Virtual Private Networks

Strong Security at What Cost?

November 2001

Ray Hunt[1] Chris Rodgers

[1]Supervisor.

Abstract

Virtual Private Networks (VPNs) are one of the most important developments in data communications in recent years, offering enterprises potentially dramatic cost savings and substantial freedom when implementing a secure Wide-Area Network (WAN). This paper examines the implications of VPN technology, which primarily involves using a shared backbone network to connect geographically dispersed sites, and requires a range of security technologies to provide confidentiality, integrity, authentication and non-repudiation to such a configuration. The various implementation and membership alternatives supported by VPNs, and their most important protocols and configuration options are also discussed. Finally, a practical investigation into the performance of a VPN environment when employing varying levels of security is documented. This investigation was conducted on a simple two-site VPN testbed, with performance measured in terms of throughput and latency for file transfers with the file transfer protocol (FTP) and the hypertext transfer protocol (HTTP). This experiment was conducted for a variety of security levels, ranging from no security to strong cryptography applied to authenticated tunnels between firewalls. It was discovered that security mechanisms can have a large impact on performance, particularly in terms of latency. This indicates that it is important to consider the performance levels a proposed VPN will be required to produce, and what hardware will be required to provide this performance, before any investment or implementation takes place.

Acknowledgements

A number of people have contributed to this report as it has progressed. Firstly, I would like to thank Professor Mike Fergusson and Associate Professor Krys Pawlikowski for providing feedback on early versions of this report, and to Jane McKenzie for providing technical writing advice. My gratitude also goes to Theuns Verwoerd, who has been generous with his time and advice throughout the year, particularly when it came to conducting the experiments. Lastly, I would like to thank Associate Professor Ray Hunt, who has gone out of his way to accommodate me this year, and has always made himself available to provide me with support and guidance.

Contents

1 Introduction **5**

1.1 What is a VPN? . . . 6

1.2 The Need for VPN Services . . . 6

1.3 The History of VPNs . . . 7

1.4 Report Structure . . . 7

2 VPN Security **8**

2.1 Cryptography . . . 9

2.2 Integrity Checksums . . . 9

2.3 Authentication . . . 10

2.4 Access Management . . . 11

2.5 Tunnelling . . . 11

2.5.1 VPN Tunnelling Protocol Requirements . . . 12

2.6 Intrusion Detection . . . 14

2.7 Remote User Security . . . 14

2.8 Packet Filtering . . . 14

2.9 Public Key Infrastructure (PKI) . . . 15

2.9.1 Implementing a PKI . . . 15

2.9.2 Digital Certificates . . . 16

2.9.3 Digital Signatures . . . 16

3 VPN Topology **17**

3.1 VPN Types and their Memberships . . . 17

3.1.1 Intranet VPNs . . . 17

3.1.2 Extranet VPNs . . . 17

3.1.3 Mobile / Dial-up VPNs . . . 18

3.2 VPN Implementations . . . 19

3.3 Virtual LANs . . . 20

3.3.1 V-LAN Membership . . . 20

3.3.2 Customer-Provider Edge (CPE) Nodes 20
3.4 Hardware Components 20

4 Protocols employed by VPNs 22
4.1 Internet Protocol Security (IPSec) 22
4.1.1 Encapsulated Security Payload (ESP) 24
4.1.2 Authentication Header (AH) 24
4.1.3 Internet Key Exchange (IKE) 24
4.1.4 IPSec Security Associations (SAs) 25
4.1.5 Using IPSec for Authentication 26
4.1.6 Conflicts Between IPSec and NAT 26
4.1.7 Conflicts Between IPSec and QoS 27
4.1.8 Applying IPSec to Overlay VPNs 27
4.1.9 IPSec vs Transport Layer Security (TLS) 27
4.1.10 Next Generation Internet Security (NGISec) 27
4.2 Network Address Translation (NAT) 28
4.3 Border Gateway Protocol (BGP) 29
4.4 Multi-Protocol Label Switching (MPLS) 29
4.5 Layer Two Tunnelling Protocol (L2TP) 30
4.6 Remote Authentication Dial-In User Service (RADIUS) 30

5 VPN Configuration and Operation 31
5.1 MPLS VPNs 31
5.1.1 The VPN-IPv4 Address Family 32
5.1.2 Security Issues 33
5.2 Provider-Provisioned VPNs (PP-VPNs) 33
5.3 Quality of Service (QoS) 34
5.4 Reliability and Survivability 35

6 Evaluation of Performance in a VPN Environment 36
6.1 Previous Work 36
6.2 Method 37
6.3 Results 39
6.3.1 The Impact of Firewalls 40
6.3.2 The Impact of Authenticated Tunnels 41
6.3.3 The Impact of Encryption 42
6.3.4 The Impact of Routers 42
6.4 Discussion 42

6.4.1 Further Work 43

7 Conclusion **44**

A Glossary **45**

Chapter 1

Introduction

The need for secure communication across distributed computer networks is an important concern for users of data communication systems, and is particularly significant for large enterprises that need to share sensitive information between branch offices and remote users. Secure communications links may also be required between an enterprise and its suppliers and customers. Such needs have traditionally been met by maintaining a private Wide-Area Network (WAN) using facilities such as leased lines, Integrated Services Digital Network (ISDN), Frame Relay and other network infrastructures to support a closed community of users at the relevant locations.

The Internet has experiencing rapid growth in recent years, and opportunities to subscribe to managed networks have also emerged, so the option of utilising these facilities instead of costlier[1] private networks has become both feasible and attractive. The use of shared networks also offers greater flexibility: it is not unusual for telecommunications providers to take up to three months to install and activate a dedicated link [Ven01], whereas subscribing to an existing network is both faster and simpler. The use of shared networks provides connectivity to sites not directly accessible via the private network, as well as providing alternate routing facilities, adding robustness to the company network that would be prohibitively expensive to provide using dedicated means. However, using a shared network to transport sensitive and private data introduces new concerns, primarily over security and performance. Since almost anyone can transmit data across a public network, there is a real risk of traffic being monitored, captured or altered as it is in transit. Performance can at times be poor, as using a public network means competing for finite resources.

Virtual Private Networks (VPNs) have evolved as a compromise for enterprises desiring the convenience and cost-effectiveness offered by shared networks, but requiring the strong security offered by private networks. Whereas closed WANs use isolation to ensure data is secure, VPNs use a combination of encryption, authentication, access management and tunnelling to provide access only to authorised parties, and to protect data while in transit. Mobile staff can use dial-up accounts from an Internet Service Provider (ISP) as an alternative to maintaining costly dial-up equipment and dedicated phone lines [PF00].

Despite the benefits and efficiency gains offered by VPNs, widespread adoption has been hampered by the lack of inter-operable implementations from various vendors, partly due to the fact that VPNs and their scope are poorly defined [GLH+00]. Confusion is also caused by the use of the term VPN to describe a wide variety of distinct network solutions.

[1] Using a shared infrastructure is less expensive than a dedicated infrastructure due to economies of scale; the costs of the links are shared between the different users of the network.

1.1 What is a VPN?

The term Virtual Private Network (VPN) has been used to describe a wide range of networking services and configurations, ranging from services as simple as a leased-line, to complex architectures providing secure, authenticated WAN access over a public network. For the purposes of this report, a VPN is defined as a WAN with the following characteristics and functionality:

- *A VPN provides a private WAN over a public network, typically the Internet.* Connections to this WAN may be through permanent connections between sites, or through on-demand connections via the Public Switched Telephone Network (PSTN).
- *A VPN interconnects dispersed nodes of the network.* These nodes may represent branch offices or employees working from home or from the road. If the VPN provides an Extranet service (see Section 3.1.2), nodes may be offices from other companies, such as customers, suppliers or trading partners.
- *A VPN provides privacy and integrity for data as it traverses the shared network.*
- *A VPN requires user authentication before allowing communication or granting access to VPN resources.* Authentication should be applied in conjunction with suitable access management procedures in order to regulate access to the various facilities provided within the VPN.
- *A VPN should provide Quality of Service (QoS) and multicasting [Wil00] facilities when appropriate.*

Many of these services may be provided by either the service provider or by the customer, depending on the requirements of the VPN customer, with issues such as cost, security and scalability to be considered. For example, having the customer provide certain services may result in lower service costs, but may require the purchase of hardware and the recruitment of technical expertise to install and maintain the necessary infrastructure.

The term “VPN” is often applied generically in the literature, as if describing a single specific solution. However, as Sections 3.1 and 3.2 demonstrate, a variety of VPN options are available to cater for different requirements and budgets. Each distinct VPN solution has its associated strengths and weaknesses, so it is not anticipated that a single solution will become dominant, but rather that a range of approaches will continue to address various needs.

1.2 The Need for VPN Services

Secure data communication requires a trusted path between computer networks in different locations. As discussed previously, the use of shared networks to transport private data introduces security concerns. This risk is seen as sufficiently serious that some industries, such as certain sectors within the health-care and medical industries, are forbidden by U.S. law to transmit sensitive data over public networks such as the Internet [BM00].

Despite the security concerns associated with shared networks, large numbers of organisations need to interconnect geographically separate locations, and for many implementing a dedicated WAN is not feasible. Specifically, dedicated networks are too expensive for smaller organisations. VPNs offer a compromise, minimising the amount of infrastructure to be purchased, installed and managed, while still providing the strong security associated with a dedicated network. Minimising capital investment also lessens the “technology risk” associated with purchasing hardware for such a rapidly-evolving technology.

VPNs are ideal for supporting an increasingly dispersed and mobile work-force, particularly for supporting telecommuters and roaming sales staff. They also offer an efficient means for supporting collaborative efforts with business partners.

VPNs are gradually gaining widespread hardware and desktop support. For example, Windows 2000 caters for end-to-end security providing facilities such as IPSec support. In addition, many Network Interface Card (NIC) manufacturers are currently producing hardware that offers on-board IPSec encryption under Windows. Other examples include firewalls and routers that provide VPN configuration facilities and support for VPN functionality such as tunnelling, encryption and Quality of Service (QoS).

1.3 The History of VPNs

VPNs first appeared in 1984, when the service was offered to United States users by Sprint, with MCI and AT&T producing competing products soon after [Bel93]. In their initial form, VPNs were offered as a flexible, cost-effective approach to the problem of connecting large, dispersed groups of users. Private networks were the main alternative, and although equivalent functionality could be achieved via the PSTN, this was a limited solution due to the length of full national telephone numbers and the lack of in-dialling capabilities from the PSTN [WSCL+88]. Such VPNs acted as a PSTN emulation of a dedicated private network, using the resources of the PSTN in a time-sharing arrangement with other traffic.

Many forms of VPNs have existed in the technology's relatively short life-time. The level of variety has resulting from the distinct domains of expertise of the companies developing and marketing VPN solutions. For example, hardware manufacturers may offer customer premise equipment-based solutions, whereas an ISP would be more likely to offer a network-based solution.

The level of confusion and lack of inter-operability between competing products lead to the submission of a VPN framework to the IETF in 1999, which defined VPNs as "the emulation of a private WAN facility using IP facilities" [GLH+00].

A number of different VPN implementations have been proposed to operate over a variety of underlying infrastructures and protocols, including Asynchronous Transfer Mode (ATM) [KOT95], Multi-Protocol Label Switching (MPLS) [RR99], Ethernet [SSP01], IP [GLH+00], and heterogeneous backbones [LBT96].

1.4 Report Structure

This paper aims to provide an overview of the operational characteristics and the set of protocols, hardware and other mechanisms that are employed by VPNs, as well as some of the major operational concerns and challenges they currently face. Chapter 2 documents the key tools that are used to provide security to VPN traffic and data. Chapter 3 discusses the range of VPN solutions that are available, as well as the various networking devices that are used in their construction. Chapter 4 discusses the contribution of the most important protocols to VPNs, with particular emphasis paid to any implications they have for reliable and secure VPN operation. Chapter 5 documents some of the choices that are available when implementing a VPN with regard to configuration and operation. Chapter 6 documents the practical investigation conducted to determine the performance costs of security to VPNs. Chapter 7 summarises the key concepts.

Chapter 2

VPN Security

VPN customers generally expect their VPN network to offer a level of security that is at least equivalent to that offered by a private WAN configuration. Providing such a level of security in a reliable, efficient and cost-effective manner is a complex task, with many factors to consider. This chapter investigates the security concerns that arise during VPN operation, and the techniques and tools that are currently available to provide protection against their associated threats.

Security concerns that arise when transmitting data over shared networks can be divided into the following categories:

1. *Privacy.* Can unauthorised parties gain access to private VPN traffic? Privacy can be achieved through the use of encryption (see Section 2.1).
2. *Integrity.* Can VPN traffic be altered without detection? Integrity is generally achieved through the use of checksums (see Section 2.2).
3. *Authentication.* Can I be certain that the sender or recipient is really who they claim to be? Authentication is usually performed by confirming that the other party has knowledge or is in possession of some shared secret or unique item (see Section 2.3).
4. *Non-repudiation.* Can the sender or receiver later deny their involvement in a transaction? Non-repudiation, also known as data origin verification, is often achieved through the use of digital signatures (see Section 2.9.3). When non-repudiation measures are in place, the recipient cannot deny having received the transaction, nor can the sender deny having sent it.

Data transmission is not the only area of VPN operations for which security concerns exist. Factors such as physical security, access management and the ability of techniques such as social engineering [Den00] to circumvent existing security measures must also be considered, and appropriate policy decisions implemented and strictly enforced to counter these threats.

There are two main trust models applicable to the use of a shared backbone network:

1. *Untrusted Service Provider.* In this scenario, the customer does not trust their service provider to provide security for VPN traffic, preferring instead to employ CPE devices which implement firewall functionality, passing data between these devices using secure tunnels (see Section 2.5). The role of the service provider in this scenario is solely that of a connectivity provider.
2. *Trusted Service Provider.* In this scenario, the customer trusts the service provider to provide a secure, managed VPN service. Specifically, the customer trusts that VPN packets will not be misdirected, their contents inspected or modified while in transit, or subjected to traffic analysis by unauthorised parties [GLH+00].

The decision on which of these trust models is appropriate for a given implementation will depend on a number of considerations, including the scale and budget of the enterprise, and the sensitivity of the data and operations that the VPN supports.

2.1 Cryptography

Cryptographic algorithms are essential for achieving secure communications over shared networks, as a would-be attacker may obtain and store all the packets of a data stream belonging to a specific communication as it traverses the shared network. However, if the session is encrypted, then this data would provide no value unless one of the following can be achieved:

- *Decryption without possession of the cryptographic key.* This is equivalent to breaking the cryptographic algorithm used, and generally requires some form of brute-force attack on the key-space – a computationally infeasible task without access to extraordinary amounts of computational power [Den00].
- *Exploit some design or implementation flaw to obtain the cryptographic key.* Such vulnerabilities are rare, and are generally repaired rapidly upon their discovery. This requires users to monitor the status of all programs being used, particularly the operating system, and to apply any patches which are released to repair such problems, thereby removing the associated vulnerability from their systems.

Deploying encryption facilities can be expensive, particularly if specialised hardware devices are required. In addition, the use of encryption may have implications for aspects of VPN operation besides cost, such as adverse effects on performance, and the hindering of protocol analysis, fault monitoring, content filtering and other network management utilities.

