applications work with minimal intervention for the user. The reality though is slightly different. The only realistic way the support can capture the traffic on the way through the IP stack is to redirect traffic based on name resolution to a local resource. Once the port redirector is enabled, the name of a remote site will be forced to connect to localhost through the use of a host file. This means the mini support must have the ability to write changes to the hosts file, which in a hardened corporate desktop may not always be possible. Also, in most implementations some administrative permission is required on the local desktop to install the mini support. The main advantage of the protocol redirection system is that it can support any application that works over fixed TCP or UDP ports (and in some implementations, applications using dynamic ports may be supported, like MS Outlook).

- *Remote control VPN*: this category is sometimes merged with the first category, "application-layer VPN", but we prefer to distinguish them. Remote control VPNs are the most flexible form of SSL based VPN, but they also have the highest overhead. They work by enhancing a remote control protocol like Windows Terminal Services, and by adding SSL VPN functionality and Web Browser support. This means that any application can be added to the SSL VPN by adding the application to the remote control desktop. As a stand-alone application, this has serious limitations, because applications that reside on the local desktop cannot be used directly. This is why most remote control VPNs are partnered with other SSL VPN technologies. On the positive side though, they can offer features like the ability to read and update a document held centrally without ever having to download the entire document. When traveling and using VPN over low speed connections, or when the connection quality is poor this may be very advantageous (because connections are restarted without loosing any work). The central element of a remote control VPN is the application-layer proxy, typically provided in the form of a dedicated VPN appliance. The proxy offers a single point of administration while acting as a sentinel to the private network behind the firewall. A remote control VPN acts as an intermediary between remote client requests and server-based applications. It terminates incoming connections from remote users at the application layer, processes the data and then translates the data to the appropriate backend application protocol, like:

  - Remote Desktop Protocol (RDP) for Windows applications residing on Windows Terminal Servers
  - X.11 over SSH for UNIX or Linux applications
  - 3270 over Telnet for mainframe and AS/400 applications
  - HTTP/HTTPS for web servers.

  During this termination gap, the VPN appliance analyzes application information, applies security policy and serves as a gatekeeper between the Internet and the private network. A remote control VPN runs client and server versions of an application on a single server, eliminating the need for a client on the remote PC.

  For example, for Windows applications, a remote user launches a browser and enters the URL of the remote control VPN appliance. SSL is used to encrypt all data from the user's browser to the remote control VPN. SSL provides strong security, and most Web browsers incorporate it. The remote control VPN enforces policy during the termination gap by polling external authentication and policy servers, such as Active Directory or Lightweight Directory Access Protocol, to certify user identities and authorize specific application access. Once the remote control VPN appliance has verified that the browser sent legacy application data to the proxy via a thin-client application protocol, the proxy terminates and translates this protocol to Microsoft RDP and delivers it to the application server. Remote control VPNs also work well with Web-based and intranet applications. The proxy terminates, examines and rewrites HTTP requests. Remote users then receive Web application resources as defined by policy and security.

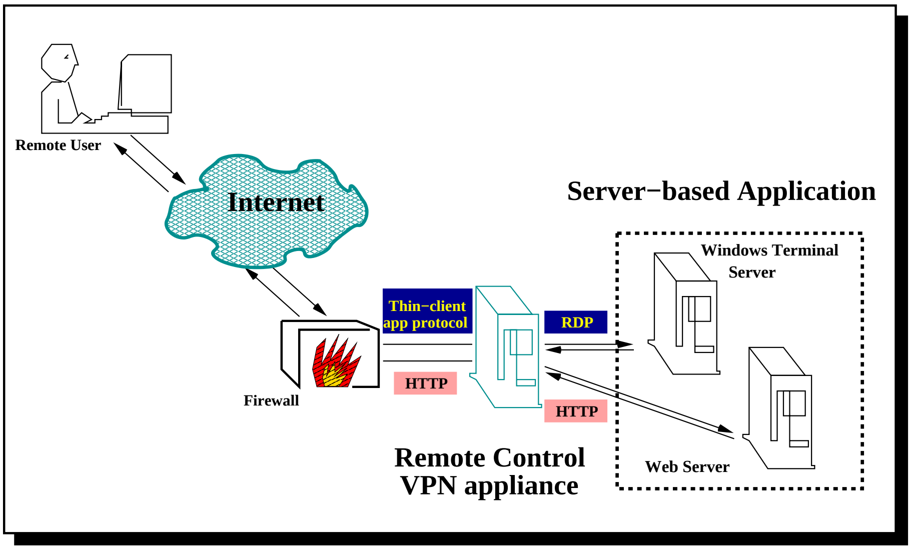


Figure 2.6. Remote control SSL VPN

Many efforts have been made in attempting to privilege one of the two technologies: IPsec and SSL over the other, but as we will see in the next section, both have distinct pros and cons. The following comparison focuses on key areas that should be considered when determining which type of VPN best meets the clients' needs.

2.3.5 IPsec versus SSL VPN

IPsec and SSL are two effective ways to provide secure communications over the Internet and both can be considered for building VPNs. Yet several fundamental differences exist [23], essentially because IPsec operates at the network layer whereas SSL operates at the application layer.

Deployment: For instance IPsec offers a global security to all the applications, no matter whether they use the TCP or UDP protocol, but it requires that an IPsec stack be installed. The IPsec secure connection is application independent, which means once the connection is established, all applications can be accessed and operated from any point on the network. Not so for SSL host site devices. SSL only offers security to SSL-aware applications. Each application has to be individually configured to work with the SSL host site device. Thus an SSL host site device can be much more expensive and time consuming than deploying an IPsec host site device. On the opposite SSL is largely deployed and used, since most web browsers support HTTPS (i.e. HTTP over SSL). The SSL client-less aspect is a great advantage to companies having a mobile workforce and connectivity from various locations.

In practice, building an SSL VPN or an IPsec VPN has more to do with clients needs. IPsec is well suited for site-to-site connections where broad and persistent connections are needed. SSL, on the other hand, is well suited for applications where the system needs to connect individuals to applications and resources.

- *Traffic type dependency*: IPsec does not stick to a specific type of application (e.g. legacy applications can benefit from it) or transport protocol. On the opposite, SSL only supports TCP services, and often only HTTP or POP3/IMAP/SMTP flows.

- *Security*: with regard to security, if we drill down to the details of IPsec and SSL VPN, they are much the same, just implemented differently. However, because of the way the SSL VPN is deployed, it can be less secure. As any Web-enabled machine can be used to access a SSL based VPN, two-way authentication is not available. Anyone with the correct username and password can access the SSL VPN from any PC connected to the Internet. Nevertheless, with strong authentication, security problems can be mitigated. In general, IPsec VPNs offer better end-user authentication and data encryption at the network layer rather than the session layer, making them much more secure than a SSL VPN solution.

- *Scalability*: also IPsec VPN solutions scale better in terms of applications, while SSL VPN solutions scale better in terms of users. Since IPsec VPNs are application independent. As soon as a new application is added to an existing system, VPN network access is enabled. However, adding new locations and new users does require deployment of additional hardware or software at the remote site. In addition, some configuration at the host site is required to add new locations or new users. SSL VPNs require either a proxy server or Web-enabled applications. If a mission-critical application (such as CRM or ERP) is deployed, it must be Web-enabled or configured to work with the SSL proxy server. A considerable amount of configuration will be required on a per-application basis. However, as SSL VPNs do not require a remote site client, adding new locations and users is easy. All it requires is adding the particular location or user to the VPN authentication database. An authorized user can access VPN resources from any Internet-enabled machine.

- *Mobility*: although widespread deployment is yet to take hold, mobile IP network deployment is growing steadily. A side effect of a mobile IP network is that the client source address can change as a client moves between cells and networks. This has the effect of breaking an IPsec VPN connection, but because SSL VPNs are not bound to the source IP address, connections can be maintained as clients move.

- *Network Address Translation (NAT) problem*: traditionally Hide Network Address Translation (Hide NAT) has caused issues with IPsec VPNs. Vendors have generally overcome these issues by developing vendor specific NAT traversal mechanisms based on payload encapsulation in UDP packets. Although these mechanisms normally function well, they break and interoperability between vendors deployments. SSL VPNs do not suffer such issues because they are not tied to the IP layer.

- *Administrative control*: since IPsec requires software installation on the clients side, administrative control is possible.For instance administrators may not want users to access sensitive data from some web sites, due to the unknown security state of these sites.

- *Header overheads and performances*: SSL VPNs offer some performance advantages over IPsec VPNs when considering the header overhead. If we ignore the setup operations, the header overhead of IPsec on a packet is between 50 and 57 bytes (including the new IP header, the ESP header and the trailers), representing a 10% increase for a 500 byte packet. On the opposite, SSL VPNs add only 5 bytes of data to each packet, just a 1% increase with 500 byte packets. Of course, the setup operations cannot be ignored, but these are roughly similar in size for IPsec and SSL connections. Also, because SSL VPNs work at a higher layer, they suffer much less from the packet fragmentation issues normally associated with IPsec VPNs. Finally, SSL has built in compression mechanisms, not IPsec.

Neither technology is the perfect solution, it depends on the type of access required by clients as well as other criteria, such as performance, manageability, or ease of integration[11]. In spite of their differences, both technologies can:

- determine if an information needs to be encrypted before going through the Internet,
- negotiate encryption keys, including authentication of each side,
- encrypt data and send it to the destination, and
- check data integrity to determine if data has been tampered while in transit.

Therefore both technologies can be used in a VPN environment and *we consider both IPsec and SSL based VPNs* in our VPN web service architecture proposal.

2.3.6 Other Tunneling Protocols

Since we are only interested in the two tunneling protocols discussed above, we do not detail the other possibilities, but only give a quick glance on them.

- GRE: the Generic Routing Encapsulation protocol (GRE), described in RFC1701, allows an arbitrary network protocol A to be transmitted over any arbitrary network protocol B, by encapsulating the packets of A within GRE packets, which in turn are contained within packets of B.
- PPTP: the Point to Point Tunneling Protocol (PPTP) consists of two communication channels between the client and server:
  - a *control channel* where link-management information is sent. This is in fact a standard TCP connection to port 1723 (server).
  - a *data channel* over which (possibly encrypted) private network traffic is carried. This data channel uses a dedicated IP protocol number, 47, reserved to GRE.

  Currently PPTP is the most common VPN connection method in a Windows environment. Microsoft's implementation of PPTP uses the Point-to-Point Protocol (PPP) to initially encapsulate the data, then encrypts this with Microsoft's Point-to-Point Encryption (MPPE).

  Authentication is provided by Windows' built in dial-up authentication protocols; MS-CHAP, MS-CHAP v2 (Microsoft Challenge Handshake Authentication Protocol) and EAP (Extensible Authentication Protocol). These protocols provide a means of authenticating both the client and the VPN server by means of a user name and password,

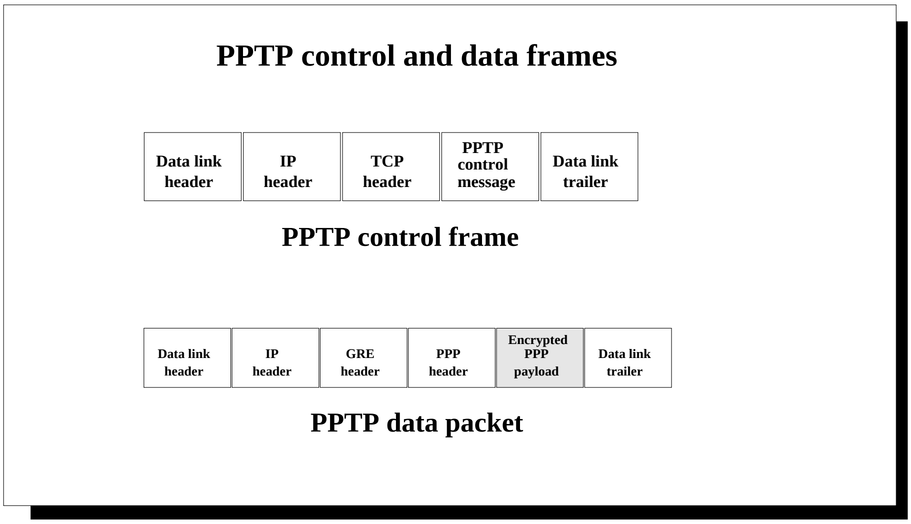


Figure 2.7. PPTP control and data frames

in the case of MS-CHAP, or a computer certificate or smart card in the case of EAP. Once data is encrypted, it is encapsulated again, this time inside a GRE packet, which provides the information necessary to transmit PPP information over the Internet. We now have the original data encrypted and enclosed within a PPP data packet, then further encapsulated within a GRE packet for transmission across the Internet. PPTP uses the separate, unencrypted control channel to carry the commands used to open, close and maintain the connection.

- L2TP: this new VPN technology, Layer 2 Tunneling Protocol (L2TP), combines some of the features of Microsoft's PPTP with Cisco's L2F. In the Windows implementation, L2TP wraps the whole thing up in Microsoft's version of the IPsec encryption and authentication protocols.

  IPsec is used both to encrypt both the control and the data packets via the exchange of a security certificate or a pre-shared key. Standard PPP authentication (MS-CHAP, EAP) is still used to validate the user with a user name and password combination. Using IPsec authentication provides an extra level of security by assuring that the machines at both ends of the VPN are known and trusted. In theory L2TP could be used without IPsec, but in practice it is not very feasible since it uses no other form of encryption.

  L2TP/IPsec differs considerably from PPTP. L2TP VPNs do not use a separate, unencrypted control channel to send control information as PPTP does. Instead of that, the control information needed to start, maintain and close a VPN tunnel is sent through the same port, like data, using L2TP control data instead of the PPP encapsulated data.

  Data is first encapsulated in a PPP packet similarly to the former method, and a PPP header added. An L2TP header containing the necessary information to convey the data through the Internet is next, followed by a UDP header. UDP, part of the TCP/IP suite of protocols, is the protocol L2TP VPNs use to transmit and receive data through

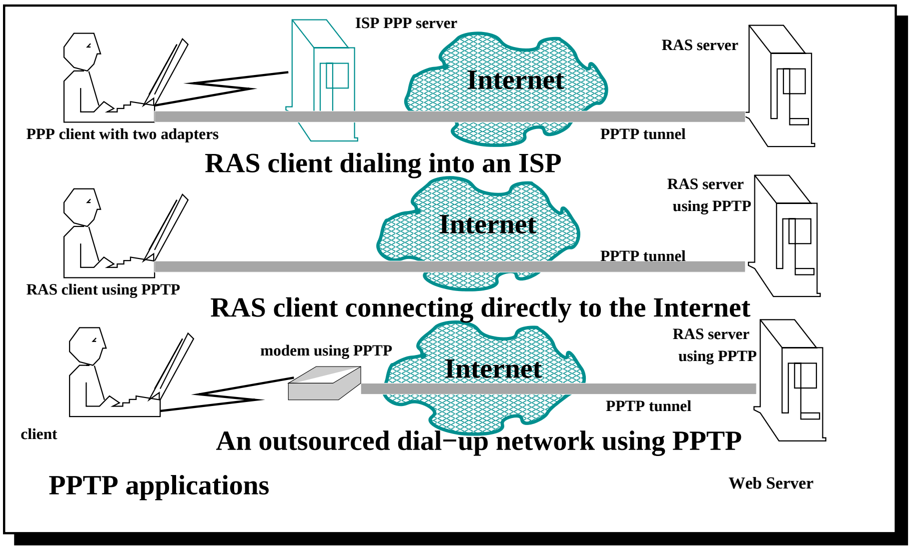


Figure 2.8. PPTP applications

ports. Assuming IPsec is used, L2TP will then encapsulate and encrypt the above contents, adding an authentication trailer which will allow the receiving computer to verify the sender. The encrypted payload is then provided with an IP header for source and destination addresses, and a data-link header and trailer specific to the form of network the VPN will traverse, just as is done with PPTP encapsulation.

In general, IPsec combined with L2TP is considerably more secure than the PPTP method of creating a VPN.

2.3.7 Different Connectivity VPN Models

IPVPNs are typically viewed as falling into three major categories: remote access VPNs; site-to-site VPNs (intranets), and business-to-business (B2B) VPNs (extranets) [50]. Here is the three forms of secure communications that could be handled by a VPN solution in more details:

- Remote Access and Dial-up services: remote-access, also called a Virtual Private Dial-up Network (VPDN), helps mobile and seamless communications for road-warriors in accessing the company's Intranet from remote locations by user-to-LAN connections. Typically, a corporation that wishes to set up a large remote-access VPN will outsource to an Enterprise Service Provider (ESP). The ESP sets up a Network Access Server (NAS) and provides the remote users with desktop client software for their computers. The telecommuters can then dial a toll-free number to reach the NAS and use their VPN client software to access the corporate network. Authentication on many access servers is performed by RADIUS. Upon successful authentication of the user, the secure connection enables the user to have access to the company's propriety.

  A good example of a company that needs a remote-access VPN would be a large firm with hundreds of sales people in the field. Remote-access VPNs permit secure, encrypted connections between a company's private network and remote users through a

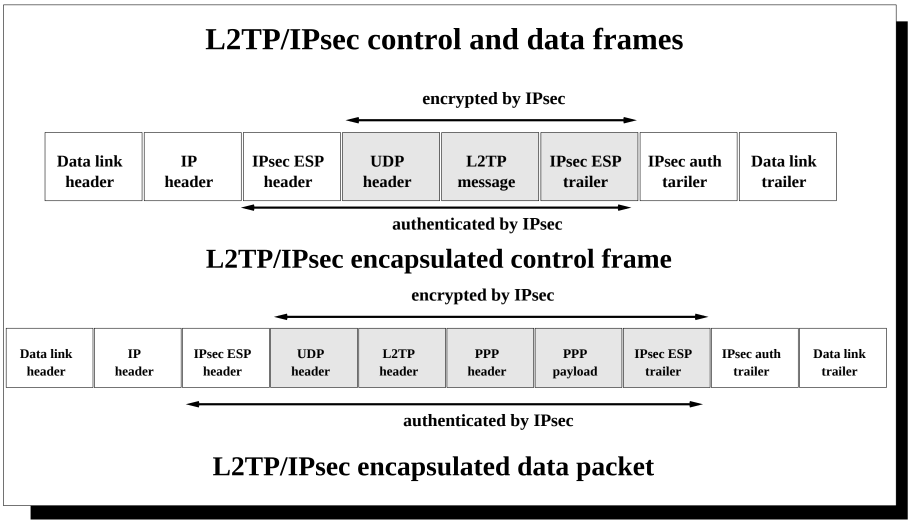


Figure 2.9. L2TP/IPsec control and data frames

third-party service provider. In a remote-access VPN, tunneling normally takes place using PPP. Part of the TCP/IP stack, PPP is the carrier for other IP protocols when communicating over the network between the host computer and a remote system. Remote-access VPN tunneling relies on PPP. L2TP usually aims at providing services to the home offices and telecommuters who dial-in to a specific local RAS. Alternatively, PPTP model focuses on the mobile user, who may dial-in to any local ISP. The user initiates a connection to any of the VPN servers located in the company's Intranet, after establishment of the connection with the ISP. Then, the access privileges are established after the authentication server validates the user. In this model a Remote Access Server (RAS) does not participate in establishment of the VPN connection. So, RAS configuration is not need with PPTP model.

Thus AAA (Authentication, Authorization and Accounting) servers are used for more secure access in a remote-access VPN environment. When a request to establish a session comes in from a dial-up client, the request is proxied to the AAA server. AAA then checks the following:

- Who you are (authentication)
- What you are allowed to do (authorization)
- What you are actually doing (accounting)

The accounting information is especially useful for tracking client's activity while accessing network services. This information is used later for security auditing, billing or reporting purposes.

- Intranets services: it helps interconnecting local area networks (LANs) located in multiple geographic areas over the shared network infrastructure in a secure and cost effective way. Typically, this service is used to connect multiple geographic locations of a single company. Several small offices can be connected with their regional and main offices. This service provides a replacement for the expensive dedicated links. Adding new geographical sites to this kind of architecture is much easier, since new sites can be connected to the VPN with little effort. In general Intrantes services takes the form of site-to-site VPN model, where IPsec is normally the encapsulating protocol that provides the framework for how to package the passenger protocol for transport over the carrier protocol, which is typically IP-based. This includes information on what type of packet you are encapsulating and information about the connection between the client and server. IPsec works well on both remote-access and site-to-site VPNs, but it must be supported at both tunnel interfaces to use.

- Extranets and E-commerce services: this type of service combines the two previous services mentioned above. This infrastructure enables external vendors, suppliers and agents to access specific areas of the company's Intranet over secure connections. The specific area is denoted as Demilitarized Zone (DMZ). When a suppliers representative connects to the company's Intranet or dialing in remotely, the firewall and authentication mechanism ensure that the connection is directed to DMZ. A Company's employee on the other hand has full access to the company's Intranet.

2.4 Various Types of IPVPN Solutions

In this section, we focus on a special category of VPN that is operated on IP infrastructure. This category is known as IPVPN. Many types of VPN are gathered under this category [2].

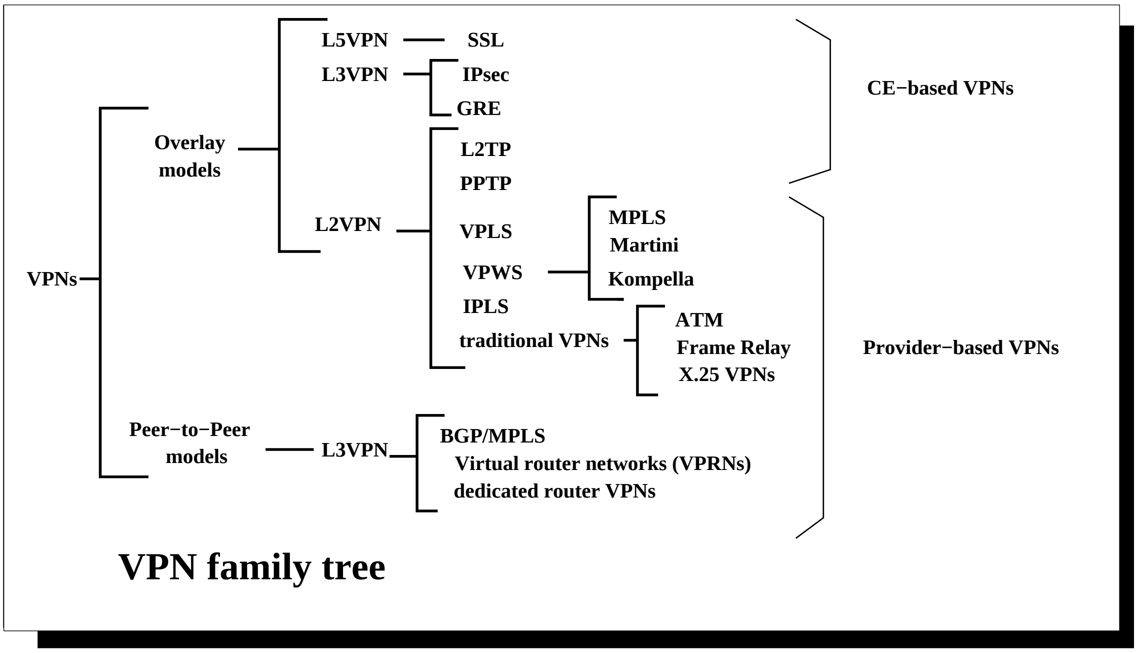


Figure 2.10. VPN Family Tree

2.4.1 IPVPN definition

With the overwhelming success of the Internet during last decade, the landscape of telecommunications has changed radically and the IP protocol has been pervasively deployed in corporate networks. For this reason, business companies made attention of creating secure, private corporate networks using the shared infrastructure of the Internet [27]. Eventually sharing network resources by using the Internet is definitely more cost effective than using dedicated facilities. Further, the Internet can be accessed from virtually anywhere in the world, with operations and administration supplied by the local Internet Service Provider.
IP-based VPNs (IPVPNs) can be defined as a VPN implementation that uses public or shared IP network resources to emulate the characteristics of an IP-based private network. They have shown potential to become the foundation for a wide range of corporate network services. An increasing number of service providers offer value-added applications and services on their IPVPN transport networks to generate new revenue and gain competitive advantage.
Different types of ISPs offer different VPN solutions [45], according to the customer needs. Customers of VPN services use shared facilities and equipment, which are managed, engineered and operated by a public network operator, either totally or partly.

2.4.2 General Classifications of IPVPN Types

Two general types of IPVPNs exist: Customer Equipment CE-based VPNs, and Provider-based VPNs (AKA Network-based VPNs).

- CE-based VPNs: In a *CE-based VPN*, knowledge of the customer network is limited to the Customer Premise Equipment. In this type, VPN access is placed at the customer equipment which provides end-to-end security. Provisioning and management of the VPN is up to the customer network administration, typically by manual configuration of the tunnels between customer equipments, without significant upgrades to the service provider's network. However, it is common for a service provider to be responsible for managing and provisioning the Customer Edge equipment, in order to reduce the management requirements of the customer. The tunnels between CE equipments may be implemented as simple link layer connections such as ATM or Frame Relay, or by means of various encapsulation formats such as GRE, IP-in-IP, IPsec, L2TP, or Multi-Protocol Label Switching (MPLS). Routing in the customer network views the tunnels as simple point-to-point links. Briefly, an CE-based VPN can be defined as an IPVPN that initiates tunneling and encryption at the edge of the customer's network for dedicated locations and on the remote user's PC for remote access users. The challenge with this approach is a problem of scalability when large-scale VPNs should be managed because of their distributed nature. More details in CE-based VPNs in section 2.4.3 and chapter 3.

- Provider-based VPNs: *Provider-based VPNs* and all related configuration, operation and control are provided by equipments of the service provider's network. Usually these types of VPNs are built on edge devices that are located at the service provider's Point of Presence (POP). Customer network is supported by tunnels, which are set up between pairs of edge routers. The tunnels may make use of various encapsulations to send traffic over the Service Provider network. Examples of tunnel encapsulations are GRE, IPsec, IP-in-IP and MPLS. There are two basic types of Provider-based VPNs: Layer 2-based VPN (L2VPN) and Layer 3-based VPN (L3VPN), discussed later. The challenge with this approach is that it requires provider's POPs to be updated to meet

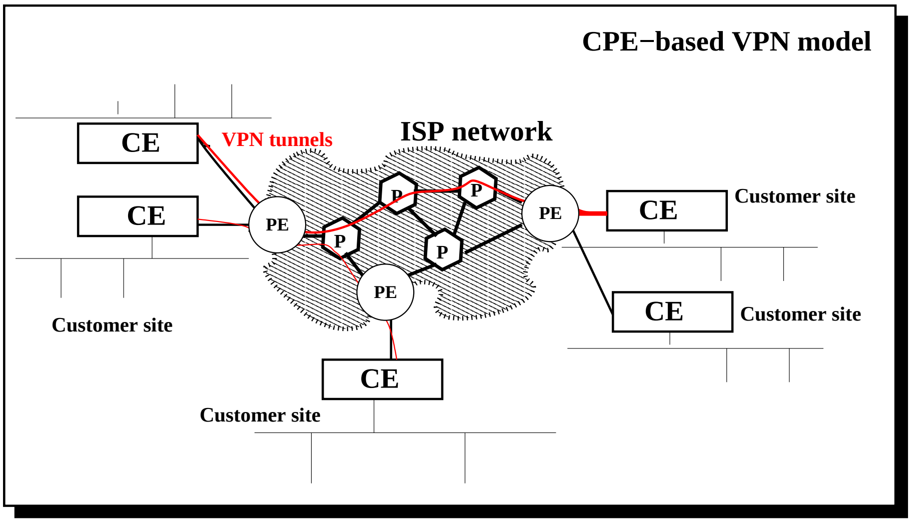


Figure 2.11. CE-based VPN model

customer's needs for changing VPN requirements. More details in Provider-based VPNs in section 2.4.4.

The previous classification of IPVPN is based on the place where IPVPN functionalities will be applied whether if it is on the customer side or on the provider side, regardless of which of the two is the responsible of IPVPN management service.

IPVPN could be also classified depending on the routing information exchanging between the customer equipments and the provider equipments. We define *Overlay VPN* when the customer and the router's provider equipment do not exchange layer 3 routing information. Otherwise, we talk about *peer-to-peer VPN*. The Overlay VPN is suitable for small-scale VPN implementations, since it becomes hard to manage in complicated topology [9]. In addition to that, the overlay VPN model provides clear separation between the customer and the provider's responsibilities. While the other model provides easier VPN service provisioning. From addressing perspective, in the Overlay VPN the addressing scheme of the transport network may be different from the customer addressing scheme. In contrast, in a Peer Model VPN the service provider and the customer share the same address space, at least up to the demarcation point of the VPN. The customer can, of course, use NAT to create a degree of address flexibility.

2.4.3 CE-based IPVPN Details

In this VPN model, VPNs are built with CEs. In the CE-based VPN, customers can use different connections from different ISPs for the VPN operation. Many technologies can be used to set up this VPN model: IPsec, PPTP, L2TP, L2F and GRE. Thus the characteristics of the VPN created depend on the tunneling technology used. The CE-based VPN scales well comparing to the Provider-based VPN, since PE and P routers do not need to get involved in the routing of the VPN. More information on CE-based VPN will be provided in chapter 3.

2.4.4 Provider-based VPN Details

As a general class, most network-based VPNs provide reliability and security options. However, security and reliability are slightly less than frame relay due to immature carrier operational processes, single points of failure between carrier backbones, and back-end exposures to the Internet (some carrier MPLS networks use common fiber and platforms for Internet and MPLS VPN services, increasing administrative security risks). Thus, customers seeking to use network-based VPNs may still need to encrypt end to end, via IPsec or other techniques (frame relay customers may need to consider this as well).
Anyhow this VPN model comes in two fundamentally different types. The first type is designed to work at layer two of the OSI reference model, the link layer (L2VPN), while the second VPN type is designed to work at layer three (L3VPN). These two types of VPN were originally suggested by the Provider Provisioned IPVPN(PPVPN) IETF working group in the RFC 2764 [38]. The PPVPN works on finding out standard ways to manage and provision VPNs in behalf of ISPs.

L2VPN

In L2VPN, layer two services are extended to clients sites. Forwarding information is based on layer two information, such as MAC address and MPLS label. A PE router implements one VPN Switching Instance (VSI) per VPN. The VSI includes layer two information and supports functions regarding the forwarding of layer two frames, cells or packets for a VPN to terminate the tunnels associated to the VPN [86]. Here are some different approaches to L2VPN: Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS) and IP LAN Service (IPLS).

- Virtual Private Wire Service (VPWS): it corresponds to a VPN service that supplies a layer 2 point-to-point service. It emulates a set of wires between the customer sites. This practically useful when customer sites are connected via a set of ATM or Frame Relay connections. A customer can keep the same layer 2 connections to the service provider, but instead of data being carried natively over an ATM or Frame Relay service, the traffic is encapsulated and routed over the provider's IP backbone. In this approach, CE routers carry out layer 2 switching information, thus the CE routers have to choose which virtual wire to use to send data to another customer site. That means that some configuration is needed on the CE sides. The VPWS offers site-to-site services

- Virtual Private LAN Service (VPLS): it corresponds to a VPN service that emulates a LAN. The Ethernet LAN at each customer site is extended as far as the edge of the provider network. Then the provider network emulates a LAN switch or bridge to connect all of the customer LANs to create a single bridged LAN. In this approach CE routers do not care of handling layer 2 switching information, instead they send all traffic destined for other customer sites to the PE routers. Thus no additional configuration of CEs is needed and only the PEs are VPLS capable. The VPLS offers multisite-to-multisite services.

- IP LAN Service(IPLS): here only IP traffic is exchanged between CE devices. IP traffic is also forwarded based on Layer 2 information.

Many solutions has been developed for each of the three approaches mentioned above:

- VPWS solutions:

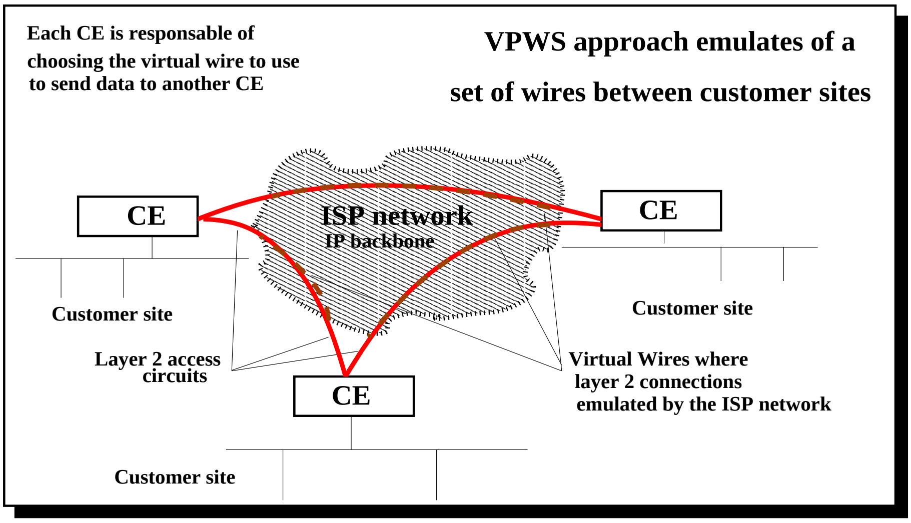

Each CE is responsable of
choosing the virtual wire to use
to send data to another CE
VPWS approach emulates of a
set of wires between customer sites
CE
ISP network
IP backbone
CE
Customer site
Customer site
Layer 2 access
circuits
CE
Virtual Wires where
layer 2 connections
emulated by the ISP network
Customer site

Figure 2.12. VPWS approach

- MPLS L2VPN: this VPN type is the simplest type of VPN compared to other types discussed below. The MPLS L2VPN uses Virtual Channels VCs between the PE devices and the CEs devices, while Label Switched Paths LSPs are used between PEs devices [77]. Each LSP needs to be cross-connected to the specified VC at PE router, this requires LSP configurations in the PE routers as it shows in the figure 2.14. The establishment of emulated VCs into the provider network also called Virtual Leased Lines (VLL).
- Martini L2VPN: this type of VPN uses Martini extensions to MPLS. The emulated VCs in this approach are known as s pseudo-wires. The pseudo-wires are created between pairs of PE devices by tunneling through such an LSP. In this solution the number of LSPs between PE routers is limited to a fixed number, thus scalability is getting much better. Martini defined a signaling pseudo-wires protocol for creating and maintaining the VCs and an encapsulation protocol for forwarding data over the pseudo-wires network. The layer 2 CE-PE connections are cross-connected with Martini pseudo-wires using the appropriate layer 2 encapsulation. This approach reduces the amount of resources consumed in the PE routers, nevertheless each pseudo-wire needs to be configured individually.
- Kompella L2VPN: this approach is more scalable than the previous ones. Each VWPS owns a unique identifier. As well each CE must be identified uniquely within the same VWPS and be configured with a separate VC for each remote CE device that it wishes to connect to. A simple algorithm is used to map the remote CE identifier to the identifier of the VC to use. Tunnels are set up automatically between PEs. Usually MPLS is used as the tunneling protocol in this approach. MPLS labels are used to identify the VCs. PE router allocates a block of MPLS labels for each CE, the block contains one label for each VC that connects the PE and CEs. Multi-Protocol Border Gateway Protocol (BGP) is used to advertise the block of labels between the PE routers, so PE routers can set up the required tunnels. For example, in the figure 2.15, when the PE2 device receives the label

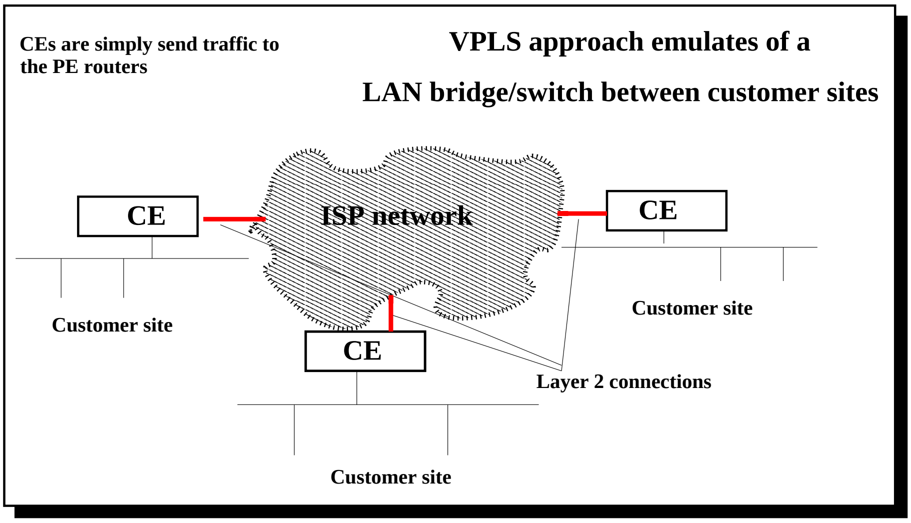

CEs are simply send traffic to the PE routers
VPLS approach emulates of a
LAN bridge/switch between customer sites
CE
ISP network
CE
Customer site
Customer site
CE
Layer 2 connections
Customer site

Figure 2.13. VPLS approach

block from PE1, it knows that the labels are allocated for sending data to CE1. There is one label for each remote CE to send data to the CE1. PE2 ignores all labels destined to other CE devices and takes care only of the label L12, where it signifies the label allocated to VC2 for CE1. Now CE2 is able to send data to CE1. To do so, CE2 sends data over VC1 to PE2. As soon as PE2 gets the data, it knows that they should be tunneled to PE1 with the label L12 since they were received over VC1. When PE1 receives the data with the label L12, the label is removed and the data is sent over VC2 to CE1.

- VPLS solutions: the VPLS is based on a full mesh of Martini pseudo-wires set up between the PE routers that are participating to a specific VPLS. Like Ethernet switches, PE devices learn MAC addresses of carried packets to successfully forward data from one CE to another. In this type of VPN, when a PE router participates to a VPLS, it uses BGP as auto-discovery mechanism to learn about VPLS membership, and sets up Martini pseudo-mires with other PEs of the same VPLS. This type of VPN does place a significant burden on the PE devices, since those later should be learn about all MAC addresses of the attached VPLSs, should be able to forward VPLS instances and should be also responsible of tunnel management and VPLS configuration.

- IPLS solutions: by snooping IP and ARP frames, each PE device learns automatically about the local CE devices. Using also the BGP auto-discovery mechanism, the PE device learns about remote PE devices in the same IPLS. For each locally CE device, a PE sets up Martini pseudo-wire to each remote PE device in the same IPLS. When the PE device receives data on this pseudo-wire, it forwards data straightly to the CE device without even looking at the IP address. Otherwise, when a pseudo-wire is signaled to a remote PE device, the IP address and the MAC address of the remote CE is included, thus each PE device could build up a table of all IP addresses and MAC addresses of all remote CE devices.