There are a number of different encryption algorithms available, many of which support varying levels of security. The algorithms most commonly applied to VPNs are the Data Encryption Standard (DES) and Triple DES (3-DES), as they are supported by all implementations of IPSec. DES has been a de-facto standard in data communications for a number of years, but the ever-increasing processing capacity of computers has reached the point where only the larger key lengths supported by DES should be considered secure, such as the 128 bit keys. For example, the 1999 `DES-III` contest, which challenged contestants to crack a message protected by a 56-bit DES key, was completed in under 24 hours. More information on the level of protection afforded by different encryption algorithms and key lengths is available in “Information Warfare and Security” [Den00].

Increased security generally comes at the cost of reduced performance, so a balance between efficiency and security must be achieved when applying encryption; the expected time for an attacker to complete a brute-force cipher attack should exceed the length of time for which the data remains valuable. If this requirement cannot be met effectively, alternative options such as dynamic key re-negotiation within a given data stream may improve security without necessarily having a significant impact on performance. Such a process forces an attacker to decrypt small blocks of data at a time, dramatically increasing the difficulty of retrieving the clear-text for the entire message.

Further detail on cryptographic techniques is available in “Cryptography and Data Security” [Den82].

2.2 Integrity Checksums

A checksum is a series of bits of fixed length, whose value is derived from a given block of data. Checksums are often appended to a block of data prior to transmission, so that the receiver can verify that the data was received in the exact condition in which it was sent. Simple checksums are merely a count of the number of bits in a transmission unit, which is insufficient for security purposes as it provides no means

of verifying the integrity of the received data, only that it is of the correct size. Hash functions introduce greater computational overhead than a simple tally, but allow verification that the transmission was received successfully, free from transmission errors or deliberate tampering en route.

The basic requirements for a cryptographic hash function are as follows:

1. *The input may be of any length, but the output is always of a fixed length.*
2. *The hash function is a one-way algorithm.* A hash function is considered to be one-way if it is computationally infeasible to find some input x, when h is known, such that $H(x) = h$.
3. *The hash function is reasonably collision-free*[1]. If, given a message x, it is computationally infeasible to find a message y, where $y \neq x$ and $H(x) = H(y)$, then H is said to be a weakly collision-free hash function. A strongly collision-free hash function H is an algorithm for which it is computationally infeasible to find any two messages x and y such that $H(x) = H(y)$.

The hash value, also known as a message digest, concisely represents the longer message or document from which it was derived. A message digest acts as a form of "digital fingerprint" of the original document, and can be made public without revealing the contents of the corresponding document.

The main role of a cryptographic hash function in VPNs lies in the provision of message integrity checks and digital signatures (see Section 2.9.3).

2.3 Authentication

Authentication of VPN users is generally performed by confirming the knowledge of some shared secret. Passwords are the most user-friendly option [KD01], although the process may incorporate tokens or smart cards if enhanced security is required. Digital certificates (see Section 2.9) are a reliable way of providing strong authentication for large groups of users.

Because passwords can be cracked[2] or sniffed[3], authentication methods which use static password authentication should be considered insecure, particularly if the authentication session is not encrypted.

There are three distinct factors to authentication:

1. *What you know.* This fact should be known only by the user and the verifier, and should be infeasible to guess or derive. A common application of this factor in VPNs is an alpha-numeric password, which is verified by an authentication server running RADIUS (see Section 4.6) or an equivalent service.
2. *What you have.* In VPNs, this factor will typically be some form of identification card, smart card or token. Such items should be unique and impractical to forge.
3. *What you are.* This refers to some unique attribute of the user, such as a fingerprint or voice-print, which is impractical to forge or imitate. Such authentication methods are rarely used in VPNs at this time.

[1] Note that it is not possible to create a perfect hash function – one which is totally collision-free, because a hash function which produces an output of length n can represent only 2^n unique inputs before having to create many-to-one mappings for the remaining inputs, of which there are a potentially infinite number.

[2] Due to the static nature of passwords, they are susceptible to brute-force attacks, or possibly more sophisticated attacks if the password is weak.

[3] Data travelling over a network can be captured and analysed relatively easily using readily available tools. Some protocols, such as FTP and Telnet, send passwords in clear-text form, so the password could be readily identified and subsequently used to impersonate the victim [RHH01].

Each of these features has associated strengths and weaknesses, and as such are suitable for different applications. The limitations of each of the individual factors can be at least partially overcome, and the resulting level of security enhanced, by verifying two or more of these factors in tandem.

An important characteristic of authentication is whether the tested value is static or dynamic. Known facts such as passwords are generally static as a matter of necessity. Held items such as identification cards will also generally be static, although some sophisticated identification cards are capable of generating dynamic authentication information, using a value such as the current time to seed the generation. The third factor, an attribute of the user, is static by definition, but is nonetheless strong due to the difficulty of forging such information.

2.4 Access Management

It is not always sufficient merely to authenticate a user; too often it is assumed that once a user has been granted access, they should have widespread access to almost all resources on the VPN [Neu99]. This is a potentially dangerous assumption, since insiders can pose as great a threat to the security of an organisation as outsiders; they may have various advantages beyond privileges and access rights, such as a superior knowledge of system vulnerabilities and the whereabouts of sensitive information. Despite the risks involved, the problem of protecting a VPN and its resources from insiders is often assigned a relatively low priority, if not ignored altogether.

This is where access management techniques can be a valuable tool. Using access management tools, such as Kerberos [KN93], an enterprise can ensure that insiders are granted access only to resources for which they have been explicitly assigned access rights.

Usually, the process of managing and restricting access to VPN resources involves maintaining an Access Control List (ACL) for each of the resources for which use is restricted to specific users or groups of users. When a user attempts to access a given resource, the access management tool consults the ACL for that resource, and will subsequently grant access only if that user is represented in the list.

2.5 Tunnelling

Secure communication across the untrusted backbone network is essential to the successful operation of a VPN. Secure tunnels carry encrypted VPN traffic across this backbone, so that the contents of packets being transported between sites in the VPN cannot be viewed by third parties.

Tunnelling in itself does not provide security, but can offer encapsulation so that the target network can use its own addressing structure. To ensure security for a VPN using tunnelling, it is necessary to deploy Ingress filters to prevent external packets with GRE formatting from being injected into the VPN [PSDY00].

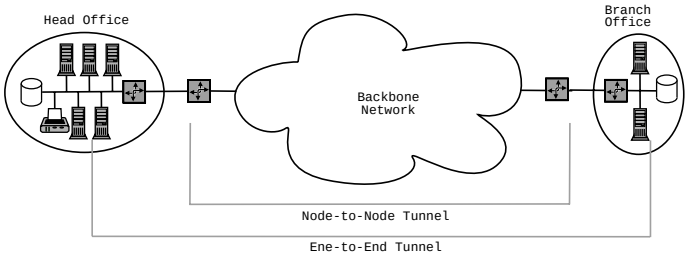

Figure 2.1: A comparison of tunnelling techniques.

Two main types of tunnelling techniques are employed by VPNs:

- *End-to-End Tunnelling,* also known as "transport model" tunnelling. The VPN devices at each end of the connection are responsible for tunnel creation and encryption of the data transferred between the two sites, so the tunnel may extend through edge devices such as firewalls to the computers sending and receiving the traffic. Secure Sockets Layer / Transaction Layer Security (SSL/TLS) is an example of a protocol which employs end-to-end tunnelling. The scope of an end-to-end tunnel is demonstrated in Figure 2.1.

  This solution is extremely secure, because the data never appears on the network in clear-text form. However, performing encryption at the end-hosts increases the complexity of the process of enforcing security policies [Shu00]; the network gateways, which would normally be responsible for enforcing security policy, are used only for forwarding the packets to their destination in this scenario, and as such they possess no knowledge of the content or purpose of the traffic. This is particularly problematic for filtering programs installed at the gateway (see Section 2.8).

- *Node-to-Node Tunnelling.* As shown in Figure 2.1, node-to-node tunnel creation and termination occurs at the gateway devices comprising the edge of the satellite networks, which are typically firewalls. Under this model, transport within the LANs remain unchanged, as it is assumed that internal traffic is inaccessible from outside the LAN. Once traffic reaches the gateway, it is encrypted and sent via a dynamically-established tunnel to the equivalent device on the receiving LAN, where the data is decrypted to recover its original format, and transmitted over the LAN to the intended recipient.

  This has an additional security advantage, in that an attacker operating a network analyser at some point on the network between the two tunnel servers would see IP packets with the source and destination addresses corresponding to those two servers - the true source and destinations are hidden in the encrypted payload of these packets. Since this information is hidden, the would-be attacker does not gain any indication as to which traffic is heading to or from a particular machine, and so will not know which traffic is worth attempting to decrypt. This also eliminates the need for Network Address Translation (NAT) to convert between public and private address spaces[4] [RMK+96], and moves the responsibility for performing encryption to a central server, so intensive encryption work does not need to be performed by workstations. This is especially important when using expensive encryption such as 3-DES, which requires hardware encryption support in order to operate without limiting effective bandwidth [KD01].

  There are two main drawbacks associated with node-to-node tunnelling:

  1. *Poor scalability.* The number of tunnels required for a VPN increases geometrically as the number of VPN nodes increases, which has serious performance ramifications for large VPNs.
  2. *Sub-optimal routing.* Since tunnels represent only the end-points and not the path taken to reach the other end of the tunnel, the paths taken across the shared network may not be optimal, potentially creating performance problems.

2.5.1 VPN Tunnelling Protocol Requirements

There are a number of IP tunnelling mechanisms available, including the Layer 2 Tunnelling Protocol (see Section 4.5), the Generic Routing Encapsulation protocol (GRE) [FLH+00], IP over IP (IP/IP), IP Security (see Section 4.1) and Multi-Protocol Label Switching (see Section 4.4). These protocols are used for packet transport across an IP network, with the transport method disjoint from the addressing of the encapsulated packets.

There are a number of desirable characteristics for a VPN tunnelling mechanism. These include the following [GLH+00]:

[4] Note that NAT will still be required when a VPN needs to connect to the Internet, to translate private VPN addresses to global IP addresses.

- *Multiplexing.* In cases where multiple VPN tunnels are required between common end-points, it is desirable for a common tunnel to be shared in order to reduce the latency and processing burden associated with tunnel establishment. A multiplexing field is therefore required so that packets belonging to different tunnels can be distinguished.
- *Signalling protocol.* Tunnel establishment can be achieved in one of two ways: via a management operation, or via a signalling protocol that supports dynamic tunnel establishment. The use of a signalling protocol is essential to many deployment scenarios, as the alternative can impose an excessive management burden. A signalling protocol would also greatly simplify the configuration process needed whenever a VPN spans multiple administrative domains.
- *Data security.* A VPN tunnelling protocol should provide support for various security requirements[5], including encryption and authentication. If tunnels are established dynamically, it is generally necessary to authenticate the party requesting that the tunnel be created.
- *Multi-protocol transport.* Many VPNs will transmit multi-protocol traffic between sites, so the tunnelling protocol which facilitates the transport of this traffic must be capable of multi-protocol transport.
- *Frame sequencing.* The ability to sequence packets in a data stream may be required in order to support the efficient operation of particular VPN protocols or applications. Such a process requires that the tunnelling mechanism support a sequencing field.
- *Tunnel maintenance.* It is necessary for VPN tunnel end-points to monitor previously-established tunnels in order to ensure that connectivity has not been lost. This can be achieved by periodically checking in-band, or through the use of some out-of-band mechanism to detect loss of connectivity.
- *Support for large MTUs.* If the MTU of the tunnel is larger than the MTU of one or more points along the tunnel path, fragmentation will be required within the tunnel, which may create choke points in the tunnel. Preferably, the tunnelling protocol will provide fragmentation and reassembly services at the tunnel end-points, so that traffic can flow smoothly through the tunnel.
- *Flow and congestion control.* These features are necessary to provide acceptable performance over networks where substantial levels of packet loss occur.
- *QoS / traffic management.* VPN customers may require specific behaviour from the network, such as guaranteed latency, bandwidth and loss rates. Delivering on such guarantees will generally be the responsibility of the VPN nodes and the backbone networks.

Facility	**Tunnelling Protocol**				
	L2TP	GRE	IP/IP	IPSec	MPLS
Multiplexing	$\surd$	$\approx$	χ	$\surd$	χ
Signalling	$\surd$	$\surd$	χ	$\surd$	$\approx$
Data Security	χ	χ	χ	$\surd$	$\approx$
Multi-Protocol Transport	$\surd$	$\surd$	χ	$\approx$	$\surd$
Frame Sequencing	$\surd$	$\surd$	χ	$\approx$	χ
Tunnel Maintenance	$\surd$	$\approx$	$\approx$	$\approx$	$\approx$
Support for Large MTUs	χ	χ	χ	χ	χ
QoS / Traffic Management	χ	χ	χ	χ	$\surd$

KEY	
$\surd$	Supported
χ	Not supported
$\approx$	Supported through extensions

Table 2.1: The capabilities of the various available tunnelling protocols.

Table 2.1 shows which of these features are provided by each of the main tunnelling protocols. As this table demonstrates, no single protocol currently provides all of these facilities.

[5] It is important to note that VPN security is not just a capability of the tunnels, as packets are also vulnerable while being forwarded into and out of these tunnels.

2.6 Intrusion Detection

An Intrusion Detection System (IDS) is a security management system for computers and networks. An IDS collects and analyses network data in real time to identify possible security breaches, which encompass both intrusions, which are attacks originating from outside the organisation, and misuse, which are attacks performed by insiders, usually in an attempt to gain access to unauthorised privileges, services or data.

An IDS may be one of the following two types:

- *Host-based IDS.* A host-based IDS resides on a single machine, and monitors the integrity of the operating and file systems, as well as analysing incoming and outgoing traffic. Host-based systems are useful for protecting dial-in VPN users such as telecommuters and roaming users (see Section 2.7).
- *Network-based IDS.* A network-based IDS is a client-server application, where client processes reside on each machine within a given network, reporting relevant information to the server, which processes this pooled information in order to identify any unusual or illegal patterns.

As discussed in Sections 2.3 and 2.4, authentication and access management services protect the VPN and its associated resources from unauthorised access. However, it is dangerous to assume that these services are infallible. An IDS is a valuable tool for the detection of abusive user or intruder activity in the event of an attack successfully compromising or bypassing the security mechanisms in place [BA96].

The use of intrusion detection is particularly relevant for Dial-up VPNs, where users access the VPN host network from remote, and potentially less secure locations, such as from home or from the road. Such users are prime targets for attack, which, if successful, would grant an attacker a level of VPN access equivalent to that normally granted to the compromised user.

Intrusion detection is also important for Extranets, where users from other organisations – whose security practices are unknown and should therefore be assumed to be weak and ineffective – are granted limited access to the VPN. An IDS will detect whether an Extranet user's account is being used to commit dangerous acts, either by the actual user or by an attacker who is posing as the user.

2.7 Remote User Security

Dial-up VPN access allows off-site staff to access VPN resources remotely by dialling in from a local ISP account (see Section 3.1.3). This represents a primary benefit of VPNs for many users, but also introduces a major security risk, as remote user machines will generally not be protected from attack by the same level of security as those residing on the host network.

Remote access leaves the VPN vulnerable if the remote machine can be successfully compromised by an attacker through the use of an exploit such as a Trojan horse, even if the user is authenticated, and communications between the VPN and the user are protected by encrypted tunnels (see Section 2.5).

The deployment of Intrusion Detection Systems (see Section 2.6) on all such machines should prevent a remote machine from becoming compromised and subsequently exploited by an attacker to gain privileged VPN access. However, implementing and enforcing a policy which requires remote users to operate a current and trusted IDS is highly problematic [KD01].

2.8 Packet Filtering

Packet filtering is the process of selectively controlling the flow of data to and from a given network. It is performed at the network gateway, generally by a firewall, and consults a previously established set of

guidelines to determine whether data packets will be allowed (granted passage through the gateway to the intended destination) or blocked (denied passage through the gateway to the intended destination). It is a means of enforcing security and appropriate Internet usage policy, by preventing unauthorised packets from entering or leaving the network.

Packet filtering can be performed on the basis of a number of packet attributes, including information which is not represented in the packet itself. Filters that examine only the header of each packet in isolation are known as basic, or static, packet filters, whereas more sophisticated techniques that examine the data segment, and evaluate a packet with regard to related packets are known as stateful, or dynamic, filters.

The packet filter can also examine the data segment of the packet. For example, if a packet is known to contain HTML data, the packet filter may wish to examine the name of the web page that it represents and compare this against a list of blocked web sites. It may also be desirable to inspect the data segment to verify that the packet is of the correct format for the port it targets. Such checks can prevent a number of denial of service attacks that rely on malformed packets [ZCC00].

Packet filtering is an important security tool for protecting VPNs; in order to preserve the integrity of the host network and connected satellite networks, it is necessary to monitor and manage incoming and outgoing traffic. Unfortunately, this places yet more burden on the network gateway devices, which may already be responsible for tunnel brokerage, cryptography and address translation. It is also difficult to reliably apply packet filtering to VPNs which employ end-to-end tunnelling (see Section 2.5).

Stateful packet filtering can impose a substantial performance overhead, especially when applied to devices that process high traffic quantities. This is a characteristic of most VPN gateways, so it is important to determine if firewall performance with VPN traffic loads will adequately support a stateful packet filtering implementation [PSDY00]. If traffic loads make stateful filtering infeasible, implementing static filtering is a less secure alternative. The risks of implementing one of the less secure filtering technologies can be partially offset through the use of suitable authentication, and if this solution is still insufficiently secure, then load-balancing technologies may be able to reduce the traffic loading on any single gateway device such that stateful filtering can be applied.

It is particularly important for stateful filtering to be applied to VPNs wherever a VPN link is used for Internet access.

2.9 Public Key Infrastructure (PKI)

A Public Key Infrastructure (PKI) encompasses a range of technologies that together provide secure and private data exchange over an insecure public network such as the Internet. A PKI provides security through use of public-key cryptography, which provides not only encryption, but also authentication and non-repudiation services. Facilities are also provided for digital certificates to identify individuals or organisations. Directory services store and, when necessary, revoke certificates.

PKI facilities are important to VPN operations, as the exchange of digital certificates is the most efficient and scalable method for performing secure shared key distribution over an untrusted network [KD01]. This mechanism can be used to provide strong user authentication that is otherwise lacking in IPSec.

The components of a PKI are discussed in `RFC 2459` [HFPS99], which defines the CCITT/ITU X.509 standard, the most widely adopted PKI specification.

2.9.1 Implementing a PKI

A VPN customer may choose to contract out their PKI requirements to another organisation, or implement their own PKI. This decision will depend primarily on the budget and security requirements for the VPN; the control offered by implementing a PKI is desirable, but will require increased investment, both initially in software and hardware, and ongoing with maintenance and support.

The Certification Authority (CA) is responsible for producing the following documents, which are crucial to the successful operation of the corresponding PKI [You00]:

1. *Certificate Policy (CP).* A CP is a formal document which specifies the requirements for issuing certificates. The CP also specifies the requirements the CA is obligated to meet, both for security and for liability purposes.

   For an IPSec end-point, the CP defines what information must be submitted to the CA to achieve certification.

2. *Certificate Practice Statement (CPS).* The CA operator must write a CPS, which defines the method of operation of the CA in order to meet the requirements outlined in the CP.

It is appropriate for the VPN customer to generate these documents even when implementing their own PKI, as they will be necessary whenever cross-certification[6] is to be performed [You00].

2.9.2 Digital Certificates

A digital certificate acts as an electronic identification to establish the credentials of a communicating party. It contains information about the certificate holder, a serial number, an expiration date and the certificate holder's public key. This information is verified by the digital signature (see Section 2.9.3) of the CA who issued the certificate.

Digital certificates in themselves do not provide authentication, as it is a trivial matter for an attacker to obtain the digital certificate of a third party and pass it off as their own. However, this possibility does not represent a practical security threat; even if a stolen certificate is mistakenly accepted, the data stream will be encrypted with the public key stored on the certificate. The attacker has no means to decrypt this traffic upon receipt, as they do not possess the corresponding private key.

Digital certificates may be employed to authenticate IPSec endpoints, whereby the two end-points wishing to negotiate a secure tunnel identify themselves through the exchange of digital certificates.

2.9.3 Digital Signatures

A digital signature acts as an electronic fingerprint; it cannot be imitated by a third party, as it is generated using the private key of the entity it identifies. As such, it can reliably authenticate the sender, as it is assumed that only the true sender possesses the given private key. It can also be used to verify that the message or document that has been transmitted has not been altered while in transit (see Section 2.2).

The process of generating and validating a digital signature is documented in [oS94].

[6] Cross-certification is the process of authentication between parties who possess certificates issued from different CAs. In such a case, the CP and CPS for each of the two CAs are compared to ensure the two certificates can be considered equal.

Chapter 3

VPN Topology

A range of VPN solutions are available to enterprises contemplating the implementation of a WAN. Each type of VPN has a number of distinguishing characteristics, in terms of membership and operation, and as such cater for specific needs. This chapter discusses the various types of VPN topologies and implementations, as well as the various networking devices that are used in their construction.

3.1 VPN Types and their Memberships

A variety of VPN implementations and configurations exist to cater for a variety of needs. Organisations may require their VPN to offer dial-up access, or to allow third parties such as customers or suppliers to access specific components of their VPN. VPNs can be classified into three broad categories: Intranet, Extranets and Mobile/Dial-up VPNs (see Figure 3.1.

3.1.1 Intranet VPNs

An Intranet connects a number of Local Area Networks (LANs) over a shared network. The purpose of an Intranet is to share information and resources amongst dispersed employees. For example, branch offices can access the network at the head office, typically including key resources such as product or customer databases. Intranet access is strictly limited to these networks, and connections are authenticated. Differing levels of access may be allocated to different sites on the Intranet, depending on their purpose.

Typically, an Intranet includes connections through one or more gateways to the Internet. These connections will generally pass through firewalls, which have the ability to filter traffic to enforce Internet usage policy and maintain security.

3.1.2 Extranet VPNs

An Extranet VPN is essentially an Intranet VPN that additionally provides restricted access to third parties such as customers, suppliers and external vendors. Such users are restricted to specific areas of the Intranet, usually denoted as the De-Militarised Zone (DMZ). It is the responsibility of the firewall and authentication and access management facilities to identify between company employees and other users, and differentiate their access privileges accordingly; employee connections should be directed to the company Intranet, whereas recognised third party connections should be directed to the DMZ.

This configuration supports a number of important e-commerce initiatives, providing opportunities for significant cost savings and efficiency gains [Ven01]. However, the increased complexity of authentication

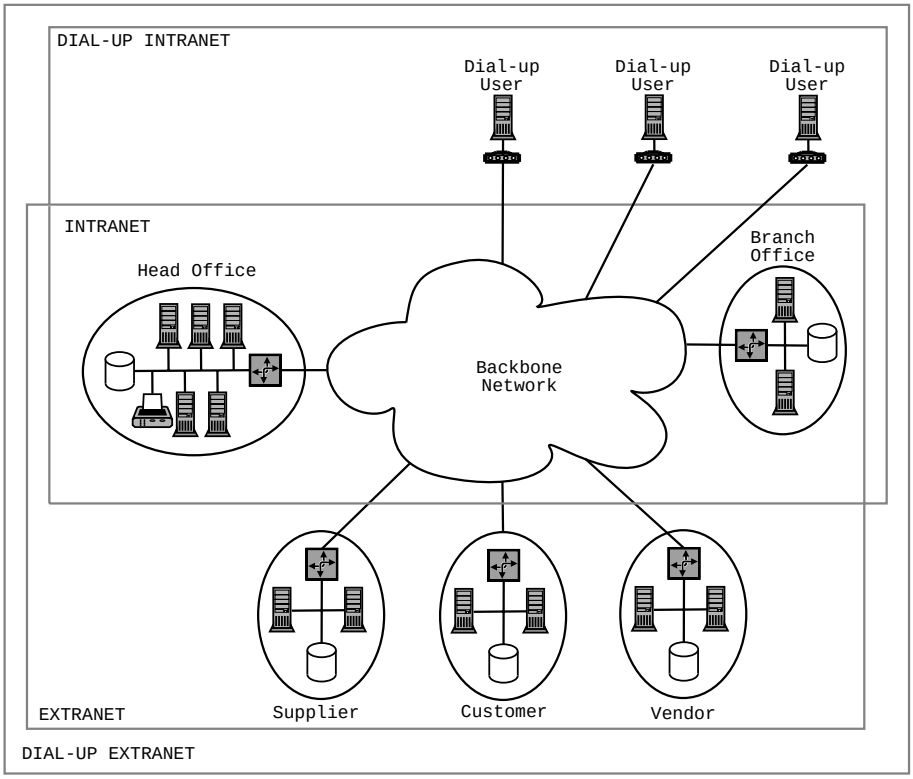

Figure 3.1: The topology and membership of various VPN types.

and access management, as well as the requirement that a separate network partition be provided to support the DMZ, may partially offset this gain.

A number of possible Extranet configurations exist, which vary in their level of security and access. The most common examples, in order from most to least secure, are as follows [Bro01]:

1. *Private Extranet.* Access to a private Extranet is strictly for members only, with no use made of shared networks. Such a configuration cannot be considered a VPN, as it is physically private.
2. *Hybrid Extranet.* A hybrid Extranet is equivalent in operation to a private Extranet, except that it utilises one or more shared networks to provide connectivity. Membership is restricted, and their access to private resources limited to relevant resources only.
3. *Extranet Service Provider.* This configuration is offered by an ISP, which builds Extranet services based on their backbone network. This is a type of Provider-Provisioned VPN (see Section 5.2).
4. *Public Extranet.* A public Extranet provides data that is globally accessible. An example is a company which provides a public web-site and possibly a public FTP site which WWW users are free to access. Such facilities are normally distinct and separate from private file servers, so that public servers cannot be used as staging points for compromising the private component of the Extranet.

3.1.3 Mobile / Dial-up VPNs

A dial-up VPN supports mobile and telecommuting employees in accessing the Intranet from remote locations. The remote employee dials in to the nearest Remote Access Server (RAS), where they are granted

access contingent upon successful authentication. The process of establishing a secure tunnel will vary, depending on which of the following approaches are taken [Ven01]:

- *Static connection.* Under this model, the RAS automatically establishes a secure connection using L2TP (see Section 4.5) to a pre-specified location inside the Intranet, providing transparent access for the user. This model is aimed at telecommuters, and requires users to dial in to a specific RAS.
- *Dynamic connection.* Under this model, the user connects to an RAS of their ISP, and then performs remote authentication with a designated server on the Intranet, after which the user is allowed access through a secure tunnel. The RAS is not directly involved in the VPN connection or tunnel establishment, so the user may connect to the Intranet via any RAS.

Deployment of a dial-up VPN can result in considerable cost savings, eliminating the need for the company to manage large modem pools, and replacing the need for toll-calls to these modems with calls to local ISP accounts [Ven01]. By taking advantage of a high-speed access infrastructure such as DSL or ISDN, some of the performance limitations typically associated with remote access can be diminished.

3.2 VPN Implementations

A number of different approaches to the problem of providing VPN links and services may be taken. In particular, a VPN may be implemented and secured at a number of the different layers of the protocol stack. Typically, VPNs are implemented at either the network or the link layer, although application layer VPNs also exist. The most significant of the currently available approaches are as follows:

- *Link Layer VPNs.* Link layer VPNs employ a shared backbone network based on a switched link layer technology such as Frame Relay (FR) or ATM. Links between VPN nodes are implemented as virtual circuits, which are inexpensive, flexible, and can offer some level of assured performance.

  Link layer VPNs are most appropriate for providing Intranet services; dial-up access is not well supported, as most ISPs provide connectivity using IP. Most of the cost savings associated with VPNs are the result of use of ISPs, so IP-based network layer VPNs are more attractive than link layer VPNs if dial-up access is required [Ven01].

  Virtual-circuit based VPNs face similar scalability issues to those of node-to-node tunnelled VPNs (see Section 2.5), so a full-mesh architecture may not be possible. Alternatives such as partial meshes or hub-and-spoke configurations address this limitation to some extent, but these solutions may produce sub-optimal behaviour, especially for routing, introducing performance concerns.
- *Network Layer VPNs.* Network layer VPNs, primarily based on IP, are implemented using network layer encryption, and possibly tunnelling (see Section 2.5). Packets entering the shared network are appended with an additional IP header containing a destination address which corresponds to the other end of the tunnel. When this node receives the packet, the header is removed and the original packet, which is addressed to some location within the given satellite network, recovered. Due to this encapsulation, the original packets could be based on any network layer protocol without affecting their transport across the shared network.
- *Overlay VPNs.* VPNs can be constructed from overlay networks, which connect subsets of resources from an underlying network and present the result as a virtual network layer to upper-layer protocols. Overlay networks rely on tunnels to provide virtual links, which are generally secured by transport-mode IPSec to provide node-to-node security [TE00].
- *Application Layer VPNs.* Application-layer VPNs are implemented in software, whereby workstations and servers are required to perform tasks such as encryption, rather than deferring these tasks to specialised hardware. As a result, software VPNs are inexpensive to implement but can have a significant impact on performance, limiting network throughput and producing high CPU usage, particularly over high-bandwidth connections [PE00].

3.3 Virtual LANs

Virtual Local Area Networks (V-LANs) [80299] provide mechanisms for partitioning a WAN (or a subset of a WAN) to appear and behave as if it is actually a LAN. A V-LAN is a network composed of physically separate nodes, connected in such a way that communication between nodes is performed transparently. This is convenient and efficient for users, but is a complex undertaking to implement and maintain.

The edge nodes of a V-LAN employ link layer bridging as opposed to network layer forwarding, so most VPN facilities can be applied to V-LANs, such as the majority of tunnelling and configuration mechanisms. However, some changes are necessary, to cater for changes in the packets and addressing information resulting from the use of a different layer for transportation. For example, some tunnelling protocols may be unusable by a V-LAN, as the use of link layer bridging introduces the requirement that the tunnelling protocol permits the transport of multi-protocol traffic [GLH+00].

V-LANs are also a convenient tool for minimising the area covered by broadcast traffic, resulting in a lower collision rate on Ethernet networks. V-LANs are also useful for security reasons, as they provide increased security over a standard switched network. However, since all network traffic will traverse the V-LAN switch, this becomes a prime target for attack. There are known attacks which will cause traffic to be moved from one V-LAN to another in most implementations [ZCC00].