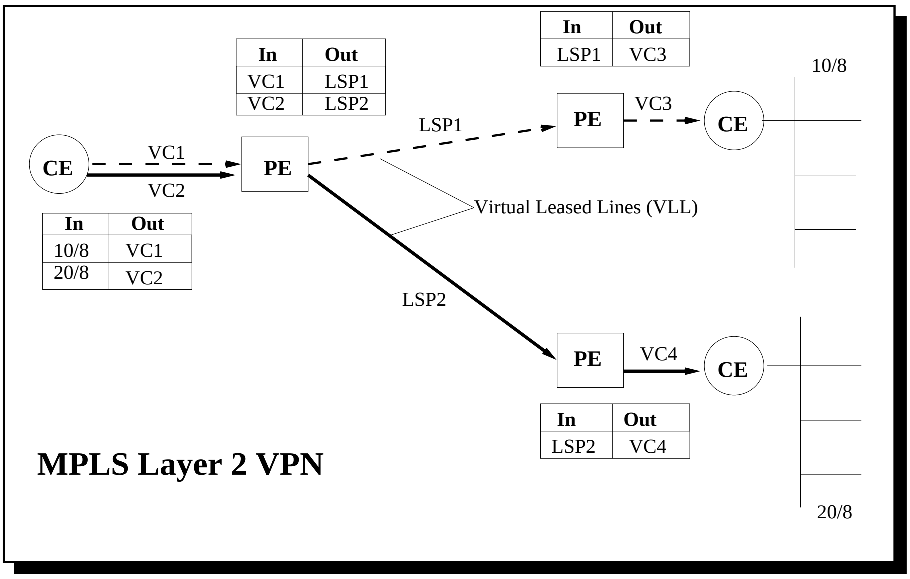


Figure 2.14. MPLS Layer 2 VPN approach

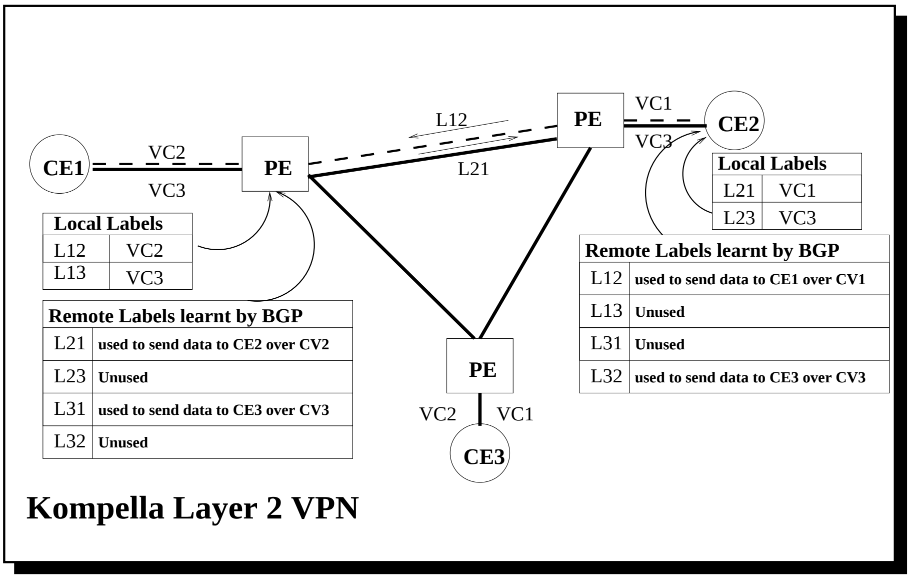
L3VPN

In the L3VPN model, forwarding decisions is based on layer 3 information such as IP addresses. This routing information is exchanged between CEs and PEs. This model has been applied under two approaches:

- BGP/MPLS VPN (also referred to RFC 2547 approach) [76]: routing information is represented in each PE as routing and forwarding table called VPN Routing and Forwarding (VRF). PE node uses BGP protocol to determine VPN distribution route across ISP core network. In addition to that, PE node uses MPLS labels to keep VPN traffic isolated when the traffic is transmitted over the ISP core network. Any tunneling technology could be used to establish tunnels over the core network. Briefly, this approach uses BGP to distribute VPN routing information across the ISP network and MPLS to forward VPN traffic from one VPN site to another.

  P routers maintain routing information of PEs, but does not maintain any information regarding the VRFs. To summarize the flow of data over a BGP/MPLS architecture; the CE forwards data to the PE. The PE looks up for the suitable route in the VRF table and adds an MPLS label to the data. The packet is forwarded over the tunnel to the remote PE. The MPLS label is used by the remote PE to identify the data which is then forwarded to the suitable remote CE.

- VPRN (AKA VR based VPN): like the previous approach, for each VPN the PE has VRF table that is used to forward data to and from the CE devices, as if we have a virtual router for each VRF table [52]. The principal different with the previous approach summarizes by using a separate routing instance for each VRF, while a single routing instance of BGP is used to advertise all VRFs in the BGP/MPLS approach. Virtual routers could use any standard routing protocol to exchange routing information



Local Labels	
L12	VC2
L13	VC3

Remote Labels learnt by BGP	
L21	used to send data to CE2 over CV2
L23	Unused
L31	used to send data to CE3 over CV3
L32	Unused

Local Labels	
L21	VC1
L23	VC3

Remote Labels learnt by BGP	
L12	used to send data to CE1 over CV1
L13	Unused
L31	Unused
L32	used to send data to CE3 over CV3

Figure 2.15. Kompella Layer 2 VPN approach

such as BGP, OSPF, etc. In this approach, tunnels could be of any type like the previous approach.

In general, there is no difference between the two approaches in the way of processing and maintaining routing information. In the other way, in the VPRN approach customer could use his own routing protocol to connect to other side, while in the other approach, the customer uses the routing protocol specified in the PE node where the BGP is running. On the other side, the VPRN approach limits the number of supported VPN by a VPRN node, while that is not the case in the first approach. Moreover, the VPRN approach consumes more resources than the other approach since it supports routing information for each VPN.

L3VPN vs L2VPN

- Scalability: L3VPN is less scalable than L2VPN, since routing information must be taken of in the ISP network.
- Flexibility: L3VPN is less flexible than L2VPN. That is also due to the routing information that are needed to be updated and maintained by the ISP according to the customer needs.
- Operational burden: in the contrary of the other properties discussed above, L3VPN eases customer's operational burden compared to L2VPN.

2.5 VPN Management Challenges

Originally, enterprises rely on service providers to establish secure connections by deploying private network technologies such as point-to-point or circuit-switched lines. In the *point-to-point leased lines* technology, service providers use physical links to connect distant sites.

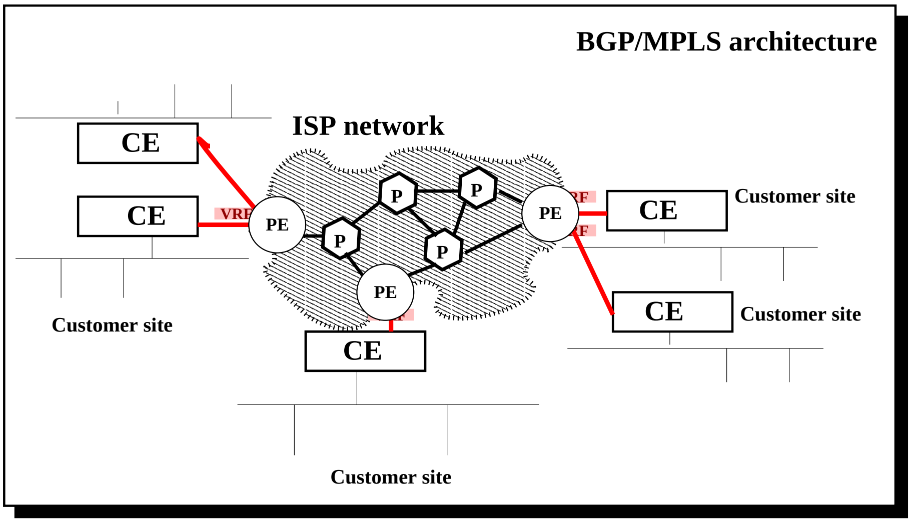


Figure 2.16. BGP/MPLS architecture

One of the most important characteristic of this technology is the high-available connectivity of the lines. In the other hand, enterprises paid for the entire line, not just the bandwidth they consumed. Moreover enterprises had wait for service providers to fix a connection when a line went down. Plus, when an unauthorized user got access to the leased line, he could spy the data streams without the knowledge of the end user. The second technology offers some improvement over the leased lines. In the *circuit-switched lines*, enterprises shared the use of the service provider's network and paid only for the bandwidth they use. Yet still the problem of privacy in question.

By using VPN technology, VPNs address the security risks of leveraging a public network through the use of a security protocol like IPsec that provides security services for encrypting and authenticating the data. But with the increasing use of IPVPNs, VPN setup and management have been a real problem for network managers for several reasons.

Moreover as the use of more advanced applications and services grows, there will be a need to extend VPN services to integrate many different types of corporate telecommunications traffic including voice, data, video and multi-media. The provisioning of such VPN services will require the deployment of advanced management services.

Detecting a fault caused by configuration errors or getting an indication of a failure in a customer VPN, should also be supported and provided by the service provider who offers VPN services.

For example, a single flaw in security implementation at any point in the network can expose the entire infrastructure, allowing malicious access to important data and files with severe consequences. In this case, management services should be available to interfere immediately in behalf of customers to localize and correct faults in a short time. Usually managing security for distributed networks on a site-by-site basis is time consuming, expensive, and unreliable, putting a big strain on already limited resources, due to the lack of dynamic efficient way in management.

In addition to security needs, there are some main challenges in providing service-enabled network management to support IPVPN services. Here are some reasons why current man-

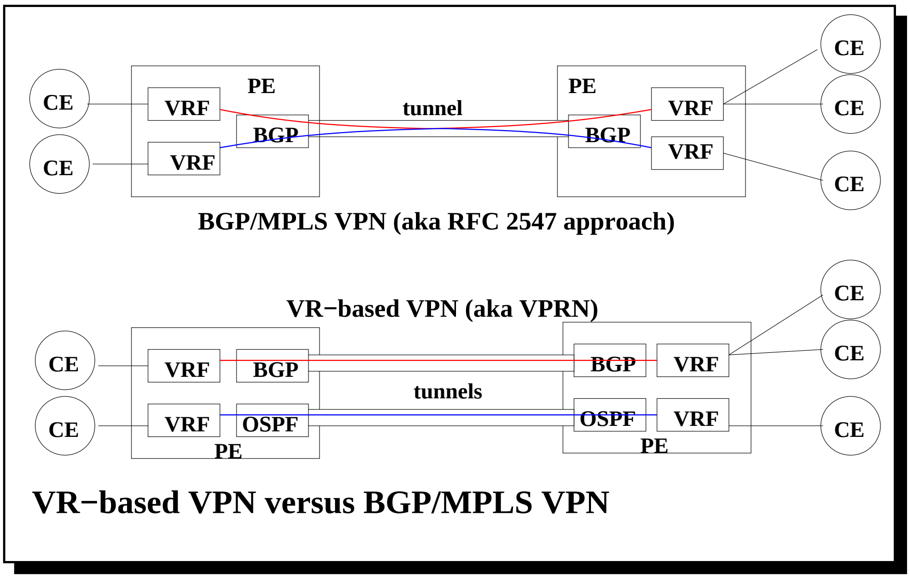


Figure 2.17. BGP/MPLS approach versus VPRN approach

agement services in VPN models have proved problematic for network managers:

- Comprehensive knowledge of the organization of the network topologies: today VPN models almost lack of having a big picture of their customers VPN topologies. Complete knowledge of network appliances is essential to configure them in a coherent way.

- Providing an integrated management of the network and services over heterogeneous technologies and protocols: service must be monitored and controlled, that is, managed, without having to specify lower-level details such as network equipment type, protocols, or management communication and control mechanisms.

- Network topologies change continuously, thus there is a need to a dynamic VPN management service: (1) manual setup is a time-consuming into larger systems installations. (2) Also setup distant VPN devices introduce risks of inconsistent policy settings that leads to incoherent configurations. (3) In addition to all of that, manual setup introduces complexity into larger system installations. Manual management does not fit well with the enterprises needs for dynamic VPNs where sites can join and leave VPNs at any time, so changes in network connections required manual updates of static routing tables.

- Plug and Play network services: without modifying network topologies. Customers should be able to pinpoint service locations easily.

- Support monitoring services: to optimize network performance by understanding traffic activities between customers and operators.

- Support of AAA module (Authentication, Authorization, Accounting): keep a track of all of the movements of the network elements.

- Secure negotiation between customers and operators: security issue is a critical and important point into any management service.
- Coherent deployment of security policies: these policies should contain information related to security issues and firewall rules.

Currently, proprietary and individual methods are often used to offer VPN manage services.

2.6 Partial Conclusions

IPVPN evolutions has been led by several working groups and big organizations like the IETF and the ITU. Yet, there are several fields, which have not been addressed so far, or partially discussed. Management solutions and requirements are still need to be developed and implemented to cover customer business needs and outsource customer management hassles. So far, we have given an overview of the current VPN models, describing their characteristics and applications. Then we analyzed current VPN models regarding management requirements and their actual challenges. In the following chapter, we presents a VPN model offering management services in a flexible and dynamic way.

Chapter 3

Our Dynamic CE-based VPN Approach

Contents

In this chapter, we introduce our dynamic CE-based VPN approach, its characteristics as well as its dynamic management features.

3.1 Introduction

In a well-managed VPN approach, the following features should be covered: provisioning, monitoring, accounting, security and maintenance.
For the rest of this document, we are going to use the term ”VPN site” to refer either to a ”Road Warrior” connected to the office from home or from a hotel, or a fixed personal computer at home, or a remote complex network like shows in figure 11.1. Each VPN site has at least one CE device. We refer to the owner of a VPN site as a ”VPN client” or a ”VPN customer”.
In this chapter, we presents our dynamic CE-based VPN approach. We will see how this approach nicely covers the previous features, while keeping a simple architecture.
The remainder of this chapter is organized as follows: section 3.2 presents our VPN approach in details, giving an overview of its components and its characteristics; the section 3.3 focuses on the benefits we get by applying our approach and how the dynamism in this approach plays a significant role from the management's viewpoint. Finally, we provide some implementation details in section 3.4 and summarize in section 3.5.

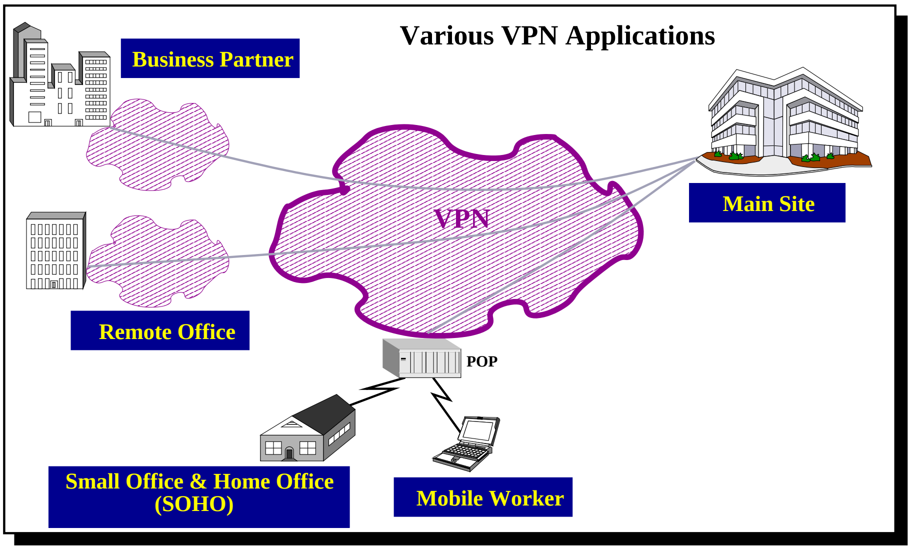


Figure 3.1. Various VPNs Applications

3.2 Architecture

3.2.1 Classification of our Approach

Our approach follows the overlay model of the CE-based VPN type (section 2.4.2 and figure 2.11), since it is independent of any ISP environment. Thus no configurations are needed in the PE and P routers of the ISP core network and the VPN service provider does not necessarily own the ISP core network.

This approach enables VPN service providers to offer IPVPN services to its VPN customers by only provisioning and configuring CE devices on behalf of the customers, even if customers are clients of different ISPs. Thus the CE devices, managed by the VPN service provider, must have a VPN support and maintain one or more VPN tunnel endpoints, in contrary to the PE-based PPVPNs where VPN support is pushed toward the PE routers of the VPN provided networks. CE devices may also have other functionalities, like firewalls, QoS classification, or NAT. Since this approach only relies on CE devices, the ED term in our context refers to CE devices (section 2.2).

Compared with other VPN solutions, our approach operates at a higher level, since it creates an abstraction layer for any WAN transport network and service, and builds a homogeneous infrastructure over heterogeneous technologies. It consists of a core, including physical networks and VPN transport networks, which is surrounded by a logical layer involved in the VPN management services (Figure 11.2). By enforcing management services into our model, VPN customers become unaware of network management hassles and complexity. Also the use of a robust management system in our approach overcomes the overlay model limits regarding the scalability. In section 3.3 we will detail the management aspect of this approach and compare it with that of other VPN models.

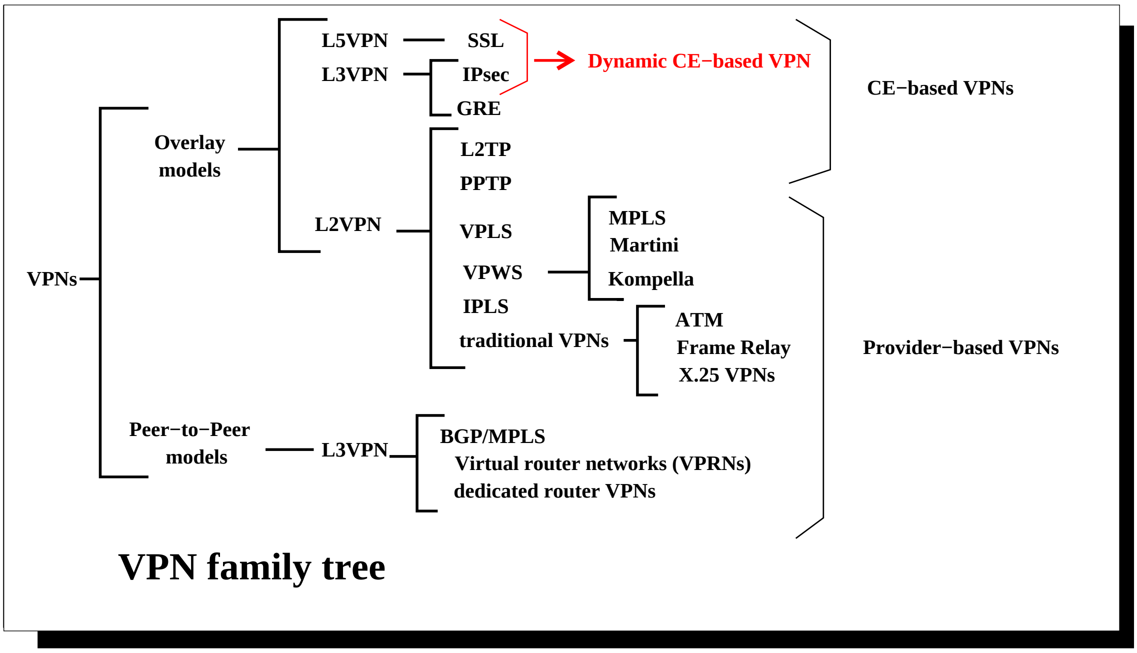


Figure 3.2. Our Approach into the VPN Family Tree

3.2.2 General Description

Basically, our dynamic CE-based VPN approach is a centralized administration management system. Creating a flexible dynamic management system does require a master point to organize the orchestra behind. Customers get access to the VPN services through its central management point, via the Internet.

Customer Related Information

For a VPN to be set up, a VPN customer must first give some information to the service provider:

- If the VPN is of type opened or closed: if the VPN is not limited to a pre-defined set of VPN sites and if any site can join it at any time, we talk about ”Opened VPN”. In the contrary, if the VPN has a set of pre-defined VPN sites and if it is only limited to those sites, then we talk about ”Closed VPN”. It is worth clarifying here that whether the VPN is of type opened or closed, the VPN is managed dynamically (i.e. sites can join or leave the VPN at any time).
- In case of a closed VPN, the VPN customer must provide the list of VPN sites participating to that VPN. On the contrary, in an opened VPN, the VPN customer might or not provide a set of sites participating to that VPN. Since the VPN is opened, the set of VPN sites can be extended at any time.
- The VPN customer can also choose the kind of topology the VPN should use. This

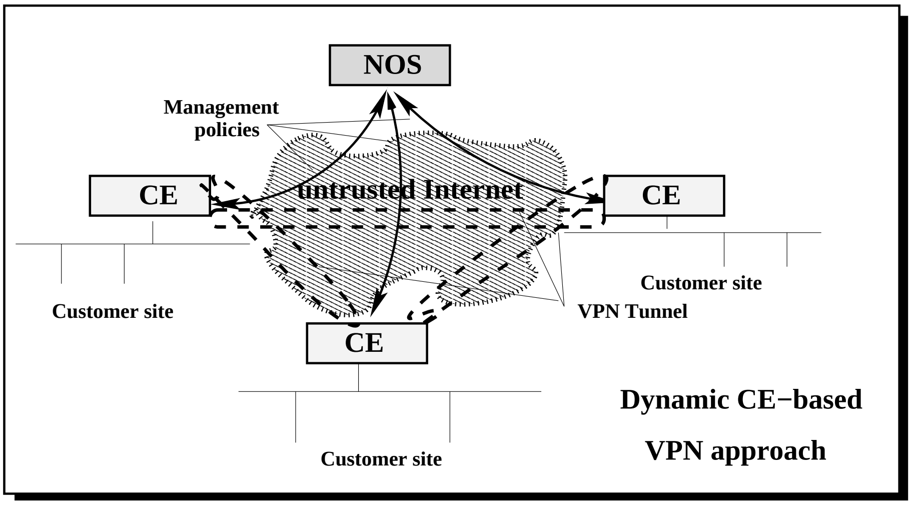


Figure 3.3. Architecture of our CE-based VPN approach.

topology can be a star VPN centered around a given VPN site, or a fully (or partially) meshed VPN. It depends on the customer requirements and traffic patterns.

Depending on the information provided by the VPN customer, the service provider then provisions and manages the customer's VPN. For every CE device in this approach, the service provider configures and maintains information related to the CE device. This includes at least two kinds of policies:

- pre-provisioning policies, for the initial provisioning of the CE device. These policies contain information required for the CE device to open secure connections with the service provider's server.
- configuration policies to secure connections between this CE device and other CE devices.

These configuration policies, and the other information related to CE devices, are maintained in a secured database server, located at the service provider's side.

Core Network versus Logical Layer

Our model includes two parts like shows in figure 11.3:

- Transport Core of the VPN: this part forms the transport layer of the environment where all traffic is transported. It might include diverse interconnection networks, virtual or physical. Since the model is aimed to VPN SPs who only offer a VPN administrative solution to customers, the assumptions made are completely different from that of the PPVPN IETF working group, where the provider, in addition to providing a VPN solution, also masters the core network that can either be Internet or a private interconnection network.
- Network Operation System (NOS) and CEs, at the Periphery: the NOS includes all the operation services. It has a secure VPN database where VPN configuration policies and

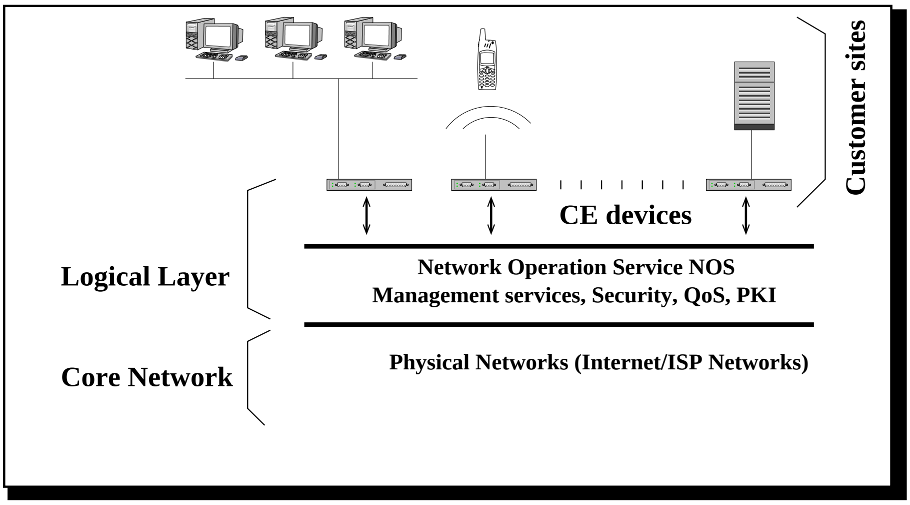

Customer sites
CE devices
Logical Layer
Network Operation Service NOS
Management services, Security, QoS, PKI
Core Network
Physical Networks (Internet/ISP Networks)

Figure 3.4. General Description of our VPN Approach

customers' information are stored. These information are created and updated by the *Virtual Network Operation Center* (VNOC) sub-component of the NOS. The VNOC is the central point that manages VPNs for customers. It creates and distributes policy configurations to the CE of each site. Those CEs are VPN enabled access gateways, and one of them is necessarily present in each site. For each site, the CE is the only gateway who can get access the VNOC, since it's the only element of the customer's site that can be authenticated by the VNOC.

VNOC Functionalities

The VNOC basic functionalities are summarized as follows:

- Complete knowledge of VPN topologies and their customer equipments: the VNOC can get any information related to a CE. During the initial phase, a CE is pre-provisioned to allow the VNOC to access its resources.

- Identification of customers' sites: each CE is given a unique identifier (ID) by the VNOC, that is used each time the CE communicates with the NOS.

- Authentication and authorization of the customers' sites by their IDs: once the CE is authenticated by the VNOC, it is authorized or not to get access to VPN services.

- Creation of policy configurations for each site: these configuration policies include information related to setting up a VPN over a tunneling technology like IPsec. Since the VNOC has information about each CE, it could easily generate a configuration file specific to each type of appliance.

- Deployment of policy configurations for each site: these policies are communicated to CEs over HTTPS. Thus messages sent to/received from the VNOC are secured by the SSL technology.

- Monitoring of VPNs and CEs: the VNOC keeps track of each VPN status and CE activities.

CE/VNOC Messages

The CE/VNOC dialog is based on XML messages sent over HTTPS. A special API has been defined and developed to handle several requests between a CE device and the VNOC. Each CE can issue requests to the VNOC to join, leave or query a VPN. For instance, in case of a "join request", if the site is authorized, the VNOC sends back the VPN configuration to the CE of the site and updates the configuration policies of the other CEs concerned by this VPN and interested in establishing VPN tunnels with that device.
At the beginning, the API was limited only to three basic functions for managing VPNs:

- `JoinVPN` message: the CE device sends this message to the VNOC when the CE device wants to join a VPN whose identification is specified.

- `LeaveVPN` message: the CE device sends this message along with VPN's identification to the VNOC when the CE device wants to leave the VPN.

- `QueryVPN` message: the CE device sends this message along with the VPN's identification to the VNOC who returns information about all CE devices belonging to this VPN.

Security Protocols

IPsec is the security protocol chosen in our VPN approach to secure site-to-site communications (i.e. communications among CE devices), while sites-to-VNOC and VNOC-to-sites communications are secured by SSL connections.
A PKI system can also be a fundamental component of the NOS. With a PKI, the NOS can issue X.509-compliant digital certificates, create key pairs and manage these keys and certificates in behalf of the VPN customers.
Later, in chapter 6, we will define another component, the Management Operation Point (MOP), to refer to the combination of the VNOC and a UDDI repository.

3.2.3 Technical Details

Initial Phase: Initialization of CE Devices

Each CE device must be pre-provisioned with some key information related to the NOS (like the VNOC address or the CA server address if it exists). This information then enables the CE device to contact the NOS, be identified, and obtain further configuration policies.
In our context, pre-provisioning is done by using a simple web console. A VPN customer uses this console to begin the initialization of its CE device. Thanks to this minimal information, the CE device then asks the NOS to further configure it, and the NOS pushes all the required configuration policies to the CE device. In particular the CE device is given a unique identifier, generated dynamically by the NOS, that will be used in all communications between the CE and the NOS.
Once this phase is over, the CE device is now able to contact the NOS and ask for VPN services.

Configuration of CE devices

A CE device that wants to join a specific VPN sends a *JoinVPN* message to the VNOC. The VNOC verifies the legibility of the CE device to participate to that VPN and triggers the suitable action, either by refusing the CE device request or by sending back the configuration policies required to secure connections with other CE devices in that VPN. These configuration policies depend on the information obtained by the VNOC from the VPN customer (e.g. the VPN star versus mesh topology, the opened/closed VPN type, and if it is closed the identifications of CE devices that belong to it). The configuration policies are updated by the VNOC each time there is a new modification related to the CE status or the VPNs to which it takes part.

Once a CE device has obtained the configuration policies for a specific VPN, it can handle the traffic destined to the other CE devices of that VPN. So this traffic can be either discarded or processed according to a suitable SA, as discussed in the following section.

Negotiation of IPsec Parameters

The configuration policies obtained previously contain security policies. These security policies indicate in particular if the traffic destined to a peer CE device must be encrypted by IPsec or not. If yes, then the IPsec parameters must be negotiated thanks to IKE (section 2.3.3).

In fact, each time the IPsec module of the CE device receives some traffic to forward, it checks if any secure channel is already established to the destination. If not the IPsec client starts negotiating an IKE phase 1 exchange. As already explained, the main purpose of IKE phase 1 is to authenticate the IPsec peers and to set up a secure channel between the peers, in order to enable IKE phase 2 exchanges later on.

In our approach, the authentication between two CE devices during the IKE negotiation relies on the use of digital signatures (let us remind that the NOS has sent the necessary information to CE devices during the initial phase). This information contains digital certificates of the CE devices that include the CEs'public keys (issued by a local CA server located at the NOS or by a third party CA server such as VeriSgin), and a digital signature from the CA.

The CA may generate a public key and a private key (a key pair) or the VPN customer applying for a certificate may have to generate their own key pair and send a signed request containing their public key to the CA for validation. The VPN customer applying for a certificate may prefer to generate their own key pair so as to ensure that the private key never leaves their control and as a result is less likely to be available to anyone else.

IKE phase 2 can now start. Its main goal is to negotiate IPsec SAs and generate the required key material for IPsec.

Exchanging Data Between Two CE Devices

After establishing IPsec SAs, an IPsec tunnel can be activated between both CE devices and traffic exchanged. As explained in section 2.3.3, two modes exist for IPsec: transport mode and tunnel mode. In our case, IPsec is used in tunnel mode. So the CE device must first analyze the private IP packets arriving from its attached local site and select/setup an appropriate SA with the appropriate destination CE device. IPsec then authenticates and encrypts the private IP packets according to the rules described in the SA, encapsulates the packet in an IPsec header and re-encapsulates the whole in a new IP header. This new header uses the local CE's non-private address as the source address, and the remote CE's non-private address as the destination address. This IPsec tunnel remains activated and only the first packet sent to the destination CE device experiences the SA and tunnel setup delay, not the following packets. Upon receiving the IPsec packet, the remote CE device decapsulates it forwards it to the appropriate next hop on its own site.

Turning Down an IPsec Tunnel

Deleting an IPsec tunnel can be caused either by the tunnel's lifetime expiration, or by a solicited action. In the first case, IPsec SAs terminate through timing out. This can happen when a specified number of seconds have elapsed, or when a specified number of bytes have passed through the tunnel.

In the second case, IPsec SAs are deleted by the VNOC. This happens when a CE device says he wants to leave a specific VPN by means of a `LeaveVPN` message sent to the VNOC. It also happens when the joining time of a CE device is expired, since joining a VPN is only granted for a certain duration, mentioned in `JoinVPN` message (this mechanism remains optional, though).

When the SAs terminate, the keys are also discarded. When subsequent IPsec SAs are needed for a flow, IKE performs a new phase 2, and, if necessary, a new phase 1 negotiation. A successful negotiation results in new SAs and new keys. New SAs can also be established before the existing SAs expire, in order to avoid any interruption.

3.3 Dynamic Management versus Manual Management

VPN management has become a serious concern, and the need for automated or dynamic management becomes a necessity, especially for large scale VPN implementations. In this section, we compare the management feature of our approach and other VPN technologies.

A successful dynamic VPN management approach should cover the following points:

- VPN device authentication: in order to manage a new VPN device, that device should have its own identification that enables to authenticate it. In a manual management process, a "shared-secret" password technique is used to authenticate the new device with another (known) member of the network. Thus to authenticate each new VPN device, a network manager may have to travel to the distant site or send authentication information via diskette or similar. In the dynamic management approach we describe here, a new VPN device communicates with the VNOC through a secure connection and performs the authentication process remotely via the exchange of digital certificates, based on the X.509 standard. The integrity of the digital certificates is guaranteed and certificates are issued by a trusted third party known as a Certificate Authority (CA). This is an improvement over the password method, but the need for the CA also adds some complexity. As a solution, some systems further automate and simplify the process by permitting X.509 certificates to be burned into VPN devices at the time of manufacture. These devices now have a hardware-strength cryptographic identity, and they avoid the remote security risk of device substitution, which might foil even the CA-issued certificates.

- VPN device authorization: with a manual management process, authorization parameters of a VPN device are stored on distant sites, thus these parameters are vulnerable because someone who accesses VPN devices on distant sites can modify these parameters. In our dynamic approach, authorization parameters are limited to the VNOC who authorizes or not a VPN device to join a VPN. Only the VPN service provide administrators can access these parameters on the VNOC.

- Security policies: these policies contain many elements such as encryption protocol types, hash functions, and key durations (AKA key rollover interval). With a manual management, this process is very hard and security policies should be set on each VPN device. On the opposite, with a dynamic approach, security policies are maintained centrally by the VNOC which insures the coherency of the security policies.

  For instance, a security policy might have several strength levels, each based on predetermined corporate policies, plus a custom setting to override the defaults. The highest-level setting might be used for highly sensitive applications such as transmitting financial information. It might feature (1) the greatest key strength for encryption (3DES, for example), (2) the shortest-duration rollover interval, and (3) per-packet (rather than per-session) authentication.

- VPN device configuration: eventually, with a manual process, each VPN device parameters should be set apart: the VPN device's IP address, the protected network behind the VPN device, and the firewall rules. In a dynamic approach, these parameters are set and sent by the VNOC, thus the probability of error is much smaller.

- VPN provisioning: with a manual process, a network manager must have a comprehensive knowledge of the network topologies to be able to distribute routing configurations and other similar parameters. In the dynamic approach, the provisioning stint is much easier since it takes place centrally, at the VNOC.

- VPN device management: any modification in VPN topologies implies the update of static routing tables in several VPN devices, with a manual management process. In the dynamic approach, this problem is solved by using special messages to signal any modification in the VPN topologies. Upon receiving such a message, the VNOC then takes in charge automatically the update of the VPN.

- VPN device monitoring: as for configuration, in a manual process, each VPN device should be observed apart, otherwise network failures could go undetected (except, of course, by the VPN users). On the opposite, in our approach the VNOC collects data from every device within the VPN cloud by means of SNMP. Thus it monitors network connections and physical devices systematically, as well as IPsec tunnels to assure the logical connectivity through the VPN topology. For instance, VPN devices can now update their own routing tables based on status messages sent regularly from other VPN devices. If a device fails, or if its physical link goes down, its status messages stop and the other devices automatically update their tables to exclude it.

- Transparently for the user: a good management platform should mask useless details to the user who should be kept away from management hassles. In a dynamic environment, this feature is not just covered for clients, but also for network administrators. In such an environment, the management burden becomes much lesser, and the system administrators happier.

- Update dynamics: with a manual process, maintaining a large VPN with hundreds of sites can take days depending on the VPN provider's workload. On the opposite, with the dynamic management approach, changes are implemented in a matter of minutes. The dynamic management approach can scale up to the real-world needs of an aggressive rollout schedule.

3.4 Implementation Details

In this section we give an overview of the technical components that form the NOS.
The VNOC Architecture is subdivided into two technical areas which, by extension, are also called architectures: the System of Exploitation Services (SES), and the System of Management Service (SMS) (figure 11.4). The relationships between the VNOC components and the other NOS components is also shown in this figure.

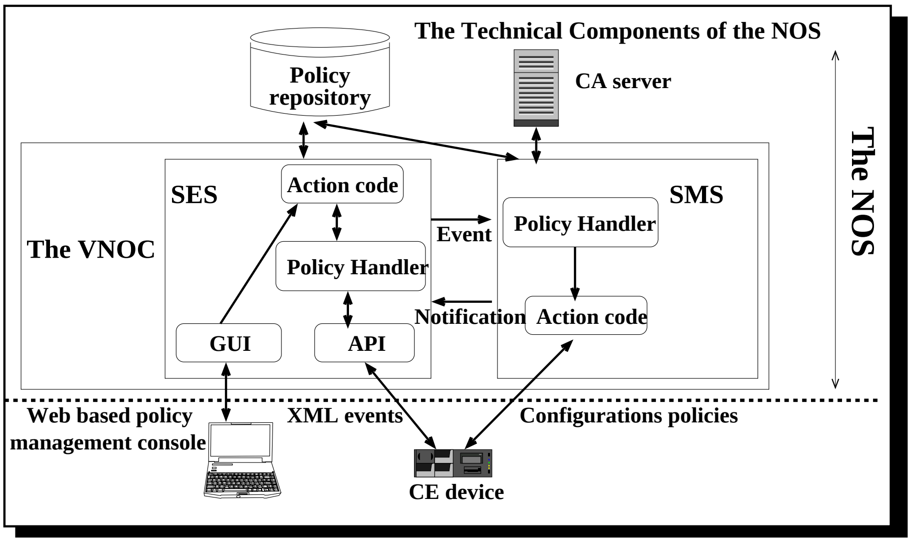


Figure 3.5. Technical Components of the NOS

3.4.1 The SMS Architecture

Essentially, the SMS has four different roles:

- It takes in charge the integration with the CE devices.
- It generates configuration policies related to the information it gets from the SES component and the information stored in the database server.
- It offers an SSH service to secure communications with CE devices.
- It also offers a DNS service to manage domains' names.

The SMS is a multi-threaded server. Services in this server are programmed in C language.

3.4.2 The SES Architecture

The SES is the principal interface of the NOS, since CE devices and the web console used in the initial phase exchange data with the NOS through the SES. The SES has two essential parts:

- The SES has a Graphical User Interface (GUI) that a VPN customer could use to configure a new CE device in the NOS. The GUI is only use during the initial phase of configuring CE devices.

- It also has the API part where it receives/sends XML messages from/to the CE devices. These messages are then treated and forwarded to the SMS where actions could be triggered and configuration policies might be sent back to the CE devices.

From a low level technical point of view, the SES has three layers (figure 11.5):

- Navigation layer: this layer takes in charge the communications with the VPN software of CE devices and web consoles. This layer is accessed by HTTPS with a simple authentication. It is realized with a Web Apache server that directs customers' requests to the suitable applications.