3.3.1 V-LAN Membership

V-LAN membership configuration is generally simple, as the only knowledge required is that of the local VPN link assignments at any given V-LAN edge node, and the identities of other edge nodes in the V-LAN. Such configuration is independent of the nature of forwarding at each VPN edge node, so any of the regular member configuration and dissemination mechanisms can be applied to V-LANs. The topology of the V-LAN is manipulated by controlling the configuration of peer nodes at each V-LAN edge node. It is likely that V-LANs will be fully meshed, however, so that traffic between two V-LAN nodes need not pass through other V-LAN nodes, as this would require the use of loop prevention mechanisms.

3.3.2 Customer-Provider Edge (CPE) Nodes

A V-LAN CPE device may be either a bridge or a router. CPE routers would peer transparently across a V-LAN without requiring any router peering with any V-LAN nodes. The same scalability issues that apply to a full mesh topology for VPNs apply here, except that the number of peering routers is potentially greater since the ISP edge device is no longer an aggregation point [GLH+00].

The broadcast domain of a CPE bridge encompasses all CPE sites, as well as the V-LAN itself. This imposes significant scalability constraints, due to the need for packet flooding, and the fact that any topology change in the bridged domain is not localised, but is visible throughout the domain. As such this scenario is generally only suited for support of non-routable protocols.

3.4 Hardware Components

A number of hardware devices are required to implement the various types of VPNs discussed in Sections 3.1 and 3.2. Many of these devices are common to standard networks, but some have additional burdens and responsibilities placed on them when applied to VPNs and their specific requirements. The main hardware devices employed by VPNs, and the implications of any additional processing that they must perform are as follows:

- *Firewalls.* VPN satellite networks must be protected from other users of the backbone network. This is typically achieved using a firewall, which provides critical services such as tunnelling, cryptogra-

phy and route and content filtering.

- *Routers.* Adding VPN functionality (see Section 1.1) to existing routers can have an unacceptable performance impact, particularly at network stress points. Specifically, MPLS VPNs address this problem by making only the perimeter (PE) routers VPN-aware, so the core routers need not maintain the multiple routing tables which introduce so much overhead on PE routers. This is discussed further in Section 5.1.

- *Switches.* Some switches offer facilities for increased separation of traffic, by allowing a physical network to be partitioned into a number of V-LANs (see Section 3.3). On a normal switch, all ports are part of the same network, whereas a V-LAN switch can treat different ports as parts of different networks if desired.

- *Tunnel Servers.* This service may be provided by a VPN router or a firewall. Assigning an existing network component this additional responsibility may have a serious impact on performance.

- *Cryptocards.* As discussed in Section 4.1, IPSec offers the computationally-expensive Triple-DES encryption algorithm, which provides strong encryption. However, it can limit effective bandwidth to around 100Mbps unless specialised cryptographic hardware is used [KD01]. On a workstation, this specialised hardware is provided in the form of an expansion card, which may be separate or integrated with the NIC. Some firewalls also offer hardware support for various encryption algorithms.

Chapter 4

Protocols employed by VPNs

A vast number of protocols are employed by VPNs for a wide range of purposes. This chapter discusses the most important protocols to VPNs, particularly those which have been designed with support for VPNs or related tasks in mind.

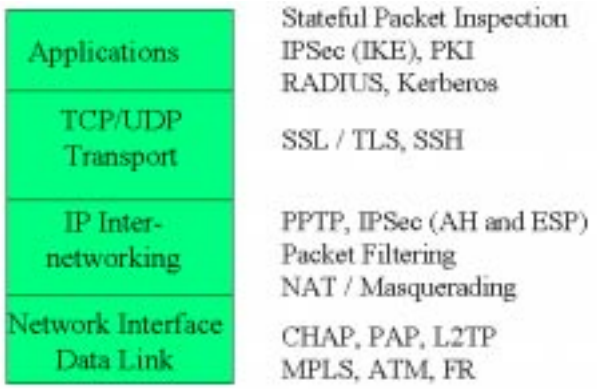

Figure 4.1: Protocols used by VPNs and their respective layers.

Due to the manner in which VPNs operate, they are often examined in the context of a VPN layered protocol model, derived from the ISO seven layer model[1]. As shown in Figure 4.1, this model segregates VPN protocols and operation into an application layer, a transport layer, an inter-networking layer and a network interface / data link layer.

4.1 Internet Protocol Security (IPSec)

Internet Protocol Security [HC98] is a suite of primarily network layer protocols, which provides inter-operable, cryptographic security for IPv6, and has also been adapted to allow use in IPv4 networks. A number of security services are provided by IPSec, including access control, connectionless integrity, non-repudiation, protection against replay attacks, confidentiality and limited traffic flow confidentiality. These services are provided at the transport layer, offering protection for IP and upper layer protocols.

The security features offered by IPSec are provided through the Authentication Header (AH - see Section 4.1.2) and the Encapsulating Security Payload (ESP - see Section 4.1.1), and through cryptographic

[1] The ISO seven layer model is discussed in “Networking Complete” [Bro01].

key management procedures (discussed in Section 4.1.3). These mechanisms are algorithm-independent, providing a modularity which permits the selection of different sets of algorithms without affecting other aspects of the implementation. For example, different user communities within a VPN may select different authentication or encryption options if required, allowing multi-level security facilities to be implemented. This flexibility is particularly important for Extranet VPNs (see Section 3.1.2), the membership of which extends to users from related organisations that operate distinct networks with their own security policies and practices. During the authentication negotiation phase prior to tunnel establishment, the two end-points negotiate to determine the highest commonly supported level of encryption. If either end fails to meet the other's minimum encryption requirements, establishment of the tunnel is abandoned.

A set of default algorithms is specified to facilitate inter-operability, which, when used in conjunction with IPSec traffic protection and key management protocols, provide strong cryptographic security. However, the development of IPSec is not yet sufficiently mature to guarantee complete inter-operability across products of various vendors [Ven01]. The presence of IPSec as a core component of IPv6 will eventually render current IPSec implementation methods obsolete as IPv6 gradually replaces IPv4, and in doing so should resolve any such inter-operability issues.

The preferred method for authentication is the use of digital certificates, which require a PKI to perform the secure key management and distribution. Unfortunately, SCEP [LMMN01], the main protocol for performing automated certificate requests, is not supported by the majority of VPN products [KD01]. This creates a weakness for IPSec: it is strong at authenticating the tunnel end-point using IKE (see Section 4.1.3), but weak at authenticating the user behind that end-point. This problem will be resolved as vendors offer greater support for PKI facilities.

IPSec can be implemented under the following topologies [You00]:

1. *Client implementation.* Also referred to as the "Bump In The Stack" (BITS), because it is inserted between the IP stack and the local network drivers. This implementation is particularly useful for legacy systems because access to the IP stack source code is not required.

2. *Gateway implementation.* Also referred to as the "Bump In The Wire" (BITW), this implementation employs IPSec-enabled equipment at the border of the network. This equipment may operate IPSec in software, such as via a router or firewall, or in hardware via a specifically designed tunnel broker, which offers superior performance, but at greater expense.

A VPN requires at least one gateway implementation at the central site or remote LAN, and a client implementation for each remote site (satellite network or remote user).

IPSec supports key regeneration[2] at specific milestones, such as every hour or for every megabyte of data, enabling the session key to be replaced as often as is required by the given security policy.

IPSec packets possess standard IP headers, and so can be routed by standard routers between VPN nodes. The IPSec packet is created by encrypting and encapsulating the original IP packet in the data segment of the new IPSec packet. The encryption algorithm and keys are negotiated and exchanged using the Internet Key Exchange (IKE) protocol (see Section 4.1.3). All implementations of IPSec are required to support the DES [oS88] and Triple-DES [KMS95] cryptographic algorithms, and some implementations additionally support algorithms such as Blowfish.

IPSec has a small number of limitations which have implications for its use in VPNs, the most serious being compatibility problems with NAT, ICMP, FTP, IKE, IP fragmentation and QoS facilities. As discussed in Section 3.1.3, mobile VPNs can be used to provide telecommuter access to the Intranet. NATs are widely deployed in home gateways, as well as in other locations likely to be used by telecommuters, so IPSec-NAT incompatibilities pose a barrier to IPSec deployment in one of its principal domains [Abo01]. This problem is discussed in Section 4.1.6.

[2] Key regeneration is the process of dynamically negotiating a new session key at some point during a previously established encrypted communication stream.

4.1.1 Encapsulated Security Payload (ESP)

The ESP header [KA98c] provides a variety of security services for IP. ESP may be applied alone, in combination with AH (see Section 4.1.2) or through the use of tunnel mode (see Section 4.1.4).

ESP can provide confidentiality, non-repudiation, connectionless integrity, replay protection and traffic flow confidentiality. The decision as to which of these facilities are to be utilised is made at the establishment of the security association (see Section 4.1.4). These should be chosen carefully, as the use of confidentiality without integrity checking or authentication may render traffic vulnerable to certain forms of active attacks, such as session hijacking, which may undermine the implied confidentiality of the service. Non-repudiation and integrity services are supported by digital signatures (see Section 2.9.3), and replay detection is provided by sequence numbers. Traffic flow confidentiality is offered in conjunction with tunnel mode, and is most effective if implemented at a security gateway, where traffic aggregation may be able to mask true source-destination patterns from traffic analysis (see Section 2.5).

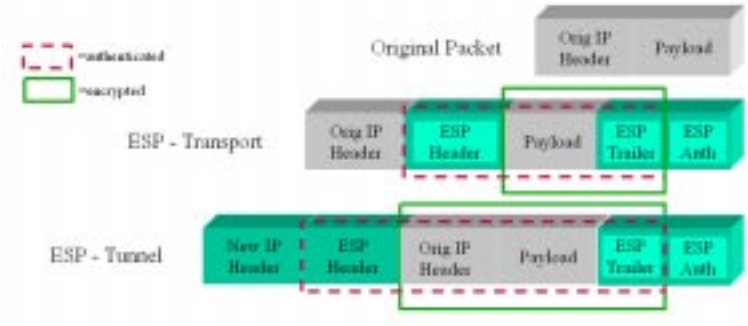

Figure 4.2: The process of affording ESP protection to IPSec packets.

Figure 4.2 demonstrates how ESP is applied to IPSec packets under the two different tunnelling modes.

4.1.2 Authentication Header (AH)

The AH [KA98b] provides connectionless integrity and data origin authentication for IP datagrams, and provides protection against replay attacks. AH provides authentication for as much of the IP header as possible, as well as for upper level protocol data. However, some IP header fields may change en route and so the value of these fields upon receipt may not be predictable by the sender. As a consequence, such fields are not afforded AH protection.

AH may be applied alone, in combination with ESP, or in a nested fashion through the use of tunnel mode (see Section 4.1.4). ESP may be used to provide the same security services, the primary difference being the extent of the coverage; ESP does not protect any IP header fields unless those fields are encapsulated by ESP in tunnel mode. The protection afforded by AH is demonstrated in Figure 4.3.

4.1.3 Internet Key Exchange (IKE)

The IKE protocol [HC98] is a sophisticated key exchange and management system, which is included in the IPSec protocol suite to provide secure key distribution services between parties wishing to communicate over an untrusted network.

IKE is a hybrid protocol composed of features from the Internet Security Association and Key Management Protocol (ISAKMP) [MSST98], Oakley [Orm98] and the Secure Key Exchange MEchanism (SKEME) [Kra96]. IKE uses parts of Oakley and SKEME in conjunction with ISAKMP to obtain authenticated keying material for security associations such as AH and ESP for IPSec. These facilities can be used for

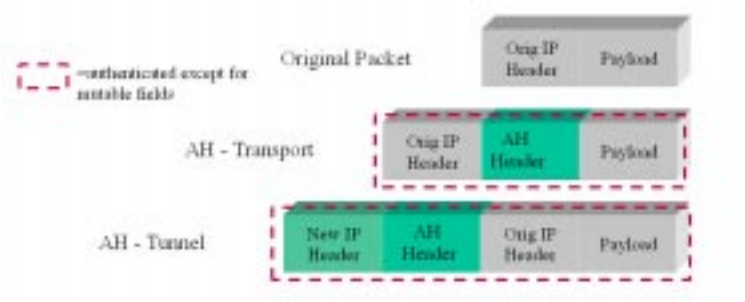

Figure 4.3: The process of affording AH protection to IPSec packets.

negotiating VPN links, as well as for providing remote users with secure access to a host or network. IKE does not require advance knowledge of the remote user's IP address, so it can readily support roaming users.

Client mode negotiation, where the negotiating parties are not the endpoints for which security association negotiation is taking place, is supported by IKE. When used in client mode, the IP addresses of the endpoints are not disclosed, so the identities of the end parties remain hidden.

4.1.4 IPSec Security Associations (SAs)

An SA is an agreement between two parties on the methods they will employ to support secure communication. This agreement is reached upon the completion of a negotiation phase that discerns the common features supported by the potentially different implementations at each end.

Security services are afforded to an IPSec SA through the use of AH or ESP, but not both. Therefore, if both AH and ESP protection is to be applied to a traffic stream, then two or more SAs must be created in order to afford the desired protection. To secure typical, bi-directional communication between two hosts or security gateways, two SAs are required; one in each direction, as SAs are uni-directional.

There are two types of security associations defined for IPSec [KA98a]:

- *Transport mode.* A transport mode SA is an agreement between two hosts. In the case of ESP, a transport mode SA provides security services only for higher layer protocols, not for the IP header or any extension headers preceding the ESP header. In the case of AH, the protection is also extended to specific portions of the IP header and any extension headers.

- *Tunnel mode.* A tunnel mode security association is essentially a transport mode SA that is applied to an IP tunnel. This mode is required whenever a security association ends at a security gateway[3], in order to avoid IPSec packet fragmentation and reassembly, and in situations where multiple paths to the same destination behind the security gateways exist. Two hosts may optionally establish a tunnel mode SA if increased security is required.

  As shown in Figure 4.2, packets in a tunnel mode SA possess both an outer IP header that specifies the IPSec processing destination, and an inner IP header that specifies the ultimate destination for the packet. If AH is employed in tunnel mode, portions of the outer IP header are afforded protection, as well as all of the encapsulated IP packet. If ESP is employed, protection is afforded only to the tunnelled packet, not to the outer header.

  Tunnel mode is commonly used to support overlay VPNs, and is required for overlay links.

[3]This restriction does not apply in situations where traffic is destined for a security gateway, such as SNMP commands, as the security gateway is acting as a host. Under these circumstances, transport mode is allowed.

4.1.5 Using IPSec for Authentication

A number of different authentication methods are offered by the various implementations of IPSec, including shared secrets such as passwords, token cards and digital certificates (see Section 2.9.2). Shared secrets are most suitable for authenticating VPN users with a small number of end-points, token cards for large Intranet implementations, and digital certificates for large Extranets, due to the scalability and cost-effectiveness of digital certificates, as well as the clearly defined processes for revoking and reissuing them.

IPSec supports Hashed Message Authentication Codes (HMAC) [KBC97] for authentication, with HMAC-MD5-96 [MG98a] and HMAC-SHA-1-96 [MG98b] algorithms the most widely used. MD5 is computationally less expensive, but SHA-1 offers superior protection, and so is generally used unless high performance is essential. A third authentication mechanism, HMAC-RIPEMD-160-96 [AK00], has recently become available. Preliminary investigation into the relative security and performance of this algorithm suggests that it compares favourably with HMAC-SHA-1-96 [rip98]. However, it is yet to receive widespread use, largely because it currently enjoys only very limited support from vendors.

4.1.6 Conflicts Between IPSec and NAT

As discussed in Section 4.1, NAT and IPSec are not completely inter-operable. The IPSec working group of the IETF is currently developing solutions to these compatibility issues, but at present there is no solution that is completely satisfactory. Stop-gap measures are undesirable, as they have the potential to create further extensibility and inter-operability issues at a later date.

NAT devices have proliferated in recent years, and increased numbers of IPv6-enabled devices will not immediately lead to their disappearance, as IPv4 is likely to remain in use by legacy systems for decades. As a consequence, bridging facilities such as NAT will be continue to be required. This is problematic for VPNs, as IPSec cannot be relied upon to operate correctly when data streams traverse NAT-bridged gateways. It is important that a stable, extensible standard for handling IPSec traffic in networks with NAT devices be defined and adopted.

The IETF Internet draft “IPSec-NAT Compatibility Requirements” [Abo01] documents a number of problems which are known to occur when an IPSec data stream attempts passage through a NAT. The most significant to VPNs are as follows:

- *Incompatibility between IPSec AH and NAT.* The AH header includes the IP source and destination addresses in the message integrity check, so changes made to these fields by NAT devices will cause the packet to be considered corrupted. This issue does not arise in ESP, as it does not include the IP source and destination addresses in its message integrity check.
- *Incompatibility between checksums and NAT.* Checksums have a dependency on the IP source and destination addresses, and so will be invalidated by passage through a NAT device.
- *Incompatibility between IKE address identifiers and NAT.* Where IP addresses are used as identifiers in IKE, modification by NAT will result in a mismatch between the identifiers and the addresses in the IP header. IKE requires such packets to be discarded.
- *Incompatibility between fixed IKE destination ports and NAPT.* A mechanism is needed to allow NAPT to demultiplex incoming IKE packets in situations where multiple hosts behind the NAPT initiate IKE SAs to the same responder. This is typically accomplished by translating the IKE UDP source port, which can result in unpredictable behaviour during key regeneration; unless the floated source port is used as the destination port for the key regeneration message, the key regeneration packets may not reach the correct destination.
- *Incompatibilities between IPSec Security Parameters Index (SPI) selection and NAT.* Since IPSec ESP traffic is encrypted and therefore opaque to NAT, the NAT must use elements of the IP and

IPSec header and the SPI to demultiplex incoming IPSec traffic. However, since the outgoing and incoming SPIs are chosen independently, NAT can not determine which incoming SPI corresponds to which destination host by inspecting outgoing traffic, so NAT may send incoming traffic to the wrong destination in situations where two hosts behind the NAT operate IPSec SAs to the same destination simultaneously.

While the majority of these problems are not intractable or catastrophic, they can severely limit the flexibility and protective power of IPSec, as well as the range of options which are viable for a given application. In many circumstances it is the higher security options which are not usable in these situations, so it may be necessary to sacrifice some security in order to maintain reliable service.

4.1.7 Conflicts Between IPSec and QoS

Many of the current QoS protocols, including the Resource ReSerVation Protocol (RSVP), MPLS, Explicit Congestion Notification (ECN), and differentiated services, do not function properly when applied to IPSec traffic streams, which hinders the adoption of IP based QoS. These incompatibilities are discussed in "NGISec-NAT and QoS compatible End-to-End Secure Communication" [Shu00].

4.1.8 Applying IPSec to Overlay VPNs

As discussed in Section 4.1.4, IPSec requires that overlay links, links between gateways and links between a host and a gateway, all employ tunnel-mode IPSec. However, the use of tunnel-mode IPSec creates problems when used in conjunction with dynamic routing in VPNs, because the tunnel-mode process binds SA selection to route selection by including routing information in information protected by the IPSec AH. As a consequence, a predefined route through the untrusted network cannot be established, so it may be necessary to replace tunnel-mode security with the weaker transport mode security association.

An alternative approach, which separates the act of tunnel encapsulation from IPSec processing, is proposed in the IETF Internet Draft "IPSEC Transport Mode for VPNs" [TE00], which also discusses the impact of this alternate use of IPSec on the IP security architecture. This solution is not perfect, however, and the draft also indicates some issues with IP/IP processing in IPSec, notably issues with IPSec encapsulation and decapsulation processing under this arrangement.

4.1.9 IPSec vs Transport Layer Security (TLS)

The TLS protocol [DA98], which is typically incorporated into individual applications, is the main alternative to IPSec for providing secure communications over an untrusted network. However, its operation at the transport layer affects its ability to provide support for VPN functionality (see Section 1.1), due to the fact that the end-points of the communication must use public addresses. IPSec, which operates at the network layer, can provide data security without modifications to relevant applications, and solves the addressing problem by using a tunnel where the original packet is encapsulated by a new header, which is discarded at the gateway of the receiving network to reveal the header containing the private address [Shu00].

IPSec is commonly used for transporting security-unaware traffic through the Internet, whereas TLS is used by security-aware applications for application layer authentication, data integrity and encryption. TLS provides true end-to-end security, but by providing the end-to-end security at the application level, security features of the operating system may be under-utilised or bypassed altogether.

4.1.10 Next Generation Internet Security (NGISec)

The incompatibilities experienced by IPSec has motivated the development of NGISec, which provides "end-to-end secure communications in LANs, VPNs, and network-to-network connections" [Shu00]. Many

of the features of IPSec are preserved in NGISec, including use of IKE (see Section 4.1.3), and it allows reuse of the existing IPSec infrastructure.

NGISec aims to ensure compatibility with other protocols, specifically NAT, ICMP and QoS protocols, and to enable the end-host to perform the majority of encryption-related tasks.

NGISec provides tunnelling facilities and end-to-end encryption in a similar manner to client-implementation IPSec, moving encryption tasks away from the gateway, which may otherwise become a bottleneck. This limits the ability of the gateway to perform policy-based tasks such as content filtering, so it may be necessary for such facilities to be installed on the end-host.

NGISec distinguishes between control packets and data packets. In order to achieve complete compatibility with NAT, control packets are decrypted at the gateway, where NAT is performed before re-encryption. It is not necessary for data packets to be decrypted at the gateway, as the network and transport layer headers remain in clear-text form, and it is claimed that since control packets make up a small proportion of total traffic, the overhead introduced by this process will be insignificant [Shu00]. Of greater concern is the fact that the gateway must possess session keys for all relevant data streams, making it an attractive target for attack. Rather than give the gateway the power to decrypt all traffic which traverses it, a preferable solution is to provide node-to-node security for control packets and end-to-end security for data packets so that the gateway only possesses the session keys for control packet streams. However, this introduces additional overhead because separate security associations must be established and key exchanges performed for the data and control flows.

This approach allows NAT to view and modify the relevant components of the packet, but will not prevent an integrity check of such a packet concluding that the packet has been corrupted, causing it to be discarded. To prevent this situation, it is necessary to reverse the effects of NAT before integrity is verified. NGISec solves this either by duplicating the information changed by NAT in the control packets, by encapsulating the original packet with extra headers, or by appending copies of the headers to the data segment of the packet. The receiving host or gateway uses this information to reverse any changes in order to restore the packet in preparation for gauging its integrity.

This operation allows the creation of SSL/TLS based VPNs, as the ability to reverse NAT solves the problem of SSL/TLS requiring the end-points to use public addresses as discussed in Section 4.1.9.

4.2 Network Address Translation (NAT)

NAT [SE01] is the process of translating an IP address used within the source network to an IP address which corresponds to the target machine on the receiving network, in a manner which is transparent to the respective end users. This process allows an enterprise to map its local addresses to one or more global IP addresses and perform a reverse mapping on the global IP addresses attached to incoming packets. This technique hides the addressing structure of the internal network from the untrusted network, and also provides a single entry point to the network at which filtering and security policy can be enforced.

NAT operates through the creation and maintenance of a table of IP address mappings. NAT can operate statically, or can translate to and from a dynamically maintained pool of IP addresses. The address translation process may be performed in conjunction with request authentication or policy-based routing[4].

NAT is an important tool for VPNs, both for the security reasons outlined above, and for allowing VPN satellite networks to employ arbitrary, possibly overlapping address spaces for internal communications. This feature is particularly useful for Extranet VPNs, whose membership includes networks from other organisations, who will have different addressing practices. However, the compatibility problems experienced between NAT and IPSec (see Section 4.1.6) hinder the use of NAT in VPNs, so it may prove desirable to seek the equivalent functionality offered by node-to-node tunnelling.

In addition, the deployment of NAT can create problems for certain applications, primarily multimedia, that

[4] Policy-based routing is a mechanism for routing traffic sent by different sets of users through different paths.

require the use of multiple service-level ports for communication. Solutions to these problems are available, but these may prove impractical for some applications and networks due to the increased processing and memory load on the affected routers, and the associated operational costs and performance implications [PSDY00].

4.3 Border Gateway Protocol (BGP)

BGP [RL95] exchanges routing information between gateway hosts, such as those connecting different backbones. BGP communicates with autonomous networks using Internal BGP (IBGP), so routers inside the autonomous network must maintain two routing tables: one for the Interior Gateway Protocol (IGP) and one for IBGP.

BGP communities can be used to control route propagation, by allowing a VPN to "mark" BGP Network Layer Reachability Information (NLRI) with a community attribute that will control route propagation in accordance with a community profile [PSDY00].

VPN membership uses a unique route distinguisher (RD) that is assigned to each VPN during provisioning. In an MPLS VPN, BGP distributes Forwarding Information Base (FIB) table updates about each VPN only to edge LSRs that have that specific VPN sites attached to them, preventing each VPN FIB appearing in all edge LSR routing tables. Within the core, a routing protocol such as OSPF distributes routing information. Label-binding information for external routes is distributed between provider edge routers using BGP multi-protocol extensions instead of the Label Distribution Protocol (LDP), which is used for internal routes, because BGP extensions easily attach to VPN-IP information already being distributed. The BGP community attribute constrains the scope of reachability information to keep very large networks from being overwhelmed by routing updates.

4.4 Multi-Protocol Label Switching (MPLS)

MPLS [RVC01] is a network traffic management protocol that establishes a specific path, known as a Virtual Circuit (VC), for data streams. A VC is identified by a label attached to each packet, so routers need not retrieve the address of the next node in the path. As its name suggests, MPLS is compatible with a number of different protocols and architectures, including IP, ATM, and FR.

MPLS allows most packets to be forwarded at the data link layer rather than at the network layer, and simplifies QoS management through the support of a connection-oriented switching approach [Ko00]. MPLS is designed to scale efficiently, providing traffic engineering and re-routing mechanisms.

Connectionless network layer protocols require each router to make an independent forwarding decision for each packet it receives, which is performed by a network layer routing algorithm that examines the packet header to make its forwarding decision. Choosing the immediate destination to forward to (the "next hop") is performed in two stages:

1. *Establish a set of Forwarding Equivalence Classes (FECs).* This involves partitioning the set of possible packets into subsets based on their destination.
2. *Map each such FEC to a next hop.* This process ensures that all packets received from a given neighbour, that belong to a particular FEC, will follow the same path.

The FEC to which a given packet has been assigned is encoded as a fixed length label. This label is sent with the packet to its next hop, so there is no need for subsequent hops to analyse the network layer header because the label is used as an index into a table which specifies the next hop. The old label is then replaced with the new label, and the packet is forwarded to its next hop with the new label.

This technique of label-driven forwarding has a number of advantages over conventional network layer forwarding (RFC 3031 [RVC01]).

These facilities can be conveniently harnessed to create a scalable MPLS backbone architecture which can support multiple VPNs concurrently. This technique is discussed in Section 5.1.

4.5 Layer Two Tunnelling Protocol (L2TP)

VPNs use L2TP [TVR+99] to create data-link layer tunnels for carrying point-to-point data link (PPP) [Sim94] connections between endpoints. If the connection has been initiated by a dial-up user connecting to the home network through a local ISP, the ISP's NAS intercepts the connection and tunnels its packets to the destination. L2TP also provides authenticated tunnelling, but does not provide message integrity or confidentiality. To do so, it must be combined with IPSec (see Section 4.1).

The two main components that are used to provide L2TP services are the L2TP Access Concentrator (LAC) and the L2TP Network Server (LNS). The LAC is a physical device, such as a modem pool, to which dial-up users establish a connection. The use of an LAC allows the processing of PPP packets to be divorced from the termination of the layer two circuit [TVR+99]. One benefit of such a separation for VPNs is that the layer two connection need not terminate at the NAS, which may require a long-distance toll charge. Instead, the connection may terminate at an LAC, which then tunnels the PPP session across the shared network. The LNS is a device that terminates, and possibly authenticates, these tunnelled PPP streams.

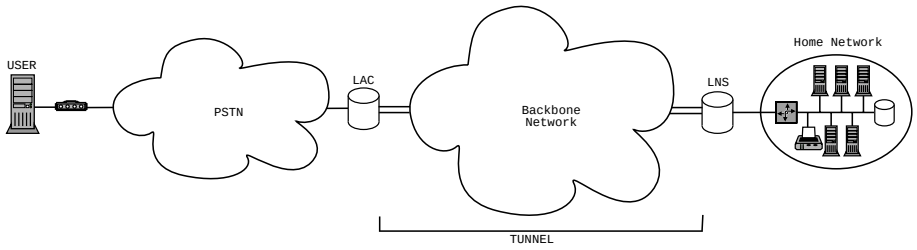

Figure 4.4: The scope and mechanics of an L2TP tunnel.

As shown in Figure 4.4, the user establishes a data link layer connection to an LAC, which transparently tunnels individual PPP frames to the NAS, where the tunnel is terminated.

L2TP is connection-oriented; the LNS and LAC maintain state for each connection that is initiated or answered by an LAC. An L2TP session is created between the LAC and LNS when an end-to-end PPP connection is established between a remote system and the LNS. Datagrams related to the PPP connection are sent over the tunnel between the LAC and LNS[5].

4.6 Remote Authentication Dial-In User Service (RADIUS)

RADIUS [ZLR+00] is an authentication tool designed for use in distributed systems, which separates the process of user authentication and authorisation from the communications process and establishes a central location for user authentication data. The process of separating security from communications enables security to be more effectively and efficiently applied and maintained. This is desirable when applied to VPNs, so that the security policy can be readily updated and modified to adapt to changing requirements, and is particularly valuable as a tool for authenticating remote VPN users.

[5] A tunnel exists between a LAC-LNS pair, and consists of a control connection and zero or more L2TP Sessions. The tunnel carries encapsulated PPP datagrams and control messages between the LAC and the LNS.

Chapter 5

VPN Configuration and Operation

5.1 MPLS VPNs

A VPN can be implemented over a provider's MPLS (see Section 4.4) backbone, which may support many VPNs concurrently. This technique models each VPN as a subset of the sites connected to the backbone, and access restrictions are enforced by the requirement that two sites may have IP connectivity over the backbone only if one or more of these subsets contain both sites [RR99]. As this implies, a given site may be a member of multiple VPNs.

The IETF Internet Draft "BGP / MPLS VPNs" [RBC+01] discusses the mechanisms available for implementing VPNs over MPLS infrastructure. This framework supports a wide range of policies, and the mechanisms are sufficiently general that these policies may be implemented by either the provider or both the customer and the provider.