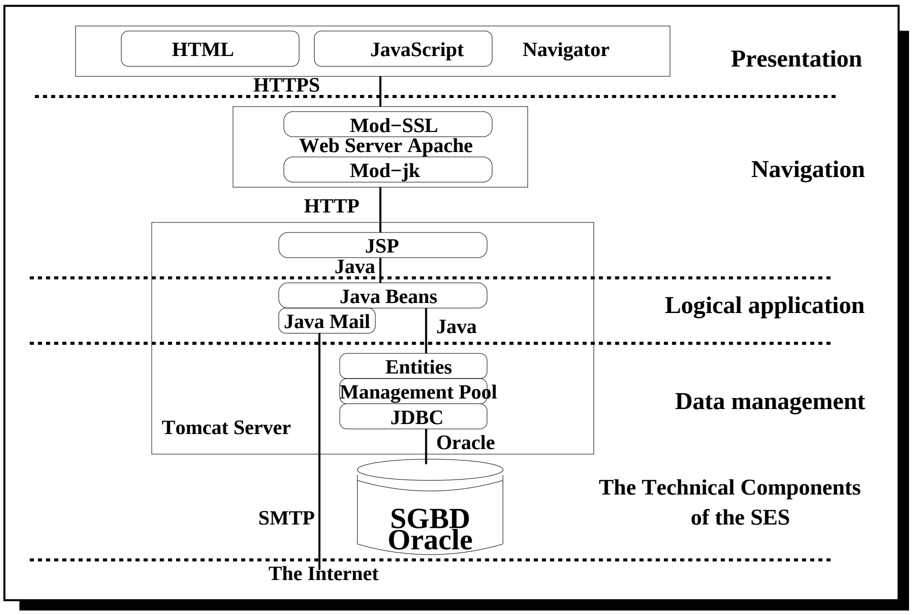


Figure 3.6. The SES Components

- Logical applicative layer: the applicative flow is treated by this layer. This layer does not contain any persistent data. In contrary, it works on information it gets from the next ”data layer” that we are going to describe. The components of this layer are Java Beans.

- Data management layer: unlike the previous layer, this layer contains persistent data. It is based on a Tomcat server with a JDBC component that interacts with the Oracle database server.

The JSP Model 2 architecture is used as a server-side implementation on the SES (figure 3.7).

In this model, the servlet acts as the controller and is in charge of the request processing and the creation of any beans or objects used by the JSP, as well as deciding, depending on

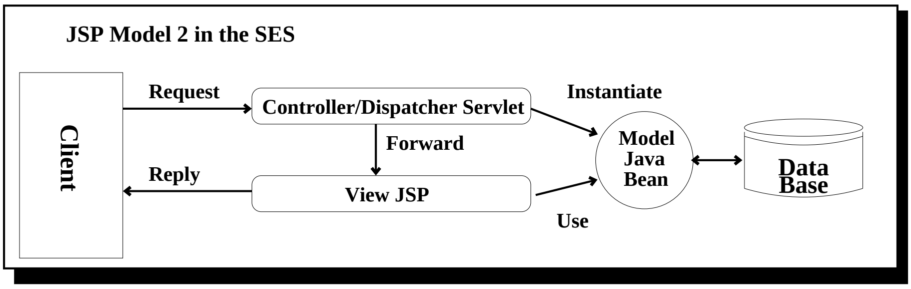


Figure 3.7. JSP Model 2 in the SES

the customer's actions, which JSP page to forward the request to. The Java Bean object is created regarding the parameters sent through the URL or the customer's request. There is no processing logic within the JSP page itself. The JSP is simply responsible for retrieving any objects or beans that may have been previously created by the servlet, and extracting the dynamic content from that servlet for insertion within static templates.
Customers are authenticated by the validation of login/passwords in the customers' requests. The Tomcat server takes in charge this authentication service with the help of the JDBC driver.

3.5 Features of our Approach: a Summary

To summarize, our approach has several distinctive features:

- ISP Independence: this service is provided by a VPN service provider independent of any ISP. This is a major asset since it does not create any ISP dependency (ISPs can change almost transparently) and enables VPN creations between sites connected through different ISPs to the Internet. This is a major asset over other VPN solutions provided by ISPs that heavily rely on their own routing infrastructure.

- Simple centralized approach: the presence of a centralized VNOC greatly simplifies the configuration, management and possibly billing aspects.

- Relying on well-known building blocks: security is always done on a point-to-point basis, using well known security protocols like IPsec (site-site) and SSL (site-VNOC).

- Fully dynamic approach: each CE device sends dynamic requests to the VNOC to join/leave a VPN. Those dynamic requests are done on behalf of clients depending on their own needs.

- Many topologies are possible for site-to-site communications: if we assume by default that a star topology, centered on the service provider, is created, this is not compulsory and other schemes are possible for other communication scenarios. For instance [3] introduces the Routed VPN (or VPRN) concept whereby CEs can route packets between various VPN branches.

- Well-managed system: this approach offers an access mechanism to authenticate and authorize users before they can benefit from the VPN service. It also provides a provisioning tool for the subscription, management, monitoring and billing processes.

> Since VPN configuration and topology are highly dependent upon a customer's organization, the CE device provisioning must address a broad range of customer specific requirements. To ensure that these devices and protocols are provisioned consistently and correctly, web technologies like the XML language are the best candidate to define configuration policies in a standard way (this will be discussed in chapter 6). In this way, all customers information are defined in standard language and stored into a data information repository (e.g. LDAP) within the NOS. When a customer's information is needed, in order to be applied to a CE, the VNOC translates this information into CE specific configurations.

In the next part of this thesis, we analyze the benefits of this approach into several domains. We first show how we can use it to offer group communication services over the Internet in a secure and cost-effective environment. We also show how easily we can secure and manage web services due to our approach. Finally we describe how we can extend this approach to achieve load balancing at the customer's side when the number of VPNs handled by the customer's CE device is increased.

Part II

Some Contributions for the CE-based VPN Approach

Chapter 4

Basic Group Communication Services

Contents

As the Internet grows, communication needs evolve and customers ask for more and more network services. The programmable network approach is one possible solution to quickly adapt existing infrastructures to new requirements. At the same time, group communication is unavoidable when dealing with collaborative work and bulk data distribution applications. Although the deployment of native multicast routing is well behind expectation [29], the important activity around application level multicasting proves there is an important need [17]. But two aspects, that often lack, are the security of communications and the dynamic management of group communication members.

In this chapter:

- We show how programmable networking can be exploited within a VPN environment to offer a secure group communication service across the Internet. In particular, we build a group communication service on top of a fully secure IPVPN environment with the help of our CE-based VPN approach.
- We show how our approach offloads security, management and administration hassles from the multicast members.
- And we propose a new simple Internet VPN Group Management Protocol, IVGMP, as an alternative to traditional multicast routing protocols. We detail our proposal, the IVGMP protocol and we give some insight on its implementation.

4.1 Background

4.1.1 Limitations of IETF Solutions for Supporting Multicast over VPNs

Two categories of related works are worth mentioning here:

Multicast study over VPNs

Our work also departs from that carried out in the PPVPN IETF working group where the VPN Service Provider, in addition to providing a VPN solution, also masters the core network and is an ISP. In general, group communication solutions developed by PPVPN providers can easily and efficiently take advantage of their own provider equipments (e.g. IP routers or MPLS-enabled infrastructure) to offer multicast-capable VPNs [70]. In our case the two entities, the VPN SP and the ISP, are different entities. It avoids ISP dependencies and enables to set up a VPN across sites connected to the Internet via different ISPs, without requiring preliminary ISP agreements and mutual confidence.

Also many efforts have been done recently in the L3VPN working group in order for IP multicast traffic within L3VPNs, in particular within BGP/MPLS IPVPNs, to travel from one VPN site to another [75]. All current work specify protocols to enable multicast traffic through the service providers' core networks and allow providers to interconnect customers mutlicast domains [8]. All proposals assume that the service provider network already supports native IP multicast forwarding [55]. Otherwise no solution is provided to enable multicast over such a VPN. Few implementations (e.g. Cisco and Juniper) are commercially available today to provide multicast in the providers core networks.

The draft [75] extends RFC 2547 VPN functionality to support multicast data transport. The key features of this approach are as follows:

- A multicast domain is a set of VPN Routing and Forwarding tables (VRFs) located on the PE devices and associated with interfaces that can send multicast traffic to each other. A VRF can belong to more than one multicast domain.
- Each multicast domain is associated with an otherwise unused multicast group address from the address space of the service provider network.
- A service provider network instance of PIM-SM runs on each PE router. These PE routers join the service provider multicast groups corresponding to the multicast domains to which they are connected.

In this approach there is a multicast tree for each multicast domain within the service provider network. This multicast tree is referred to as the multicast tunnel for the domain. Customer multicast data is transmitted across the service provider network, after being encapsulated within a packet whose outer destination address is the per-domain multicast group address (since the inner packet will have a customer site specific multicast group address). This encapsulation could use MPLS tunnels or GRE.

Each PE router runs a per-VRF instance of PIM-SM that communicates with the CE device. This PIM-SM instance treats the multicast tunnel as a multidrop interface that reaches all of the other PE routers that are associated with this domain. If a single VRF is in several multicast domains, then PIM-SM requires additional information to make the choice of which multicast tunnel to use for which customer space multicast groups. This information is provided by an out-of-band mechanism. Note that from the point of view of the CE devices, there is no change to PIM-SM or packet forwarding mechanism.

For scalability reasons, this simple mechanism does not consider whether there are actually any receivers in a particular PE routers part of a multicast domain, and which customer-space multicast groups that they are members of. If a PE router is connected to a multicast domain, it will receive all multicast traffic for that multicast domain.

These approaches differ from ours by the fact (1) they are aimed to be used in the service-provider backbones [76], while our approach is only based on edge technology, and (2) they do not address the problem of using PIM along with IPsec.

Security of multicast

Most current multicast implementations of business network services (such as media distribution, distance learning or videoconferencing) offer no security. However, this may become a major requirement to be able to get revenues from multicasting a movie on the Internet, or to prevent eavesdropping. These recent developments in terms of low bit-rate audio and video coding, and in terms of high speed ADSL, modem access network, make applications like Video-on-Demand over the Internet feasible. IP multicast can be used to minimize the network load for streaming services, but there is still one problem to solve concerning the security issues: designing a scalable key management and encryption system for multicast based traffic ([54], [14]). The concept of VPNs based on the IPsec standards suite can be used for this purpose, but this requires replacing the IKE key management with a more suitable protocol, since as we described before, IKE is based on the Diffie-Hellman key agreement protocol and works only for two communicating parties.

The standardization activities of IETF MSEC working group are still in progress [13]. The goal is to standardize protocols for securing group communications within (potentially) very large groups, when IP multicast routing protocols are used. They are trying to define a replacement for IKE. In the current MSEC drafts this goal is achieved by using a central key management authority which distributes the IP multicast keys to all receivers. MSEC focuses only on a scenario with a single sender in each multicast group and many recipients. This is the most commercially interesting scenario, because it covers all radio- and television-like applications.

To conclude, we see that the priorities and assumptions made in the MSEC and PPVPN working groups largely differ from ours. Non-surprisingly, the approaches considered differ and in fact complement each other, by addressing different needs. In the rest of this chapter, we will discuss how our approach meets easily the needs mentioned above, in particular for the multicast deployment of services in commercial environments requiring a high level of security.

4.1.2 Our Approach for Secure Group Communications

The centralized dynamic feature of our VPN approach is well suited to our needs, due to two major points:

- With a central point the VPN service provider can easily take in charge the group security management aspects that not only include authentication and access control but also the cryptographic key management, on behalf of the communication group. In addition to all other management services like deploying and monitoring services.

- VPN topologies in our approach are dynamic since a site can join or leave a VPN at any time, which fits well with the dynamic nature of a multicast group.

Therefore taking advantage of the VPN infrastructure to offer a fully secure group communication service seems reasonable. It must be noticed that *security is managed on a per-sites basis, not on a per-node basis, which is reasonable when threats arise from the Internet.*
In the following section we spot the light on the obstacles we faced when we had tried to use a multicast routing protocol in a VPN environment secured by IPsec.

Limitations of traditional multicast routing protocols in secure IPVPN environments

We first tried to deploy PIM-SM (Sparse Mode) [34], an efficient multicast routing protocol for sparse groups, in an IPVPN environment. We used `pimd` [87], an open-source PIM-SM implementation and tried to deploy it on the VPN edge devices, i.e. the hosts running IPsec in the customer network. We never succeeded for several reasons:

- The Security Associations (SA) management mechanism of the FreeS/Wan [36] IPsec implementation is currently only defined for unicast addresses [74]. This is a problem since a multicast packet, coming either from an application or from PIM-SM (for control purposes), cannot be sent as such in a VPN. For instance a VPN virtual interface built by FreeS/Wan does not set the MULTICAST flag and explicitly checks that no multicast address is used. Yet, there is no fundamental reason for which IPsec could not process multicast packets. There is much activity in the MSEC IETF working group on supporting multicast addresses in a SA, yet this is not finalized and not reflected in the current IPsec/IKE implementations [20].

- PIM-SM and IPsec ignore each other. For instance the FreeS/Wan [36] IPsec implementation manages its own routing table that cannot be seen by PIM, and there is no way to force PIM-SM packets to go through IPsec virtual interfaces. The same limitations apply to the PIM-DM (Dense Mode) and DVMRP [90] protocols.

An alternative solution could be to deploy a second host in each customer network (in addition to the VPN edge device), controlled by the VPN SP, and to deploy a multicast routing protocol supporting IP-in-IP tunneling on it (e.g. the `mrouted` DVMRP implementation). In that case only unicast tunneled packets are sent through the IPsec tunnel, which solve the problems mentioned above. This solution is not satisfactory though, since (1) it requires that *two hosts*, controlled by the VPN SP, be deployed in each customer network, (2) it generates additional traffic on the customer network (packets cross the LAN twice), and (3) it requires an additional encapsulation.
Two additional more fundamental flaws exist:

- These protocols have been developed to solve the general multicast routing problem over complex networks composed of many subnets, whereas a VPN environment creates a simple overlay network that may even be fully meshedBurner99.

- More fundamentally, from a VPN management point of view, the multicast routing protocol approach is not satisfying as it decouples the group communication service from the VPN management service. Therefore the VNOC has limited control and accounting capabilities.

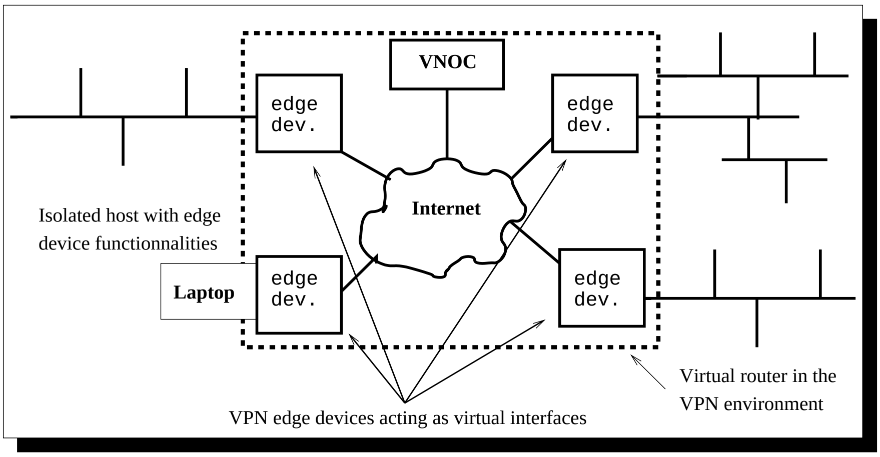

VNOC
edge dev.
edge dev.
Internet
Isolated host with edge device functionnalities
Laptop
edge dev.
edge dev.
Virtual router in the VPN environment
VPN edge devices acting as virtual interfaces

Figure 4.1. The virtual router concept.

4.2 Our Approach: IVGMP Architecture

4.2.1 The Virtual Router Concept

The CE VPN edge device within each site insures the forwarding of multicast packets issued from the local site over the Internet towards other sites participating in the VPN and vice-versa. Therefore the VPN interconnecting the various CE devices can be modeled as a single virtual router (See figure 12.1) with several virtual interfaces, one per CE device. Within this virtual router, we introduce the IVGMP protocol to manage the configuration of the virtual interfaces and the forwarding of multicast packets. IVGMP is an alternative to traditional multicast routing protocols that catches the specificities of a programmable IPVPN environment.

4.2.2 IVGMP Detailed Description

Adding a New Receiving Site to a VPN

We first assume that each site is composed of a single LAN. In order to discover new local group members, the IVGMP protocol running on the CE device relies on IGMP (Internet Group Management Protocol) and its Query/Report mechanism [33]. This mechanism is used both (1) to discover members of new groups for which a new branch must be created in a VPN and (2) to dismantle VPN branches for groups having no member any more in the site. This is done by listening to IGMP traffic on the site's LAN (See figure 12.2). In order to know if a new VPN branch is needed when an IGMP Report for group G is listened, each CE device maintains the list of multicast groups in which it already participates. In case of a new group, the CE device issues a dedicated VPN command, JOIN_VPN(G), to the VNOC. On receiving a JOIN_VPN request, the VNOC performs some policy checking to determine if this site is authorized to subscribe to this group. A confirmation message is then sent back to the CE Device and the VNOC automatically distributes the new management policies to all the CE devices. Some accounting operations may also be performed during this process (e.g. to do per group subscription billing). Finally the CE device who joined the VPN asks the VNOC for some additional information (e.g. the list of sites participating in this VPN).

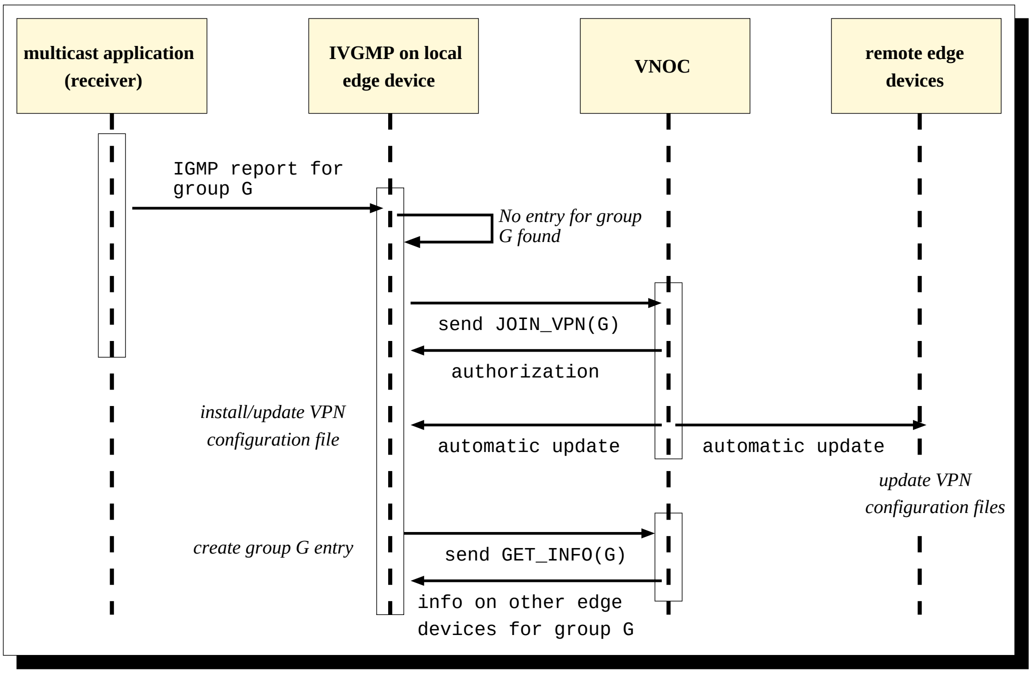

multicast application (receiver)
IVGMP on local edge device
VNOC
remote edge devices
IGMP report for group G
No entry for group G found
send JOIN_VPN(G)
authorization
install/update VPN configuration file
automatic update
automatic update
update VPN configuration files
create group G entry
send GET_INFO(G)
info on other edge devices for group G

Figure 4.2. Joining a Multicast Group as a Client.

Adding a New Sending Site to a VPN

A similar process is used to manage multicast sources. In that case no IGMP message is issued by the application and a sending host will not respond to IGMP Queries either. Thus a CE device has to listen to all multicast packets coming from the local site, check if a new branch is needed for that multicast group, G, and finally issue a JOIN_VPN(G) to the VNOC as described above.

IVGMP and Multicast Routing Protocols Interoperability

When a site is composed of several subnets, a multicast routing protocol is needed. In that case [38]:

receiver problem: the CE device will not receive IGMP messages sent by group members located in inner subnets not directly attached to him.

source problem: likewise the "top" multicast router (i.e. located on the same subnet as the edge device) will not forward any traffic coming from an inner source to this CE device if there is no receiver on the CE device's subnet.

Several solutions are possible to solve these problems:

- one can use IGMP-proxying [35] to let IGMP Report messages be forwarded by routers up to the CE device. This solution has the drawback of requiring some administration work in the site which is not always possible and desirable. Besides this solution only solves the "receiver problem".
- when there is a *small number of pre-defined multicast addresses* that can be used between the VPN sites, IVGMP can pro-actively subscribe to these groups (i.e. send

IGMP reports) each time the top multicast router performs an IGMP query. Therefore the multicast traffic from inner sources, if any, will flow up to him. This solution only solves the “source problem” but can be used along with the IGMP-proxying solution. A major drawback is the useless state information created in routers and the increased IGMP signaling.

- another possible solution is based on a dedicated application used by users that start a sending or receiving application to inform the local IVGMP of the presence of new multicast groups (“source problem”) and/or receivers (“receiver problem”). Then IVGMP can then contact the VNOC accordingly and subscribe to this group (to receive multicast traffic) if required. This solutions does not require any modification to the internal site but puts some burden on users. To avoid problems, the announcement is only valid for a limited span of time (and should be re-issued when required).

4.2.3 IVGMP Critical Appraisal

The IVGMP approach brings several distinctive benefits compared to traditional group communication approaches:

- Simple but efficient security management: security is done on a point-to-point basis, using the well known and operational IPsec/IKE framework. The security protocols dedicated to group communications currently being defined in the IETF MSEC working group are not needed.
- Centralized approach: the presence of a VNOC simplifies the configuration, management, and possibly billing aspects. Note that this VNOC is anyway needed for point-to-point IPVPN management.
- Many communication topologies are possible: so far we assumed that a star topology, centered on the sending site, was created. This is not compulsory and other schemes are possible. For instance the sending site(s) may forward traffic to a security certified node (the VNOC should have such an accreditation) that acts as a reflector to other receiving sites. This solution enables a sending site having limited upstream bandwidth to disseminate data to a large number of receivers without sacrificing security. More elaborated topologies can be envisioned, leading to the notion of VPRN, or Virtual Private Routed Networks [38] (see below).
- No dependency on inter-domain multicast routing: our solution only assumes the presence of a unicast routing service in the core network. This is an advantage in front of the slow deployment of inter-domain multicast routing [29].
- Compatible with other VPN types: our approach, since it is based on edge technology, can be used with other VPN types such as MPLS VPN environment when QoS guarantees are needed.

IVGMP also has some limitations:

- Lower communication efficiency than native multicast routing: any solution based on traffic duplication at the edge is non-surprisingly less efficient than a solution based on traffic duplication in the core network as native multicast routing protocols do. The same remark can be done to application level group communication schemes that share some similarities with our approach.

- Scalability problem: this is a direct consequence of the lower communication efficiency ([92] [31] [56]).

The scalability problem can be addressed by provisioning some sites (either a subset of the receiving sites or trusted third-party sites) as VPRN nodes, i.e. nodes that can perform traffic forwarding. Each VPRN node belongs to two VPNs for the same multicast group, one including the sending site and another one including one or more receiving sites. A hierarchy of interconnected VPNs is thus created for each multicast group, leading to a distribution of the networking load among the set of VPRN nodes. The scalability of the solution is increased while keeping a centralized (around the VNOC) management.

4.3 Implementation Aspects

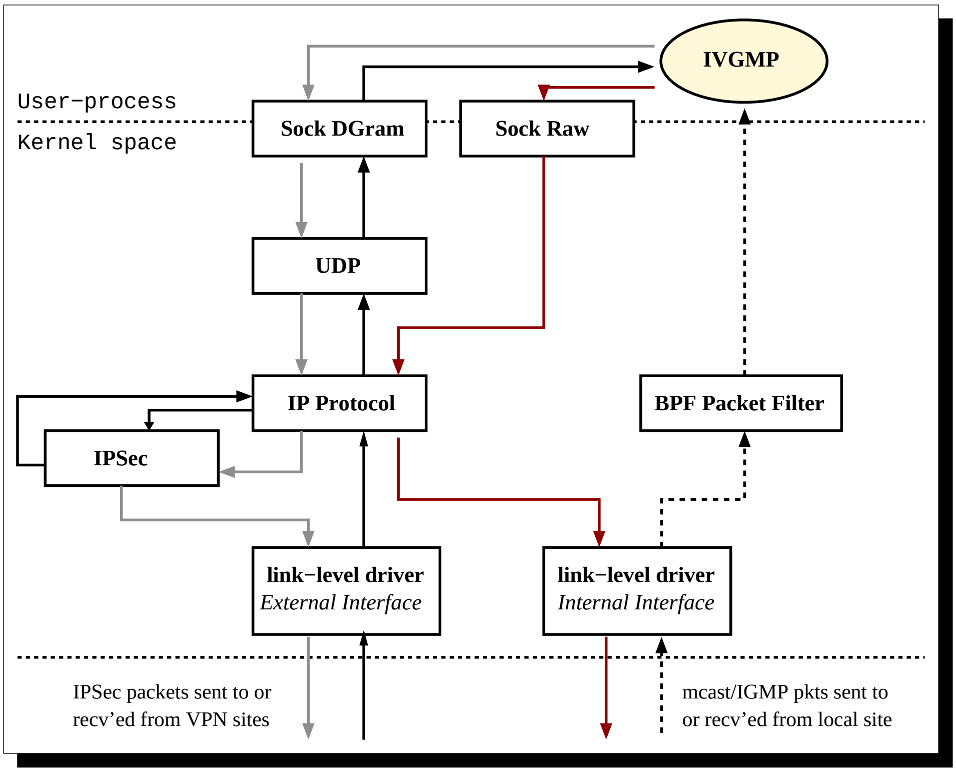


Figure 4.3. An architectural view of packet processing in IVGMP.

We have implemented the IVGMP protocol and integrated it in a PC/Linux edge device (See figure 12.3) [91]. We use a modified version of the FreeS/Wan IPsec implementation [36], the `isakmpd` implementation of IKE for OpenBSD [42], and Netcelo's VPN administration tools / VNOC [63].

The IVGMP-VNOC communication, required for instance to maintain the multicast enabled VPN list, is based on the Simple Object Application Protocol (SOAP) web service technology which offers major advantages in terms of interoperability support and firewall/proxy friendliness. More about web services in 6.

Packets are processed as follows in each CE device: a multicast packet coming from an active source in the local site is first intercepted by the BPF packet filter [48] running in the CE device and sent to the IVGMP daemon. IVGMP looks for a VPN entry that matches the destination multicast address to decide whether or not the packet should be sent to the other sites. If an entry is found, a copy of the packet is encapsulated in a unicast UDP/IP datagram for each remote site (we use a UDP encapsulation in the current prototype for simplicity), given to IPsec and sent through a tunnel to the remote CE device. In the other direction a packet coming from a remote VPN site is successively processed by IPsec, IP, UDP, and then IVGMP. This latter finally injects the original multicast packet in the local site through a raw socket.

This implementation enabled us to validate the IVGMP concept. Some values were measured related to the API used in the CE-VNOC communications 6. However other performance aspects and possible optimizations are left for future work, we focus essentially on architectural aspects.

4.4 Partial Conclusions

The approach discussed in this chapter fuses miscellaneous sparse technologies like IPVPNs, web services (SOAP), programmable networks, in the same melting pot to get out with a simple flexible way to offer fully secure group communication services over the Internet [5]. We have described IVGMP, a new solution for offering a group communication service based on the programmable IPVPN technology. The nature of this solution ensures its robustness, flexibility and full security. This solution departs from the traditional approach since it brings a group communication service on top of a fully secure infrastructure rather than the contrary.

Finally, our approach shares some similarities with the Centralized Multicast (CM) approach [51]. In CM, the data forwarding and control functions are kept separated, and the control part is centralized in distinct control elements. The control elements are arranged in a two-level hierarchy within autonomous systems and are used to set up multicast trees. In our approach too, the control part is centralized in the VNOC. The major difference yet it that CM does not address security.

Chapter 5

Using Application Level Multicast for Improved Group Communication Efficiency: the VPRN Concept

Contents

This chapter is a follow-up of the previous work on group communications in a VPN environment 4 and on application-level multicast.

In this chapter we show how to extend and optimize our approach discussed in the previous chapter, to offer full secure but efficient group communication service between several sites. This extension is based on two notions: the concept of Virtual Private Routed Network (VPRN) and an innovative application-level multicast protocol called Host-Based Multicast (HBM).

As we will discuss later the solution in this chapter is a step forward to overcome scalability issue regarding our previous solution in 4.

The rest of this chapter is organized as follows: section 5.1.2 gives a quick background and details both the concept of VPRN and the HBM overlay multicast proposal; section 5.2 discusses the merge of the IVGMP and HBM protocols in order to create a multicast-enabled

VPRN; section 5.3 introduces a performance evaluation that highlights the benefits of the concept; section 5.4 introduces related works, and finally we conclude this chapter.

5.1 Background

Many applications like collaborative work applications and bulk data distribution require an efficient group communication service. It is the only viable solution when network resources must be preserved, either because of their scarcity, of the large amount of data transmitted, or the high number of receivers. If intra-domain multicast (within a LAN or a site) is widely available, this is different for inter-domain multicast. Today many ISPs are still reluctant to provide a wide-area multicast routing service [29]. The important activity around application-level multicasting [32], that most of the time try to offer a pragmatic alternative group communication service when there is no native multicast routing, proves there is an important need. The idea is to build an application-level overlay topology, made of point-to-point tunnels between the group members, over which data is distributed. This is the reason why an application-level multicast approach is also called Overlay Multicast. In the remaining of this chapter both names are used indifferently.

But an aspect that lacks in the overlay multicast proposals is security. In the previous chapter [5] we have shown how to build a group communication service on top of a fully secure IPVPN environment. In this chapter we show how to further improve the distribution efficiency of this solution and how to reduce the stress laid on the physical infrastructure thanks to the help of application-level multicast techniques.

5.1.1 Limitations of our CE-based VPN approach

Security Versus Scalability

We believe that this approach meets many needs, in particular for the deployment of services in commercial and competitive environments requiring a high level of security. A typical example is a headquarter that needs to distribute a large confidential database to its remote offices. In this case the number of sites concerned is limited (a few tens), but communications must be fully secure (i.e. the source must be authenticated, the content encrypted and the integrity verified).

In this example, typical of the problem we address, security is the primary concern, not scalability. Contrary to the MSEC working group where scalability is also considered into its priorities.

The number of sites that take part in the VPN is limited, at most a few hundreds and usually only a few tens. Therefore having a centralized approach for VPN management (and also for overlay multicast management as we will see later on) is by no means an issue. Besides, within each site, the number of nodes, senders or receivers, is not limited, which largely increase the effective scalability (in terms of nodes). Finally the scalability in terms of the number of VPNs (rather than the number of sites for each VPN) is a different issue that can easily be addressed by having several VNOC.

Limits of the IVGMP Architecture

In the previous chapter, we proposed the Internet VPN Group Management Protocol (IVGMP) implemented on each CE device. The IVGMP functionalities are summarized by discovering multicast group members and sources located in the VPN sites and by checking the legibility of CE devices to join multicast groups. The IVGMP communicates with the VNOC to check the authorization issues of CE devices. A limitation though is that traffic replication is performed by the CE attached to the source, when the CE sends traffic to many receivers. Therefore when the number of remote sites increases, the performances quickly degrade. This is the reason why this chapter introduces the VPRN concept along with an overlay multicast solution to improve this efficiency.

5.1.2 The Benefits of the VPRN and Overlay Multicast Solutions

The Traditional VPRN Concept Versus our View of a VPRN

Many aspects introduced by VPNs have a direct analogue with those of physical networks. One of them is the way in which VPN sites are connected together and traffic is forwarded. If a fully meshed topology between the various sites is feasible, it is not the only possibility and creating a non-fully connected topology can be highly beneficial in some situations. It naturally leads to the concept of Virtual Private Routed Network, or VPRN, which emulates a multi-site wide area routed network using IP facilities [81].

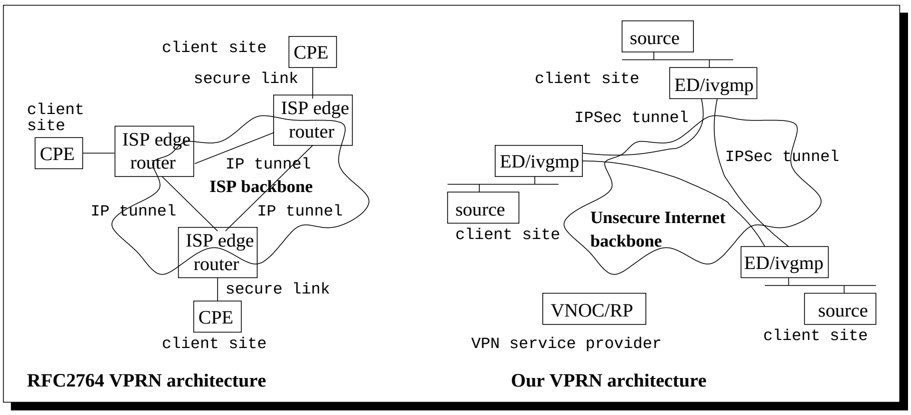


Figure 5.1. RFC 2764 versus ours VPRN architecture for group communications.

The traditional VPRN model described in RFC 2764 [38] (figure 5.1-left) considers a provider network as an opaque IP cloud where only nodes on cloud border are part of VPN description; nodes within the cloud are transparent. Users access the network via a Customer Premises Equipment (CPE) router,which is a router connecting the customer internal network to the provider's edge router, using a non-shared secure link. The provider is responsible for establishing a mesh of tunnels between the provider's edge routers that have at least one attached CPE belonging to a given VPN. This mesh represents a new dedicated network that virtualizes the physical one. Conceptually, there is a dedicated mesh per VPN, and the mesh topology is arbitrary (partially or fully meshed, depending on customer needs). The main benefit of this approach is that it moves the complexity and the configuration tasks from the CPE router to the provider's edge router. Besides the mechanisms proposed intrinsically rely on the features provided by the underlying physical infrastructure (most of the time an MPLS network).

The VPRN model discussed in this chapter differs quite a lot from the previous model. In our case (figure 5.1-right) the CE device located in each customer's site behaves as a VPRN node. The complexity and configuration tasks remain hidden to the customer since these CEs are remotely managed by the VNOC. Another difference is that the CE/VPRN nodes have a more dynamic nature (compared to an ISP edge router) and are concerned by group management (e.g. by discovering local sources and receivers with IVGMP). Besides, our

VPRN architecture is built on top of a generic IP network, without making any assumption on the underlying physical infrastructure. Likewise it does not need any ISP agreement when sites are connected through different ISPs, which is a big asset.

The Overlay Multicast Concept and the HBM Protocol

The usual Overlay Multicast goal is to offer an alternative to the lack of deployment of inter-domain multicast routing. Many protocols, largely different in their approach, have been defined, but they all share some similarities that distinguish them from traditional multicast routing [32]:

- A forwarding node in the overlay topology can be either a end-host (i.e. running the application), a dedicated server within a site, or a border router.
- With an overlay topology, the underlying physical topology is completely hidden. A directed (or often undirected) virtual graph is created between all the nodes, and metric measurements taken between these nodes (or a subset).
- In traditional multicast, the membership knowledge is distributed in the multicast routers. With an overlay multicast, group members are known either by a Rendez-vous Point (or RP), by the source, by everybody, or is distributed among members (e.g. for increased scalability).
- The overlay topology is potentially under complete control. In particular the topology creation process is often optimized using the distance metrics collected between the nodes.

In [73] the Host-Based Multicast (HBM) protocol is defined. This protocol automatically creates a virtual overlay topology between the various group members, using point-to-point UDP tunnels between them. Everything is under the control of a Rendez-vous Point, or RP. This RP knows the members, their features, and the communication costs between them. He is responsible of the topology calculation and its dissemination among group members.

The data distribution efficiency highly depends on the quality of the distribution tree. This is addressed by the periodic node-to-node (in our case site-to-site) measurements performed by HBM nodes and that are communicated to the RP. This latter then create the distribution topology, using the available metrics. If existing solvers can easily create an optimal topology, in practice the metric database only gives a partial, more or less outdated, view of the networking conditions. Yet we assume that the resulting topology is reasonably good. By default, a shared shortest-path tree is created, but other topologies are possible, for instance a per-source tree when the application is known to be single-source.

5.2 Our Approach: The IVGMP/HBM Architecture

We have described so far the various concepts and protocols. In this section we describe how they nicely fit with one another.

5.2.1 General Architecture

The VPN approach considered so far is centralized around the VNOC. Thanks to the IVGMP protocol running on each CE device, the VNOC is also responsible of collecting and distributing configuration policies and membership information (in terms of sites) for each multicast group. The HBM protocol also assumes the presence of a central RP which collects membership and distance information, and performs topology creation. Therefore *it is natural to merge the various features and add a RP functionality to the VNOC* (figure 5.2).

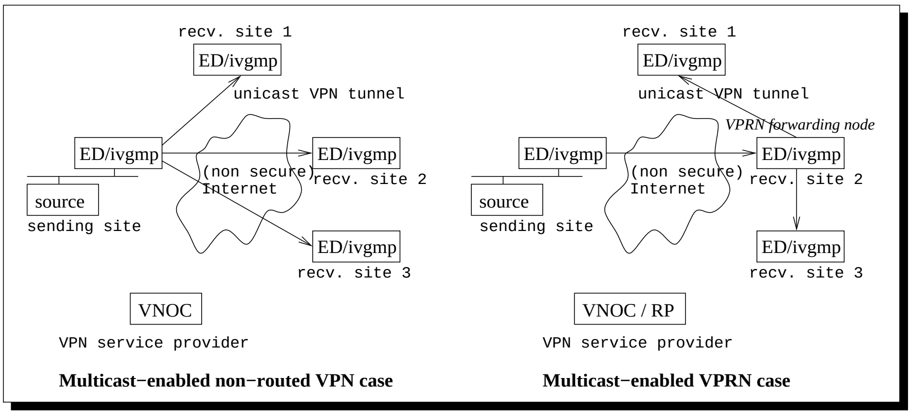


Figure 5.2. The non-routed versus VPRN approaches for group communications in a VPN environment.

Improved Scalability

Each VPN site can now act as a VPRN node and can forward traffic to its neighbors in the topology. Doing so reduces the fan-out of the site where the source settles, and because it removes a hot spot in the network, the scalability (in terms of number of sites) is significantly improved.