The level of virtual connectivity provided will generally be more restrictive than that offered by the underlying physical connectivity. For example, it is possible to restrict certain sites from having direct routes to each other, creating a partially-meshed VPN. This can be desirable in situations where all traffic must pass through a certain point for purposes such as enforcing accounting or security policy.

There are three categories of devices that are used to direct traffic between sites on an MPLS VPN:

- *Customer Edge (CE) devices.* Each satellite network connects to one or more PE devices on the backbone with one or more CE devices. A CE device may be a host or a switch, but is generally a router. CE devices at different sites have no knowledge of each other, so the customer does not perform management of the backbone, and does not handle inter-site routing issues. This is in contrast to how overlay VPNs operate, where each CE device must be explicitly configured to recognise the CE devices at all other reachable VPN sites.

- *Provider Edge (PE) devices,* also known as edge Label Switching Routers (LSRs). As discussed previously, the backbone network is capable of supporting multiple VPNs concurrently, so routers must be capable of differentiating between traffic from different VPNs and applying appropriate routing behaviour as necessary. This is achieved through the use of specially-configured PE routers, which maintain a number of separate forwarding tables. Every site to which the PE is attached is mapped to one such forwarding table[1], and when a packet is received from a particular site, the corresponding forwarding table is consulted in order to determine how the packet should be routed. Forwarding tables are populated solely with routes that connect other sites of that site's VPN(s), preventing communication between sites which have no VPN in common.

  As shown in Figure 5.1, a PE router acts as the interface to the backbone for sites, which connect via

[1] Different sites can be mapped to the same forwarding table if they have all their VPNs in common.

their CE device(s). When a PE router receives a packet from a CE device, it identifies the interface over which the packet arrived in order to determine which forwarding table to use for processing that packet.

PE routers also maintain a default forwarding table, which is consulted whenever non-VPN traffic, or traffic not specifically related to any of that router's mapped VPNs, is received.

- *Provider (P) devices*, also known as core LSRs. Each VPN-aware PE router in the backbone must maintain multiple forwarding tables, which introduces substantial overhead and creates the potential for poor scalability as the number of VPNs supported by a given backbone network becomes large. In order to address this concern, the PE routers are moved to the perimeter of the backbone, where VPN sites connect, and the heart of the backbone network is instead populated with VPN-ignorant P routers, which have no knowledge of which VPN, if any, is associated with packets it forwards. P routers are responsible solely for forwarding traffic from the PE router where the traffic entered the backbone, to the PE router where the destination site is connected.

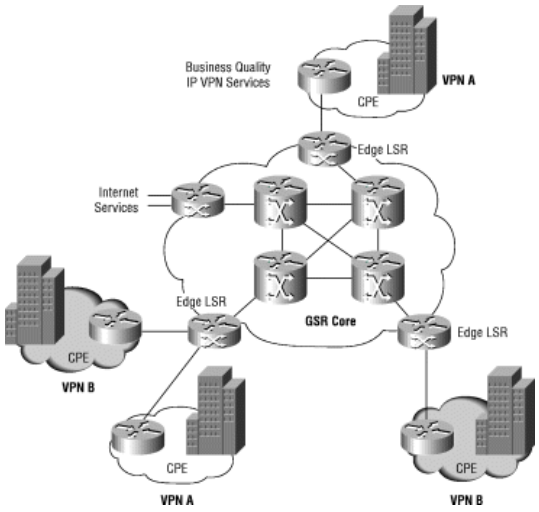

Figure 5.1: The Topology of an MPLS VPN backbone network [Sys01].

These devices are connected as shown in Figure 5.1. As this suggests, many paths between VPN sites require packets to be forwarded through a number of P routers, which have no knowledge of various VPN routes. This is achieved using MPLS with a two-level label stack; one label specifies the end site, and the other label is used in traversal of the backbone to the PE router connecting the destination site to the backbone.

VPN membership is dependent on which pre-provisioned ports on the edge LSR belong to a particular VPN. Logical ports on the provider edge router are associated with particular VPNs when provisioned. During VPN provisioning, a unique RD is assigned to each VPN in a process which is invisible to the end user, so a packet can enter a VPN only via a logical port configured for that VPN. The provider, not the customer, associates a specific VPN with each interface when provisioning a VPN. Users are able to access an Intranet or Extranet only if they reside on the correct physical or logical port and have the corresponding RD.

5.1.1 The VPN-IPv4 Address Family

As discussed previously, PE routers maintain per-site forwarding tables in order to support multiple VPNs concurrently. These are known as VPN Routing and Forwarding tables (VRFs), and they allow different

VPNs to employ arbitrary private address spaces [RMK+96] without creating any routing ambiguity on the backbone. Consequentially, a given address may correspond to different locations within different VPNs, so in order for a PE router to identify which routing table is appropriate for forwarding a given packet, it must be capable of associating data streams with particular VPNs. This is achieved through the use of the VPN-IPv4 Address Family.

A VPN-IPv4 address is a 96-bit value, comprised of a 64-bit Route Distinguisher[2] (RD) prefix, and a standard 32-bit IPv4 address suffix. This composite address structure ensures that if the same address is used in two different VPNs which are supported by the same MPLS backbone, it is possible to install a distinct route to that address for each VPN. In addition, the RD can be used to provide multiple routes to the same system, which is desirable in some policy configurations, such as when differentiating service and access levels to Intranet and Extranet users.

An RD consists of a 16-bit type field, and a 48-bit value field which is comprised of administrator and assigned number fields. The length and semantics of the administrator and assigned number fields vary depending on the value of the type field. This structure is ignored when used for route dissemination by BGP, so in such a context only the property that its value is unique is significant.

5.1.2 Security Issues

There is a clear division of duties, responsibilities and administrative boundaries in MPLS VPNs; customers cannot access PE or P routers, and the SP is not responsible for CE devices (although they may in fact support the CE devices under a separate agreement if the equipment is outsourced by the customer to the same provider).

In order to provide data confidentiality, it is a condition of this technique that backbone routers accept labelled packets from non-backbone devices only if the following conditions are met:

- *The non-backbone router received the given label from the backbone router.* This requirement ensures that labelled packets entering the backbone possess a legitimate and properly-assigned label at the top of their label stack.
- *The packet will leave the backbone before the given label is popped.* This requirement ensures that neither the underlying IP header nor labels lower in the stack will be inspected by a backbone device.

It is common for backbone routers to be configured so that all labelled packets received from non-backbone routers are automatically discarded, to prevent packets reaching VPN sites from unsupported sources.

Security under this configuration is equivalent to that provided to VPNS by Frame Relay or ATM backbones, and is similar to the trusted provider model discussed in Chapter 2, as the customer must rely on the provider to ensure that data does not enter or leave the VPN unless authorised. This model requires that the provider ensures the security of backbone devices from tampering or traffic capture, and that routing protocols should not allow connections with untrusted peers.

It is important to note that this technique does not provide cryptographic data protection, although the underlying architecture is such that encryption between CE end-points can be employed if desired.

5.2 Provider-Provisioned VPNs (PP-VPNs)

Implementing and maintaining a secure, reliable VPN requires a substantial investment in equipment and expertise, which may discourage smaller organisations from partaking of VPN technology. This creates an opportunity for ISPs to offer Provider-Provisioned VPN services [CSG+01] to customers, taking over

[2] The Route Distinguisher contains no information about the route origin or about the relevant set of VPNs; it is used solely for the creation of distinct routes to a common IPv4 address.

much of the management overhead and equipment requirements and so making VPN technology practical for smaller enterprises.

Under such an arrangement, the various VPN sites are connected to the SP's IP network backbone, with the constraint that traffic may travel between two sites only if a subnet[3] contains both sites. The SP acts as a "one-stop shop" for the provision, billing, administration and management of the VPN [BA96]. Details of the underlying network infrastructure and methods for data transport across it are hidden from the customer, so that operation is transparent.

Policies of operation for such VPNs can be the sole responsibility of the customer or the SP, or shared between them as required to achieve an optimal balance between cost and flexibility for the customer.

The partial or complete out-sourcing of provisioning duties for a VPN introduces concerns about how information is shared and security is managed. The SP should be able to view some of the network management information for the purposes of internal management. Relevant information will include traffic events, statistics events, security logs and configuration information [BA96].

The SP stores and manages information related to the private and customer directories of the PP-VPN networks, and it is the responsibility of the SP to ensure the integrity and confidentially of this data. The SP is also responsible for performing service usage authorisation checks, to confirm that a user is authorised to use a given resource before granting access. Authentication and access control protection also needs to be provided to protect the user directories[4] and customer directories[5].

5.3 Quality of Service (QoS)

IP networks currently possess no direct mechanism for specifying a desired quality of service for different traffic streams, and so service responses cannot be differentiated between traffic from the various users of the network. Hence they are said to offer a "best-effort" response service, whereby all traffic is processed on a first-in-first-out (FIFO) basis. As the network load varies, the network's response will in turn vary accordingly. Network congestion can result in unacceptably poor performance for a VPN, potentially hindering mission-critical applications.

This situation is unacceptable for enterprises who require dependable levels of performance from their VPNs. The goal of QoS is to make available a variety of selectable service response levels to accommodate these needs. Service responses may represent some form of superior service level, or they may provide a predictable service response, unaffected by external conditions.

QoS provides both bandwidth management and congestion management or avoidance. This not only allows different types of traffic to receive different levels of service based on previously established agreements, but also maximises network efficiency [Ko00]. This is particularly important for applications such as voice and video, which require very specific service levels in order to operate effectively, such as an upper bound on loss and delay rates. The requirements for data and multimedia are very different; multimedia can generally cope with some loss, but due to its real-time nature cannot tolerate delays. Data is more tolerant of delay, but can not tolerate loss, as dropped or corrupted packets must be retransmitted.

The provisioning of distinguished levels of service requires the ability to provide a differentiated service response within the network, and the ability to control – and when necessary, limit – the level of service-qualified load admitted into the network, in order to ensure that the response level required by resources who have already reserved bandwidth can be satisfied. Many different service response mechanisms exist which can be used over IP, but the ability to provide a completely consistent response level is a current

[3] A subnet is a subset of all sites connected to the backbone.

[4] The end user may maintain a private directory in which rules and policies for the request and receipt of VPN calls are defined [BA96].

[5] The customer has access to the customer management service and maintains the customer directory, which stores information about closed user groups, equipment and resources of the communication services, billing policy, network sites and end-points, and end-user authorisation codes [BA96].

area of weakness. A simple but manageable QoS strategy is to assign a given QoS level to an entire VPN, so that traffic for that VPN is granted the corresponding level of performance. FR and ATM networks can accomplish this, but such a facility of IP is problematic, because routing protocols like OSPF, which are used in the construction and maintenance of routing tables, are incapable of communicating QoS or resource utilisation information. Without knowledge of the commitments already made by the network, it is not possible to guarantee QoS.

Further detail on QoS for IP networks is provided in "Next Steps for the IP QoS Architecture" [Hus00].

5.4 Reliability and Survivability

VPNs require the support of backbone networks which offer equivalent reliability to that offered by the public switched telephone network. Fault-tolerance and redundancy are critical factors for providing high-availability network connectivity, which must continue to operate not only during planned network upgrades and changes, but also when connections fail unexpectedly due to component fault or attack.

Redundancy aids the ongoing provision of network services in both situations; by eliminating single points of failure, networks can be relied upon to be highly resilient - a key requirement for mission-critical applications which are often reliant on VPNs.

Graceful degradation paths should be planned and disaster recovery contingencies tested. A VPN should provide quick, secure recovery of link and node failures. Upon detecting a failure, restoration schemes include switching to pre-planned backup routes or dynamic protocols that search for available backup routes given spare capacity. The use of "chaining" a number of components together to achieve functionality, including routers, gateways and firewalls, creates more opportunities for failure. Combining all of the necessary VPN functions (see Section 1.1) into a single special-purpose VPN component with added redundancy features such as multi-homed connections and reliable router recovery mechanisms is a far more robust solution [PSDY00].

It may be necessary to provision the VPN infrastructure through more than one ISP, either out of necessity, due to ISPs' limited Points of Presence (PoPs), or as a multi-homing availability technique. In such cases, facilities for inter-provider provisioning and VPN identification will be required.

Chapter 6

Evaluation of Performance in a VPN Environment

The widespread deployment and successful operation of VPNs relies on a key benefit being achieved and maintained; that they are cheaper than the implementation and maintenance of a functionally-equivalent dedicated network, without sacrificing the security of the data and applications that the VPN supports. The relative cost is probably the most hyped characteristic of VPNs, particularly by telecommunications companies looking to provide VPN services to customers. In addition, tools for providing strong security enjoy strong support from networking vendors.

However, one consideration which has received very little attention, but nonetheless represents information of vital importance to anyone contemplating the implementation of a VPN, is the operational performance of the VPN in relation to its dedicated equivalent. This is important, because it should not be assumed that a VPN will be able to support all the distributed applications that a dedicated network can; each of the security measures discussed in Chapter 2 has a computational overhead associated with it, which together have the potential to impair the efficient operation of some applications. This overhead will generally manifest itself in the form of reduced throughput, increased packet delay, or some combination of the two.

The remainder of this chapter documents the experiments that were carried out in an attempt to quantify the level of overhead imposed by specific examples of the discussed security mechanisms.

6.1 Previous Work

The area of quantifying the performance overhead of network security tools appears to have received very little attention to date, with a literature search yielding only a single paper: "Performance Evaluation of Software Virtual Private Networks (VPN)" [PE00], which compares a number of different software VPN tools. This paper reports that "software VPNs may have a significant impact on performance, producing high CPU usage and limiting network throughput."

In addition, "A Layered Framework Strategy for Deploying High Assurance VPNs" [PSDY00] discusses steps for conducting an investigation into the performance overhead of encryption. Specifically, the authors advocate the use of a "VPN Decision-Making Framework", which encourages customers to consider a number of factors, including encryption, firewall considerations, existing network infrastructure, survivability requirements and the need for any infrastructure-dependent applications. In addition, the importance of considering the impact of encryption processing on network performance is stressed. The authors claim that it is "essential to determine the level of processing needed at peak times in order to effectively handle a given number of secure connections." Some advice on how to perform such an investigation is offered, but no such test was reported.

One possible reason for the lack of attention received by this area is that the outcome of any such experiments will be, by necessity, highly proprietary; the choice of hardware devices and network topology may impact significantly on the results, as there is no "typical" network topology, configuration or behaviour on which to base an experiment. Despite this limitation, it is considered that conducting such an experiment would yield results which, while not necessarily being directly applicable to all possible VPN implementations, would provide a set of indicative values which would aid in the design of subsequent case-specific investigations.

6.2 Method

As mentioned previously, the experiments aimed to quantify the level of performance overhead incurred when implementing examples of the security features discussed in Chapter 2. In order to achieve this, a simple two-site VPN, consisting of a single connection between a client machine and a server machine, was implemented. This network was used as a testbed for measuring the levels of throughput and latency experienced over a number of file transfers. Microsoft Internet Information Server (IIS) version 3.0 and Microsoft FTP server 3.0 were installed on the server for file transfers using HTTP and FTP respectively. HTTP sessions were established by the client through Internet Explorer version 5.5, and FTP sessions were established from the MS-DOS command line interface.

Ten different files were transferred between the machines, ranging in size from 100kb to 10Mb, and the (unweighted) mean throughput and latency recorded. Throughput was measured using Microsoft Network Monitor version 1.1, using the time stamps on the first and last packet of the data stream to derive an accurate measure of the time taken to transfer the given quantity of data. Latency was measured using the `ping` tool during file transmission[1] to measure the level of latency currently being experienced by that link. This task was repeated five times, with the VPN modified to increase the level of security afforded to the connection after each iteration.

The following configurations were evaluated[2]:

1. *Unsecured connection.*

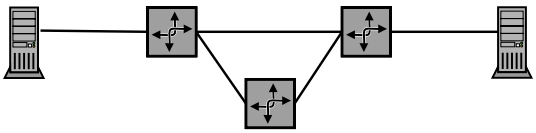

Figure 6.1: The two machines connected via a routing triangle.

As shown in Figure 6.1, this configuration connected the client and server via three Cisco 2514 routers arranged in a "routing triangle". All routers were configured to dynamically determine routing paths using the Router Information Protocol (RIP) [Mal98], and their clock rates were set to the maximum supported line speed of 64kb/s.

[1] Note that `ping` reports the time taken for the return trip, so the level of latency reported here may be slightly higher than that actually experienced by packets within the data stream. However, since the traffic flows were predominantly unidirectional, it is expected that the discrepancy would be minimal.

[2] The network diagrams which describe the topologies evaluated conform to the key given in Figure 6.6.

2. *Firewall-protected connection.*

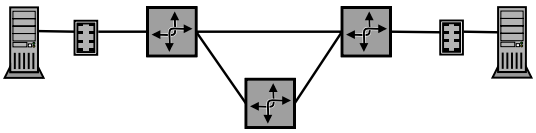

Figure 6.2: The two machines connected via firewalls and routers.

As shown in Figure 6.2, this configuration added a firewall between each computer and its gateway router, with the previous routing policy maintained. Both firewalls were assigned the following policy:

- Both incoming and outgoing `ping` (ICMP) traffic was allowed.
- Both incoming and outgoing `trace-route` traffic was allowed.
- Both incoming and outgoing HTTP traffic was allowed via a proxy service, with filtering of ActiveX and Java applets disabled.
- Both incoming and outgoing FTP traffic was allowed via a proxy service.
- All other protocols were disallowed.

3. *Authenticated, tunnelled connection.*

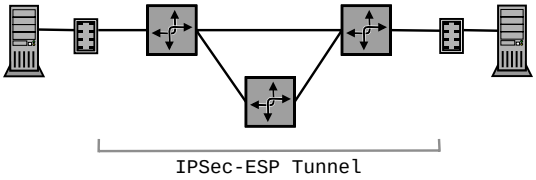

Figure 6.3: The two machines connected via an IPSec tunnel.

As shown in Figure 6.3, this configuration was identical to the previous configuration, except that the firewalls were reconfigured to tunnel all traffic between the client and server using IPSec-ESP (see Section 4.1.1), with HMAC-SHA-1-96 employed for authentication.

4. *DES-encrypted connection.*

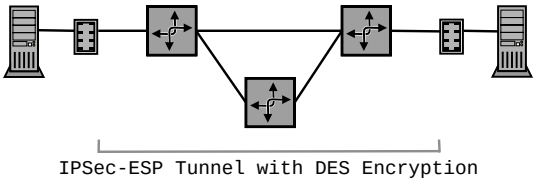

Figure 6.4: The two machines connected via a DES-encrypted IPSec tunnel.

As shown in Figure 6.4, this configuration was identical to the previous configuration, except that the firewalls were reconfigured to encrypt the data segment of all tunnelled packets between the client and the server using DES.

5. *3DES-encrypted connection.*

As shown in Figure 6.5, this configuration was identical to the fifth configuration, except that the firewalls were reconfigured to encrypt the data segment of all tunnelled packets between the client and the server using 3-DES.

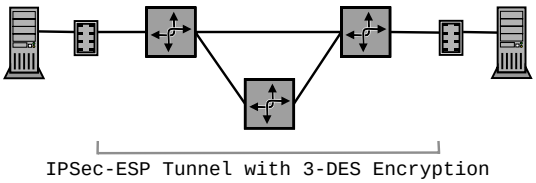

Figure 6.5: The two machines connected via a 3DES-encrypted IPSec tunnel.

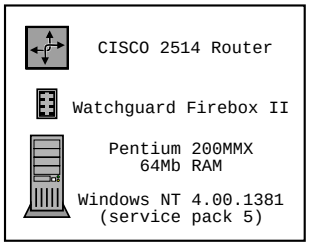

Figure 6.6: The components used in the VPN testbed construction.

As discussed in Section 6.3.1, the results of these experiments suggested that the routers may have acted as a bottle-neck to the VPN, so the five different configurations were re-tested with the routers removed, so that the firewalls were connected via a hub only, allowing the backbone connection to operate at LAN speed.

6.3 Results

The mean levels of throughput and latency observed on each of the configurations are given in Tables 6.1 and 6.2 respectively. The standard deviations are given in brackets.

Configuration	**Throughput** (with routers)		**Throughput** (without routers)	
	FTP	HTTP	FTP	HTTP
1	7.61 (0.007)	5.85 (0.250)	684.23 (29.18)	6.03 (0.300)
2	7.32 (0.092)	5.87 (0.368)	109.21 (9.053)	6.017 (0.260)
3	7.01 (0.119)	5.80 (0.345)	111.73 (11.25)	5.972 (0.19)
4	7.02 (0.090)	5.57 (0.362)	96.44 (6.522)	5.816 (0.31)
5	7.04 (0.085)	5.60 (0.256)	102.88 (14.113)	5.777 (0.26)

Table 6.1: The mean level of throughput achieved for each configuration (measured in kb/s).

As demonstrated by Figures 6.7 and 6.8, the VPN configuration had no significant effect on HTTP throughput, regardless of whether routers were employed, with even the largest observed performance degradation occurring when DES encryption was added to the routed VPN, showing no statistically significant difference (paired T-test, $T(18) = -1.1997$, $p = 0.2646$). It is believed that the throughput results for HTTP were not representative of the capabilities of the underlying protocols or hardware, but rather that performance was hampered by the internal limitations of the HTTP client software employed by Internet Explorer.

Similarly, very little change in throughput was observed for FTP as the VPN configuration was modified, although a drop in throughput of 83.2% (paired T-test, $T(18) = 73.328$, $p < 0.01$) was experienced when firewalls were added to the VPN without the routed WAN component.

Configuration	**Latency** (with routers)	
	FTP	HTTP
1	248.3 (55.32)	242.4 (40.72)
2	321.6 (41.29)	332.4 (32.76)
3	606.5 (124.62)	647.2 (137.36)
4	789.0 (58.65)	875.4 (126.26)
5	809.1 (54.90)	985.6 (135.02)

Table 6.2: The mean latency incurred by each configuration (measured in milliseconds).

The routed VPN displayed interesting properties in terms of latency. As shown in Figure 6.9, the level of latency incurred increased significantly as security features were added. The level of latency, and the potential reasons and implications of this latency, are discussed in the remainder of this section, in the context of the security tools that created this latency.

When the router backbone was removed from the VPN, the latency dropped to an immeasurably small level (under 10ms) for even the most computationally-expensive firewall configuration, so the results have not been presented here. It is believed that this result is probably due to the low level of loading imposed on the firewalls; only a single session originating from a workstation-class machine was generated, which is dramatically less than a hardware firewall is capable of supporting. A VPN generally has a much higher loading at the gateway, so the performance of the different firewall configurations when it is more heavily utilised would be a logical next step for evaluation. This is discussed further in Section 6.4.1.

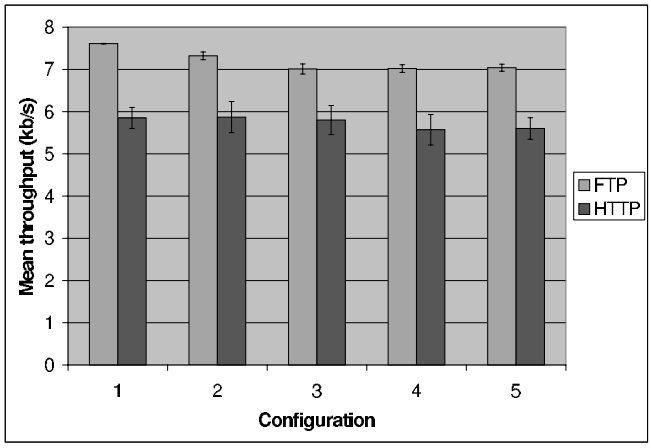

Figure 6.7: The mean level of throughput achieved for each configuration. Error bars show the standard deviation.

6.3.1 The Impact of Firewalls

The introduction of firewalls to the VPN caused a drop in FTP throughput of 3.8% (paired T-test, $T(18) = 9.8994$, $p < 0.01$) for the routed VPN, and 83.2% for the VPN without routers. There was no significant change in the throughput of HTTP for either the routed VPN (paired T-test, $T(18) = 1.6355$, $p = 0.691$), or the VPN without routers (paired T-test, $T(18) = 0.1609$, $p = 0.906$).

On the routed VPN, the latency increased by 29.5%, or 73.3ms, for FTP traffic (paired T-test, $T(18) = 4.846$, $p < 0.01$), and 37.1%, or 90ms, for HTTP traffic (paired T-test, $T(18) = 3.0668$, $p < 0.05$) with the introduction of firewalls.

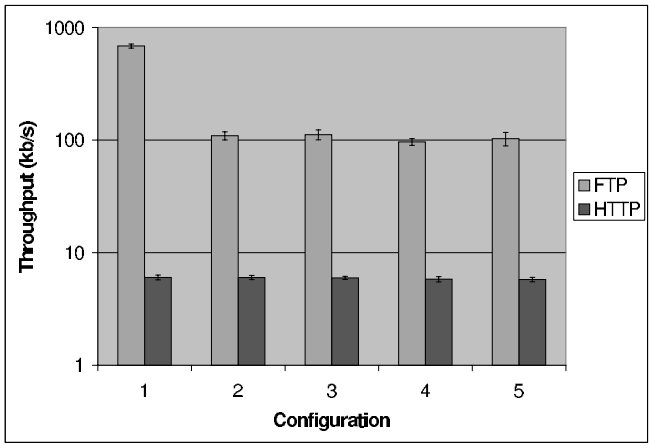

Figure 6.8: The mean level of throughput achieved for each configuration, without routers. Error bars show the standard deviation.

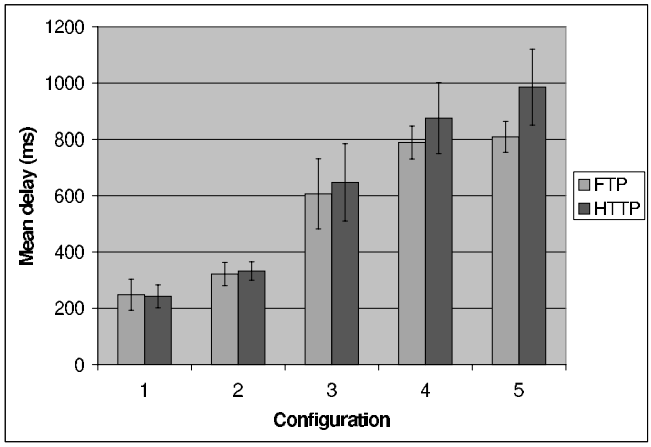

Figure 6.9: The mean level of latency incurred by each configuration. Error bars show the standard deviation.

These results indicate that the processing cost associated with inspecting traffic to determine if it conforms to security policy imposes a large burden if conducted within the LAN, but will have a more negligible impact if conducted at the LAN/WAN gateway, where firewalls are traditionally located. The observed increase in latency is sufficiently small (under 100ms) that it is unlikely to pose problems for any applications.

6.3.2 The Impact of Authenticated Tunnels

When the firewalls were configured to create authenticated tunnels, throughput dropped slightly on the routed VPN for FTP, by 4.2% (paired T-test, $T(18) = 2.458$, $p < 0.05$), but produced no significant change to HTTP throughput (paired T-test, $T(18) = 0.587$, $p = 0.572$). The VPN without routers experienced no significant change in throughput for FTP (paired T-test, $T(18) = 0.6552$, $p = 0.5287$) or HTTP (paired T-test, $T(18) = 0.4301$, $p = 0.6772$) with the introduction of tunnelling.

The use of tunnels also increased the latency of the connection for both FTP, by 88.6%, or 284.9ms

(paired T-test, T(18) = 6.2331, $p < 0.01$), and HTTP, by 94.7%, or 314.8ms (paired T-test, T(18) = 8.075, $p < 0.01$).

These results indicate that, while having little effect on throughput, the process of encapsulating and decapsulating traffic, and verifying checksums in order to maintain an authenticated tunnel carries a substantial processing cost, causing the level of latency over the network to increase substantially.

6.3.3 The Impact of Encryption

The addition of DES encryption had no significant effect on the throughput of FTP over the routed VPN (paired T-test, T(18) = .02547, p = 0.805), but caused a drop in throughput of 13.7% on the VPN without routers (paired T-test, T(18) = 3.9333, $p < 0.01$).

HTTP traffic experienced no significant change in throughput with the addition of encryption to the routed VPN (paired T-test, T(18) = 1.5056, p = 0.1664), or the VPN without routers (paired T-test, T(18) = 1.063, p = 0.3155).

The introduction of encryption had a more noticeable effect on latency, with FTP traffic over the routed VPN delayed by an additional 30.1%, or 182.5ms (paired T-test, T(18) = 3.975, $p < 0.01$) under DES, and an additional 33.4%, or 202.6ms (paired T-test, T(18) = 4.6119, $p < 0.01$), for 3-DES. The choice of encryption algorithm had no significant effect on latency for FTP (paired T-test, T(18) = 0.7539, p = 0.4702).

HTTP traffic experienced a similar increase in latency, with DES imposing a 35.3%, or 228.2ms, latency increase (paired T-test, T(18) = -5.0301, $p < 0.01$), and 3-DES imposing a 52.3%, or 338.6ms, latency increase (paired T-test, T(18) = 17.8688, $p < 0.01$). The choice of encryption algorithm had a significant effect on HTTP traffic latency, with 3-DES imposing an additional latency of 12.6%, or 110.2ms, to that imposed by DES (paired T-test, T(18) = 2.5038, $p < 0.05$).

These results show that encrypting traffic poses a substantially greater burden than merely tunnelling it, although the effort of tunnelling is largely wasted if it is not combined with encryption. These results also indicate that the decision as to whether to employ encryption will have a much greater influence on VPN performance than the choice of which cryptographic method to employ, as 3-DES – a much more computationally-expensive process than DES – performed only slightly worse than DES in terms of its effect on performance. This suggests that if encryption is to be used, the strongest available method should generally be employed.

6.3.4 The Impact of Routers

As can be seen in Table 6.1, the use of a routing backbone had a dramatic effect on FTP throughput. This is primarily caused by the fact that the Cisco 2514 routers used to build the routing backbone support a maximum line speed of 64kb/s, whereas the connection would otherwise operate at LAN speed, in the Mb/s range. These results reinforce the belief that VPN performance is heavily dependent on the performance of the backbone network, making the choice of connectivity provider and SLA an important consideration.

6.4 Discussion

Possibly one of the more interesting insights to be gained from these experiments is the potentially large impact the performance of the backbone network can have on the VPN as a whole. Table 6.1 suggests that the firewalls could sustain far higher loading before their performance degrades to that of the WAN component of the VPN. The backbone network in this case is miniscule compared to an enterprise-level backbone, however, so the results gained here in relation to backbone performance should not be assumed to be indicative of what is available from service providers. Specifically, the unrealistically close proximity

of the source and destination machines, both in terms of physical distance and router hops, may have had a distortional effect on the observed performance.