Dynamic Aspects

The group membership dynamic triggers both VPN updates (e.g. by removing tunnels set-up to/from sites that no longer participate in the group) and VPRN/distribution topology updates (e.g. to avoid forwarding traffic to the site that left the group). There is a risk of topology partitioning (and of packet losses) when a forwarding site leaves the group, until the topology is updated. Yet, and this is a major difference with the Overlay Multicast general case, nodes considered here are well administered routers (the CE), instead of end hosts that are far less stable. Therefore the CE departures are almost always negotiated, and appropriate measures can easily be taken.

CE-to-VNOC/RP Security

Our architecture gives security the priority over other services. If site-to-site security is addressed by IPsec, site-to-VNOC (or RP since the VNOC acts as a RP too) communications must also be secure. To that goal, each CE device establishes a secure communication channel with the VNOC based on SSL and certificates. The CE and the VNOC first authenticate one another, and the establish a secure SSL connection. Remote configuration and other control operations can then take place, using the SOAP approach [16].
The VNOC must be able to deploy all the IPsec features and ensure that key exchange can be handled properly on the VPN CEs. In order to enable secure communications between two CE devices, the VNOC supports a framework for automatic key management, IKE [44].

The IPsec Security Associations (or SAs) generated dynamically by the VNOC, are created between two VPN sites to exchange keys as well as any details on the cryptographic algorithms that will be used during a session.

5.2.2 Detailed Description

A more detailed analysis of the VPRN/IVGMP/HBM integration exhibits several slight differences with the initial HBM proposal. In this section we detail the operation of HBM in this environment and highlight the specificities related to its integration in a VPRN environment.

CE device Functionalities

In addition to its VPN and IVGMP functionalities, a CE device now needs to participate in the metric evaluation with the other CEs of the group. The list of such CEs is necessarily present on each CE since IPsec tunnels are created between them. This is in line with the full membership knowledge assumption of HBM where each node potentially knows all other nodes.

By default, metric evaluation consists in issuing `ping ECHO_REQ/ECHO_REPLY` messages within the IPsec tunnels. It assumes the presence of a fully meshed VPN, otherwise some destinations could not be reached. This is a reasonable assumption since each tunnel between two CEs is in fact shared by all the unicast or multicast traffic between them, and there is a high probability that both sites have already exchanged some packets before.

The metrics are periodically and asynchronously collected by the CE, and sent to the VNOC/RP using the secure SSL channel. Once again, using SOAP is in line with the XML approach used by HBM for control messages.

A CE device is rather different from the end-host assumed in HBM:

- a CE device is rather stable when compared to a traditional end-host (usually a PC). A CE is a well administered router, that rarely reboots or crashes (at least in theory). This feature greatly improves the overlay multicast solution, since node stability, especially in case of a forwarding node, is of high importance.
- a CE is a small embedded PC, usually running a dedicated Linux OS, and has less processing power than a traditional end-host. The VPRN approach adds some processing on the CE (metric evaluation, packet forwarding), yet most of the work (topology creation, database management and configuration distribution) is performed by the VNOC, not the CE.

VNOC/RP Functionalities

Periodically the VNOC/RP calculates a new topology, taking into account new networking conditions (e.g. a congested path between two CE devices can lead the VNOC/RP to find an alternate path). This topology update is also performed in case of membership modification (e.g. when a new site joins the group). The new topology is then communicated to the concerned CE. Each topology update message consists in an updated VPRN configuration, instead of the new list of neighbors of a node as in HBM.

A major difference with the initial HBM proposal is the fact that group departure is by default implicit, since a site always subscribes to a VPN for a limited span of time. Because of this soft-state approach, each CE must periodically subscribe to the group, sending a new `JOIN_GROUP` message to the VNOC, otherwise the site is automatically removed. This is different from the original HBM proposal which follows an explicit leave model, plus a partition recovery mechanism in case of ungraceful departures (e.g. after a crash). Having a soft-state model enables the VNOC/RP to asks a CE device that implicitly leaves a group to keep on forwarding packets until the new topology has been updated.

5.3 Performance Evaluations

5.3.1 Experimental Conditions

We implemented both the IVGMP and HBM protocols in C++/Perl on PC/Linux machines, and carried out several experiments to assess the benefits of the VPRN approach. The experiments reported in this paper are simulations based on a large interconnection transit-stub network, composed of 600 core routers, 3nd generated by the Georgia Tech Model (GT-ITM) [94]. Some of these routers are interconnection routers, others, at the leaf of the topology, are access routers connecting the client sites. We then choose $N \in 10; 243$ sites randomly among the 243 possible leaves.
We compared (1) the multi-unicast solution, where the CE of the sending site unicasts a copy of each packet to the $N - 1$ remote sites of the VPN, which corresponds to the non routed VPN approach, and (2) the VPRN solution where an optimized shared tree is created between the N sites of the VPN. This tree is constrained so that the maximum degree of each node is 6.

5.3.2 Metrics Considered

The quality of the topology is judged with several metrics [95]. Some of them evaluate the *resource consumption*:

Cost: this is the sum of the *delay* over all the physical links of the distribution topology. With the multi-unicast case, this is the average for all possible sources among the N possibilities since there is a different distribution topology per source. With the shared tree case, by definition the distribution topology remains the same for all possible sources.

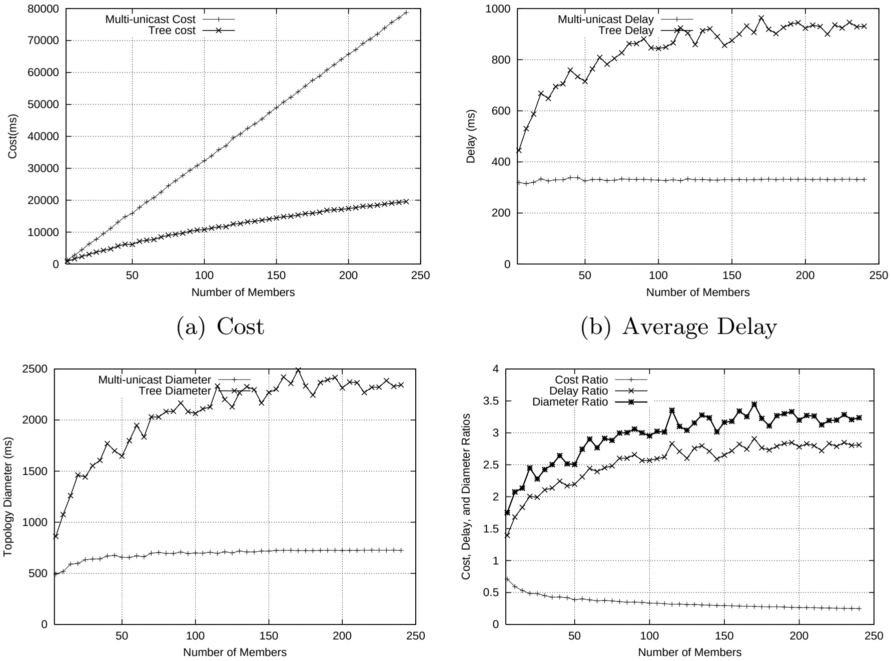
Link Stress: this is number of copies of a given packet that cross a given physical link. The ideal stress, only achieved with native multicast routing is 1. The maximum stress is an important information since it highlights the presence of hot spots within the physical network.

while others evaluate *performances*:

Average Delay: this is the average delay between two sites over the distribution topology. With the multi-unicast case, this is the average for all possible sources.

Diameter: this is the maximum delay between the two farthest sites over the distribution topology. The optimal case is achieved with the multi-unicast solution since all paths between the source and receivers are direct.

Each point in the figures (except for the stress distribution figure) is the average over N sites chosen randomly among the 243 possibilities. Note that during simulations, the physical links are only limited by their unidirectional delay (assumed constant), not by their bandwidth, which leads to underestimate the effect of the link stress.



(a) Cost

(b) Average Delay

(c) Diameter

(d) $\frac{shared\_tree}{multi\_unicast}$ Ratios

Figure 5.3. Multi-unicast versus constrained shared tree comparison

5.3.3 Results and Discussion

Figures 5.3 (a)-(d) and 5.4 (a)-(d) compare performance metrics (average delay and diameter) and resource consumption metrics (cost and link stress) with the multi-unicast and constrained shared tree topologies. Figure 5.4 (c) is a zoom that emphasizes the tail distribution, but that does not show that most links have a stress at most equal to 5, and 325 links a stress equal to 1.

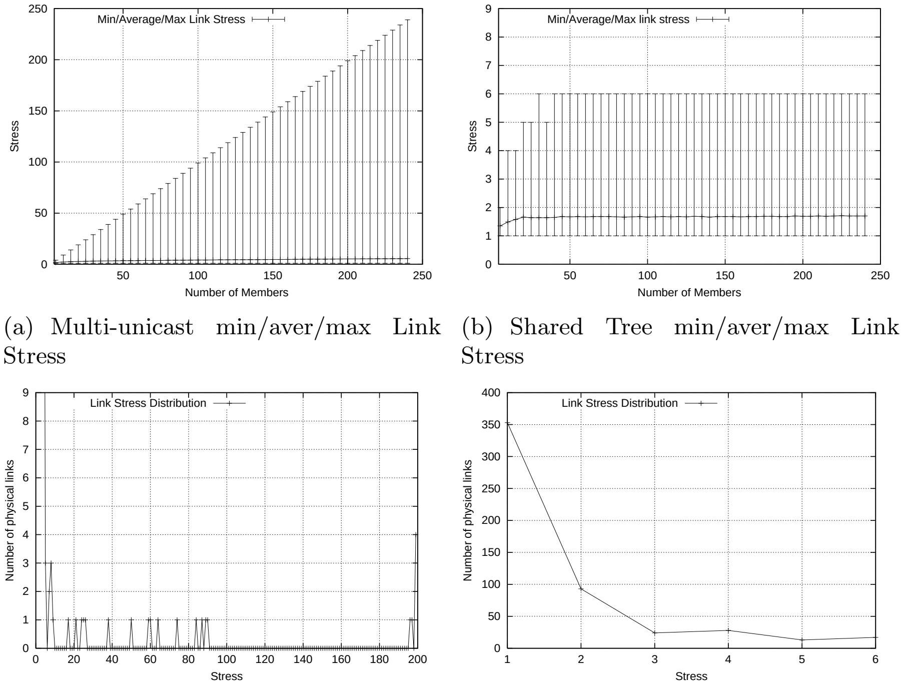
The multi-unicast topology is quickly limited by the fan-out of the sending site (equal to $N - 1$) which creates a high stress on the first few links. This is visible on figures 5.4-(a) (look at the maximum stress) and (d) (stress distribution, showing a tail distribution at $x = N - 1$). Consequently, the multi-unicast approach is definitely not a reasonable solution when there are more than a few tens sites (especially as these tests underestimates the effects of the link stress).

On the opposite the constrained tree, by construction, limits this maximum stress (at most 6 neighbors in these experiments), no matter what is the number of sites, N. The price to pay is a higher maximum delay/diameter of the topology, but experiments show that the stretch factor when compared to the unicast case remains inferior to 3.5 which is fairly reasonable.

Note that in case of a single source application, the distribution topology can be optimized using a per-source tree. This is made possible by the total control over the topology at the VNOC/RP. The only difficulty is inform the VNOC of this application specific feature.

5.4 Partial Conclusions

In this paper we only considered degree-bounded shared tree overlay topologies. Other overlay topologies are possible. For instance [82] discusses several algorithms/heuristics to optimize both the topology diameter and bandwidth usage at the overlay nodes.



(a) Multi-unicast min/aver/max Link Stress

(b) Shared Tree min/aver/max Link Stress

(c) Multi-unicast link stress distribution with N = 200 sites (ZOOM)

(d) Shared tree link stress distribution with N = 200 sites

Figure 5.4. Multi-unicast versus constrained shared tree link stress comparison.

[75] describes a solution to offer a multicast service over MPLS/BGP VPNs, using PIM within the VPN and customer routers at different sites. This solution differs from ours by the fact (1) it is aimed to be used in the service-provider backbones specified in [15], while our approach is based on edge technology, and (2) it does not address the problem of using PIM along with IPsec.

As mentioned before, a PPVPN can easily and efficiently offer a group communication service. [70] describes, at a high level, how the VPN can exploit either a multicast routing service in the provider's network, or an MPLS-enabled infrastructure.

Finally, our approach shares some similarities with the Centralized Multicast (CM) approach [51]. In CM, the data forwarding and control functions are kept separated, and the control part is centralized in distinct control elements. The control elements are arranged in a two-level hierarchy within autonomous systems and are used to set up multicast trees. In our approach too, the control part is centralized in the VNOC. The major difference yet it that CM does not address security.

In this chapter we show how to build a fully secure and efficient group communication service between several sites. It details both the underlying motivations and the architecture proposed. It is a follow-up of work we performed on offering a group communication service in an IPsec IPVPN environment [5] and on the HBM application-level multicast protocol [73]. We show that these proposals, that both follow a centralized approach, naturally fit with one-another and lead to the concept of Virtual Private Routed Network, of VPRN. This concept enables us to largely improve distribution efficiency, in particular by reducing the stress laid on the physical infrastructure, over the multi-unicast approach used so far to distribute packets between the sites. It is worth noting that centralized overlay multicast approaches, often criticized, find a perfect field of application in VPN environments, and the security benefits brought by the whole architecture largely compensate any possible scalability limitation.

Finally IVGMP and HBM have both been implemented and simulations carried out to quantify the gains made possible by the VPRN approach.

Chapter 6

Managing and Securing Web Services with VPNs

Contents

Web Services constitute a set of technologies that many believe will change the web communication landscape within the next few years. They offer standardized and easy communications for distributed systems over the Internet. However their dynamic and distributed nature requires a well-managed system, and pending security issues prevent their widespread adoption. Meanwhile there is a big rage toward the use of VPNs to secure communications in a cost-effective environment like the Internet. In this chapter we explain how to merge these two technologies in a new powerful hybrid model that: (1) enables an easy management of web services, (2) provides web services security thanks to the use of dynamic and programmable VPNs, and (3) remains simple and fully integrated. The proposed solution provides an easy and fast way to accomplish a well-manged secure web service over the Internet rather than a complex sophisticated architecture.

This chapter is organized as follows: in section 6.1 we discuss web services and highlight the pending management and security issues; we detail our proposal in section 13.2 and discuss some design and performance aspects in section 13.3; finally we conclude.

6.1 Background

There is clearly a mutual need between the web service and the VPN technologies. The proposed model fuses several security technologies, IPsec and SSL, and the web services technology into the same melting pot.

The resulting hybrid model: (1) enables an easy management of web services, (2) provides web services security thanks to the use of dynamic and programmable VPNs, and (3) remains simple and fully integrated.

6.1.1 Requirements for Standard Technologies in our VPN Approach

The control operations between the VNOC and an ED rely on several XML control messages sent over HTTPS. Obviously, the use of a specific XML language between the VNOC and an ED creates an interoperability problem, since each ED must know this language to be able to take part in site-to-site VPNs. Therefore the use of the VPN technology for managed a web service seems to be a good candidate, since it provides a way to access VNOC services in a completely independent manner using a standard language like SOAP. Creating a VPN web service will create a way to query, invoke and communicate with the VNOC services without having to care about the VNOC specifications.

6.1.2 Web Services Evolution and Challenges

The WWW is more and more used for application-to-application communications [25]. Web services are a new breed of web applications. They are self-contained, self-describing, modular applications that can be published, located and invoked across the Web [59]. Web services perform functions which can be anything from simple requests to complex business processes. Once a web service is deployed, other applications (and other web services) can discover and invoke this service. Actually, a web service is a veneer for access services in other middleware platforms. Web services provide a standard means of interoperability between different applications, running on a variety of platforms and frameworks. One of their strength is the use of standard web technologies [64] (e.g. XML, SOAP, WSDL, UDDI), as will be explained latter on.
However other issues like the security and dynamic management of web service architectures are still under discussion. We now introduce web services and focus on the limitations of current solutions.

Web Service Architecture

A web service architecture is also referred to "Service Oriented Architecture", since the service is aimed to fulfill a specific client needs [89]. A typical web service consists of three elements:

- a Service Requester (the client), who requests the execution of a web service,
- a Service Discovery Agency, who typically operates as a repository where a service provider publishes its services, and
- a Service Provider (later referred to as a web service platform), who provides a set of services.

In a typical scenario, a service provider first describes a web service and publishes it to a service discovery agency. A service requester retrieves the web service description from service discovery agency, and invokes the web service implementation from the service provider. The role of the service discovery agency becomes more important when a service requester may ask multiple services supplied by different service providers.
The use of standard technologies was the main factor of the web service success [65]. Among them is the use of the XML language. XML provides a metalanguage in which you can write

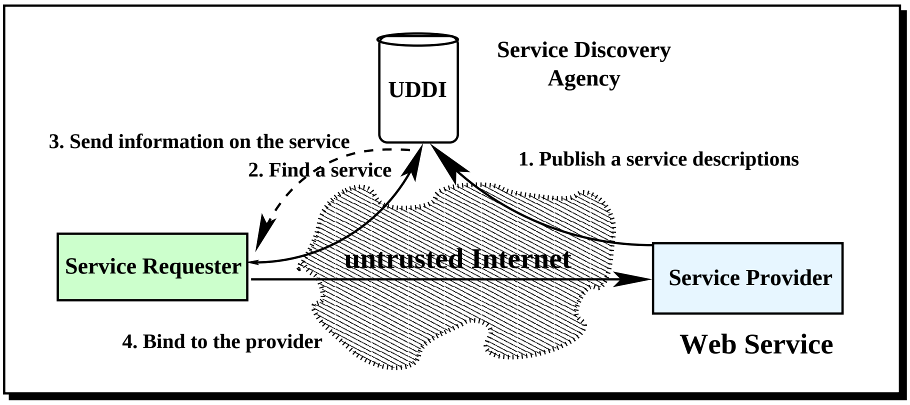


Figure 6.1. Web Service Architecture.

specialized languages to express complex interactions between clients and services or between components of a composite service. XML has many advantages over other languages used for describing data. One of them is the fact that XML is not a programming language, and therefore it can be used to exchange data between different programming languages.
The second standard technology is HTTP. HTTP is a ubiquitous protocol, running practically everywhere on the Internet. Behind the facade of a web server, the XML message are converted to a middleware request and the results are converted back to XML.
In addition to that, a fully functional web service also requires:

- SOAP: it is a specification protocol that defines a uniform way for passing an XML message [60]. It is the envelope that is used to warp up the XML message before sending it on the wire. A SOAP message contains the name and parameters of the method to call on the service provider, and returns the values to the service requester. SOAP provides a light-weight messaging format that works with any operating system, any programming language, any platform, and which is firewall-friendly.

- UDDI: it looks like a repository service [66] and it offers a mechanism to discover services published by providers. It has two kinds of clients: service providers (who want to publish a service and its usage interfaces), and service requesters (who want to obtain some services).

- WSDL: it is a definition language [88] used to describe where a web service resides and how to invoke it. The WSDL document itself is written in XML and specifies the types of operations that the web service provides, when invoked with correctly formed arguments.

The hosting web services platform is typically either J2EE or .NET. Platforms handle runtime issues on behalf of web services, and because of their complexity they need to be carefully managed.

Evolution of Web Services and Open Challenges

The evolution of web service technologies can be divided into three phases:

- The first phase concerned the basic standards: XML, SOAP, UDDI and WSDL. Thanks to the efforts of such standardization groups as W3C and WS-I (Web Service Interoperability Organization), these standards are now rather mature.
- The second phase addresses the security and reliability issues. This phase is still in an intermediary stage, despite the big efforts of several working groups such as OASIS (Organization for the Advancement of Structured Information Standards), WS-I and others. For instance WS-I is working on critical web services specifications like XML Digital Signature, XML Encryption, HTTP-R, SAML (Security Assertion Markup Language), and XACML (eXtensible Access Control Markup Language).
- The third phase addresses the provisioning, monitoring and system management. This phase is really in the very early stages of discussion, even if it is of utmost practical importance.

Since the functional requirements are essentially covered, challenges have moved to the management and security aspects.

Management Challenges:

A successful approach to web service management requires a set of flexible, inter-operable security primitives that, through policy and configuration, enable a variety of secure solutions, in addition to a set of policies for deploying, operating and monitoring web services. In this work we concentrate on the external management operations (i.e. the management operations between a web service platform and the client environments) rather than on the internal ones. External management operations include:

- client identification: clients are given a unique identity when placed under management control. Without these identities, monitoring all instances of the web service environment would be complex.
- client authentication and authorization: the goal is to authenticate clients and give them the right or not to invoke services from a web service platform.
- monitoring operations: the goal can be to determine whether a web service instance is working correctly, to get the list of clients currently hosted on the web service platform, the list of web service instances currently responding or not, or the average dispatch time for messages sent between web service instances.
- configuration deployment: configuration policies must be deployed in order to secure communications between a client and a web service platform. By having enough information about clients, generating suitable configuration policies that conform to the clients environments is much easier.

Yet the current web services standards do not handle all above issues. A significant limitation is the management of authorization and security events, as well as the explicit capabilities of monitoring, configuring and identifying.

Security Challenges:

Most of the current efforts concern security issues. Many languages are created for this purpose (but are not yet finished). One of these languages, known as Web Services Security Language (WS-Security), defines SOAP extensions that can be used to provide integrity and confidentiality [47]. Other languages use XML extensions for these purposes:

- XML Signature [39] describes how to digitally sign an XML document and provides integrity, signature assurance, and non-repudiation for web data.
- XML Encryption [40] describes how to encrypt an XML document and represents the encrypted content of XML data and the information that enables a recipient to decrypt it.
- XML Key Management Specification (XKMS) [41] describes an XML-based protocol for delegated trust and specifies protocols for distributing and registering public keys (used in conjunction with XML Signature).
- XACML [68] provides a specification for policies to access XML documents, based on objects (elements to be accessed in the XML document), subject (the user), action (read, write, create, delete)
- SAML [67] describes authentication, attributes, and authorization decision
- Service Provisioning Markup Language (SPML) [69] uses for exchanging user, resource, and service provisioning information

These standards do not address the larger problem of how to secure a web service, but how to secure XML data, no matter whether it is part of a web service or not. Moreover they do not address the issue of transport protocol security, although transport is an integral piece of end-to-end secure messaging required.

Other technologies are used for establishing secure connections over HTTP like SSL and TLS [28]. Basically transport security for web service messaging consists in using SOAP over HTTP/SSL or TLS.

Using IPsec is another way to secure web service messages. It provides peer authentication, confidentiality and integrity of IP packets. However fundamental differences exist between SSL/TLS and IPsec as explained in section 11.2.1.

No matter whether the security technology is for point-to-point or end-to-end communications, a high level of security management is required to prevent conflicts between web service instances and to deploy security policies without any interoperability problem. Additionally, these policies should be generated in a dynamic way, without any human intervention, each time a client requests a web service.

Related Works in Security and Management Services

Security remains an important barrier to customer adoption of web services, and in spite of many efforts, no single standard for security exists. Many documents and drafts have been written in this domain, yet the vast majority remains highly theoretical, as opposed to practical, real-world implementations.

The WS-Security document, which is now developed in the OASIS standards body, should be published for public comments and is expected to be completed within a few months. The WS-I (Web Services Interoperability) organization was formed recently and to promote open standards for web services interoperability across platforms, applications, and programming languages, and it will come up with security specifications, in particular WS-Security. Yet most standards require some time before they are truly useful from an interoperability point of view, and web services security standards are no exception.

As for management issues, no serious efforts have been done, and most of them are proprietary approaches from major application developers and providers. Lately a new OASIS Web Services Distributed Management (WSDM) Technical Committee was created as a way to bridge the gap between OASIS and other organizations such as the W3C Web Services Architecture Working Group and the Distributed Management Task Force (DMTF). This includes using web services architecture and technology to manage distributed resources. The first Web Services Distributed Management (WSDM) V1.0 Specification is expected in January 2004. It will include WSDL descriptions of manageable resources and the associated XML schema. This document will also define the explicit manageability of the components of the Web Services Architecture (WSA) as defined by the W3C.

In addition to these organizations, business companies such as Microsoft, Actionl, Amberpoint, already realized the urgent need for web services management solutions. For instance Microsoft will soon release its new .NET-based management solution, Microsoft Operations Manager (MOM 2004).

Following sections discuss with great details several related works concerning security aspects of web services. A complementary question is: "why should we use web services while other middleware platforms like RMI, CORBA or DCOM provide great implementation vehicles for services"? The strengths of the web as an information distributor, namely simplicity of access and ubiquity, are important in resolving the fragmented middleware world where interoperability is hard to achieve. The web complements these platforms by providing a uniform and widely accessible interface over services that are more efficiently implemented in a traditional middleware platform.

6.2 Our Approach: VPN Web Services Architecture

6.2.1 Introduction to the VPN Web Service Architecture

The proposed architecture, that merges the web service and VPN technologies, includes all the components of any web service. It is based on a *central Management Operation Point (MOP)* that handles all the management aspects of both the web service and the clients, but is not involved in the web service processing itself. The MOP includes two components:

- a UDDI register where the service descriptions are published, and
- a VNOC where the management of the web service occurs.

In this architecture, we distinguish the *management operations* of the web service platform and client environments, from the *business operations* where clients request the execution of a web service. The MOP is only concerned by management aspects. There is only one management interface in the MOP, shared by all web service platforms and clients, while there might be several business interfaces, one at each web service platform.

In order to unify all operations, a dedicated web service is created between the MOP and each web service platform or client environment for management operations. To avoid confusions, we now distinguish:

- the *management web service*, provided by the MOP to the entities he manages, and
- the *business web services* (or just web services) provided by a web service platform to its clients.

We therefore make a *recursive use of the web service concept* in our architecture. Naturally, WSDL descriptions are provided for the management and business services. SOAP is used in both cases, and the management functionalities can be discovered in the same way as web services.

The traffic within the management web service itself is already secured by SSL.

To simplify management tasks of the various business web services, and to address security requirements, the MOP creates a *dynamic VPN for each business web service.* This VPN is a star VPN that spans:

- the web service platform (center of the star).
- all clients of this business web service: they join (or leave) the VPN dynamically and invoke the web service. They are at the periphery of the star VPN.

From now on we will refer to the business web service created over a VPN by a *VPN web service.*
In this architecture a VPN web service is created by the MOP for each web service offered by a web service platform. Nevertheless a web service platform could use only one VPN web service for all his web services, but this case is not considered in this work because it may require to add AAA functionalities to the platform which contradicts our goals.

6.2.2 VPN Web Service Phases

In this section we describe more in details the various steps required to set up a VPN Web Service, as well as several types of SOAP messages used to establishing VPN web services. Two phases can be identified during the setup process:

Signaling Phase and VPN Branch Setup:

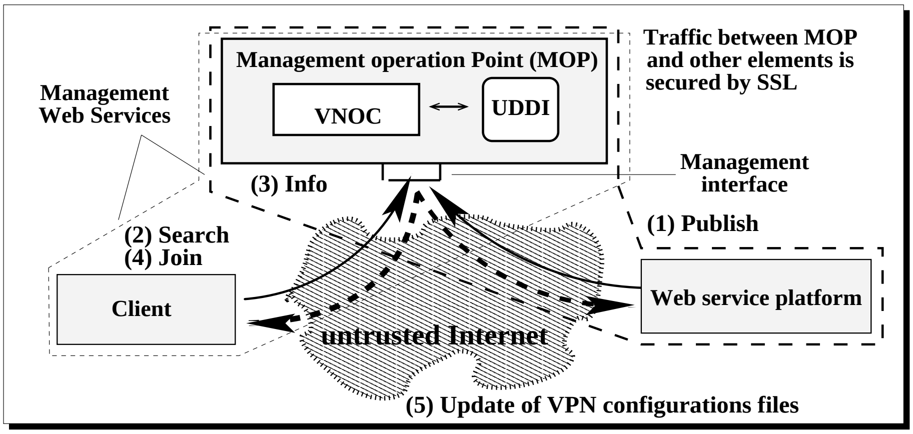


Figure 6.2. Building Web Service VPN: Signaling Phase.

This phase concerns the management web services, and messages are exchanged between the MOP and the web service platform or the clients. So we can distinguish two kinds of messages:

- The messages exchanged between the MOP and a web services platform: the web service platform first contacts the MOP by sending a *Publish* SOAP message (1), with a WSDL description file attached that contains details of the business interface and the provided services. The SOAP message is processed by the VNOC who verifies the identity and authorization of the web services platform (we assume a list of authorized platforms exists). If OK, the VNOC registers the new service description file into its UDDI register, and associates a new star VPN to the web service platform.

- The messages exchanged between the MOP and a client environment: a client initiates the signaling phase by sending a *Search* SOAP message (2) to the management interface to look for information about a business interface. Once the client is identified by the VNOC, an *Info* SOAP message (3) is returned to the client with the description file of the business interface requested, plus the identification of the associated VPN web service. The client then sends a *Join* SOAP message (4) to the MOP to join the VPN web service. Upon receiving this message, the VNOC adds the client's identification to the VPN members and distributes the new (IPsec or SSL) configuration policies (5) to the client and the web service platform. A VPN tunnel is finally established between them.

Here we defined three types of SOAP messages:

- *Search message*: it is sent by a client environment and contains the client's identification and the kind of service the client is looking for.

- *Info message*: it is the answer to the search message. It contains one or more WSDL description file(s) corresponding to the service requested.

- *Join message*: this message is sent by a client to join VPN web services whose identification is provided.

Two other types of SOAP messages exist:

- *Leave message*: it is sent by a client who wants to leave one (or more) VPN web services identified by their identifications. Upon receiving it, the MOP updates the configuration policies of the corresponding web service platform(s) to delete the VPN tunnels for this client.

- *GetVPN message*: this message enables to get the VPN web services identifications that a client has joined.

All the identifications for the VPN web services, the clients and the web services platforms are generated manually. This manual configuration is due to the need of verifying the requester identities which can be done via traditional ways, like mails or telephone calls and the need of determining the required security level.

Data Transfer Phase:

Once the client and web service platform have received VPN configuration policies, the client can then invoke service thanks to the information found in the WSDL description file obtained previously. When the client sends his first SOAP message to the business interface, it invokes the establishment of the tunnel between the client and the business interface. Once established, the tunnel will remain active till the client sends a *Leave* message to the management interface, and this later has updated the VPN configuration policies.

Depending on the security policies deployed in the previous phase, the traffic between the client and the web service platform is secured either by IPsec or by SSL, depending on the configuration policies. The choice of the security protocol(s) depends on the clients security needs when they register to the VPN service. The use of IPsec or SSL will not

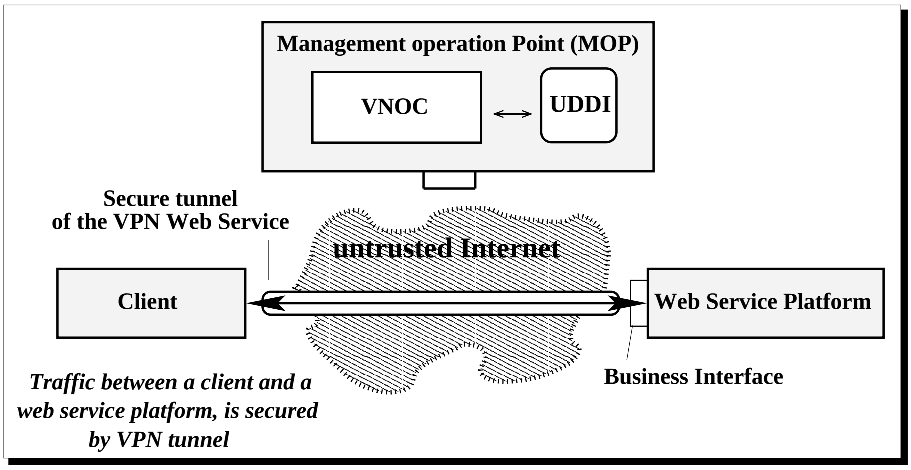


Figure 6.3. Building Web Service VPN: Data Transfer Phase.

prevent the use of other technologies such as XML signature or SAML to secure point-to-point communications, but they add another level of security.
Another benefit is that a VPN web service contains only authorized members who are interested in invoking its services. Thus a web service platform does not care about the authentication and access control of clients any more. These security aspects are performed by the VNOC on behalf of the web services platforms.

Example

Here is a small scenario of how this could function in whole:
First of all, a web service platform gets its identification from the management point as well as a VPN's identification of the star VPN that is centered around the web service platform. When a web service platform is ready to publish its services to the UDDI of the management point, it sends a *Publish* message that includes a WSDL description file and its identification. The VNOC arranges the WSDL into the UDDI register, after receiving the message via the management interface.
After publishing services description files, a client could get a file description from the management point. Again the client should get its own identification to be able to contact the management interface. Once the client gets its identification, it could send a*Search* message asking for a special service. The VNOC sends back an *Info* message including WSDL description files according to the service requested by the client and the VPN identifications of the corresponding services. The client now is freely to ask to join a VPN web service. As soon as the management point gets a *Join* message from the client, it sends back configuration policies to establish VPN tunnel between the client and the matching web service platform. Now the VPN web service contains the web service platform and that client. Platform's services could be invoked by simple Invoke messages issuing by the client. The messages between the web service platform and the client are completely exchanged via the secure tunnel. Moreover, since the client has been already authenticated by the management point during the previous phases, the web service platform does no more care of the client identification hassles. Sure this does not prevent the use of another security point-to-point level.

To summarize, the the roles of the MOP are:

- including a UDDI service where the VNOC registers the service descriptions provided by the web service platforms,
- handling SOAP messages for the different types of requests (publishing description services, finding a service, joining a VPN, querying a VPN, etc.),
- deploying VPN configurations: it includes managing the security policies for IPsec/SSL, as well as the certificates with the help of a certificate server,
- controlling the client accesses to VPN web services,
- and updating VPN configurations.

6.3 Implementation and Evaluation of a VPN Web Service Platform

In order to evaluate this architecture we implemented a great part of the system. The MOP includes the VNOC (UDDI is missing in this prototype) and uses a J2EE platform to handle runtime issues. The management interface runs in front of a Java Apache server, and the clients use a Perl program to send SOAP messages to the management interface.

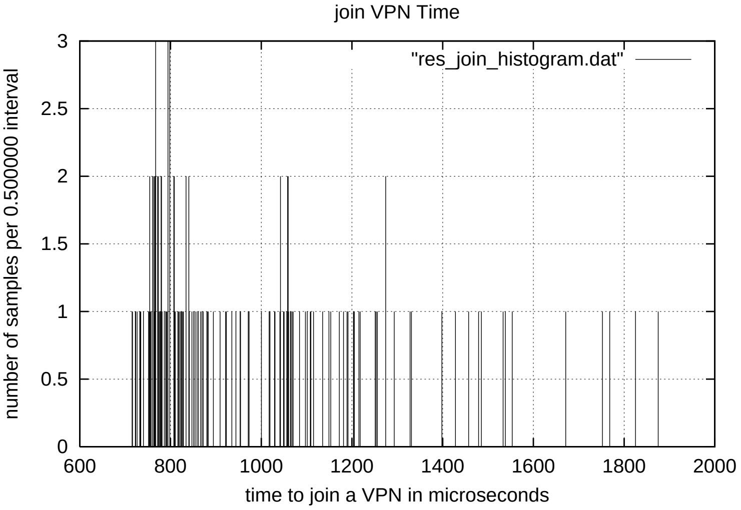


Figure 6.4. Processing time of a Join message (histogram).

We measured the time required to handle management operations on the VNOC, in a real, operational environment. We focused on the three major operations: *Join*, *Leave* and *GetVPN* services (figures 6.4, 6.5 and 6.6). Experiments have shown that processing *Join* (figure 6.4) and *Leave* messages require respectively 978 microseconds and 938 microseconds on average. This is twice as long as the processing of *GetVPN* messages that only require 492 microseconds. This is normal because *Join*/*Leave* operations imply modifications in the VPN database handled by the VNOC.

A problem that affected performance, especially when the number of clients increases, is related to the basic authentication within SOAP messages. We used the SOAP::Lite model

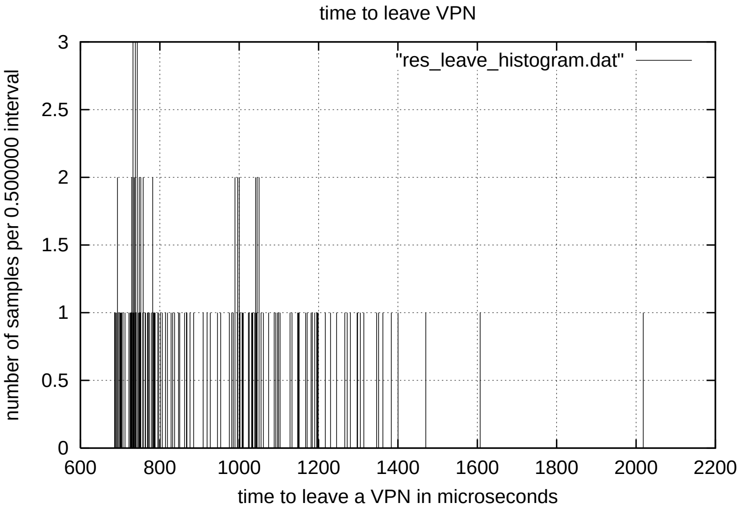


Figure 6.5. Processing time of a Leave message (histogram).

to implement the Perl client program. Since SOAP was developed to send authentication information just in case where the server asks for them, the client sends its first SOAP message without any authentication information. This causes the management interface to send back a "401: authentication error" message, and the client must reply with the missing information. We solved this problem by enforcing SOAP messages to always send authentication information in the first message sent to the management interface.

We then measured the time required to set up a VPN web service. This is the time between receiving a *Join* message from a client, and updating the VPN configuration files in the client's side. This time is in fact strongly related to some VNOC parameters that add an additional Δtime to the update process in the current VNOC implementation. In our tests (figure 13.5) the Δtime ranges between 10 to 20 seconds and we measured an average update time of 24 seconds. This is the waiting time before a client can participate to the VPN web service. It may seem long, but once the client is connected, service invocation is only determined by the web service platform, no longer by the VPN web service framework. Besides we are currently working on this aspect to reduce this initial latency.

We finally performed scalability tests. Figure 13.6 shows the time required to process a *GetVPN* SOAP message when the number of clients accessing the management interface increases. The results are good up to around 20 simultaneous clients, and then the average processing time largely increases. A first reason for this behavior is the fact that for each *GetVPN* message received, the management interface must invoke the database. Yet, since this is a shared database, the number of concurrent accesses is limited and some *GetVPN* messages cannot be handled immediately. A second reason that we suspect, is a negative impact of the Java and the J2EE platform on performances, but further investigation is needed here.

Our experiments have shown that the overall performance level is largely impacted by many different technological aspects. Yet we are confident that performance can still be largely improved, and efforts in this complex area are continuing. Therefore this section should be understood as a first evaluation of a prototype, not as a performance analysis of an optimized

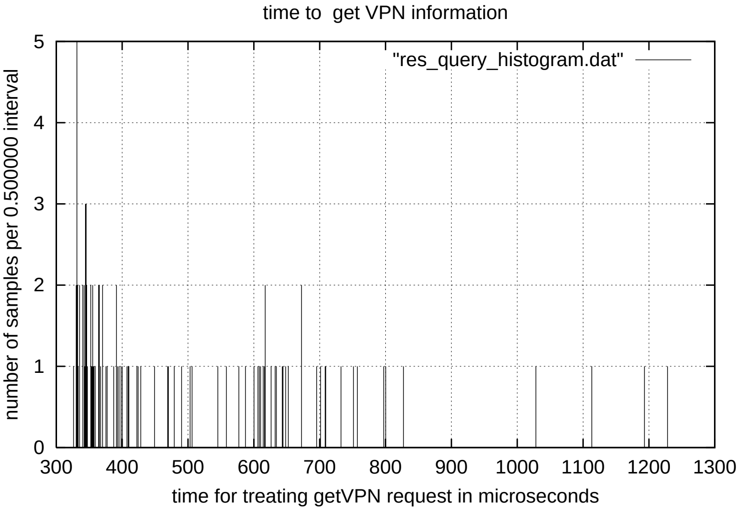


Figure 6.6. Processing time of a GetVPN message (histogram).

solution.

6.4 Partial Conclusions

Due to their flexibility and simplicity, web services gained much interest during the last few years. However the distributed and dynamic nature of web services requires advanced management capabilities. VPNs play a prominent role in the framework we propose to manage and secure these services, since VPNs provide the infrastructure over which business web services take place. The centralized MOP performs the client authentication and authorization aspects, and manages policy and security configurations at the clients. Therefore it seamlessly takes in charge lots of hassles found in traditional web services.
The implementation of VPN web service shows the feasibility of the approach. Yet achieving a good performance level requires an appropriate tuning of many building blocks, which is a complex task. If we are not satisfied by some aspects, most problems are now identified and left for future works for further investigation.

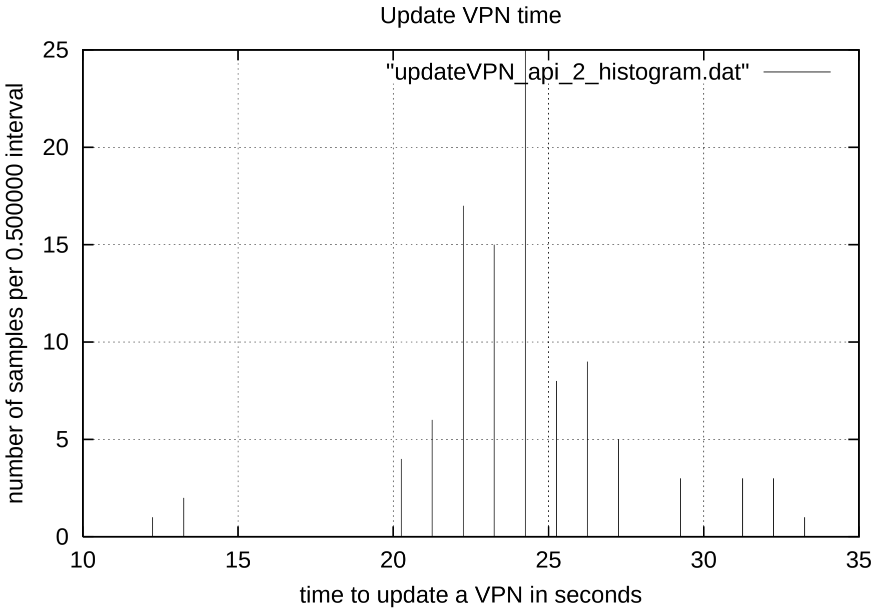


Figure 6.7. Update time of a VPN web service (histogram).

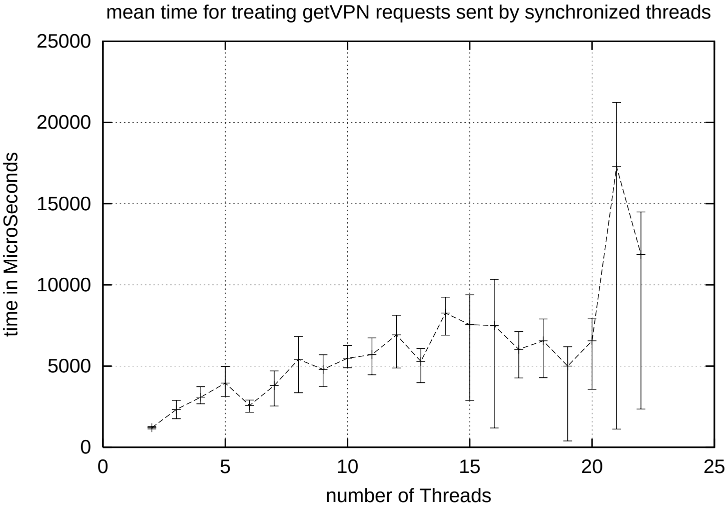


Figure 6.8. Processing time of GetVPN messages sent simultaneously by several clients.

Chapter 7

Adding Load Balancing for Higher Availability/Performance

Contents

Since our VPN approach is categorized under the CE-based VPNs, the CE devices are the only vulnerable elements in this model and requirements for configuration managements are unique for CE devices. There are two important points to take into consideration when managing such model: Firstly, it is highly desirable to detect CE device failures rapidly and to enable that the site experiencing the failure continues to operate smoothly.Secondly, when the number of communications increases for a given VPN site, it becomes a necessity to find a suitable solution to low the charge on the CE device attached to that VPN site.
In this chapter, we show two possible approaches to achieve both a high level of availability and performance, and load balancing with our CE-based model.

7.1 Background

Many mechanisms reply to the demand for high availability in VPN environments. These mechanisms are based on dynamic routing protocols to detect a case of failure among VPN routers. It is worth mentioning that the main goal behind developing such protocols was to detect VPN routers failure that might occur in the ISP core networks and react rapidly to insure the best delivery service for clients.

With a dynamic routing mechanism routers must communicate using a routing protocol. The routing daemon, on each router, adds and deletes information into the routing tables as routes change over time. It also adds routing policies choosing which routes to insert into the kernel's routing table. In case of multiple routes to the same destination, the daemon chooses which route is the best. In the other side, if a link has gone down, the daemon deletes routes connected to that link, and finds alternatives if they exist.

Dynamic routing protocols are categorized as Interior Gateway Protocols (IGPs) and Exterior Gateway Protocols (EGPs). IGPs are used within an autonomous system, in other words, within a single controlling entity, whereas EGPs are used between autonomous systems. For instance, Open Shortest Path First (OSPF) is an IGP that collects and advertises information about its neighboring routers. OSPF uses multicasting to reduce the load on systems not participating in OSPF routing. In a VPN environment, OSPF uses GRE tunnels to simulate the private lines over the Internet, and then secures the GRE tunnels using IPSec (probably transport mode). IPsec could not be used immediately to tunnel OSPF, because router interfaces at tunnel endpoints will, in general, not belong to the same subnet, so OSPF will not work. Moreover IPsec could not tunnel multicast traffic as we discussed before in section 4.

The Dynamic Multipoint VPN (DMVPN) proposal is a new IGP developed by Cisco. It allows users to better scale large and small IPSec VPNs by combining GRE tunnels, IPSec encryption, and Next Hop Resolution Protocol (NHRP).

The Border Gateway Protocol (BGP) is an EGP. BGP is commonly used within and between ISPs. It enables groups of routers to share routing information so that efficient routes can be established.

The Hot Standby Router Protocol (HSRP), is a Cisco protocol that is designed to be used over multi-access, multicast or broadcast capable LANs (e.g., Ethernet). In particular, the protocol protects against the failure of the first hop router when the source host cannot learn the IP address of the first hop router dynamically.

The first approach that we are going to discuss in section 7.3 differs radically from the mechanisms discussed above since it is based on the dynamic nature of our VPN approach. The benefit we get of our approach is that is easy to implement, uses minimal network overhead and consumes minimal memory resources compared to the other protocols mentioned above. On the opposite, the second approach is quite similar in its architecture to these mechanisms.

7.2 Definitions

Before discussing our approaches, we start by introducing the basic terms and concepts used in the proposed architectures and then provide overviews of the two architectures.

Cluster Concept

A cluster consists of one or more nodes, co-operating to provide certain services by hosting some resources on one of the cluster nodes. The most common cluster resource is an IP address, often combined with an application resource (e.g. database, web server). A user accesses a cluster service as if it was provided by one single node, and has no way of knowing which of the nodes in the cluster is currently hosting the service.

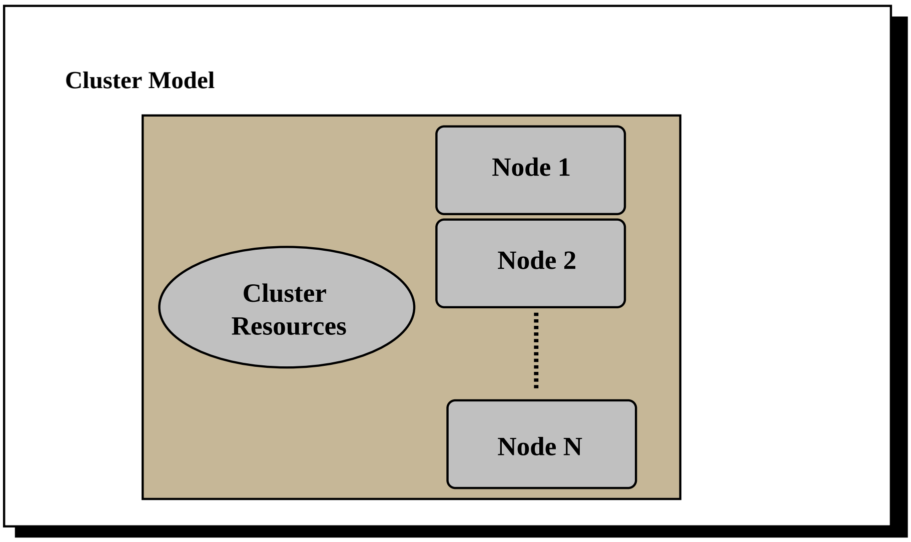


Figure 7.1. Cluster Model

High Availability or High Performance

We distinguish between two notions: high availability and high performance. We define a *high available cluster* as follows: in a high available cluster, the service is hosted by only one node. If the node hosting the service fails, its resources are taken over by another node, which can cause a service interruption for the users. The users may have to disconnect and re-connect to the service (this depends more on the user's application than on the clustering technology). In a Fault Tolerant cluster, service interruptions are minimized so as to be invisible to the end-user, but this is not always possible without modifying the client applications.
We define a *high performance cluster* when all the nodes in the cluster host the service at the same time. Adding additional nodes to the cluster improves the performance as seen by the end-user. There are several dispatching technologies that can be used to intercept client requests and redirect them to the least-loaded or next node, depending on the dispatching algorithm. There is often a need to monitor the nodes in the cluster to ensure that the user requests are dispatched to a node that is still active, and to use a highly-available dispatcher. Each node also keeps its own "private" resources, such as an IP address. Other nodes in the cluster will not attempt to host these resources if the node fails. However the client can take advantage of the presence of nodes to switch over following host (i.e. address) when the previous one fails.

7.3 Our First Approach: Load Balancing and High Availability Architecture

To insure the high availability of a VPN site within a VPN, and to reduce the load over the CE device of that site, we adopt the concept of a *high available cluster* by using several CE devices for the same VPN site, instead of one CE device. Thus if the VPN site belongs to several VPNs, each CE device in the VPN site takes into responsibility some VPN configurations. In this way, each VPN is handled only by one CE device of the VPN site. Note that all the tunnels of a given VPN are necessarily handled by the same CE device of a site. This feature is insured by the VNOC which has a global overview of all VPN sites and the VPNs to which they belong.

For each VPN site, we suppose the presence of a VPN agent located behind several CE devices in the same VPN site. This agent has several functionalities:

- It observes and monitors the CE devices of the VPN site where the agent is located.
- It takes in charge the communications with the VNOC on behalf of the CE devices.
- It sends corrective actions to the VNOC.

As soon as the agent detects that there is a CE device goes down in the local VPN site, it chooses another CE device which works good, and transfers all jobs from the inoperative CE to the second one. This procedure necessities a kind of interaction with the VNOC, in addition to new actions that we developed for this purpose with the help of the web service, here are some of them

- *DisableSite*: this action is used to take off a CE device from a set of VPNs to which it belongs.
- *EnableSite*: this action is the reverse action of the previous one. It is used to set a CE device as active.
- *JoinVPNasBackup*: the agent sends this type of request to the VNOC with at least two parameters; the inoperative CE device and the other CE device that the agent chose to replace the first one. When the VNOC gets the request, it joins the active CE device to all VPNs where the inoperative CE device belongs. But at same time, the VNOC indicates by flag that the new CE device is just joining those VPNs as a backup. In that way, once the inoperative device comes back alive, it gets back its jobs.
- *SwitchSite*: this action is used to switch the role between two CE devices. Thus VPN jobs will be transferred from one CE device to the other. In other way, the VNOC will update VPN configurations files on both devices.
- *GetSiteBackup*: it is used to get information about the backup devices for a specific CE device.

The agent uses a set of the previous actions to ask the VNOC to transfer the jobs from the inoperative CE to another CE device that works correctly. The VNOC treats the agent's requests and updates information of all VPNs to which the inoperative CE belongs. The VNOC starts by updating its own database then sending new VPN configurations to all damaged CE devices to insure perfect VPN functionalities with the VPN site where the failure occurred. Actually while transferring VPN functionalities from the inoperative CE device to the active one, the VNOC sets a flag to indicate that the active CE device is just a backup till the inoperative device gets back working again.

Once the agent detects that the inoperative device gets back working, the previous procedure is repeated. The VNOC gets back all VPN functionalities from the backup device to that device and updates VPN configurations on other VPN sites.

7.3.1 Agent Routing Tables

The agent should update its routing table to be able to forward traffic coming from the protected networks of the local site and destined to remote sites. Since the agent is behind several CE devices, a correct routing table is essential for the agent to decide which CE device to forward traffic to. So in case where there is a failure with a CE device in the local site and once the VPN functionalities are completely transferred to the active device, the routing table is no more up to date. The agent should update his routing table by its own. This operation is done by interviewing the VNOC by using two additional actions:

- *GetSiteDistantProtectedNetwork*: this message is used to get the IP addresses and masks of a remote site.
- *GetSiteInternalIPAddress*: this message is used to get the internal IP address of a remote site.

By using these actions, the agent gets the needed information to update his own routing tables. Thus traffic destined to the inoperative CE device will be forwarded to the active one, till the previous device comes back to live again.

7.3.2 Scenario

To understand the real benefit we get from our approach, here is a small scenario of how this could function in whole. Let's consider a VPN between two VPN sites. The first VPN site is a simple site with one CE device and a second VPN site includes two CE devices and one agent, as explained in the figure 7.2.

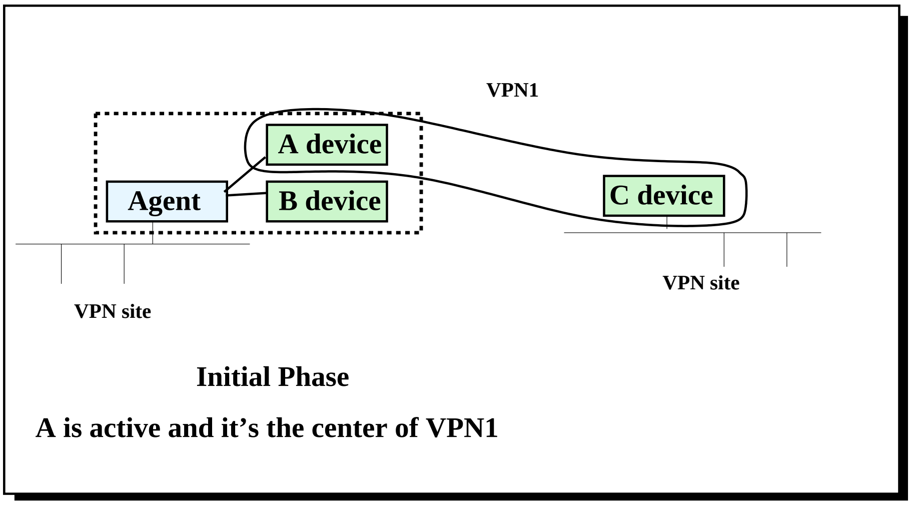


Figure 7.2. Initial Phase

We suppose having a star VPN, VPN1 between A and C where A is the center of this VPN. The initial status of VPN1 is shown in table 7.1, while the agent routing table should look like the table 7.2.

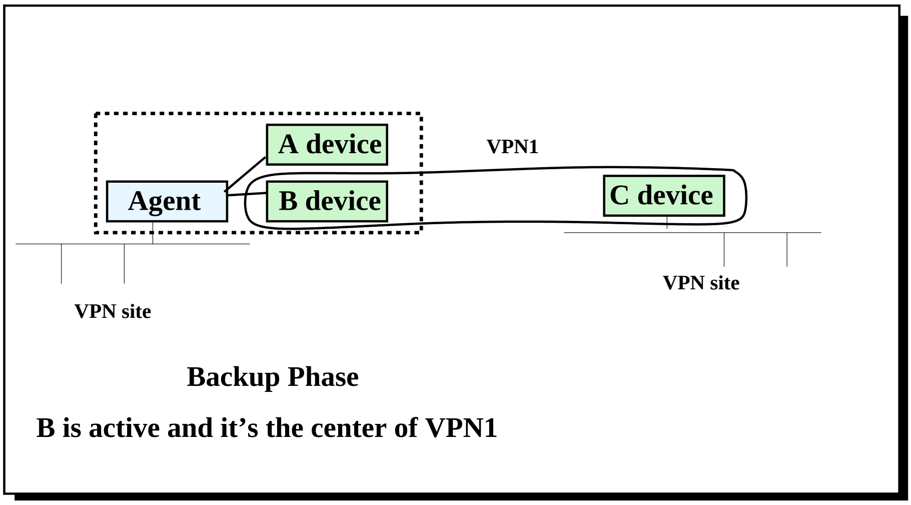


Figure 7.3. Backup Phase

VPN
Name : VPN1 Members : A C Center : A

Table 7.1. Initial Situation

As soon as the A device goes down, the agent tries to transfer VPN functionalities to an active device like B, as shown in the figure 7.3.

To do so, the agent gets the list of all VPNs that the A device belongs to, from the VNOC by using the *GetVPN* message, discussed before in a previous chapter. It sends a *DisableSite* message to the VNOC to disactivate VPN functionalities of the A device. The device B then will be added to the set of VPNs included A, as a backup with symbol (A - B), by sending *JoinVPNasBackup* to the VNOC. Then the agent sends a *SwitchSite* message to the VNOC so the B device will get in charge VPN functionalities of A. Now B becomes the new star of the VPN1 like shows in the table 7.3.

The agent then updates its routing table by sending the previous actions discussed in 7.3.1 to the VNOC, thus all traffic destined to C site will be forwarded to B device instead of A, as it shows in table 7.4.

IP address of remote protected network	Mask address of remote protected network	gateway
IP address of C site	Mask address of C site	internal IP address of A device

Table 7.2. Initial Routing Table

VPN
Name : VPN1 (A - B) Members : A (inactive) C B Center : B

Table 7.3. Backup Situation

IP address of remote protected network	Mask address of remote protected network	gateway
IP address of C site	Mask address of C site	internal IP address of B device

Table 7.4. Routing Table After Backup

When the agent detects that A device becomes active again, it sends a *GetVPN* message to get the list of VPNs where the A device should be activated. For each VPN, the agent sends a *GetSiteBackup* message to find out the backup device. In our context, the *GetVPN* message returns VPN1, and the result of the *GetSiteBackup* message for that VPN is the device B, with the help of the flag (A - B). Once the agent discovers the backup of VPN1, it switches the role between A and B by sending a *SwitchSite* message to the VNOC. Thus A becomes again the star center of VPN1. The agent provokes the VNOC to activate the A device by sending an *EnableSite* message. Now the VPN1 status returns to its initial status like it was in the table 7.1.

Once the A device is activated the VNOC update VPN configuration files on both A and B devices. Again the agent has to update its routing table to reforward traffic for site C over the A device, and to return the initial routing table 7.2.

The same procedure is applicable for N devices in the same VPN site. No matter if a backup site goes down, cause another active device could replace it, till it becomes alive again. Here is another scenario where we have two VPN sites each one with an agent and several CE devices. The figure 7.4 shows the VPN status in many positions where one of the CE devices goes down or becomes alive.

7.3.3 Implementation Aspects

Implementing this approach required the development of special API **??** to insure the switching VPN functionalities from an inoperative CE device to an active one. Again, web services are used to implement the API between the VNOC and the agents. SOAP is used over SSL to secure the exchanged traffic. While VPN agents are implemented using Perl language.

7.4 Redundancy versus Memory Overhead

In our first approach we assumed that CE devices of the same VPN site do not share the same VPN states or policies. This feature is a two-edged sword.

The advantage of this feature is that a CE device contains only VPN states of these VPNs where the CE device is active and belongs to. That means if the CE device does not belong to a VPN, there is no VPN state for this VPN on the CE device. This is rather good when

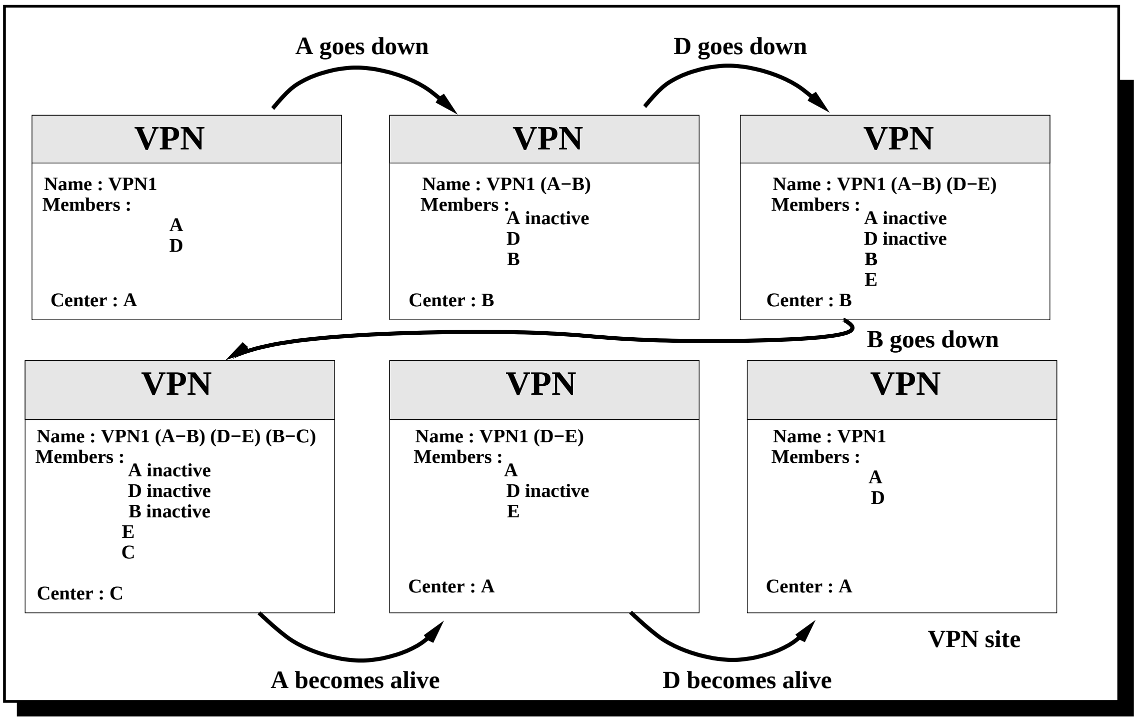


Figure 7.4. Transferring VPN functionalities in case of CE devices failure

the client VPN site belongs to a big number of VPNs. In this way, each CE device of that VPN site has VPN states that are completely different from those states on other CE devices. Once a CE device crashes down, the VPN agent contacts the VNOC to switch the VPN states to an active CE device in the same VPN site. In this case there is no redundancy of VPN states on the CE devices of the same VPN site. So there is no overhead consummation of memory when the number of VPNs becomes too big.

On the other hand, this feature has its disadvantage since it would be logical that a CE device can also keep the VPN connection going without disruption when a device failure occurred. The disruption happens since the VPN tunnels between an inoperative CE device and all distant CE devices are lost and needs to be reestablished. This operation takes around thirty seconds to a minute while the VNOC switches the suitable VPN states to an active CE device and while keys are shared to re-establish the VPN tunnels. If the CE devices had shared VPN states before the failure occurred, the time to re-establish VPN connections would be less than a second. Thus sharing VPN states between CE devices reduce the latency of a CE device failure, but at the same time it overheads CE memories when the number of VPN states becomes important. In our second approach we show how we achieve load balancing and high performance depending on another strategy.

7.5 Our Second Approach: Load Balancing and High Performance Architecture

In this approach we inherit the concept of a *high performance cluster*. One way of implementing a high performance cluster is to make each node in the cluster receive each packet, and then each node discards packets that it knows will be handled by another node in the cluster.

Since the equipment using the cluster must not require modification, each node will receive all the packets only if a broadcast or shared Ethernet-MAC address is used. Again, broadcasting all packets sent to the cluster to other hosts on the same LAN could cause interference, a shared MAC address is the most favorable solution.

Several Linux hosts can share the same Ethernet-MAC address as long as neither tries to forward packets or sent ICMP 'destination unreachable' messages. In general, if a host receives an IP packet destined for an IP address that is not local it will be forwarded or dropped. Equally the sender cannot distinguish between a node hosting or routing several IP addresses and the shared MAC-address configuration. Thus an ARP request for any IP address on either node will return the shared MAC address, and any packet sent to this MAC address will be received by all nodes but dropped on one.

Additional tools are required to implement a high performance cluster. The cluster must provide one virtual service address, which must then be shared by all nodes in the cluster. A kernel module must be created that will filter traffic so that each node gets a fair share of the traffic (and thus the load).

7.5.1 Hubs or Switches

In the basic Ethernet architecture, all packets sent on the network are seen (and filtered) by each node. This allows a network analyzer to see all the exchanges on the network. A network build using Hubs works in the same way, and a network analyzer still works as expected.

However in a network built with switches, not all packets are seen by all nodes. The switch tries to improve network performance (and security) by tracking the MAC addresses connected to each port of the switch. Thus once a switch has detected that a MAC address is connected to one port, any packets sent to that MAC address will not be forecast to other ports. This is a frequent problem with network analyzers.

When a shared MAC address is used with a switch, the switch gets confused. Very few switches expect this form of configuration. On occasion, the switch can crash, but in any case the switch will filter on the MAC address, which prevents the cluster software form working properly.

7.5.2 Router Example to Better Understand

To better understand our architecture, we prefer to start by a simple example of a router acting as a firewall. Therefore, we quickly remember the basic concepts of firewalls.

Firewalls

Traditionally there are two types of firewall, proxy-based firewall or packet-filter firewall. The packet filter firewall is the most widely used, since it does not require any changes in the network, the client applications or in the user's behavior. A proxy-based firewall does not route any packets. Instead users must connect to the proxy on the firewall, and it is then the proxy that connects to the outside world. A Packet Filter firewall behaves like a router, in that it routes packets from one interface to another, normally without modifying them. A Packet Filter also inspects the source and destination addresses of the packets and any appropriate flags, and compares these parameters against filtering rules defined by the administrator. An extension of the simple packet filter is the 'stateful' packet filter. A stateful packet filter remembers the parameters of outgoing packets and automatically (temporarily) updates the filtering rules to allow the replying packet to pass. This allows the administrator to implement very strict and precise rules controlling the packets that pass.

Now backing to our example, we show how traffic should be filtered to implement a four node high performance router cluster as shows in 7.5. Imagine the cluster is routing packets between an internal and external network, possibly filtering packets like a firewall. The cluster has two IP service addresses, one on each network. The cluster also has 2 Ethernet MAC service addresses, again one for each network.

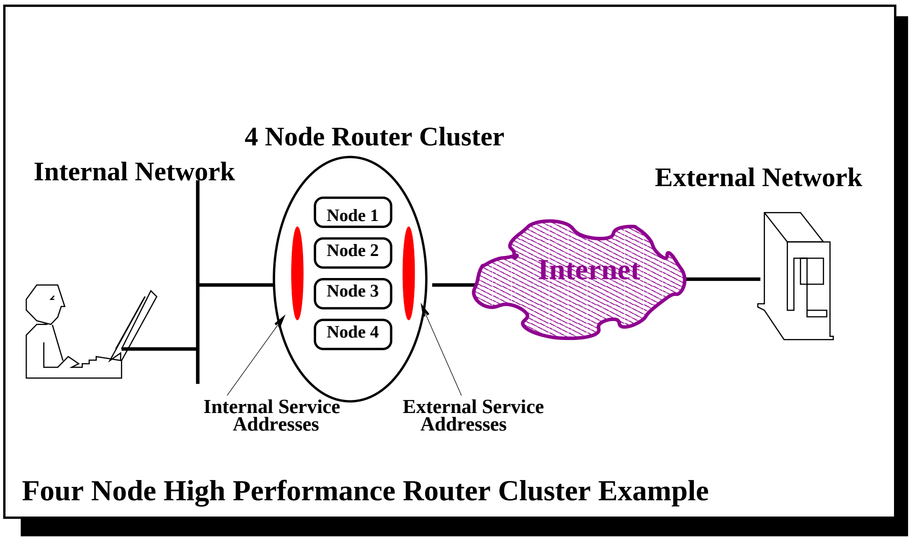

4 Node Router Cluster
Internal Network
External Network
Node 1
Node 2
Node 3
Node 4
Internet
Internal Service Addresses
External Service Addresses
Four Node High Performance Router Cluster Example

Figure 7.5. Four Node High Performance Router Cluster

When a host on the inside wants to send a packet to the outside, its routing table tells it to send the packet to the internal cluster service address. The internal host then sends an 'arp who-has' message on the internal network to determine the MAC address corresponding to the internal service address. All the nodes in the cluster reply to the arp message, indicating the same shared MAC address. The node then sends its packet to that MAC-address. However the packet sent does not contain any reference to the IP service address, it is used uniquely to determine the MAC-address.

When the cluster nodes (all) receive the packet from the internal host, it can be filtered on the address of the internal host (the source-address), the port on the internal host (source-port), the address of the external destination host (the destination-address) and the port of the correspondent on that host (e.g. 80 for HTTP, 22 for SSH). If the cluster is implementing a 'stateful' firewall, we will require the reply (if any) to pass through the same node. The reply packet will contain the same addresses and ports, however the fields will be inverted, thus the source becomes the destination.

It is obvious that we must use a filtering algorithm that includes both the source and destination address (and/or ports), and that is not dependent on the order of the information. Using both port and IP address information provides a more even distribution of the load over the cluster, however port information only exists for UDP and TCP communications. Therefore we use a hashing function that takes the source and destination address and produces a number between 1 and 4.

Each node in the cluster is configured to accept packets that produce a particular hash-value, thus one node will accept to process the packet from the internal node and passes it on to the normal packet-handling routines, where it may again be filtered to verify that it complies with any firewall configuration. The firewall software may additionally store the address information from the packet, to create a state table to help filter returning packets.

If the packet passes the firewall filtering it will be transmitted on the external interface. When a reply is generated, all the nodes in the cluster will again receive it. The same hash algorithm will be applied and the same node will accept the packet, allowing the packet to be processed by the routing and firewall software. If it again passes the firewall filters, it will be transmitted on the internal network.

If one of the nodes in the cluster fails, another node must take over its role. This means that the replacement node must accept two hash-values. If 3 nodes fail, the remaining node must accept all hash-values. Thus the hash values are the real cluster resources, and the cluster must act so that all the hash values are (almost) continuously "active".

If only 3 nodes are active in a cluster, then with a hash-value from 1 to 4, one node must take half the cluster load. However if a larger range of hash-values is used, say 0 to 31, then the load can be more evenly divided between the remaining nodes in the cluster. This can be automated by broadcasting the hash-values active on each node, and performing a takeover of a hash-value if a node detects that it is not already active.

In this scenario, no packets are ever sent or received with the IP service addresses. Each node must be accessible using alternative host-based IP addresses.

7.5.3 VPN Gateway Cluster

In this section, we show how the architecture outlined in the previous section would be sufficient to provide a high performance VPN Gateway and how does a VPN Gateway differs from a normal router.

The figure 7.6 shows a road warrior connecting to an internal mail server to download mail, while an internal client connects to the Internet.

Tunnel Initiated by the Road Warrior

The road warrior first establishes an IPSEC tunnel with the cluster, and then uses this tunnel to send packets into the internal network. The IPSEC tunnel is established between the external cluster service address and the Internet address of the PC, called the 'peer endpoint address'. The road warrior typically connects to the Internet via a dial-up Internet connection, and so the peer endpoint address is probably assigned dynamically by whatever ISP used. The peer endpoint address is always a routable public IP address, as opposed to private, non-routable address often used for a small company's internal network.

Once the tunnel is set up, the road warrior uses an internal IP address, assigned by the network administrator of the internal network. Thus when the mail server receives a packet, the source address can be verified to belong to the internal network in general, and the road-warrior user specifically.

The implementation of a shared MAC-address High Performance cluster in this scenario is complex. The tunnel endpoint is hosted by one of the cluster nodes. This node can be chosen

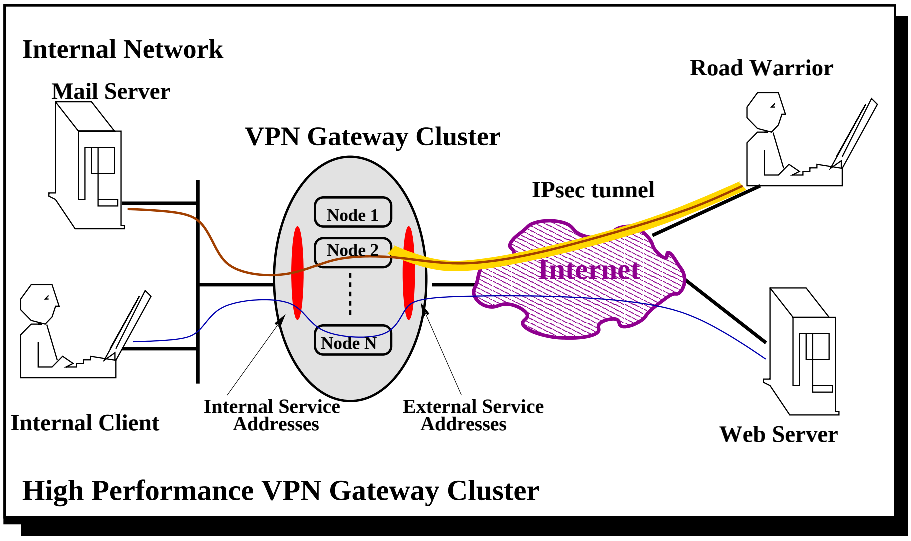


Figure 7.6. A High Performance VPN Gateway Cluster

by generating a hash-value from the source address, which is the peer endpoint address, of the packets sent from the road-warrior to the cluster service IP address. Packets recovered from the tunnel can be sent from the node hosting the endpoint without any problem.
However reply packets, which can only be handled by the tunnel hosting the endpoint, will probably not generate a hash-value corresponding to the node hosting the tunnel endpoint. To get the same hash-value, the hashing function should be applied on the destination address of the tunnel endpoint, that is usually different from the destination address of the replied packets and is not specified in the replied packets. Actually a tunnel does not necessarily correspond to one isolated IP address as in the road warrior example. In reality, a tunnel corresponds to a subnet, thus making any attempt at generating a hash-value from packets to be forwarded over a tunnel very difficult.
The destination address on these reply packets must match the subnet definition corresponding to the tunnel, and the tunnel corresponds to a peer tunnel endpoint address. Thus when the tunnel is established, an additional input filter could be activated with the subnet definition. When a packet is received by the internal network that matches that subnet, it is accepted on the node hosting the tunnel since it has the additional filter definition. However the other nodes in the cluster do not have the additional filter and apply the normal hash-value filter. One other node could also accept the packet, based on its hash-value, however additional filters could prevent these packets from being accepted on any node not hosting the tunnel endpoint.

Tunnel Initiated by the VPN Gateway Cluster

In addition the inverse scenario must be considered, where it is the VPN Gateway Cluster that initiates the tunnel. The tunnel is initiated as required to send a packet received from the internal network. However the parameters of the tunnel, which include the peer address and the subnet definition, are known when the VPN software is activated. Thus a table can be generated statically, containing the subnet definition and the hash-value that would be generated if a packet was received from that peer. Thus when a packet is received on the internal interface that corresponds to a subnet listed in this table, the filter can use the hash-value listed in the table instead of generating a hash-value from the addresses in the packet. This allows all the nodes except one to drop the packet.

For any packet received on the external interface with a destination address that would normally be routed to the internal network, the normal hash-value algorithm can be applied. Similarly any packet received on the internal interface that would be sent to the external network can also be treated in the same manner. In fact, any packet received on any non-IPSEC interface, and that would be forwarded to any non-IPSEC interface, can be treated using the basic hash-value mechanism.

7.6 Partial Conclusions

Most often dynamic routing protocols can take up to a minute to learn about a device failure. Our first approach, however, does not need to rely on dynamic routing to recover from a connection failure. It gets benefit of the dynamic feature of our VPN approach and the redundancy at the physical level by using several CE devices on the same VPN site. VPN agents in our approach constantly monitor CE devices, so a device failure is rapidly identified and fail over can be accomplished in a few seconds. However a redundancy at the logical VPN layer could be added to like in our second approach, by sharing the same VPN states among CE devices of the same VPN site to minimize that down time, although memory overhead becomes more noticeable 7.4. But the benefits we get by applying our second proposal become more important when the MOP handles lots of VPN sites, since the VPN gateway in this model handles by itself the load balancing without interfering with the MOP. However, a mix of the two strategies could come up with better solution.

Part III

Discussion, Conclusions, and Future Work

Chapter 8

Discussion, Conclusions and Future Works

Contents

We have so far introduced and analyzed our CE-based VPN proposal for offering dynamic, secure and well-managed services in an easy and transparent way for customers. We also applied and implemented our proposal in various domains. In this chapter, we conclude the dissertation by revisiting the lessons learned from this work and presenting direction for future researches in this area. We discuss several key points that were raised and classify them in decreasing order of importance from our point of view. For each of them, we identify our contributions and, when applicable, future works.

8.1 General Discussion

In this dissertation, we have presented the CE-based VPN model that shifts the VPN support from ISP core networks to customer equipments CE devices. Our proposal builds an overlay network over physical access and interconnection networks, and offers an abstraction level between the information system (e.g. customers) and physical networks (e.g. ISP networks), creating a complete independence between them. This overlay network is built over the Internet by using IPsec tunnels. To configure and maintain these tunnels in a large scale environment, a central management point called the NOS has been developed for this purpose. The NOS creates secure SSH management tunnels to VPN gateways in order to provision, configure and manage them dynamically. This dynamic feature saves the need for manual interventions, and simplifies the management of VPN topologies.

We achieved many benefits when applying our VPN proposal in two vast domains: the Group Communication Services and the Web Services. In *Group Communication Services*, we focused on many challenges that are usually neglected in the group communication services proposals like security of multicast. We showed how our approach could effectively solve these challenges and significantly reduce time consuming tasks when building group topologies.