The two measured indicators of performance, throughput and latency, have very different effects on networked applications. Throughput will primarily affect applications which require sustained access to large quantities of bandwidth, such as FTP. Low throughput will increase the time taken for such applications to perform certain tasks. In contrast, latency will have a greater impact on real-time applications such as VoIP or streaming video; excessive latency will cause the continuous communication facilities of such applications to degrade or even fail, reducing the quality and usability of these services. On the routed VPN, the observed levels of latency were sufficiently high that real-time services may be adversely affected.

One conclusion that should be drawn from these findings is that it would be dangerous to start building a VPN without considering the performance levels it will be required to produce, and how these needs are likely to change in the future. Once these needs are established, tests with various products should be conducted to confirm that these needs can be met before investing heavily in specific hardware or other facilities.

6.4.1 Further Work

There are two key confounding factors to these experiments: the inability of the testbed configuration to impose a load on the firewalls sufficient to gauge their behaviour while heavily utilised, and the simplicity and low performance of the routed backbone. VPN gateways generally perform the majority of security-related tasks, so the issue of performance of these devices while under a heavy load and a large number of concurrent connections, is one which would warrant further investigation. It is believed that the performance overhead of tunnelling and cryptography may be greater when the gateway device is experiencing more realistic utilisation patterns.

Chapter 7

Conclusion

VPNs have evolved as a cost-effective solution to the problem of providing network connectivity to geographically dispersed locations. They offer significant cost savings to the acquisition of dedicated private links, which are prohibitively expensive for many customers, particularly smaller organisations. However, the use of a VPN introduces a number of new concerns, primarily resulting from the use of a shared, and therefore untrusted, backbone network. This leads to more complex operations in a number of areas, including management and configuration, security, and network performance. In particular, the suite of security tools that are commonly used to protect VPNs (see Chapter 2) require a substantial investment in time and other resources to implement and maintain.

As discussed in Section 3.1, a variety of approaches are available when implementing a VPN, providing the customer with some freedom in terms of the level of investment in hardware and responsibility for security and administration they will undertake. The outsourcing of configuration and administrative tasks reduces the need for specialised equipment or expertise, but can give the customer less control over the operation of the VPN as the requirements of the organisation evolve.

As discussed in Section 4.1, some of the most important security mechanisms are not yet sufficiently mature to offer reliable inter-operability under certain conditions, a situation which may be fatal to selected applications or services. Such conflicts must be resolved in order to support reliable, efficient VPN operation.

The results of the practical investigations suggest that the performance overhead of security can be significant, so the performance implications of any proposed security mechanisms should be investigated carefully prior to any substantial investment in VPN infrastructure.

VPNs are undoubtedly an important technology, and one which will receive increasing levels of interest from customers, and support from vendors, as business operations continue to become more globalised and more dependent on computers and data communication systems. However, it is clear that a number of potential pitfalls exist at present, so it is important that prospective VPN customers perform a thorough analysis of their specific requirements, investigate any suitable options and carefully plan the different implementation stages in order to achieve a successful VPN implementation.

Appendix A

Glossary

Due to space constraints, technical terms are not defined here, and only acronym expansions are provided. Details on specific terms are available at `http://www.emory.edu/ITD/RA/vpn/glossary.html` and `http://whatis.techtarget.com`.

Acronym	Meaning
AAA	Authentication, Authorisation and Accounting
AH	Authentication Header
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
BITS	Bump In The Stack
BITW	Bump In The Wire
CE	Customer Edge
CIDR	Classless Inter-Domain Routing
CPE	Customer-Provider Edge
CRL	Certificate Revocation List
DES	Data Encryption Standard
DMZ	De-Militarized Zone
DNS	Domain Name System
ECN	Explicit Congestion Notification
ESP	Encapsulated Security Payload
FIB	Forward Information Base
FR	Frame Relay
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
GSR	Gigabit Switching Router
HMAC	Hashed Message Authentication Code
HTTP	Hyper-Text Transfer Protocol
IBGP	Internal Border Gateway Protocol
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IKE	Internet Key Exchange

Acronym	**Meaning**
IP	Internet Protocol (version 4)
IPv6	Internet Protocol version 6 (or Internet Protocol next generation)
IPSec	Internet Protocol Security
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
LAN	Local Area Network
LDP	Label Distribution Protocol
LLC	Logical Link Control
LSP	Label Switched Path
LSR	Label Switching Router
L2TP	Layer 2 Tunnelling Protocol
MAC	Media Access Control
MIB	Management Information Database
MD	Message Digest algorithm
MPLS	Multi-Protocol Label Switching
MPOA	Multi-Protocol Over ATM
NAPT	Network Address Port Translation
NAS	Network Access Server
NAT	Network Address Translation
NIC	Network Interface Card
NHRP	Next Hop Resolution Protocol
NLRI	Network Layer Reachability Information
OSPF	Open Shortest Path First
OTP	One-Time Pad
PE	Provider Edge
PKI	Public Key Infrastructure
POP	Point Of Presence
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunnelling Protocol
PP-VPN	Provider-Provisioned Virtual Private Network
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RD	Route Distinguisher
SCEP	Simple Certificate Enrolment Protocol
SHA	Secure Hashing Algorithm
SHTTP	Secure Hyper-Text Transfer Protocol
SLA	Service Level Agreement
SPD	Security Policy Database
SPI	Security Parameters Index
SSH	Secure SHell
SSL	Secure Socket Layer
TLS	Transaction-Layer Security
V-LAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
VPLS	Virtual Private LAN Segment
VPN	Virtual Private Network
VRF	VPN Routing and Forwarding table
WAN	Wide Area Network

Bibliography

[80299] IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks. In *LAN MAN Standards Committee of the IEEE Computer Society, USA*. IEEE Std 802.1Q-1998, March 1999.

[Abo01] Bernard Aboba. IPsec-NAT Compatibility Requirements. `http://www.ietf.org/`, June 2001. (Internet Draft).

[AK00] N. Provos A. Keromytis. RFC2857: The Use of HMAC-RIPEMD-160-96 within ESP and AH, June 2000. (Proposed Standard).

[BA96] Nora Boukari and Ali Aljane. Security and Auditing of VPN. In *Proceedings of Third International Workshop on Services in Distributed and Networked Environments*, pages 132–138. IEEE, 1996.

[Bel93] Ron Bell. Virtual Private Networks - The Major Issues, Problems and Opportunities. In *IEEE Colloquium of Virtual Networking*, pages 2/1 – 2/7. IEEE, 1993.

[BM00] Mark Becker and Greg Machler. Virtual Private Networks for the enterprise. *Enterprise Systems Journal*, 15(1):38–43, January 2000.

[Bro01] Kari Brooks, editor. *Networking COMPLETE*. Sybex, 2nd edition, March 2001.

[CSG+01] R. Callon, M. Suzuki, B. Gleeson, A. Malis, K. Muthukrishnan, Eric Rosen, Chandru Sargor, and Jieyun Jessica Yu. A Framework for Provider Provisioned Virtual Private Networks, July 2001. (Internet Draft).

[DA98] T. Dierks and C. Allen. RFC2246: The TLS Protocol Version 1.0 , January 1998. (Proposed Standard).

[Den82] Dorothy Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

[Den00] Dorothy E. Denning. *Information Warfare and Security*. Addison-Wesley, 4 edition, January 2000.

[FLH+00] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. RFC2784: Generic Routing Encapsulation (GRE), March 2000. (Proposed Standard).

[GLH+00] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. RFC2764: A Framework for IP Based Virtual Private Networks, February 2000. (Informational).

[HC98] D. Harkins and D. Carrel. RFC2409: The Internet Key Exchange (IKE), November 1998. (Proposed Standard).

[HFPS99] R. Housley, W. Ford, W. Polk, and D. Solo. RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, January 1999. (Proposed Standard).

[Hus00] G. Huston. RFC2990: Next Steps for the IP QoS Architecture, November 2000. (Informational).

[KA98a] S. Kent and R. Atkinson. RFC2401: Security Architecture for the Internet Protocol, November 1998. (Proposed Standard).

[KA98b] S. Kent and R. Atkinson. RFC2402: IP Authentication Header, November 1998. (Proposed Standard).

[KA98c] S. Kent and R. Atkinson. RFC2406: IP Encapsulating Security Payload (ESP), November 1998. (Proposed Standard).

[KBC97] H. Krawczyk, M. Bellare, and R. Canetti. RFC2104: HMAC: Keyed-Hashing for Message Authentication, February 1997. (Informational).

[KD01] Christopher M. King and Curtis E. Dalton. VPNs: THE GOOD THE BAD & THE UGLY. `http://www.infosecuritymag.com/articles/may01/cover.shtml`, May 2001.

[KMS95] P. Karn, P. Metzger, and W. Simpson. RFC1851: The ESP Triple DES Transform, September 1995. (Experimental).

[KN93] J. Kohl and C. Neuman. RFC1510: The Kerberos Network Authentication Service (V5), September 1993. (Proposed Standard).

[Ko00] Denny Ko. The right protocol. *Telephony*, 239(18):84–88, October 2000.

[KOT95] Takao Kato, Kazuhiko Omachi, and Shiro Tanabe. BVPN (Broadband Virtual Private Network): A Flexible, High-Speed, Enterprise Network Architecture. In *Proceedings of the Fifth IEEE Computer Society Workshop of Future Trends of Distributed Computing Systems*, pages 420–424. IEEE, 1995.

[Kra96] H. Krawczyk. SKEME: A Versatile Secure Key Exchange Mechanism for Internet. In *IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security*. IEEE, 1996.

[LBT96] David Lewis, Lennart H. Bjerring, and Ingi H. Thorarensen. An Inter-domain Virtual Private Network Management Service. In *Network Operations and Management Symposium*, volume 1, pages 115–123. IEEE, 1996.

[LMMN01] Xiaoyi Liu, Cheryl Madson, David McGrew, and Andrew Nourse. Cisco Systems' Simple Certificate Enrolment Protocol (SCEP). `http://www.ietf.org/`, February 2001. (Internet Draft).

[Mal98] G. Malkin. RFC2453: RIP Version 2, November 1998. (Standard).

[MG98a] C. Madson and R. Glenn. RFC2403: The Use of HMAC-MD5-96 within ESP and AH, November 1998. (Proposed Standard).

[MG98b] C. Madson and R. Glenn. RFC2404: The Use of HMAC-SHA-1-96 within ESP and AH, November 1998. (Proposed Standard).

[MSST98] D. Maughan, M. Schertler, M. Schneider, and J. Turner. RFC2408: Internet Security Association and Key Management Protocol (ISAKMP), November 1998. (Proposed Standard).

[Neu99] Peter G. Neumann. Risks of Insiders. *Communications of the ACM*, 42(12):160, December 1999.

[Orm98] H. Orman. RFC2412: The OAKLEY Key Determination Protocol, November 1998. (Informational).

[oS88] US National Bureau of Standards. Data Encryption Standard. `http://www.itl.nist.gov/fipspubs/fip46-2.htm`, January 1988.

[oS94] US National Bureau of Standards. DIGITAL SIGNATURE STANDARD (DSS). `http://www.itl.nist.gov/fipspubs/fip186.htm`, May 1994.

[PE00] C. Javier Castro Pena and Joseph Evans. Performance Evaluation of Software Virtual Private Networks (VPN). In *Annual IEEE Conference on Local Computer Networks*, pages 522–523. IEEE, 2000.

[PF00] Suketu Pandya and Elizabeth M Ferrarani. VPNs for everyone. *Enterprise Systems Journal*, 15(12):36–40, December 2000.

[PSDY00] Samuel Patton, Bryan Smith, David Doss, and William Yurcik. A Layered Framework Strategy for Deploying High Assurance VPNs. In *Fifth IEEE International Symposium on High Assurance Systems Engineering*, pages 199–202. IEEE, 2000.

[RBC+01] E. Rosen, Stephen John Brannon, Christopher J. Chase, Jeremy De Clercq, Paul Hitchin, Dave Marshall, and Vijay Srinivasan. BGP / MPLS VPNs, February 2001. (Internet Draft).

[RHH01] Chris Rodgers, Ray Hunt, and Brendon Harris. Networking Systems — Design, Analysis and Applications: Threats to TCP/IP Network Security, July 2001. COSC407 Research Project.

[rip98] Information technology - Security techniques - Hash-functions - Part 3: Dedicated hash-functions. *International Organization for Standardization, Geneva, Switzerland*, ISO/IEC 10118-3 1998.

[RL95] Y. Rekhter and T. Li. RFC1771: A Border Gateway Protocol 4 (BGP-4), March 1995. (Draft Standard).

[RMK+96] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. RFC1918: Address Allocation for Private Internets, February 1996. (Best Current Practice).

[RR99] E. Rosen and Y. Rekhter. RFC2547: BGP / MPLS VPNs, March 1999. (Informational).

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. RFC3031: Multiprotocol Label Switching Architecture, January 2001. (Proposed Standard).

[SE01] P. Srisuresh and K. Egevang. RFC3022: Traditional IP Network Address Translator (Traditional NAT), January 2001. (Informational).

[Shu00] J. Shukla. NGISec-NAT and QoS compatible End-to-End Secure Communication. `http://www.ietf.org/shadow.htm`, November 2000. (Internet Draft).

[Sim94] W. Simpson. RFC1661: The Point-to-Point Protocol (PPP), July 1994. (Standard).

[SSP01] Tissa Senevirathne, Som Sikdar, and Neena Premmaraju. Ethernet Over IP - A Layer 2 VPN Solution using Generic Routing Encapsulation (GRE), July 2001. (Internet Draft).

[Sys01] Cisco Systems. Intranet and Extranet Virtual Private Networking - Delivering New World Services With VPNs. `http://www.cisco.com/warp/public/cc/so/neso/vpn/vpnsp/ievpn_rg.htm`, March 2001.

[TE00] Joe Touch and Lars Eggert. Use of IPSEC Transport Mode for Virtual Networks. `http://www.ietf.org/`, November 2000. (Internet Draft).

[TVR+99] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. RFC2661: Layer Two Tunneling Protocol L2TP, August 1999. (Proposed Standard).

[Ven01] R. Venkateswaran. Virtual Private Networks — Various Services and Implementation Scenarios. In *IEEE Potentials*, volume 20, pages 11–15. IEEE, March 2001.

[Wil00] Beau Williamson. *Developing IP Multicast Networks*, volume 1. Cisco Press, 2000.

[WSCL+88] D. Wood, V. Stoss, L. Chan-Lizardo, G. S. Papacostas, and M. E. Stinson. VIRTUAL PRIVATE NETWORKS. In *International Conference on Private Switching Systems and Networks*, pages 132–136. IEEE, 1988.

[You00] Roger Younglove. Virtual Private Networks – How They Work. *Computing & Control Engineering Journal*, 11(6):260–262, December 2000.

[ZCC00] Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman. *Building Internet Firewalls*. O'Reilly and Associates Inc., 2000.

[ZLR+00] G. Zorn, D. Leifer, A. Rubens, J. Shriver, M. Holdrege, and I. Goyret. RFC2868: RADIUS Attributes for Tunnel Protocol Support, June 2000. (Informational).