In *Web Services*, we showed that our model is suited to web services business needs and leverages many of web services deployment limitations. A key point is that new services can be added quickly without making any change to the implementation of the existing interfaces.

We now discuss several key aspects:

8.1.1 Security

Discussion

A well-secure VPN model always requires a well-managed system. In general, security is not a single technology but an integration of several technologies combined with management policy that provides protection balanced with acceptable risks. How to control security from the management perspective, is the most important task of any VPN model. Careful network design and configuration is required. Three key points to achieve this security:

1. The security policies should be generated on behalf of customers, without their knowledge or involvement.
2. The update of security policies on VPN gateways, should also be achieved with a reasonable small delay.
3. The ability to create coherent security policies is a major step towards establishing a reliable VPN.

Contributions and Future Works

The dynamic and centralized feature of our model solves the above problems by deploying the suitable configuration files to VPN gateways and insuring the update of these files regarding the VPN topologies. Using the NOS as a central point that collects information on customers, makes security issues easier to handle than in the general case of communications where many additional hard problems must be solved. Integrating a dynamic access capability into the NOS highly facilitates security management, since the NOS is informed rapidly when there is any VPN topology changement. Thus it can update configuration files on the concerned

VPN gateway, insuring the establishment and the availability of tunnels. It is worth noting that centralized overlay approaches, often criticized, find a perfect field of application in VPN environments, and have many security benefits.

8.1.2 Scalability

Discussion

Many VPN proposals target a high scalability, which is required for instance with large peer-to-peer applications. This scalability is usually achieved with a hierarchical overlay topology. Scalability in our approach is not an explicit target in. Consequently our approach, as discussed in this thesis, is specifically designed to handle a small number of VPNs.
However, three different levels of scalability are considered in our approach:

1. Scalability in terms of the number of tunnels required by a VPN on a given CE device. The number of tunnels required for a VPN increases as the number of VPN sites increases, with the (default) meshed or star VPN topologies.

2. Scalability in terms of the number of VPNs managed by a CE device.

3. Scalability in terms of the number of VPN customers managed by the NOS. Indeed, the amount of data stored on the NOS and the processing load when a high number of clients try to access the NOS may exceed the ability of a single point to manage all the VPNs. The more services the NOS provides, the more scalability problem it has.

Contributions and Future Works

Through this thesis, we discussed the scalability of our VPN proposal, even if we do not regard it as a key requirement. We explained how the scalability can be improved by:

- Building a hierarchical VPN topology: in chapter 5, we showed how to apply this solution thanks to the concept of Virtual Private Routed Network, of VPRN. We also showed how this solution largely improved distribution efficiency with the help of an application-level multicast scheme, HBM.

- Applying the concept of load-balancing on CE devices: by using several CE devices on the same VPN site, a VPN load is shared between them. This solution is effective, since applying cryptographic security mechanisms on a CE device are memory and CPU demanding tasks.

We have addressed the two kinds of scalability in terms of VPN gateways overhead. But other aspects must be considered, and in particular the NOS load and the number of VPNs managed by the NOS. To that purpose, future works will consider the possibility of having multiple NOSs, one primary NOS and one or more secondary NOSs (not for backup) that will manage different VPNs. The primary NOS is responsible of VPN topology creation. The first contact with a VPN site is via the primary NOS, and then the primary NOS directs the new site to the 'closest' secondary NOS.

8.1.3 Robustness

Discussion

Fault-tolerance and redundancy are critical factors for providing high-availability connectivity, which must continue to operate not only during the planned network upgrades and changes, but also when connections fail unexpectedly due to component failures or attacks.

If our VPN model offers a way to alleviate management problems, it also creates other instability problems. For instance, a VPN model based on CE devices (usually PCs or workstations) is intrinsically less robust than one based on provider core networks where dedicated and well administered routers are used. There is a high risk, as the number of tunnels, handled by a VPN gateway, increases, that the topologies be partitioned after the VPN gateway failure. Of course, it depends on which distribution topology (e.g. tree or ring) is used. For instance, a VPN gateway failure is a major issue if this VPN gateway is the center of a VPN topology. Robustness requires that VPN gateway failures be rapidly discovered and the reparation be rapidly done.

Contributions and Future Works

Our CE-based model has a major advantage over provider-based models, since it does not rely on a specific connection network or a specific ISP network. For instance, in our model, the packets sent by a VPN gateway, might not follow the same path to reach the remote VPN gateway. Thus, when a failure occurs in the connection network itself, some packets may be lost, during the unstability period only, before an alternate route is discovered. This is not the case if the VPN is attached to a specific path with PPVPN model.

VPN gateways are not the only points of failure, the NOS is also vulnerable to failures. In practice, one or more secondary backup NOSs, located outside of the VPN service provider's network and periodically synchronized with the central NOS, are used for high reliability purposes.

In Chapter 7, we addressed the robustness aspect by using a fast detection and repair mechanism for VPN gateway failures. An agent located behind the CE devices of a VPN site is responsible of detecting a CE failure. Upon detecting a failure, restoration schemes that include switching to active VPN gateways, are rapidly sent by the NOS. The dynamic feature of our model plays an important role for repairing such a problem. We also distinguished between two notions: high availability and high performance, and we proposed two kinds of architectures, one for each perspective. The cluster architecture based on a high-performance notion, introduces some redundancy, which as a side effect, also improves its availability.

8.1.4 Performance

Discussion

Although creating an automatic VPN topology is more effective for creating good data delivery topologies, many aspects will affect performance in our model:

- Signaling overhead: the automatic creation and update of VPN topologies has negative effects on performance, since topologies must reflect the dynamic networking conditions and the data flow generated dynamically between the VPN gateways and the NOS add a significant overhead (section 13.3).
- Low Internet performance: Heavy-use applications may not be able to maintain adequate throughput on a VPN over the Internet. If the performance of the Internet becomes unsupportable, one possible alternative is to use a provider-based VPN that uses private backbones instead of the public Internet infrastructure.

Contributions and Future Works

Performance has not been deeply studied in this thesis. Yet we have introduced the idea of dynamic topology adaptation at the NOS, and we improved our model performance by applying the concepts of VPRN, load-balancing, high-availability and high-performance. We also measured some parameters like the processing time to join or leave a VPN and the processing time for treating a request on the NOS. Yet achieving a good performance level requires an appropriate tuning of many building blocks and further investigation works that are left for future works.

8.1.5 Relationship with Active and Programmable Networks

Many efforts have been done in several related domains like programmable networks and active networks [93] [72]. Our VPN model can be considered as an application of programmable networks, since configuration policies, like IPsec configuration files and filtering rules, are pushed and executed on VPN gateways. It is important to distinguish between programmable and active networks, although few references make this difference. An active network [19] is a network in which the nodes perform custom operations on the agents that pass through the node. These agents carry miniature programs that are executed at each node they traverse. In programmable networks, those agents carry data that are used by programs located on the nodes. In general, programmable networks are considered as a subset of active networks.

8.1.6 Other Related Works

In the last few months, two interested drafts were published: the first draft [26] proposes an architecture to provision CE devices on behalf of customers and the parameters that are needed to be provisioned, but it does not describe which protocol to use for the provisioning. It is therefore complementary to our approach.

The other draft [21] identifies many requirements for L3VPN environments, that include several topics like security, privacy, manageability, interoperability and scalability. General requirements for PPVPN were also discussed in RFC 3809 [62]. But they were not specific to any particular type of PPVPN technology, but rather apply to all PPVPN technologies.

8.2 Another field of application: NAT and its Challenges

For the rest of this chapter, we are going to discuss the applications of our approach in other domains, in particular NATs and Firewalls.

8.2.1 Discussion

Most enterprises today sit behind Firewalls and also use private IP addressing behind NATs (Network Address Translators) [83]. These NATs and Firewalls cause significant problems for many applications.

In our context, let's imagine our CE-based VPN model with two peers, one of them, Peer1, being behind a NAT router. The NOS deploys IKE configuration policies to both peers without knowing that one of them is behind a NAT. Unfortunately, when a connection is originated by Peer2 to Peer1, to negotiate IKE parameters, Peer2 will use the wrong address, trying to communicate with Peer1. In this case, there needs to be some rule that tells Peer2 what to do with the outgoing traffic to Peer1, otherwise its traffic to Peer1 will be lost and no connection will be established. If the NOS had enough information related to the NAT router, it would send the correct address and port in the IKE policies to Peer2.

A Simple Traversal of UDP through NATs (STUN) [80] was developed to help clients determine which kind of NAT they are behind. It also returns the public source IP and port number of packets it receives.

8.2.2 Contributions and Future Works

Our approach could evolve to offer peer-to-peer communications where one or both peer is behind a NAT. This can be achieved by:

- Using a *management tunnel* between the NOS and the VPN gateways: until now, we have adopted the web service technology to send secure messages between the NOS and the VPN gateways. In this case, VPN gateways were always the initiators of the communications with the NOC. By setting up a management tunnel between the NOS and a VPN gateway, the NOS can supervise, monitor, configure and manage the VPN gateway and be the initiator of the communications with the VPN gateways.

- Integrating a STUN client into the VPN gateways and using a STUN server: the STUN server could be either integrated in the NOS or an external STUN server anywhere on the Internet. A VPN gateway will send STUN requests to the STUN server to determine the NAT type.

- Sending the type of NAT that the VPN gateway is behind to the NOS: after getting enough information about the type of NAT with the help of the STUN server, the VPN gateway sends this information to the NOS. Thanks to this information, the NOS can configure other VPN gateways that wish to communicate with that VPN gateway. Currently the NOS can propagate dynamic IP addresses to peer VPN gateways. This evolution requires the NOS to also propagate port information. According to the type of NAT, the NOS determines whether or not to send IKE configuration policies to the other peer to help both peers to establish an IPsec tunnel between them.

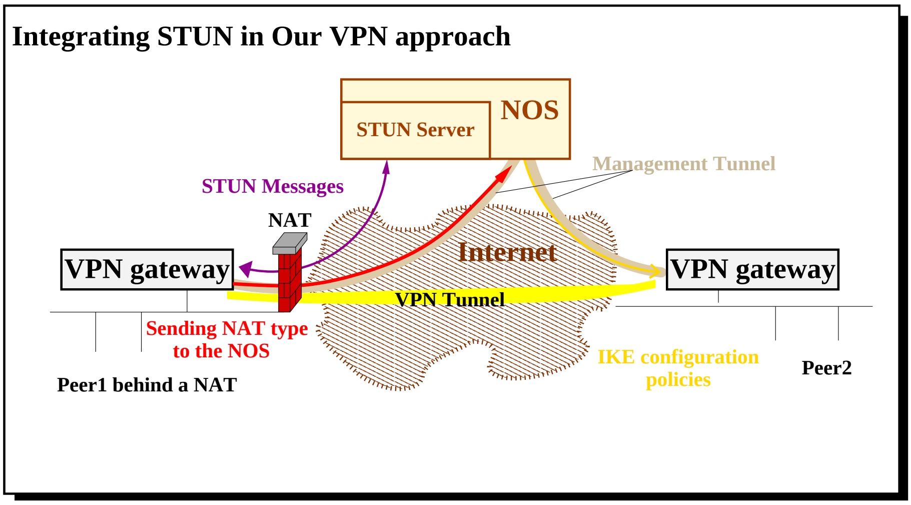


Figure 8.1. Integrating STUN in The NOS for IKE Applications

However, STUN will not work with symmetric NATs, as the mapping of external ports and addresses varies based on the destination. Thus, the external port that the STUN server reports back to the STUN client cannot be used with other hosts.

The issue of NAT Traversal is a major problem for the widespread deployment of many services like VoIP (more information in Annex 9). Yet, our approach could be a simple way to deal with this dilemma, and a more detailed investigation could prove insightful results.

8.3 Another field of application: Introducing Dynamic Firewalls Management to our CE-based VPN Model

Threats on the Internet are increasing at a staggering rate, which means that network management poses many new challenges. One popular tool for accomplishing this task is firewalls. There are two points that we would like to address here: (1) the security improvements of our VPN model and (2) the general need for a dynamic firewall management system.

8.3.1 Improving Filtering in our VPN Model

Discussion:
Our CE-based VPN model lacks to control the services accessible over the IPsec infrastructure. So far, a tunnel created dynamically by an application in our model, was also available for use by any other clients on the same network. When we established a secure tunnel between two VPN sites, we opened the possibility to all machines in the VPN sites to access any service turned over these sites. The resulting security is only suited to intranet environments, where all of the sites participating in the VPN belong to the same administrative domain and are trusted.

Contributions and Future Works:
We easily solve this problem by adding a detailed firewall configuration to the VPN gateway. This solution yields a high robustness, does not incur VPN configuration duplications, and is simple to implement. Packet filtering techniques include stateful packet filtering, which allows incoming connections to be treated differently from outgoing connections, which is not possible when configuring an IPsec tunnel. Therefore we chose to allow applications to indicate their service addresses when dynamically creating or joining a VPN. This service address is then used to generate packet filter rules that are configured on the VPN gateways, as shown in the figure 8.2.

More precisely, a VPN gateway receives requests from the VPN site clients to get the right of using a service over the VPN. Once the VPN gateway gets sufficient information, it sends a message to the NOS asking it to generate the appropriate packet filtering rule for that service or application. This message contains the following information: the type of protocol used by the service, the IP address of the client who asks for the service, the port number of the service, the identification of the VPN where the firewall rules will be applied and the service duration time.

The IP address of the service as well as the identification of the VPN are mandatory parameters in that message. As soon as the NOS receives the VPN gateway request for enabling that service, it verifies the access rights of the VPN gateway for that service and the local users who may access the services offered by this VPN. If the verification is positive, the NOS sends back packet filtering rules to enable that service for the requested VPN.

Thus if the application is trusted, it can be assumed that it will not be unduly lax in providing appropriate information for the services offered and the participation addresses. In consequence the IPsec tunnels configured between sites will not be accessible to untrusted elements on any site.

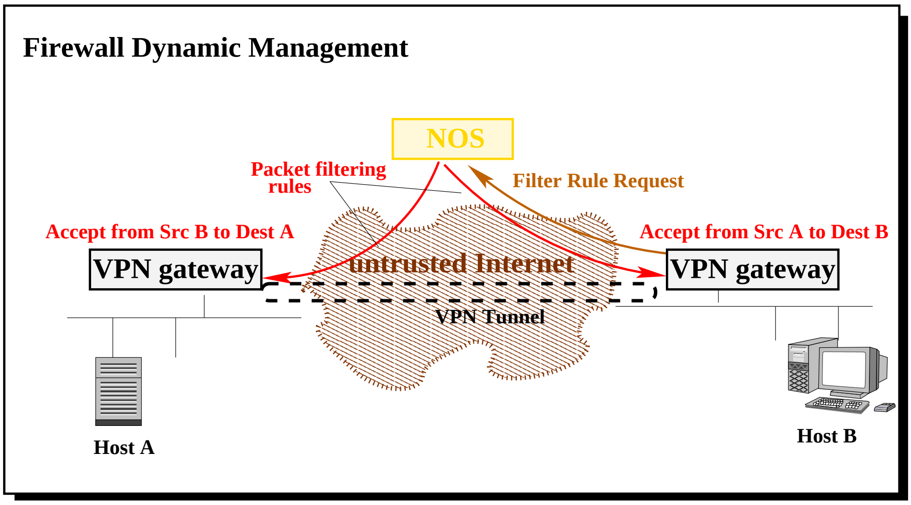


Figure 8.2. Applying Dynamic Management for Firewalls in our VPN Model

Our dynamic VPN management model can be used to develop many types of distributed applications in an extranet environment. Dynamic VPN creation, when associated with dynamic packet filtering management, allows applications behind the VPN gateway to securely communicate with partner applications on other sites, while preventing unplanned access between other applications on the same sites.

8.3.2 Needs for Dynamic Firewall Management

Discussion:
Changing firewalls rules on a big distributed system is a big chore. Because firewalls are rules-based, configuration and manageability are important features, and designing a system that scales well in a large distributed environment is still a challenge [24]. A survey of the definition and the different types of firewalls can be found in annex [22]. Here are some key points that must be considered when managing firewalls:

- Detecting misconfigurations: any firewall misconfiguration can result in the network being vulnerable to attacks. With the vast number of configuration settings, errors can easily creep into the system. Configuration errors can become even more prevalent when multiple administrators make changes to the firewall settings. A well-managed solution should compare the firewall configuration file with identified security policies to ensure that the firewall configuration is correct.

- Deploying coherent configuration rules: large enterprises will have numerous firewalls to deploy, and will likely want to implement similar policies in many locations. Consistent and coherent firewall rules will be important for insuring a good functionality of the firewalls.

- Providing monitoring: in large network environments, using multiple applications to monitor network activity can delay the problem resolution. A fast identification of firewall issues is needed.

Contributions and Future Works:
Our VPN model can address the above challenges, and a central management system seems to be the best candidate for the following reasons:

- Gathering all firewall-related events, alerts, and other activities into one central location makes monitoring tasks much easier. It avoids having the look at numerous event logs to identify when attacks are underway, or when configurations are out of date. Additionally, the integrity of this information is easier to guaranty if everything is stored in a secure central repository.

- By using a central management point for firewalls, rule deployment and threats or configuration errors can be automated. For example, if the firewall cannot start up properly due to network problems, a notification message can be sent to the central point.

- Deploying firewall rules is done by a single secure entity, which avoids rule coherency problems.

8.4 Another field of application: Security and VoIP

8.4.1 Discussion

IP networks are now used to handle an increasing number of voice calls [84]. Yet, the dark side of the convergence of voice and IP networking is that it is a good melting pot for attacks and threats, not only to customers but also to operators.
Until now, VoIP security has not been particularly addressed, but as VoIP usage becomes widespread, enterprises and home users will become subject to the same security risks that have affected data networks. This is largely due to the fact that next-generation voice networks are IP based and all IP protocols for sending voice traffic contain flaws [79].
Security is a critical feature for deploying VoIP services [61], since it requires the availability of administration and management systems to support customers distributed throughout the network, eliminate single points of failure, ensure the highest degree of service availability and deploy coherent secure policies.
Moreover, service providers must find an acceptable method for sending VoIP traffic through firewalls, which can inhibit and even block VoIP packets due to their inherent security functions. NATs also create similar problems [78]. Another significant challenge is the testing of VoIP services in a highly distributed and diverse environment.
The above challenges are configuration problems, that requires a well-managed system to overcome.

8.4.2 Contributions and Future Works

VoIP seems to be a good field of application of our VPN model. By adopting a media proxy server into the NOS, or by interacting with the proxy servers that help to route SIP requests, the NOS will be able to get information about two end-hosts that intend to communicate using VoIP. The NOS can get this information (1) by identifying end-hosts initiating SIP sessions, during the authentication phase (2) and by having information regarding the VPN that the end-hosts belong to. Using this information the NOS can react rapidly to deploy the suitable VPN configuration policies, as well as other services like generating firewall and NAT rules on behalf of the end-hosts (as discussed in sections 8.3,8.2). As soon as the policies are deployed, the end-hosts can establish a VPN tunnel between them and send voice packets over it. Briefly, we believe that the use of our VPN model cuts costs and simplifies the network management requirements for the deployment of VoIP services.

Chapter 9

Annex

Contents

9.1 Firewalls

In general, a network firewall is some dedicated hardware or software that prevents unauthorized access from the outside network to internal individual hosts (personal firewall) or groups of hosts. Some firewalls also blocks the outgoing flows from unauthorized programs in the protected domain.
Firewalls typically adopt one or more of the following methods:

- *Packet Filtering*: this type of firewalls inspects each packet going through the firewall (packet filtering is usually hardware implemented) and accepts or reject packets based on a configured set of access policy. It is done at the network layer (layer three) or the transport layer (layer four) of the OSI reference model. Packet filtering rules or filters can be configured to allow or deny traffic based on one or more of the following variables: Source or Destination IP address, Protocol type (TCP/UDP) and Source or Destination port.

  Most network routers implement some sort of packet filtering, its cheap and fast but difficult to configure for heightened levels of security. This is because packet filters look at source and destination IP addresses, ports and protocols but not at content or purpose of the packets (e.g. it does not does not know that the packet is an outgoing Outlook email). Packet filtering firewalls are prone to different kinds of attacks: the IP spoofing, ICMP tunneling and buffer overruns.

- *Stateful Packet Inspection (SPI)*: instead of inspecting each data packet independently, SPI filtering looks at certain characteristics of flow of data packets and compares against its sets of configured rules. This allows more intelligent decisions but usually requires user intervention from within the firewall software and anyway is more costly in terms of resources needed since a state is maintained for each flow. A connection table tracks individual flows, enabling policy checks that extend across series of packets. For example, TCP ACK packets not preceded by a TCP SYN packet with a correct sequence number can be blocked. The SPI firewall examines packets from the network layer to the application layer of the OSI model. SPI rules or filters can be configured to allow or deny traffic based on one or more of the following variables: Source or Destination IP address, Protocol type (TCP/UDP), Source or Destination port and Connection state.

When a packet passes through the firewall, it is examined and fed into a dynamic state table where it is stored. The packet is compared to rules and filters then it is evaluated using the state table to know if it is already part from a valid and established connection. Thus the connection state is derived from gathered information from previous packets.

- *Application Level Proxy Server*: with a pure application proxy, no traffic at all goes through the firewall. Instead, the application proxy behaves like a server to clients on the trusted network and like a client to servers outside the trusted network. Thus an application proxy is an intermediary application that secures the data traffic going in and out of a system from a particular application (e.g. MS Outlook). When an application needs to send data to the Internet, the proxy server performs the connection and pass/fails the transaction based on earlier user configuration.

  Application proxy firewalls permit as much granularity as anyone could desire. For example, lists of specific URLs can be blocked from certain subnets, or FTP clients can be restricted from PUTs, but permitted to execute GETs. Added advantages of Application-layer operation include the ability to require strong authentication before connecting and the ability to create detailed logs of security events.

  The most significant weakness of application level proxy is that either the client software is modified or the user is instructed to follow special setup procedures in order to make call to the actual server through the proxy server.

- *Transparent Proxy Server*: this solution is motivated by the desire to force clients to use the proxy, whether they want or not, or because we want them to use the proxy without their knowledge, or because we do not want to update the settings of hundreds of applications. With this kind of proxy, a client thinks that he is talking to the origin server when he sends a request to the server. Actually the client's request is intercepted by the proxy who takes the broker's role between the client and the server. With this model, the server itself knows that a proxy is involved in the middle since it receives the request with proxy IP address, not with the client IP address. The big disadvantage of this approach is to teach the transparent proxy how to redirect connections to the suitable servers.

- *Adaptive Proxies (A.K.A. dynamic proxies)*: combining the merits of both application gateways/proxies and packet filtering, adaptive proxies work by inspecting the first part of a connection at the application layer to make allow or deny decisions. Then subsequent packets are inspected at the network layer and allowed or denied at that layer. Like stateful packet inspection, adaptive proxies examine packets and then feed the information into dynamic state tables where the information is stored. The data in the table is then used to evaluate subsequent packets from the same connection. If packets are considered part of a new connection, they are passed back up to the application layer and inspected there. In this way, it is always the adaptive proxy, working at the application layer, that inspects and makes allow or deny decisions for each new connection. Only after the adaptive proxy on the application layer has approved the session, does it pass to the less secure but faster packet filter on the network layer.

- *Circuit-level Gateway*: unlike a packet filtering firewall, a circuit-level gateway does not examine individual packets. Circuit-level gateways monitor TCP or UDP sessions. Once a session has been established, it leaves the port open to allow all other packets belonging to that session to pass. The port is closed when the session is terminated. In many respects this method of packet screening resembles application gateways/proxies and adaptive proxies, but circuit-level gateways operate at the transport layer (layer 4) of the OSI model.

- *Network Address Translation (NAT)*: works from within a network router to translate the service provider's assigned IP address to multiple addresses within the internal network. This hides the IP address of each individual computer in the network from scanners software in the Internet. Describing NAT devices firewalls is misleading. NAT does shield internal addresses and their structures from outside view, but it does not provide control over outbound traffic, perform authentication, or prevent spoofed inbound packets. Thus NAT technology could not be considered as one of the firewall methods mentioned above, unless it was implemented with a firewall method. Application proxy firewalls and circuit-level gateways transparently perform NAT. The network interface that connects to the untrusted network is all that the untrusted network sees, so the addresses and the structure of the trusted network are shielded from view. In themselves, packet filters and stateful inspection firewalls do not translate IP addresses and port numbers, though there is no reason that NAT can not be implemented separately.

9.2 NATs

There are four types of NATs, as defined in [46]:

1. Full Cone NATs: they map active sessions on an internal IP address and port to a specific external IP address and port. In another word, all requests from the same internal IP address and port are mapped to the same external IP address and port. Furthermore, sessions in this type can be initiated by internal and external hosts. Thus any external host can send a packet to the internal host, by sending a packet to the mapped external address.

2. Restricted Cone NATs: in the case of a restricted cone NAT, the external IP address and port pair is only opened up once the internal host sends out data to a specific destination IP. So all requests from the same internal IP address and port are mapped to the same external IP address and port. Unlike a full cone NAT, only internal hosts can initiate a session through restricted cone NATs. For instance, an external host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.

3. Port Restricted Cone NATs: this type of NAT goes much further than the previous type and extends this limitation to the port number. In this setup, packets from external hosts must contain the source IP address and a port number previously used by an internal host. Specifically, an external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.

4. Symmetric NATs: also called stateful inspection, it goes even further and assigns a unique external port to every session that an internal host initiates with an external

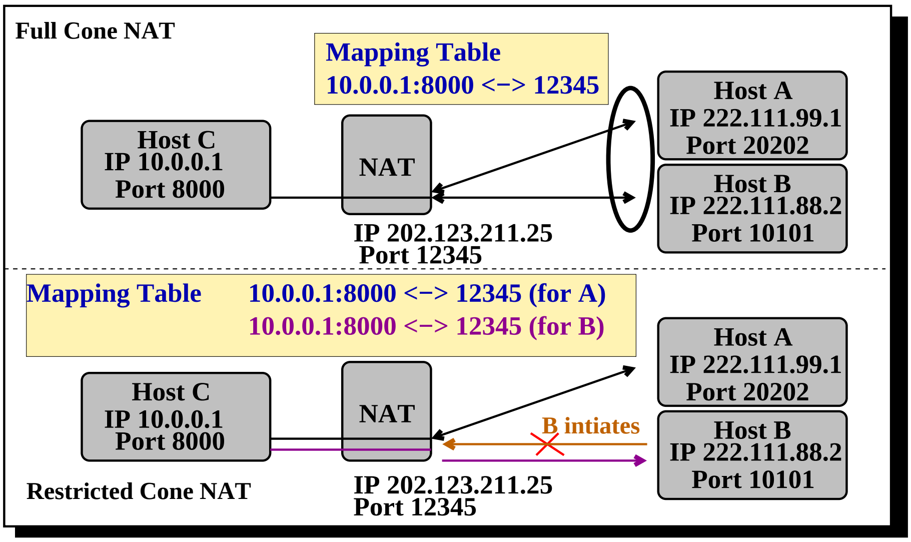


Figure 9.1. The Full Cone NAT and The Restricted Cone NAT

host. So all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port. If the same host sends a packet with the same source address and port, but to a different destination, a different mapping is used. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.

9.3 NATs and VoIP

Conventional VoIP protocols only deal with the signaling of a telephone connection. The audio traffic is handled by another protocol and to make matters worse, the port on which the audio traffic is sent is random. The NAT router may be able to handle the signaling traffic, but it cannot know that the audio traffic is related to the signaling and should hence be passed to the same device the signaling traffic is passed to. As a result, the audio traffic is simply discarded.

At first, for both the calling and the called party, everything will appear just fine. The called party will see the calling party's Caller ID and the telephone will ring while the calling party will hear a ringing feedback tone. When the called party picks up the telephone, both the ringing and the associated ringing feedback tone at the other end will stop as one would expect. However, none of the party will hear the other one.

Part of the problem with NAT and VoIP starts by the presence of many different kinds of NAT architectures as discussed above. Each of these NAT architectures requires specific techniques and solutions for VoIP applications. In addition, a multimedia session signaling protocol like SIP, by its design, is difficult to operate through NAT. Because its purpose is to establish a flow of packets, it carries the end-hosts IP addresses within the signaling messages, which is known to be problematic with NAT.

There are two parts to a SIP based phone call [85]. The first is the signaling that is the protocol messages that set up the phone call, and the second is the actual media stream,

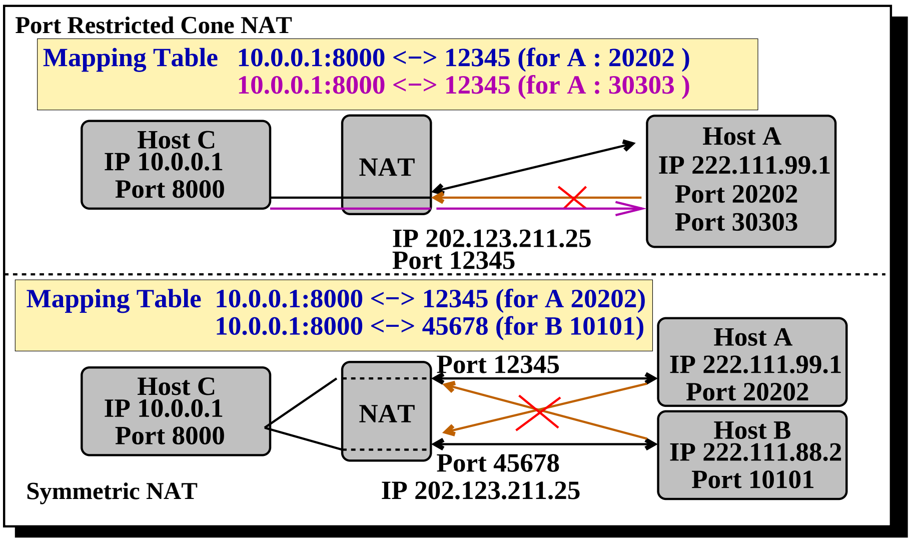


Figure 9.2. The Port Restricted Cone NAT and The Symmetric NAT

Real-Time Transport Protocol (RTP) packets that travel directly between the end devices. The real problem is in the second part. In the case of RTP, the SIP message body contains the information that the endpoints need in order to communicate directly with each other. The end point clients fill in that information according to what they know about themselves. A client sitting behind a NAT knows only its internal IP address and port number, and that is what it includes in the Session Description Protocol (SDP) [43] body of the outgoing SIP message. When the destination endpoint wants to start sending packets to the originating endpoint, it will use the received SDP information containing the internal IP address and port number of the originating endpoint, and since this internal IP address is non routable, packets cannot reach the host.

To solve this problem the client must find the NAT mapped public IP address and port number and include this information into the SDP message instead of its internal information.

To determine its external IP:port, the client asks a server, usually called NAT Probe, sitting outside the NAT on the public Internet, how it sees the source of a packet coming from him. STUN protocol was develop for setting up such kind of server. With the help of this protocol, a STUN client sends test messages to a STUN server on the Internet. From the STUN server's replies, the client learns the type of NAT used, as well as the outbound address and port associated with the client.

Part IV

Résumé en français

Résumé étendu français

Avertissement : ce résumé vise à permettre à un lecteur de comprendre l'essentiel du travail effectué dans cette thèse. En revanche il ne couvre pas la totalité du travail. Ainsi il ne comporte qu'une introduction au domaine, la présentation de notre proposition, l'étude de deux aspects qui nous ont semblé important, et une discussion synthétique finale.
Pour tous les autres aspects, à savoir l'état de l'art du domaine et les autres contributions de notre approche, le lecteur est invité à se référer à la version anglaise de ce document.

Chapter 10

Introduction

Contents

10.1 Contexte du travail

Le contexte des études est celui de l'évolution vers des réseaux multi-services et en particulier celui de l'évolution de l'Internet. Nous pouvons dire que l'Internet est rentré dans sa deuxiéme génération avec l'introduction du Web et l'ouverture commerciale du réseau. Cela a engendré une explosion du trafic et du nombre d'utilisateurs et de réseaux, ce qui a engendré des évolutions technologiques pour le rendre scalable. Cette deuxiéme génération a également vu naître l'usage de l'Internet en temps de que réseau d'infrastructure pour l'interconnexion de sites distants d'entreprise. L'Internet offre une infrastructure suffisamment performante et étendue pour que les entreprises saisissent cette opportunité. Les réseaux privés virtuels IP, connus sous le nom (IP Virtual Private Networks) IPVPNs se sont fortement développés. Les VPNs sont sources d'économies car ils remplacent des lignes louées et des appels longue distance. Ils présentent d'autres avantages comme la possibilité d'intégrer au réseau privé des sites qui ne pouvaient l'être jusqu'alors et ils permettent d'étendre les applications de l'intranet en utilisant l'infrastructure Internet à moindre coût.

10.1.1 Les Points Clés d'un Système VPN Bien Administré

Trois points de clés pour réussir l'administration d'un VPN, sont: la sécurité, l'administration et le dynamisme. Le déploiement de réseaux privés sur des infrastructures partagées voire publiques comme Internet nécessite des précautions impératives pour protéger les systémes connectés à l'infrastructure partagées et les échanges qui vont prendre place sur cette infrastructure. Les solutions retenues s'appuient sur plusieurs protocoles de sécurité, tel que L2TP, PPTP ou encore IPsec. IPsec, une suite de protocoles de sécurité définie par l'IETF (RFC2401-RFC2412), est le protocole utilisé dans notre approche grâce á ces avantages sur les autres protocoles.

D'autre part, l'étude d'une solution VPN nécessite une bonne maîtrise de l'architecture du réseau à modifier ou à construire. L'administration est un élément clé au déploiment des VPNs et tout particulièrement tout ce qu'il concernce les services de groupes de communications. La mise en place d'une solution bien administrée est reliée avec les points suivants:

- Politique: le système d'administration doit avoir le pouvoir de convertir les politiques de configuration á des règles appliquées sur les passerelles VPNs. La sécurité et l'atténuation des attaques sont aussi basées sur les politiques.

- Configuration: l'approvisionnement et la configuration des équipements VPN doivent être faits dans une maniére automate et appliqués sur les passerelles VPNs.

- Déploiement rentable: dans un système administré, il faut avoir un méchanism pour envoyer, deployer et mettre-à-jour les fichiers de configuration.

- Connaisance des topologies VPNs: plusieurs types de topologies doivent être supportés. On distingue les VPN maillés et les VPN en étoile qui correspondent respectivement à des topologies de liaisons virtuelles entre sites soit de type n vers m (réseau maillé) soit de type n vers 1 (réseau en étoile).

- Réajustement rapide: agilité et une forte réactivité aux changements dans les topologies VPNs doivent déclencher le mis-à-jour des fichiers de configuration sur certaines passerelles VPNs.

- Surveillance: l'environment sécurisé doit être mise sous contrôle et sous surveillance. Une automatisation de la supervision et de l'administration des équipements VPN par des actions à distance sur les équipements VPN installés sur les sites des utilisateurs finals.

En générale, l'administration est avant tout une problématique de connectivité réseau, et d'infrastructure IP. Certains problémes spécifiques liés à la technologie utilisée compliquent la mise en oeuvre, par example, la diffusion des fichiers de configuration IPsec bien conhérents entre les différents équipements VPN distribués sur l'Internet.

En même temps, l'Internet entre aujourdhui dans sa troisiéme génération, celle qui verra la mise en oeuvre effective de la convergence avec les Telecom (fixes et mobiles) et l'audiovisuel, la généralisation de la mobilité et l'offre de modéles de service réseau évolués. Ces évolutions qui ont provoqué la naîssance de nouvelles applications, citons notamment la généralisation du 'peer-to-peer', les jeux distribués, la distribution de contenu multimédia et en particulier la vidéo à la demande. Le déploiment de ces applications, où le trafic quelles génèrent devient majoritaire, requiert en particulier une gestion du trafic plus dynamique se basant sur une mesure permanente de l'évolution de l'état du réseau.

De manière générale, l'évolution vers l'Internet multi-services et les VPNs requierent la conception d'une nouvelle architecture de réseaux qui prend en charge l'administration dynamique et la sécurité du trafic circulés sur l'Internet. Cela a constitué les principales motivations de nos travaux de recherche.

On s'est intéressé en particulier à divers domaines couvrant les services de groupes de communication et les services Web.

Dans le domaine d'une groupe de communication, on s'est intéressé à la distribution du trafic multicast vers les différents clients distribués sur l'Internet. Le multicast IP a été un sujet chaud de recherche et développement pendant de nombreuses années. Malgré cela il reste nombre de points ouverts qui rendent son déploiement difficile tel que la sécurité. D'autre part, une groupe de communicattion qui se compose de différents membres agissant

à différentes échelles de temps, a besoin d'une conception optimale pour traiter ces différents membres et leurs interactions de maniére dynamique et unifiée. La conception de notre approache pour les groupes de communication, a été notre premier domaine de travail.
Deuxiéme domaine abordé était dans le domaine des services web. Les services web prennent leurs origines dans l'informatique distribuée et dans l'avénement du Web (en particulier le commerce électronique sur l'Internet). Ils offrent des services applicatifs publiés, localisés et exécutés sur l'Internet, et accessibles via des protocoles standardisés du web par des entités distantes (applications ou utilisateurs finaux). Néanmoins des obstacles majeurs à l'adoption des services web restent, à savoir la sécurité et surtout l'administration dynamique des services web.

10.1.2 L'Architecture Conceptuelle

Le modèle conceptuel d'architecture de notre approche est un modèle services multi niveaux basé sur plusieurs concepts et modèles:

Les Réseaux Intelligents Distribués (Distributed Intelligent Network), DIN:
Le modèle DIN fournit un environnement flexible et facile à faire évoluer ([57] , [58]). Ce modèle est basée sur la fourniture d'un réseau logique unique qui fournit aux équipements télé-administrés installés sur les sites en réseau ou personnels des clients, un ensemble de services administrés.
Ce modèle inclut trois composants:

- Réseau Logique: pour les communications entre les différents entités. Ce réseau est un réseau de recouvrement qui se compose sur plusierus réseaux d'interconnection et qui fournit un ensemble de services d'administration aux clients, comme la configuration, la surveillance, la qualité de service et le téléphone sur IP.

- Réseaux Physiques: ces réseaux sont l'ensemble des réseaux d'interconnection et télécommunications qui sont utilisés par le réseau logique pour envoyer et recevoir des informations.

- Système d'Information: inclus les portables des sites personnels et les passerelles des sites réseaux. Ces équipements sont utilisés pour approcher les services d'information aux clients.

Dans notre cas du service VPN administré un modèle DIN est utilisé sur l'Internet comme des réseaux physiques, et qui est adapté aux sites VPN et aux élément du réseau qui sont des équipements VPN et qui correspondent au système d'information. Le réseau logique dans ce cas, prend en charge les relations avec les clients et gére leurs services dans une mainére indépendante des réseaux physiques utilisés.

Tunnel Broker, TB: Un Tunnel Broker [30] est l'équivalent d'un FAI virtuel, qui fournit un service de connexions IP à des utilisateurs. En appliquant le concept du 'Tunnel Broker' dans notre approche, les composants du modèle sont:

- Le 'Tunnel Broker' qui administre les équipements VPN par envoi d'informations de configuration.

- Le 'Tunnel Client' qui est un équipement VPN de site utilisateur qui initialise une demande de tunnel.

- Le 'Tunnel Server' qui est un équipement VPN de site utilisateur qui reçoit des demandes d'établissement de tunnel émises par des 'Tunnel Client'.

Quand un utilisateur du site souhaite se connecter á un utilisateur d'un site distant, le 'Tunnel Client' si il n'a pas les informations nécessaires, va se connecter via Internet au 'Tunnel Broker' pour lui demander des informations pour établir le tunnel VPN. Sur réception d'informations du 'Tunnel Broker', le 'Tunnel Client' va pouvoir établir le tunnel vers un 'Tunnel Server'. Le 'Tunnel Server' reçoit du 'Tunnel Broker' des informations de configuration qui vont lui permettre d'accepter des mandes d'établissement de tunnels émises par des 'Tunnel Client'.

Policy Based Networking, PNB: Le PNB est l'administration d'un réseau basé sur des politiques qui définissent des services pour le trafic dans le réseau, comme la sécurité et la qualité de service. Le but est de réduire la complexité de la configuration d'un réseau et de ses noeuds. Les politiques sont dérivées de spécifications de niveau de service (Service Level Specifications), SLS. Une SLS est une spécification qui concerne le comportement d'un réseau et qui est dérivée d'un agrément de niveau de service SLA, qui est un contrat entre un opérateur et un utilisateur ou entre deux opérateurs. Notre approache applique un modèle PNB basé sur un serveur de politique qui saisit les politiques et qui les transmet á un serveur qui traite les politiques appelé PDP (Policy Decision Point). Le PDP transmet les informations de configuration des politiques á des Policy-Enforcement-Point (PEP) qui sont des équipements de réseau.

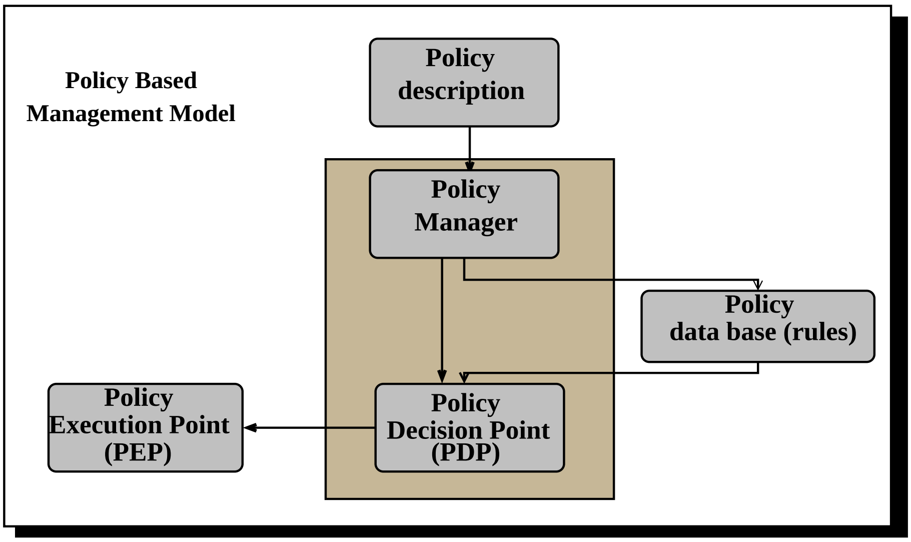


Figure 10.1. Le Modèle PBN

10.2 Buts de ce travail et organisation du document

Dans ce mémoire on a propsé une solution VPN proposant l'accés au VPN sous forme de service. Cette solution est centralisée où tout, y compris la gestion d'adhésion des clients et la création de la topologie VPN, est de la responsabilité d'un unique point d'administration.

Cette solution explique comment construire un réseau VPN en intégrant les sites, les postes isolés et les nomades, à l'aide du fonctionnement dynamique de cette approche.
Dans cette thèse, nous présentons une proposition pour établir un service alternatif de communication de groupe qui déplace le support de multipoint depuis les routeurs vers les extrémités. Nous proposons un protocole, appelé IVGMP (Internet Virtual Group Management Protocol) fonctionne au niveau applicatif et fournit un service efficace de distribution multipoint de données sur les VPNs. Avec cette approche, on offre un service de communication de groupe à tous les hôtes, même ceux situés dans un site qui n'a pas accès, pour une quelconque raison, au routage multicast natif. En plus, le trafic du groupe sera diffusé en toute sécurité à travers les tunnels VPNs.
Nous présentons aussi la contribution de notre approche dans le domaine des services web qui permet de déployer et surveiller les services web dans une maniére flexible et inter-opérable. Nous montrons comment notre approche offre à notre avis suffisamment de souplesse pour permettre d'administrer les politiques de sécurité entre les différents éléments du service web, tout en évitant les conflits d'interopérabilité. De plus la solution est dynamique et permet de générer les politiques de configurations sans intervention humaine.
Les travaux présentés ont donné lieu aux articles ([5], [3], [7], [4], [6]).

Chapter 11

Notre Approche: CE-based VPN Modèle

Contents

Les réseaux VPN Internet sont utilisés dans plusieurs types d'applications : Intranet étendus, Extranet, Accès distants.
L'utilisation du réseau public Internet comme réseau de transport IP sous jacent rend attractif les IPVPN par:

- des coûts de communication très bas,
- des points d'accès au réseau Internet partout dans le monde, fixes et mobiles,
- et la disponibilité d'accès à Internet à haut débit (ADSL, GPRS).

Dans la suite du document, on va utiliser le terme 'Site VPN' pour faire référencer au réseau local, une branche d'entreprise, un réseau partenaire, ou même un utilisateur (cas des travailleurs nomades ou á domicile).

11.1 Le Principe de Notre Approche

Le principe de notre approche est de gérer automatiquement la configuration d'équipements VPN qui ont la charge de gérer des tunnels VPN dans une manière dynamique et complètemente transparente pour les clients. Ces tunnels sont établis entre les équipements VPN. Ils sont utilisés pour transporter les paquets échangés entre des machines d'utilisateurs finals. Les tunnels empruntent le réseau Internet et assurent une sécurité robuste des échanges de données: authentification forte des équipements VPN source et destination, intégrité et confidentialité des données échangées.

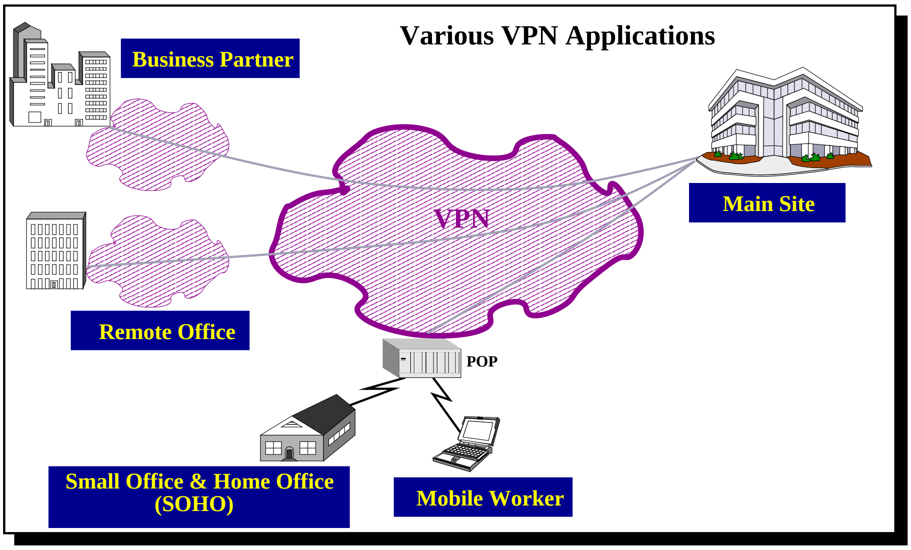


Figure 11.1. Différents Applications VPNs

11.2 Description de Notre Approche

Dans [5] nous avons défini une architecture VPN basée sur des tunnels IPsec entre des sites distribués sur l'Internet. Les VPNs sont dynamiques et leur topologie varie en fonction de l'ajout ou du retrait d'un site au VPN. *Tous les services fournis* (authentification des sites, ajout et retrait d'un site, configuration) sont réellement *dynamiques.*
Un deuxiéme aspect essentiel de cette architecture est son caractére centralisé , nécessaire afin d'appliquer les politiques á tous les sites de façon cohérente. Le centre d'administration, ou NOS (Network Operation System), prend en charge l'application des politiques VPNs sur les passerelles VPN (ou Customer Device, CE) des sites.
Un seul CE existe pour chaque site VPN et il constitue le point d'entrée et de sortie des tunnels VPNs de ce site. Les CEs jouent un rôle actif dans cette architecture: en particulier ils envoient des requêtes au NOS pour participer ou quitter un VPN. Ainsi, si un CE demande á participer á un VPN, et si le site de cet CE y est autorisé, le NOS lui envoie en retour les fichiers de configurations nécessaires. Le NOS informe également les autres CEs concernés par ce VPN afin pour établir les tunnels, et toutes les communications site-á-site sont alors sécurisées par IPsec.
Pour résumer les caractéristiques de cette approche :

- *Indépendance vis-á-vis des opérateurs FAIs ou ISPs:* ce service peut être appliqué entre des sites liés á différents ISPs. Dans les services VPNs plus traditionnels, établir un VPN entre plusieurs sites appartiennant à diffèrents opérateurs est hautement complexe.

- *Approche centralisée:* la présence d'un centre d'administration facilite le contrôle et la configuration des VPNs, ainsi que la maîtrise des services de facturation.

- *Utilisation de briques standards:* le trafic entre les sites VPNs est protégé par le IPsec, pendant que le trafic entre un site VPN et le NOS est protégé par SSL. D'autre part, l'architecture conceptuelle de notre approche est basée sur les travaux d'organismes de

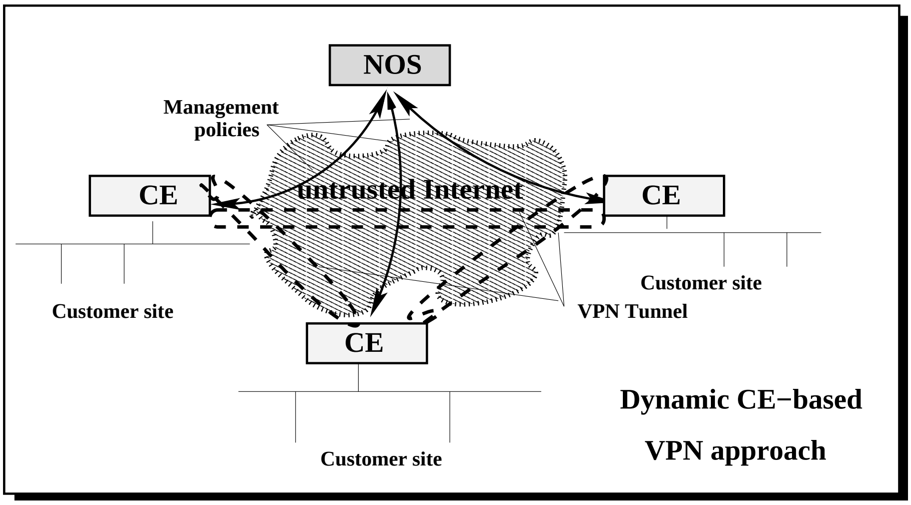


Figure 11.2. L'Architecture de Notre Approche CE-based VPN

standardisation internationaux et des modèles standards comme c'était expliqué à la section 10.1.2.

- *Approche dynamique:* chaque site envoie dynamiquement des requêtes au NOS afin de participer ou quitter un VPN, selon les besoins des utilisateurs, et le NOS met á jour immédiatement la configuration du VPN.

- *Plusieurs topologies sont possibles pour les communications site-á-site:* dans cet article seuls les VPNs en étoiles sont considérés, néanmoins mais d'autres topologies sont envisageables. Les services de groupes de communication (multicast) sont également possibles [5][3].

- *Administration:* l'authentification et le contrôle d'accés des utilisateurs sont effectués par le NOS qui offre aussi un outil de configuration et de surveillance des VPNs.

11.2.1 Utilisation de SSL et IPSec

IPsec et SSL sont les protocoles de sécurité les plus utilisés sur l'Internet [71][12]. Néanmoins ces deux protocoles présentent de grandes différences [23]:

- *niveau d'opération :* IPsec est un protocole de niveau réseau tandis que SSL est un protocole de niveau applicatif. Cette différence est le point de divergence essentiel.

- *périmètre sécurisé :* opérant à des niveaux différents, SSL offre des services de sécurité limités à TCP tandis que IPsec supporte n'importe quel trafic, TCP, UDP ou autre. De même SSL sécurise une application donnée tandis qu'IPsec sécurise plusieurs applications simultanément.

- *support d'installation :* un grand avantage de SSL est l'absence de logiciel supplémentaire côté client, un grand nombre de navigateurs supportant nativement HTTPS (HTTP sur SSL). A l'inverse IPsec requiert à ce jour le déploiement de logiciels spécifiques.

D'un point de vue sécurité, il n'y a pas une grande différence entre SSL et IPsec qui partagent les mêmes algorithmes. Etablir un VPN SSL ou un VPN IPsec dépend alors essentiellement des besoins des utilisateurs [11] et les deux types de VPNs sont utilisés dans notre approche et dans la suite de ce travail.

11.2.2 Le Réseaux de Base et la Couche Logique

Notre approche inclus deux parties (figure 11.3):

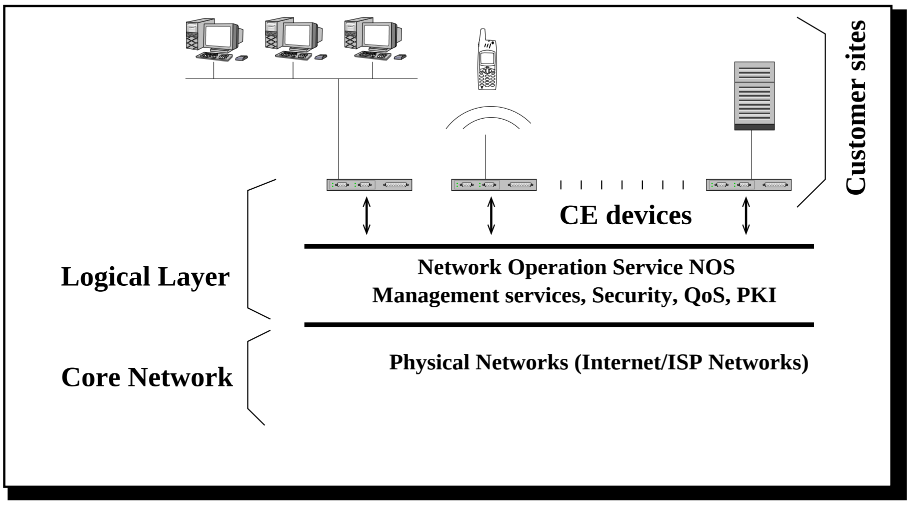

Customer sites
CE devices
Logical Layer
Network Operation Service NOS
Management services, Security, QoS, PKI
Core Network
Physical Networks (Internet/ISP Networks)

Figure 11.3. La Description de notre Approche VPN

- La base du transport des VPNs: cette partie fait la couche transport de notre approche, où le trafic VPN est envoyé. Notre approche établit des communications VPN sans problème d'interopérabilité avec les types de réseaux d'interconnection utilisés dans cette base.
- Network Operation System (NOS) et CEs: le NOS inclut tous les services d'opération. Il a une base de données sécurisée où tous les politiques de configurations VPN et les informations concernants les clients sont stockés. Ces informations sont crés et mis-à-jour par le VNOC (Virtual Network Operation Center) qui est le composant principal du NOS. Le VNOC est le point central qui prend en charge l'administration des VPNs. Il crée et déploie les politiques de configuration sur l'équipement CE du chaque site VPN. En rappelant que les équipements CE sont les seuls qui peuvent accéder au VNOC, et donc, ils sont les seuls qui sont authentifiés par le VNOC.

Dans ce cadre le VONC du NOS peut couvrir:

- la gestion de les topologies VPN: le VNOC doit avoir des informations initiales reliés à chaque équipement CE.
- l'identification des équipements VPN: chaque CE a son propos ID, fourni par le VNOC utilisé pour les communications avec le NOS.

- l'autentification et l'autorization des équipements VPN: dès qu'un CE est autentifié par le VNOC, il peut ou non avoir l'accès aux services VPN.
- la création des politiques de configuration pour chaque équipement VPN: grâce à la vision du VNOC sur les équipements VPN, le VNOC peut créer facilement les fichiers de configuration IPsec qui sont utilisés pour établir des tunnels VPN.
- la gestion du déploiment des configurations VPN sur les équipements VPN: tous les messages envoyés/reçus de VNOC sont sécurisés avec SSL.
- la modification des configurations des équipements VPN en cas d'évolution de la politique VPN.
- la sauvegarde permanente pour restauration rapide en cas d'incident sur les équipements VPN.

11.2.3 Les Messages CE/VNOC

Le dialogue CE/VNOC est basé sur des messages XML envoyés sur HTTPS. Un API spécial était défini et implementé pour traiter les requêtes entre le VNOC et un équipement VPN. Chaque CE peut envoyer des requêtes au VNOC pour participer, quitter ou avoir des information sur un VPN. Par example, si un CE envoie une 'join' requête, et si il était autorisé á participer au VPN, le VNOC envoie des fichiers de configuration VPN au CE et tous les autres CEs concernés par ce VPN et qui s'intéresse de monter des tunnels VPN avec ce CE. Au début, l'API était limité aux trois fonctions pour administrer les VPNs:

- message `JoinVPN`: un CE envoie ce type du message au VNOC quand le CE veut participer au VPN. Le CE joint l'identification du VPN dans le message.
- message `LeaveVPN`: Ce message est envoié au VNOC avec une identification du VPN, pour quitter ce VPN.
- message `QueryVPN`: Le CE envoie ce type de message au VNOC avec une identification du VPN, pour avoir des informations sur ce VPN.

11.3 Détails d'Implémentation

Les sous systèmes principaux du système VNOC sont les suivants:

- Le système d'exploitation du service (SES) permet aux administrateurs, aux clients et aux utilisateurs finaux de définir, exploiter et utiliser les ressources du service IPVPN.
- Le système de mise en oeuvre du service IPVPN administré (SMS) pilote les équipements VPN.

11.3.1 L'Architecture SMS

Le sous système de mise en oeuvre des équipements VPN a comme fonctions principales:

- l'approvisionnement des équipements VPN,
- la configuration dynamique des équipements VPN,

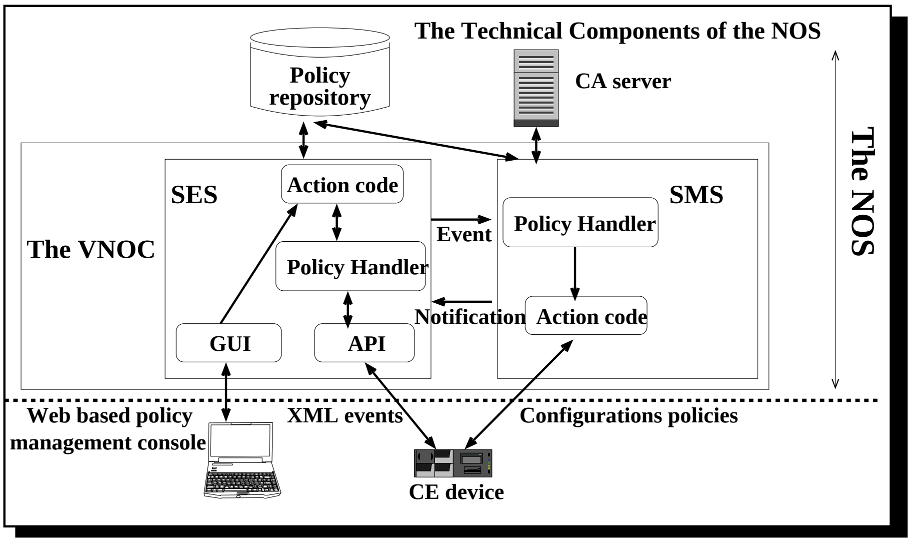


Figure 11.4. Les Composants Techniques de NOS

- et la collecte d'informations d'usage.

Le sous systéme de mise en oeuvre du service SMS est constitué de serveurs de traitement qui communiquent avec:

- le serveur de données (SGD) pour la gestion des données,
- le SES via le système de communication interne,
- le proxy IP-UP pour la gestion des équipements VPN supportant le NAT traversal,
- et les équipements VPN administrés via le système de communication externe.

11.3.2 L'Architecture SES

Le SES est composé de systèmes de traitement qui interagissent avec:

- des consoles Web sur Internet utilisées les configuration à distant des équipements VPN,
- des applications d'utilisateurs qui gèrent des VPN dynamiques,
- le SMS pour la transmission des politiques de VPN,
- le SGD qui est le système de gestion des données, pour le stockage des informations.

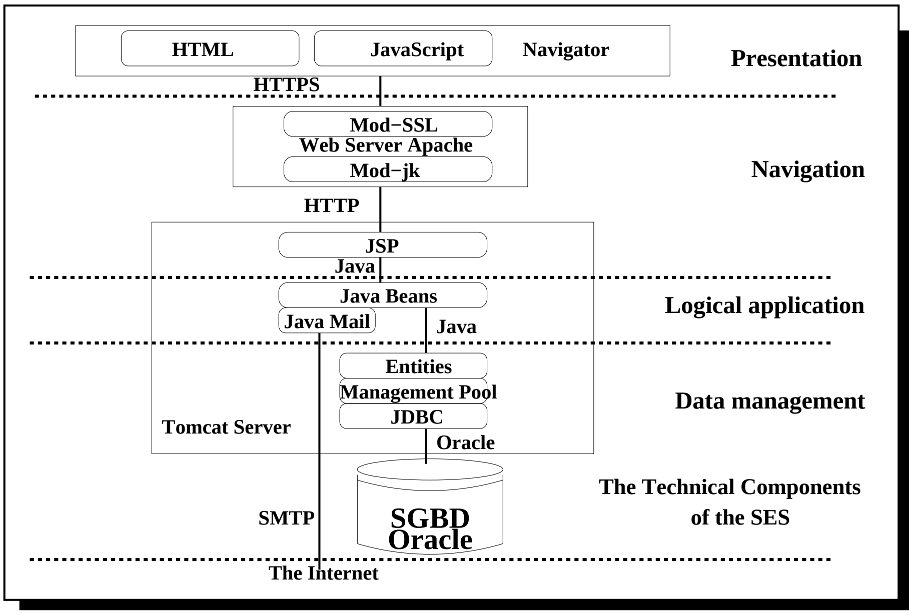


Figure 11.5. Les Composants de SES

Chapter 12

Notre Contribution: dans les Services de Groupes de Communication

Contents

Les réseaux de données se répandent avec des débits de plus en plus élevés. Ce fait a favorisé l'apparition des applications à participants multiples, comme la téléconférence ou l'enseignement à distance. Aujourd'hui le déploiement à grande échelle des services de groupes de communication subit de deux grands obstacles: l'administration et la sécurité des groupes. D'autre part, la diffusion d'informations à un groupe réalisée par 'multicast' est bloqué par les problématiques de sécurité, il faut donc sécuriser les applications de communication multipoint. Le problème principal de sécurisation des applications de groupe est le facteur d'échelle, la complexité croit en fonction du nombre des membres du groupe et la façon utilisée à l'administration des groupes. Dans la suite de ce chapître, on vise à développer une architecture globale de sécurité permettant la communication multipoint dans un environnement sécurisé basé sur notre approche VPN. Cette solution permet le maintien de la sécurité des transmissions du groupe tout en permettant aux utilisateurs de le rejoindre ou de le quitter à tout moment.

12.1 Etat de l'Art: Services de Groupes de Communication

Actuellement, il n'existe pas d'approche basée sur un modèle de sécurité prenant en compte à la fois les besoins spécifiques de sécurisation d'une application multicast, mais également la connaissance de l'environnement (le réseau, ses services, ses capacités en matière de sécurité). Le facteur d'échelle joue aussi sur la complexité et la dynamisité des solutions à apporter. De plus, la sécurisation des échanges multicast doit rester la plus transparente possible pour l'utilisateur. Et le dysfonctionnement d'un équipement d'un membre de groupe, ne doit ni entraîner l'arrêt du réseau, ni mettre en péril sa sécurité. La définition et la conception d'une architecture et des protocoles associés pour atteindre ces objectifs constituent l'objet de notre approche.

12.1.1 Limitations de Support Multicast sur les VPNs

On classifie deux domaines importants:

Multicast sur VPNs

Les solutions dévélopées par des fournisseurs PPVPN pour le déploiement du multicast, pourront bénéificier d'avoir la maîtrise des réseaux d'interconnection pour ouffrir des équipements VPN capable à gérer du trafic multicast ([70], [75]). La plupart du travail actuel dans ce domaine, est concentré à founir le service multicast pour des réseaux privés d'interconnection, administrés par des opérateurs ([8], [55]).
Par example, le draft [75], supporte le transport du trafic multicast, est basé sur les caractéristiques suivantes:

- un domaine multicast est un ensemble de tables de routage (VRFs), disponibles sur les équipements PE des opérateurs et associés aux interfaces qui supportent le trafic multicast. Un tableau VRF peut appartient à un ou plusieurs domaines multicast.

- chaque domaine multicast est associé à une nouvelle adresse multicast, appartienne à la plage d'adressage du réseau opérateur.

- le protocole PIM-SM est executé sur chaque routeur PE. Les routeurs PE participent aux différents groupes multicast selon les domaines de multicast où ils sont attachés.

Cette approche crée une arbre de diffusion multicast pour chaque domaine multicast du réseau opérateur. Le traffic multicast des clients est transmit par le réseau opérateur, encapsulé dans un paquet en utilisant une adresse destination de groupe multicast. L'encapsultaion est faite dans des tunnels MPLS ou GRE. Par contre, une instance VRF de PIM-SM est executée sur chaque routeur PE et qui communique avec un équipement CE et assure l'acheminement du traffic multicast pour un ou plusieurs domaines multicast.
Pour des raisons de scalabilité, cette approche ne prend pas en consideration si des récepteurs sont actuellement présents derrières les routeurs PE et si ils sont toujours intéressants de recevoir du traffic multicast concernant leurs domaines multicast. Dans ce cas là, chaque routeur PE attaché au domaine multicast, va recevoir du traffic multicast destiné à ce groupe, même si il y a aucun récepteur qui s'instéresse à ce traffic au moment de la transmission du traffic multicast.
Ces approches sont différentes de la notre pour deux raisons principales (1) l'effet que ces approches sont destinées à l'utilistaion dans des réseaux opérateurs [76], pendant que la notre est basée sur la technologie aux extrémités (2) ces approaches n'addressent pas le problème d'utiliser IPsec sur PIM.

Sécurité du Multicast

L'état de l'art de la sécurité multicast garantit l'accès restreint à des données diffusées, par le principe du chiffrement avec une clé secrète qui est partagée entre les membres du groupe de récepteurs autorisés. Cette approche pose un problème dans le cas de scénarios de distribution de contenus à des groupes dabonnés diffusés sur l'Internet, puisque elle nécessite un schéma de contrôle d'accès dépend de la fiabilité de chacun des membres du groupe et une méthode du chiffrement pour le trafic multicast ([54], [14]). Les réseaux privés virtuels VPNs basés sur le standard IPsec font des bons environements pour la distribution des données multicast, mais cela requiert de remplacer le protocole de gestion des clés IKE par un autre protocole qui prend en charge la diffusion d'une clé partagée pour un groupe multicast. Les activités de standardisation du groupe IETF MSEC sont en cours de développement [13]. Ces activités sont faites pour trouver un replacement du protocole IKE dans les cas des applications où on a un expéditeur pour chaque groupe multicast. Pour conclure, les priorités de notre approche sont différentes de celles des groupes MSEC et PPVPN. Dans les paragraphes suivants on décrit notre simple approche pour ouffrir un service du déploiment du trafic multicast sécurisé.

12.1.2 Notre Approche pour la Sécurité des Groupes de Communications

Notre solution est simple et indépendent de la position de chaque récepteur. Elle consisterait à utiliser notre approche basée sur les réseaux privés virtuels VPNs pour sécuriser le trafic multicast.

Notre approche concilie le besoin de sécurité et la gestion d'accès des membres du groups en impliquant les opérations de sécurité dans le VNOC entre la source et les récepteurs afin d'assurer que les données arrivent chez les abonnés en étant protégées par des clés. La source effectue un seul chiffrement pour un site VPN quel que soit le nombre de récepteurs dans le site distant. La sécurité est aussi garantie par IPsec. Ce principe est appliqué dans une architecture à trois plans: la gestion des clés, la gestion des abonnés et la distribution des données.

Le caractère dynamic et centralisé de notre approche atteint deux points principaux pour faire le but de nos besoins:

- Avec un point centralisé l'opérateur du service VPN peut facilement prendre en charge les aspects administratives pour la sécurité d'un groupe de communication. Ceux-ci incluent et n'est pas restreint aux services suivants: le service d'authentification, le contrôle d'accès, l'administration des clés de cryptages, le service du déploiement et le service de surveillance.

- Les topologies VPNs dans notre approche sont dynamiques, puisque un site VPN peut joindre ou quitter un VPN à tout moment. Ce qui convient bien à la dynamicité des groupes multicast.

Pour ses raisons précédentes, notre approche VPN semble à fournir une puisante infrastructure pour sécuriser les services de groupes de communication. En particulier, pour les services de communication qui sont menacés par des dangers liés aux environements, comme l'Internet.

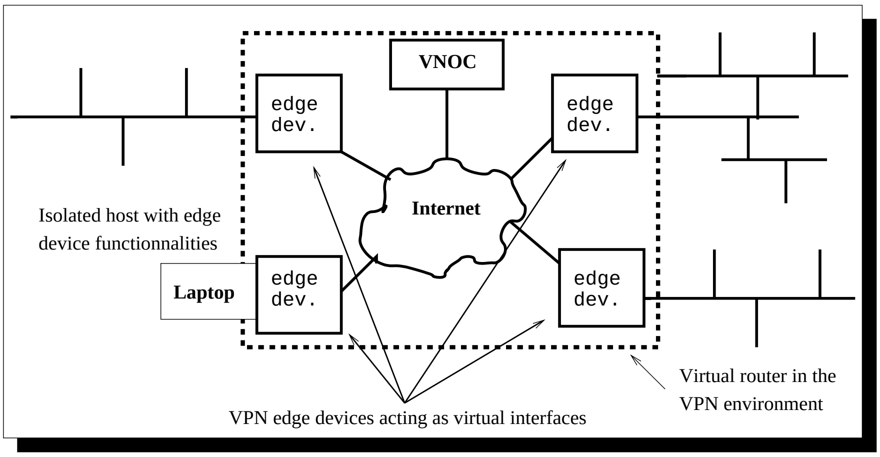

VNOC
edge dev.
edge dev.
Internet
Isolated host with edge device functionnalities
Laptop
edge dev.
edge dev.
Virtual router in the VPN environment
VPN edge devices acting as virtual interfaces

Figure 12.1. Le concept du routeur virtuel.

12.2 Notre Approche: Architecture IVGMP

12.2.1 Le Concept du Routeur Virtuel

Le support VPN dans chaque équipement CE d'un site VPN assure le cheminement du packets multicast du site local jusqu'aux sites distants qui participent au même VPN, à travers l'Internet. L'interconnection VPN de plusieurs équipements CE fait le modèle d'un seul routeur virtuel (Voir la figure 12.1) avec plusieurs interfaces virtuels, un pour chaque équipement CE. On introduit le protocole IVGMP dans ce routeur virtuel pour administrer la configurations des interfaces virtuels et pour permettre la transmission des packets multicast. IVGMP est un protocole alternatif de ceux du routage multicast, dans un environement IP VPN.

12.2.2 Description Détaillée d'IVGMP

L'Ajout d'un Nouveau Récepteur dans un Site VPN

D'abord, on assume que chaque site VPN est composé par un seul LAN. Le protocole IVGMP, situé sur chaque équipement CE, a l'objet de trouver les nouveaux membres d'un groupe mulicast dans le site local attaché au CE. Il utilise le mécanisme du IGMP (Internet Group Management Protocol) [33] et ses propres messages Query/Report pour découvrir les nouveaux clients. Le protocole IVGMP écoute le trafic IGMP dans le site local pour savoir si il y a des nouveaux clients qui ont l'attention de joindre un groupe multicast, ou pas (Voir la figure 12.2). Dans le premier cas, IVGMP fait le nécessaire pour établir une nouvelle branche VPN pour accéder au reste du groupe, si le site local n'appartient pas à ce groupe. IVGMP envoie une requête JOIN_VPN au VNOC pour participer au groupe. Le VNOC reçoit la requête, vérifie le droit du site pour joindre le groupe et envoie une lettre de confirmation au site. Le VNOC déploie des nouveaux politiques de configuration sur le site local et sur les autres sites participés au même groupe. Si le IVGMP découvert que aucun client s'intéresse au groupe multicast, le IVGMP démonte la branche VPN avec le reste des sites participent à ce groupe, si la branche existe déjà.

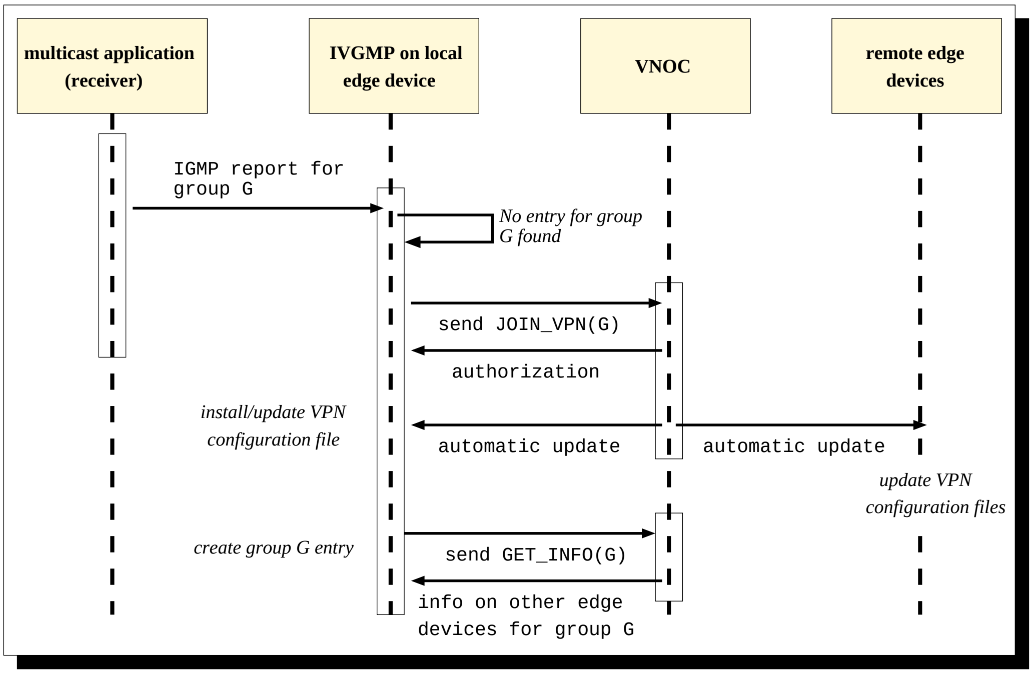

multicast application (receiver)
IVGMP on local edge device
VNOC
remote edge devices
IGMP report for group G
No entry for group G found
send JOIN_VPN(G)
authorization
install/update VPN configuration file
automatic update
automatic update
update VPN configuration files
create group G entry
send GET_INFO(G)
info on other edge devices for group G

Figure 12.2. Joindre un groupe multicast (Client).

L'Ajout d'un Nouveau Emetteur dans un Site VPN

Un procés similaire se déroule pour l'adminstration des sources multicast. Dans ce cas là, le IVGMP écoute tous les paquets multicast émis par le site local, vérifie si une nouvelle branche VPN doit être établir pour un groupe multicast G et enfin envoie un message JOIN_VPN(G) au VNOC.

L'Interopérabilité d'IVGMP et les Protocoles du Routage Multicast

Quand le site VPN est composé par plusieurs sous-réseaux, un protocole du routage multicast est nécessaire pour le cheminement du trafic multicast [38]:

problème récepteur: l'équipement CE ne va pas recevoir des messages IGMP émis par des membres de groupe situés dans les sous-réseaux intérieurs qui ne sont pas attachés directement au CE.

problème émetteur: le routeur multicast le plus proche de l'équipement CE, situé dans le même sous-réseau de celui de CE, ne va pas transmettre le trafic multicast, venu d'une source interne, si il n'y a pas un récepteur dans le sous-réseau de CE.

Plusieurs solutions sont possibles pour résoudre ce problème:

- L'utilisation d'un proxy IGMP [35] pour cheminer les 'Report' messages d'IGMP jusqu'au l'équipement CE. L'inconvenient de cette solution qu'elle nécessite l'administartion de quelque équipiements dans le site client, qui n'est pas favorable. D'autre part, cette solution adresse seulement le problème récepteur.
- En cas où on a une liste prédéfinie des adresses multicast utilisés entre des sites VPN, IVGMP peut s'incrire à ces groupes, en envoyant des messages IGMP 'Report' à chaque

fois qu'un routeur multicast envoie un IGMP 'Query'. Dans ce cas, le trafic multicast envoyé par des sources internes, sera transmettre jusqu'à l'équipement CE. L'inconvenient de cette solution est d'avoir plein du traffic de signalement d'IGMP.

- Une autre solution est basée sur des applications dédiées, utilisées par les clients et qui informent l'IVGMP de la présence d'une source multicast ou des nouveaux clients pour des groupes multicast. L'IVGMP peut donc contacter le VNOC et inscrire le site local au groupe multicast. Cette solution met un peu de charge sur les clients, mais évite des modifications dans le site local.

12.3 Implémentation d'IVGMP

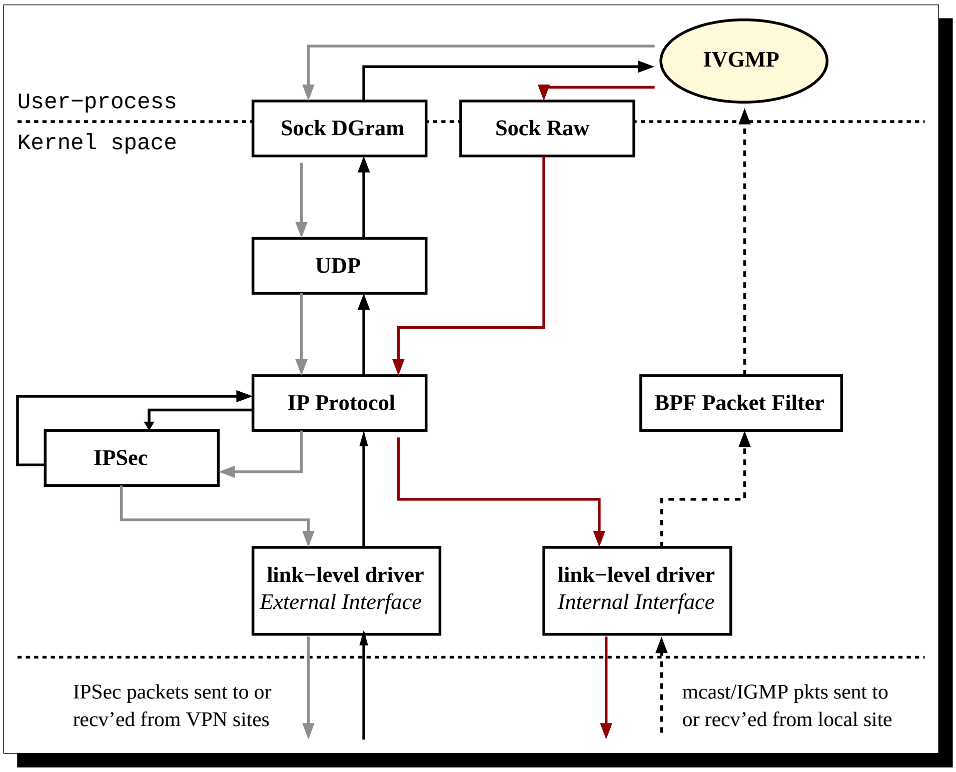


Figure 12.3. Vue architecturale du traitement des paquets IVGMP.

On a implémenté le protocole IVGMP dans des équipements PC/Linux (voir la figure 12.3). On a utilisé une version modifié de FreeS/Wan IPsec, l'implémentation IKE pour OpenBSD [42], et les utils d'administration de Netcelo (le VNOC) [63].
Les communications IVGMP-VNOC utilisées pour maintenir la liste des VPNs pour le fonctionnement du service multicast, sont basées sur le protocole SOAP 'Simple Object Application Protocol' des services web. Ceci offre également lavantage dune grande souplesse pour faire face aux problèmes actuels d'interopérabilité relatifs aux pare-feux/ proxies 6.
Voici les démarches du traitements des paquets multicast sur chaque équipement CE: Un paquet multicast, venu d'une source active dans le site local, est intercepté par la filtre BPP [48] tournée sur l'équipement CE et envoyé au démon IVGMP. IVGMP cherche une entrée VPN qui correspond à l'addresse multicast de destination pour vérifier si le paquet doit être transmis aux autres sites distants. Si une entrée est trouvée, une copie du paquet

sera encapsulée dans un segment UDP/IP pour chaque sites distants et puis délivrée au IPsec qu'il l'envoie sur le tunnel convenable. Dans l'autre sens, les paquets reçus par les sites distants sont traités par IPsec, IP, UDP et puis IVGMP. Ce dernier injecte l'orginal du paquet multicast sur le site local à travers les raw sockets.

12.4 Autres Travaux dans le Domaine

L'approche discutée dans ce chapître, adopte plusieurs technologies comme l'utilisation des réseaux VPNs sur IP, le service web (SOAP) et les réseaux programmables pour offrir une solution simple et flexible pour sécuriser les services de groupes de communication sur l'Internet [5]. En même temps, notre approche a des points de similarités avec l'approche CM 'Centralized Multicast' [51]. Dans l'approche CM, le trafic des données est séparé de celui du contrôle et la partie du contrôle est centralisée dans plusieurs noeuds distribués. La différence majure entre cette approche et la notre, est la sécurité, puisque le CM n'addresse pas la sécurité du trafic des données.

Chapter 13

Notre Contribution: dans les Services Web

Contents

Les services web prennent leurs origines dans l'informatique distribuée et dans l'avènement du Web (en particulier le commerce électronique sur l'Internet). Ils offrent des services applicatifs publiés, localisés et exécutés sur l'Internet, et accessibles via des protocoles standardisés du web par des entités distantes (applications ou utilisateurs finaux) [59]. L'adoption de standards comme XML, SOAP, WSDL et UDDI a largement contribué au succès car cela résout le problème d'interopérabilité entre les différentes plates-formes [64]. Néanmoins des obstacles majeurs à l'adoption des services web restent, à savoir la sécurité et surtout l'administration dynamique des services web. Dans ce chapitre, on va discuter les différents obstacles des services web actuels, et nous proposons un modèle hybride qui fusionne les services VPNs dans un architecture des services web. La suite de ce chapiter est organisé ainsi : la section 13.1 discute les problèmes d'administration et de sécurisation des services web; nous introduisons notre proposition au sein de la section 13.2, puis rendons compte d'une première évaluation du système en section 13.3. Finalement nous présentons des travaux associés puis nous concluons.

13.1 Etat de l'Art: Services Web

13.1.1 Architecture d'un Service Web

Un service web typique comporte trois éléments (figure 13.1):

- un fournisseur de services web, qui offre un ou plusieurs services.
- un annuaire de services, où les fournisseurs publient la description des services fournis, et
- un client, qui demande l'exécution d'un service web. Ces clients interrogent l'annuaire, récupèrent les descriptions des services voulus, puis les invoquent.

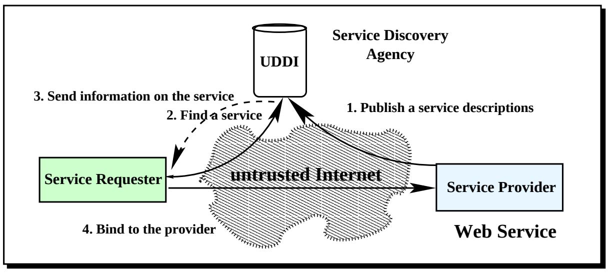


Figure 13.1. Architecture d'un Service Web.

L'utilisation de protocoles standards est une des raisons de la réussite des services web. C'est le cas de XML et HTTP, bien connus, mais aussi des protocoles suivants :

- SOAP : Ce protocole définit un modèle d'échange de messages XML via HTTP entre clients et fournisseurs de services web. Ces messages contiennent le nom et les paramètres du service web à exécuter, et en retour contiennent les résultats. En plus, SOAP bénéficie aussi de la tolérance des pare-feux au trafic HTML.
- UDDI : Ce protocole d'annuaire permet de trouver le service web recherché (de façon manuelle ou automatique) [66], mais aussi d'en annoncer la disponibilité. Les clients d'un annuaire UDDI sont soit un fournisseur qui publie ses services web, soit un client qui recherche un service.
- WSDL [88] : Cette norme dérive de XML et décrit l'interface d'utilisation d'un service web (méthodes et propriétés des composants de l'application), le système de communication sous-jacent (un service peut proposer une communication via SOAP, via HTTP GET ou POST, ou via MIME), et ses détails de déploiement.

Les plates-formes des services web sont à ce jour soit "J2EE", soit ".NET". Ces plates-formes sont complexes et nécessitent une administration précise.

13.1.2 Evolution des Services Web et Challenges Associés

L'évolution de service web a suivi trois phases :

- La première phase a concerné les principaux blocs fonctionnels: XML, SOAP, WSDL, et UDDI. Grâce aux efforts d'organisations de standardisation tels le W3C et le WS-I (Web Service Interoperability Organization), ces protocoles sont maintenant largement matures.

- En revanche la deuxième phase, qui concerne les problèmes de sécurité et de fiabilité, est loin d'être achevée, même si de nombreux efforts sont fait par OASIS (Organization for the Advancement of Structured Information Standards), WS-I et d'autres organisations (section 13.1.2).
- La troisième phase concerne le provisionnement, la surveillance et l'administration des services. Cette phase en est encore à ses débuts, malgré l'urgence.

Plusieurs challenges restent donc ouverts, et nous allons maintenant les détailler.

Challenges Liés à l'Administration

Un bon système d'administration nécessite un ensemble flexible et inter-opérable de primitives permettant de déployer et surveiller les services web. Dans cet article nous nous concentrons uniquement sur les opérations externes d'administration, entre les fournisseurs et leurs clients, à savoir :

- l'identification des clients: chaque client sous contrôle du système possède un identifiant unique.
- l'authentification et autorisation des clients: l'authentification des clients est la première étape, la deuxième consistant à autoriser ou non l'accès aux services.
- les opérations de surveillance: un système de surveillance est essentiel pour connaître à tout moment l'état des services web, la liste des clients de chaque fournisseur de service web, voir le temps moyen pour l'exécution d'un service web.
- le déploiement des fichiers de configurations: des fichiers de configurations sont à déployer pour assurer des communications sécurisées entre clients et fournisseurs.

Malheureusement les standards actuels ne permettent pas de gérer ces aspects.

Challenges Liés à la Sécurité

Les services web n'échappent pas aux éternels problèmes de sécurité. Les flux SOAP étant véhiculés sur le port 80, les pare-feux traditionnels qui ne pratiquent pas d'inspection du contenu, sont à priori incapables de détecter la majorité des attaques. L'essentiel des efforts dans ce domaine ont donné naissance à plusieurs langages de sécurité (mais qui ne sont pas encore mis en oeuvre) :

- Le WS-Security (Web Services Security Language) définit les extensions SOAP pour sécuriser les échanges des messages SOAP.
- XML Signature [39] définit la syntaxe (compatible XML) permettant de signer électroniquement tout ou partie d'un document (texte, image, plus généralement ressource Web) représentable par une URI (Universal Resource Identifier), et donc notamment tout ou partie d'un document XML pour assurer l'intégrité et la non répudiation des données.
- XML Encryption [40] spécifie le processus pour le cryptage de données et la représentation du résultat dans XML. Ces données peuvent être des données arbitraires (y compris un document XML), un élément XML ou le contenu d'un élément XML.
- XKMS [41] traite des services de gestion des clés et certificats (utilisé conjointement avec XML Signature).

- XACML [68] spécifie les politiques utilisées pour accéder aux documents XML selon leur contenu, le sujet et l'action (lire, écrire, créer, effacer).

- SAML [67] est un dialecte XML pour exprimer des informations d'authentification sur des systèmes sécurisés et pour définir des droits utilisateurs génériques.

- SPML [69] vise à normaliser le mode d'invocation et d'administration d'une plate-forme de provisionnement.

Tous ces standards ont pour but de sécuriser les données XML, mais ils ne visent pas à sécuriser ni le service web lui-même, ni le trafic du service web. Aussi le trafic SOAP doit-il être transporté sur HTTP/SSL ou TLS (Transport Layer Security) pour une sécurité de bout-en-bout [28]. IPSec [49] permet lui aussi de sécuriser les messages web dans un service web, avec une approche ici point-à-point.
Mais aucune de ces solutions n'offre à notre avis suffisamment de souplesse pour permettre d'administrer les politiques de sécurité entre les différents éléments du service web, tout en évitant les conflits d'interopérabilité. De plus la solution doit être dynamique et permettre de générer les politiques de configurations sans intervention humaine.

13.2 Notre Approche: Services Web VPN

13.2.1 Architecture

Notre architecture est basée sur l'intégration des services VPN et services web. On y retrouve en particulier tous les composants d'un service web ordinaire. *Le MOP, le centre d'administration centralisé, prend en charge les fonctions d'administration des services web. Cependant le MOP n'intervient pas dans les processus des services web eux-mêmes.* Le MOP contient deux éléments :

- un annuaire UDDI, qui contient les descriptions des services web, et

- un VNOC qui assure les opérations d'administration.

Notre architecture crée un service web spécifique aux opérations d'administration. On peut donc distinguer:

- les services web 'commerciaux' (ou *business web services*), fournis par les fournisseurs des services web à leurs clients, et

- le service web d'administration (ou *management web service*), fourni par le MOP á chaque élément des services web commerciaux.

Il s'agit donc d'une application récursive (!) de la notion de service web. Nous trouverons donc à la fois un fichier WSDL pour décrire les services d'administration fournis par l'interface d'administration, et un fichier WSDL pour décrire les services web commerciaux. A chaque fois SOAP est utilisé pour les échanges de messages.
Afin de simplifier les opérations administratives liées aux services web commerciaux, ainsi que les problèmes de sécurité, *le service web d'administration adopte la technologie VPN.* Ainsi le MOP crée dynamiquement un VPN pour chaque service web commercial, chaque VPN ayant une topologie en étoile et contenant:

- un fournisseur de services web: ce fournisseur est le centre du VPN,

- les clients du service web commercial: ces clients, à la périphérie du VPN, peuvent exécuter les services web associés à ce VPN, et quitter à tout moment le VPN.

Dans la suite nous utiliserons le terme 'service web VPN' pour dénoter le service web commercial associé à un VPN.

13.2.2 Les Deux Phase d'Etablissement d'un Service Web VPN

Nous présentons maintenant plus en détails les deux phases d'établissement d'un service web VPN, et les messages SOAP utilisés.

Phase de Signalisation et d'Installation du VPN

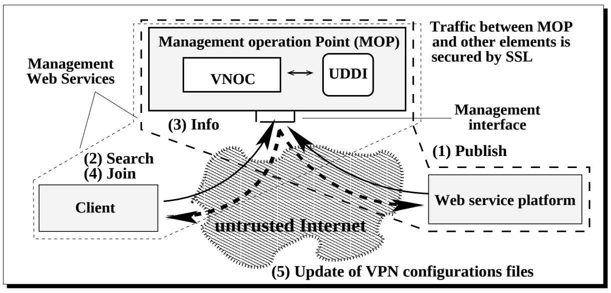


Figure 13.2. Création d'un Service Web VPN: phase de signalisation.

Cette phase concerne le service web d'administration. On identifie deux types de messages :

- les messages échangés entre le MOP et un fournisseur du service web: le fournisseur envoie au MOP un message SOAP du type *Publish*, avec un fichier WSDL décrivant en détails son interface commerciale et les services fournis par celle-ci. Le message SOAP est traité par le VNOC qui vérifie l'identité du fournisseur et son autorisation (on suppose qu'existe une liste des fournisseurs qui ont le droit d'utiliser le service d'administration). Si la réponse est positive, le VNOC enregistre le fichier de description des services dans l'annuaire UDDI, et créé un VPN centré sur ce fournisseur.

- les messages échangés entre le MOP et les clients: le client envoie un message SOAP du type *Search* au MOP afin de trouver des informations sur un service web. Une fois le client identifié et autorisé par le VNOC, un autre message SOAP du type *Info* est retourné au client avec le fichier de description du service demandé et l'identifiant du service web VPN associé. Le client envoie alors un second message SOAP du type *Join* au MOP afin de participer au service web VPN. Le VNOC ajoute alors l'ID du client aux autres membres du VPN et met à jour les fichiers de configuration du VPN afin qu'un tunnel puisse être établi entre le nouveau client et le fournisseur de services.

Deux autres messages SOAP existent :

- *Leave*: afin de quitter un ou plusieurs services web VPN, le client envoie ce message au MOP, avec l'ID des VPNs. Le MOP met alors à jour les fichiers de configuration côté fournisseurs afin de désactiver les tunnels correspondants au client.

- *GetVPN*: ce message sert à récupérer la liste d'IDs des service web VPNs auquel fait parti un client donné.

Les IDs fournisseurs (voir des clients) des services web VPNs sont stockés dans une base de données de l'opérateur VPN, après inscription au service. Cette génération des IDs constitue une phase initiale incontournable dans laquelle on vérifie l'identité de chacun (par mail ou téléphone) ainsi que le niveau de sécurité désiré.

Phase de Transfert de Données

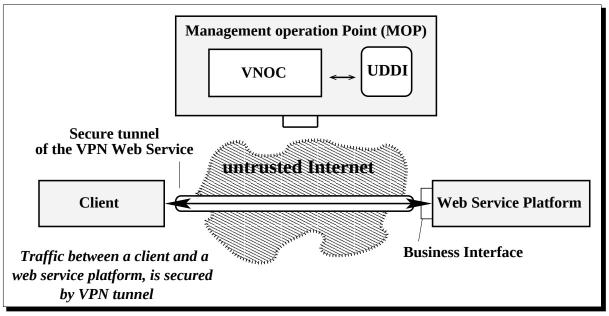


Figure 13.3. Création d'un Service Web VPN: phase de transfert de données.

Après déploiement des fichiers de configuration du VPN sur le client et le fournisseur, et obtention du fichier de description du service web, le client est désormais prêt à exécuter les services web fournis. L'établissement du tunnel est faite dès que le client envoie le premier message à l'interface commerciale pour exécuter le service web. Ce tunnel restera ensuite actif jusqu'à ce que le client envoie un message *Leave* au MOP.

Selon les politiques de sécurité déployées dans la phase précédente, le trafic entre le client et le fournisseur de service web sera sécurisé par IPsec ou SSL. Le choix entre ces deux technologies dépend des niveaux de sécurité demandés par les clients au moment de leur inscription au service VPN. D'autre part l'utilisation d'IPsec ou de SSL n'empêche pas l'utilisation d'autres technologies comme XML signature ou SAML qui ajoutent un niveau de sécurité supplémentaire.

Un avantage de l'utilisation du service web VPN est qu'il ne contient que les clients qui ont été autorisés à exécuter un service web. Les fournisseurs des services web sont ainsi déchargés des taches d'authentification/autorisation/accounting de leurs clients, ces aspects étant gérés par le VNOC.

13.3 Implémentation et Evaluation des Services Web VPN

Nous avons implémenté une grande partie de cette approche à des fins d'évaluation et validation de l'architecture. Dans ce prototype le MOP contient uniquement un VNOC (pas de base UDDI) et est construit sur une plate-forme J2EE (qui gère les opérations d'administration). Les clients utilisent un programme Perl afin de générer des messages SOAP et les envoyer à l'interface commerciale. L'interface elle-même tourne sur un serveur Apache Java.

Nous avons mesuré le temps nécessaire au traitement d'opérations d'administration sur le VNOC au sein d'un environnement opérationnel. Nous nous sommes concentrés sur trois opérations : *Join*, *Leave* et *GetVPN*. Ces tests ont montré que le traitement d'un message *Join* (figure 13.4) et *Leave* nécessite respectivement 978 microsecondes et 938 microsecondes en moyenne. Ceci est deux fois plus long que le traitement d'un message *GetVPN* qui ne prend que 492 microsecondes. Cela s'explique par le fait que les opérations *Join*/*Leave* impliquent des modifications au sein de la base de données VPN gérée par le VNOC.

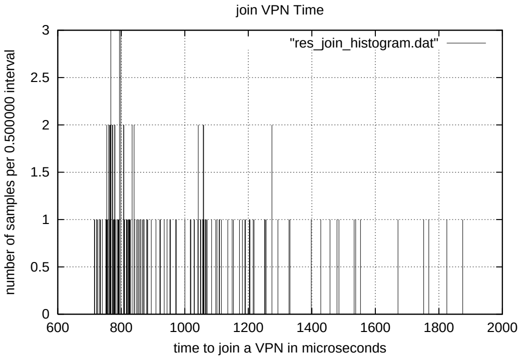


Figure 13.4. Temps de traitement d'un message JOIN (histogramme).

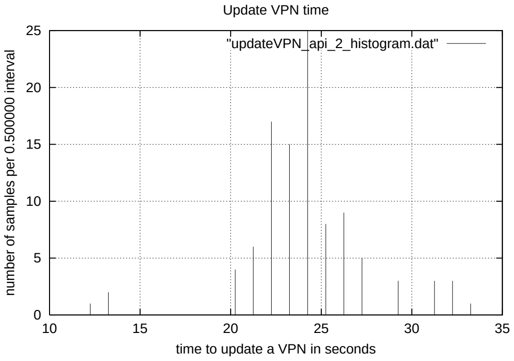


Figure 13.5. Mise à jour d'un service web VPN (histogramme).

Un problème qui affecte les performances, tout particulièrement lorsque le nombre de clients augmente, est le mécanisme d'authentification au sein de SOAP. Nous avons utilisé le modèle SOAP::Lite au sein du programme client Perl. Puisque SOAP a été développé pour n'envoyer d'informations d'authentification que sur demande expresse du serveur, le premier message émis par le client n'inclue aucune information d'authentification. L'interface d'administration se doit donc de renvoyer un message d'erreur "401: authentication error" au client, qui répond alors avec les données demandées. Nous avons résolu ce problème en obligeant le client à ajouter systématiquement des informations d'authentification au premier message SOAP généré.

Nous avons alors mesuré le temps requis pour mettre en place un service web VPN. Ce temps est l'intervalle s'écoulant entre la réception d'un message *Join* d'un client, et la mise à jour des fichiers de configuration chez le client. Nous avons découvert que ce temps est en fait fortement lié à certains paramétrages actuels du VNOC qui ajoutent un délai additionnel Δ au processus de mise à jour. Dans nos tests (figure 13.5), ce délai Δ varie entre 10 et 20 secondes, ce qui conduit à un temps moyen de mise à jour de 24 secondes. C'est le délai que doit attendre un client avant de participer à un service web VPN. Cela peut paraître long, mais (1) une fois le client connecté, alors l'invocation du service est seulement limitée par la plate-forme du service web, et non pas le VPN associé; de plus (2) nous travaillons sur ce point afin de réduire cette latence.

Nous avons finalement étudié le passage à l'échelle de notre solution. Le figure 13.6 montre le temps de traitement de messages SOAP *GetVPN* lorsque le nombre de clients augmente. Les résultats sont bons jusqu'à environ 20 clients simultanés, puis le temps moyen de traitement augmente largement. Une première raison à ce comportement est le fait que pour chaque

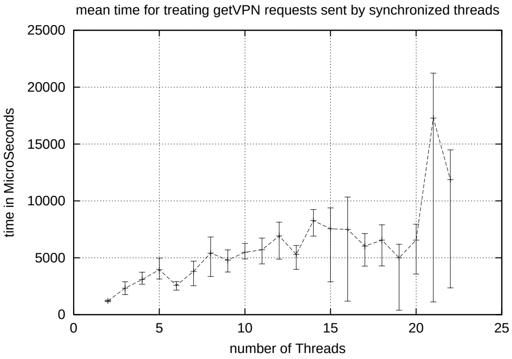


Figure 13.6. Temps de traitement de messages GetVPN envoyés simultanément par plusieurs clients.

message *GetVPN* reçu, la base de données VPN doit être accédée. Comme cette base est partagée, le nombre d'accès concurrents est limité et certains messages ne peuvent être traités de suite. Une deuxième raison probable est l'impact négatif de la plate-forme Java/J2EE sur les performances, mais ceci nécessite un approfondissement.

Nos expériences ont montré que le niveau de performances global est largement impacté par différents aspects techniques. Nous sommes confiants que les performances peuvent encore largement être améliorées et nos efforts dans ce domaine complexe se poursuivent. Par conséquent cette section doit davantage être vue comme une première évaluation d'un prototype, et non comme une étude de performances d'une solution optimisée.

13.4 Autres Travaux dans le Domaine

Un but de l'informatique distribuée est de permettre à une application donnée d'accéder à une fonction d'une autre application sur une machine distante aussi simplement que si l'appel était local, indépendamment des plates-formes et des langages utilisés. Trois standards sont aujourd'hui utilisés : CORBA/IIOP (Common Object Request Broker Architecture / Internet Inter-ORB Protocol), DCOM (Distributed Component Object Model) et RMI (Remote Method Invocation). Ces modéles offrent des mécanismes de récupération d'espace mémoire, de sécurité, de gestion du cycle de vie des objets, mais restent complexes, peu compatibles avec les pare-feux, et difficilement interopérables entre eux. Ils restent donc souvent confinés à l'intérieur des entreprises.

Les services web résolvent ce problème et tirent le meilleur profit de l'Internet en tant que plate-forme de distribution d'informations ubiquitaire et simple d'accès. Mais les problèmes de sécurité sont encore un obstacle important à leur adoption, et en dépit des efforts fournis aucun standard unique n'existe. De nombreux documents ont été écrits dans ce domaine, mais la vaste majorité d'entre eux restent purement théoriques. Le document WS-Security, qui est développé au sein de l'organisme de standardisation OASIS, devrait être publié d'ici quelques mois. L'organisme WS-I (Web Services Interoperability) a été formé récemment afin de promouvoir des standards ouverts pour l'interopérabilité des services web, et il devrait fournir des spécifications de sécurité.

Concernant les aspects d'administration, aucun effort sérieux n'a été fait, et la plupart sont des approches propriétaires émanant des principaux fournisseurs. Récemment un nouveau comité technique au sein d'OASIS a été créé, le Web Services Distributed Management

(WSDM) Technical Committe, afin de faire le lien entre OASIS et d'autres organismes de standardisation tels le W3C Web Services Architecture Working Group et le Distributed Management Task Force (DMTF). Ceci inclue l'utilisation des technologies service web afin d'administrer des ressources distribuées. La première spécification Web Services Distributed Management (WSDM) V1.0 est attendue début 2004.

Outre ces organismes, plusieurs compagnies telles Microsoft, Actionl, Amberpoint, ont dors et déjà compris l'urgence des besoins en matière de solutions d'administration. Ainsi Microsoft doit livrer sa nouvelle solution d'administration pour .Net, Microsoft Operations Manager MOM 2004, en été 2004.

Enfin les précédentes sections ayant dors et déjà discuté avec force de détails plusieurs travaux liés à la sécurité des services web, nous ne les reprendrons pas ici.

Bibliography

[1] *Cisco Secure VPN Client Solutions Guide*, 2002. Cisco Systems Inc., Number: OL-0259-02.

[2] *Guidelines on IPVPN Deployment: models, architecture and technologies*, April 2002. Eurescom Project Report.

[3] L. Alchaal, V. Roca, A. El-Sayed, and M. Habert. A vprn solution for fully secure and efficient group communications. July 2003. 8th IEEE Symposium on Computers and Communications (ISCC'03), Kemer - Antalya, Turkey.

[4] L. Alchaal, V. Roca, A. El-Sayed, and M. Habert. *A VPRN Solution for Fully Secure and Efficient Group Communications.* INRIA, Rhône Alpes, April 2003. Research Report number RR-4799.

[5] L. Alchaal, V. Roca, and M. Habert. Offering a multicast delivery service in a programmable secure ip vpn environment. October 2002. Fourth International Workshop on Networked Group Communication (NGC'02), Boston, USA.

[6] L. Alchaal, V. Roca, and M. Habert. De l'utilisation des vpns pour l'administration et la sécurité des services web. June 2004. 3ème Conférence sur la Sécurité et Architectures Réseaux - SAR 2004, La Londe, Cote d'Azur, France.

[7] L. Alchaal, V. Roca, and M. Habert. Managing and securing web services with vpns. July 2004. the second IEEE International Conference on Web Services - ICWS 2004, San Diego, California, USA.

[8] K. C. Almeroth. The evolution of multicast: From the mbone to interdomain multicast to internet2 deployment. January 2000. IEEE Network.

[9] D. G. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. October 2001. Proc. 18th ACM SOSP, Banff, Canada.

[10] L. Andersson and T. Madsen. *PPVPN terminology.* PPVPN Working Group, September 2003. draft-andersson-ppvpn-terminology-04.txt, work in progress.

[11] Array Networks Inc. *SSL VPN vs IPSec VPN*, January 2003. white paper.

[12] J. Barrett. *A Response to the Feature on IPv6 vs SSL.* Root Prompt Org., June 2000. http://rootprompt.org.

[13] M. Baugher, R. Canetti, L. Dondeti, and F. Lindholm. *MSEC Group Key Management Architecture*, June 2004. Work in Progress, <draft-ietf-msec-gkmarch-08.txt>.

[14] M. Baughter, R. Canetti, T. Hardjono, and B. Weis. *IP Multicast issues with IPsec*, June 2003. work in progress, <draft-ietf-msec-ipsec-multicast-issues-01.txt>.

[15] U. Blumenthal and B. Wijnen. User-based security model (usm) for version 3 of the simple network management protocol (snmpv3). April 1999. IETF Request for Comments, RFC 2574.

[16] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer. *Simple Object Access Protocol (SOAP) 1.1, May 2000. W3C Note.* http://www.w3.org/TR/SOAP/.

[17] I. Brown, J. Crowcroft, M. Handley, and B. Cain. Internet multicast tomorrow. December 2002. The Internet Protocol Journal (IPJ), vol 5 no 4.

[18] M. Burner and R. Stadler. Virtual active networks- safe and flexible environments for customer-managed services. October 1999. Tenth IFIP/IEEE International Workshop on Distributed Systems Operations and Management (DSOM'99), Switzerland.

[19] K. Calvert. *Architectural Framework for Active Networks*, July 1999. Active Networks working group.

[20] R. Canetti, P.C. Cheng, D. Pendarakis, J. Rao, P. Rohatgi, and D. Saha. *An IPSec-based Host Architecture for Secure Internet Multicast*, February 2000. Network and Distributed System Security Symposium (NDSS00), San Diego, USA.

[21] M. Carugi and D. McDysan. *Service requirements for Layer 3 Virtual Private Networks.* L3VPN Working Group, July 2004. draft-ietf-l3vpn-requirements-02.txt, work in progress.

[22] B. Chapman and E. Zwicky. *Building Internet Firewalls.* O'Reilly Edition, ISBN 1-56592-124-0, 1995.

[23] Check Point Software Technologies Ltd. *IPSec Versus Clientless VPNs for Remote Access*, September 2002. white paper, http://www.checkpoint.com.

[24] B. Cheswick and S. Bellovin. *Firewalls and Internet Security.* Wesley Edition, ISBN 0-201-63357-4, 1994.

[25] J. Clabby. *What are Web Services.* Inc. InformIT Division, September 2002. http://www.informit.com.

[26] J. De Clercq, O. Paridaens, A. Krywaniuk, and C. Wang. *An Architecture for Provider Provisioned CE-based Virtual Private Networks using IPsec.* L3VPN Working Group, February 2004. draft-ietf-l3vpn-ce-based-02.txt, work in progress.

[27] R. Cohen. On the establishment of an access vpn in broadband access networks. February 2003. IEEE Communications Magazine.

[28] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*, January 1999. IETF Request for Comments, RFC 2246.

[29] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the ip multicast service and architecture. *IEEE Network*, pages 78–88, January 2000.

[30] A. Durand, P. Fasano, I. Guardini, and D. Lento. *IPv6 Tunnel Broker*, January 2001. IETF Request for Comments, RFC 3053.

[31] A. Elsayed and V. Roca. Improving the scalability of an application level multicast protocol. *10th International Conference on Telecommunications (ICT'2003)*, February 2003.

[32] A. Elsayed, V. Roca, and L. Mathy. A survey of proposals for an alternative group communication service. *IEEE Network, special issue on Multicasting: an enabling technology*, January 2003.

[33] B. Fenner. *Internet Group Management Protocol, Version 2*, November 1997. IETF Request for Comments, RFC 2236.

[34] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*, November 2001. IETF PIM working group, work in progress,<draft-ietf-pim-sm-v2-new-04.txt>.

[35] B. Fenner, H. He, B. Haberman, and H. Sandick. *IGMP-based Multicast Forwarding (IGMP Proxying)*, November 2001. Work in Progress, <draft-ietf-magma-igmp-proxy-00.txt>.

[36] FreeS/Wan org. *FreeS/Wan project home page: an open-source implementation of IPSEC and IKE for Linux*. http://www.xs4all.nl/ freeswan/.

[37] B. Gleeson. *Uses of IPsec with Provider Provisioned VPNs*. PPVPN Working Group, August 2001. draft-gleeson-ipsec-ppvpn-00.txt, work in progress.

[38] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. *A Framework for IP based Virtual Private Networks*, February 2000. IETF Request for Comments, RFC 2764.

[39] S. Gokul. *XML Digital Signatures*. Inc. InformIT Division, August 2002. http://www.informit.com.

[40] S. Gokul. *XML Encryption*. Inc. InformIT Division, August 2002. http://www.informit.com.

[41] S. Gokul. *XML Key Management (XKMS)*. Inc. InformIT Division, September 2002. http://www.informit.com.

[42] N. Hallqvist and A. Keromytis. Implementing internet key exchange, ike. In *Usenix Annual Technical Conference*, June 2000.

[43] M. Handley and V. Jacobson. *SDP: session description protocol*, April 1999. IETF Request for Comments, RFC 2327.

[44] D. Harkins and D.Carrel. The internet key exchange (ike). In *IETF Request for Comments, RFC 2409,*, November 1998.

[45] J. Harrison. *VPN Technologies - a comparison*. Data Connection Ltd., February 2003. http://www.dataconnection.com.

[46] C. Hitema. *Short Term NAT Requirements for UDP Based Peer-to-Peer Applications*, February 2001. IETF Draft, work in progress.

[47] L. Houston. *SOAP Security Issues.* Sun com., December 2001. http://sunonedev.sun.com/building.

[48] V. Jacobson and S. McCanne. A bsd packet filter: A new architecture for user-level packet capture. In *Usenix Winter Conference, San Diego, California*, pages 259–269, January 1993.

[49] S. Kent and R. Atkinson. *Security Architecture for the Internet Protocol*, November 1998. IETF Request for Comments, RFC 2401.

[50] Z. Kerraval. *The Evolution of Virtual Private Networks*, October 2002. white paper, Yankee Group.

[51] S. Keshav and S. Paul. Centralized muticast. In *7th Int. Conference on Network Protocols (ICNP'99), Toronto, Canada*, October 1999.

[52] P. Knight, H. Ould-Brahim, and B. Gleeson. *Network based IP VPN Architecture using Virtual Routers.* L3VPN Working Group, April 2004. draft-gleeson-ipsec-ppvpn-00.txt, work in progress.

[53] D. Kosiur. *Building and Managing Virtual Private Networks.* John Wiley & Sons Inc., ISBN 0-471-29526-4, 1998.

[54] P. S. Kruus and J. P. Macker. Techniques and issues in multicast security multicast security. *MILCOM 98*, October 1998.

[55] V. Kumar. *MBone: Interactive Multimedia on the Internet.* New Riders Publishing, ISBN 1-56205-397-3, 1996.

[56] M. Lerner and K. Elsayed. Topics in internet technology: Scalability and policy for the services-enabled internet. May 2002. IEEE Communications Magazine.

[57] N. Lippis. *The Distributed Intelligent Network Architecture*, October 2003. Lippis Report Vol 12.

[58] N. Lippis. *Distributed Intelligent Networking Part 2: Network Services*, October 2003. Lippis Report Vol 13.

[59] V. Machiraju, A. Sahai, and A. van Moorsel. Web service management network: An overlay network for federated service management. August 2002. IEEE/IFIP IM 2003, Colorado Springs, USA.

[60] N. Mitra. *SOAP Version 1.2 Part 0: Primer.* W3C Working Group, December 2002. work in progress.

[61] D. NAG and S. Scoggins. *Security In Carrier VoIP Applications*, April 2004. white paper.

[62] A. Nagarajan. *Generic Requirements for Provider Provisioned Virtual Private Networks (PPVPN)*, June 2004. IETF Request for Comments, RFC 3809.

[63] Netcelo S.A. http://www.netcelo.com.

[64] E. Newcomer. *Understanding Web Services: XML, WSDL, SOAP, and UDDI.* Addison Wesley Professional, ISBN 0-201-75081-3, 2002.

[65] A. Nghiem. *The Basic Web Services Stack.* Inc. InformIT Division, February 2003. http://www.informit.com.

[66] OASIS Standard. *UDDI Spec Technical Committee Specification Version 3.0*, July 2002.

[67] OASIS Standard. *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1*, September 2003.

[68] OASIS Standard. *eXtensible Access Control Markup Language (XACML) version 1.0*, February 2003.

[69] OASIS Standard. *Service Provisioning Markup Language (SPML) Version 1.0*, June 2003.

[70] D. Ooms and J. De Clercq. *Overview of Multicast in VPNs*, February 2002. work in progress, <draft-ooms-ppvpn-mcast-overview-00.txt>.

[71] A. Patrick. *SSL VPNs: The next hot mantra.* Network Computing company, February 2003. http://www.nc-india.com/features/.

[72] K. Psounis. Active networks: Applications, security, safety, and architectures. 1999. IEEE Communications Surveys.

[73] V. Roca and A. Elsayed. A host-based multicast (hbm) solution for group communications. July 2001. The First IEEE International Conference on Networking (ICN01), Colmar, France, pages 610619.

[74] A. Rodriguez, J. Gatrell, J. Karas, and R. Peschke. *TCP/IP Tutorial and Technical Overview*, 2001. IBM corp. Document Number GG24-3376-06.

[75] E. Rosen, Y. Cai, D. Tapan, I. Wijnands, Y. Rekhter, and D. Farinacci. *Multicast in MPLS/BGP VPNs*, February 2002. work in progress, $< draft-rosen-vpn-mcast-03.txt >$.

[76] E. Rosen and Y. Rekhter. *BGP/MPLS VPNs*, March 1999. IETF Request for Comments, RFC 2547.

[77] E. Rosen, A. Viswanathan, and R. Callon. *Multi-protocol Label Switching Architecture*, January 2001. IETF Request for Comments, RFC 3031.

[78] J. Rosenberg. *Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Multimedia Session Establishment Protocols.* MMUSIC Working Group, February 2004. draft-ietf-mmusic-ice-01, work in progress.

[79] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. *SIP: Session Initiation Protocol*, June 2002. IETF Request for Comments, RFC 3261.

[80] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, March 2003. IETF Request for Comments, RFC 3489.

[81] R. Scandariato and P. Lago. *Dynamic VPRN Provisioning: an Information Model and Architecture*, June 2000. Politecnico Technical Report DAI-SE-2000-06-14.

[82] Sherlia Y. Shi and Jonathan S. Turner. Routing in overlay multicast networks. In *Proceedings of IEEE INFOCOM'02*, June 2002.

[83] P. Srisuresh and M. Holdrege. *IP Network Address Translator (NAT) Terminology and Considerations*, August 1999. IETF Request for Comments, RFC 2663.

[84] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan. *Middlebox communication architecture and framework*, August 2002. IETF Request for Comments, RFC 3303.

[85] B. Sterman and D. Schwartz. *NAT Traversal in SIP*. IEC Annual Review of Communications, Vol. 56, ISBN 1-931695-22-9, November 2003.

[86] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. *Layer Two Tunneling Protocol 'L2TP'*, August 1999. IETF Request for Comments, RFC 2661.

[87] University of South California. *PIM-SM implementation home page, University of South California*. http://netweb.usc.edu/pim/pimd/.

[88] W3C Standard. *Web Services Definition Language (WSDL)1.1*, March 2001.

[89] W3C Working Group. *Web Services Architecture*, November 2002. work in progress.

[90] D. Waitzman, C. Partridge, and S. Deering. *Distance Vector Multicast Routing Protocol*, November 1988. RFC 1075, BBN STC, Stanford University.

[91] G. R. Wright and W. R. Stevens. *TCP/IP illustrated: the implementation, vol. 2*. Addison-Wesley, ISBN 0-201-63354-X, 1995.

[92] L. Wuu and H. Chen. A scalable framework for secure group communication. *Proceedings of the First International Conference on Networking-Part 2, 225-238*, July 2001.

[93] S. Yadav, S. Bakshi, D. Putzolu, and R. Yavatkar. The phoenix framework: A practical architecture for programmable networks. 1999. Intel Journal.

[94] E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *Proceedings of IEEE INFOCOM'96*, March 1996.

[95] Beichuan Zhang, Sugih Jamin, and Lixia Zhang. Host multicast: a framework for delivering multicast to end users. *IEEE INFOCOM'02, New York, USA*, June 2002.